UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

FERNANDO JOSÉ GÓMEZ PAREDES

ALTERNATIVES AND COMPLEMENTS TO POLCA SYSTEM, IN QRM CELLS CONFIGURATION: SIMULATED EXPERIMENTS

SÃO CARLOS-SP
Março de 2021

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

FERNANDO JOSÉ GÓMEZ PAREDES

ALTERNATIVES AND COMPLEMENTS TO POLCA SYSTEM, IN QRM CELLS CONFIGURATION: SIMULATED EXPERIMENTS

Tese apresentada ao Programa de Pós-graduação em Engenharia de Produção da Universidade Federal de São Carlos, como parte dos requisitos para obtenção do título de Doutor em Engenharia de Produção.

Orientador: Dr. Moacir Godinho Filho

SÃO CARLOS
2021

FICHA CATALOGRÁFICA

Gómez Paredes, Fernando José

Alternatives and complements to POLCA system, in QRM cells configuration: simulated experiments – São Carlos, 2021.

Número de páginas: 191

Área de concentração: Gestão da Produção

Orientador: Prof. Dr. Moacir Godinho Filho.

Tese de doutorado de Engenharia de Produção - Universidade Federal De São Carlos, Centro de Ciências Exatas e de Tecnologia, Programa de Pós-Graduação em Engenharia de Produção.

1. Production control; 2. Card-based system; 3. Job shop; 4. Indirect load; 5. POLCA

UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

Folha de Aprovação

Defesa de Tese de Doutorado do candidato Fernando José Gómez Paredes, realizada em 05/03/2021.

Comissão Julgadora:

Prof. Dr. Moacir Godinho Filho (UFSCar)

Prof. Dr. Matthias Thürer (Jinan University)

Prof. Dr. Nuno Octávio Garcia Fernandes (IPCB)

Profa. Dra. Juliana Keiko Sagawa (UFSCar)

Prof. Dr. Murís Lage Júnior (UFSCar)

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

O Relatório de Defesa assinado pelos membros da Comissão Julgadora encontra-se arquivado junto ao Programa de Pós-Graduação em Engenharia de Produção.

A los Pirroros del hoy, Pacote y Pungunga
para que pueda verlos volar y les enseñen a los del mañana: MaFe y JoMi

AGRADECIMENTOS

Quero agradecer a Deus pela imensa oportunidade de cumprir o sonho de estudar uma pós-graduação no Brasil. Espero que este trabalho consiga render mais frutos na sua obra, porque eu apenas fui um instrumento de fazer esta pesquisa.

Agradeço ao meu orientador, o prof. Moacir Godinho Filho. Fico muito grato por todos seus conselhos e por sua guia profissional na minha formação. Ele contribuiu muito para minha formação pessoal e profissional. Agradeço muito por ter sido uma inspiração durante todo o processo, e por materializar esse sonho de estudar no Brasil.

Estendo meus agradecimentos aos professores Mathias Thürer e Nuno Fernandes. Suas contribuições no processo de doutorado têm ajudado a encontrar níveis inimagináveis com a minha pesquisa. Agradeço também aos meus professores da casa da UFSCar: Juliana Sagawa e Murís Lage Júnior, pela sua ajuda na minha formação com vários encontros. Expresso também minha gratidão a todos os professores, secretários e técnicos administrativos do PPGEP, DEP da UFSCar, pelo seu apoio na formação, nos processos e na mínima ajuda que precisei para desenvolver minhas atividades. A vossa ajuda é incalculável.

Agradeço a minha família, minha esposa Daiana e sua família, pelo seu apoio (e agradeço ao doutorado por encontrar ela), assim como meus pais Ricardo e Silvia. Eles me ajudaram a manter a fé no resultado possível de terminar esta etapa. Também aos meus irmãos Ileana e Carlitos, pelo seu apoio em todo momento. Incluo também com eles, a toda minha família da Paróquia Santa Luzia de São Carlos, com seus administradores (Pe. Valcir, Pe. Carlos, e Pe. Walace), os pais brasileiros (Célia e Belo), meus irmãos de caminhada (Jorge, Joseilton e os outros moradores), o Ministério de Música MMSL como o grupo Unidos pela Fé: obrigado padrinhos por todo o que me acolheram nessa grande família.

Agradeço também a todos os meus colegas da sala PLACOP, GePReLT, agora GOSC. Eles colaboraram muito com diversas conversas desde o café até grandes discussões. Destaco aos meus irmãos Luana e Flávio, por tantos momentos importantes para desenvolver-me na pesquisa. Na passagem pela UFSCar, expresso a minha gratidão aos membros da Bateria UFSCar, porque permitiram criar outras habilidades pessoais.

A forma que a ciência seja relevante, agradeço a FANTEL de El Salvador e CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) do Brasil pelo financiamento da pesquisa.

FIGURES INDEX

TABLES INDEX

RESUMO

Alternativas e complementos ao sistema POLCA, em ambientes de células QRM: experimentos por simulação

O *Paired-cell Overlapping Loops of Cards with Authorization* (POLCA) é o Sistema de Controle de Produção (PCS) que o *Quick Response Manufacturing* (QRM) sugere para alta variedade e baixo volume. No entanto, o ambiente sugerido comumente descreve o *Make-to-Order* (MTO), que requer um PCS para controlar muitos roteamentos de jobs, como descreve o *General Flow Shop* (GFS), com tempos de processo variáveis em cada ordem. Nessas condições, liberar ordens por meio de listas de autorização apresentam alguns problemas para lidar com essa variabilidade de tempo, para liberar jobs. Uma das soluções tem sido utilizar PCSs alternativos como Controle de Balanço por Navegação Baseada em Cartão (COBACABANA). Outra solução para esse problema é que o POLCA incorporou outros complementos ao mecanismo de liberação de pedidos, como o *Release-and-Flow* POLCA (RF-POLC), o *Advanced Resource Planning* e o COBA-POLCA. No entanto, esses complementos não tiram proveito do conceito original do POLCA: os laços sobrepostos de células emparelhadas. Portanto, esta pesquisa propõe um mecanismo de liberação de ordens para POLCA baseado na carga indireta do chão de fábrica, denominado IL-POLC, que utiliza os laços de células emparelhadas para agregar a carga indireta. Para tanto, esta pesquisa utiliza uma revisão sistemática da literatura para identificar os fatores que influenciam o funcionamento dos PCSs no ambiente citado. Em seguida, a pesquisa apresenta dois experimentos de simulação para comparar o IL-POLC proposto. O primeiro experimento compara o IL-POLC com o POLCA original e o RF-POLCA, detalhando fatores específicos para a parametrização do POLCA. O segundo experimento compara IL-POLC com COBACABANA e COBA-POLCA. Como resultado da revisão da literatura, esta pesquisa apresenta um framework para descrever como cada fator influencia a tarefa dos PCSs. Este framework foi usado para definir fatores nos experimentos. O IL-POLC proposto mostra que a liberação de trabalhos por condições de carregamento indireto é melhor do que a liberação do pedido original da POLCA e RF-POLCA que usa as listas de autorização por datas de liberação. O IL-POLC resolve a variabilidade dos prazos de entrega pela condição de carga indireta no mecanismo de liberação de ordens. De acordo com o segundo experimento, IL-POLC é melhor do que os outros dois PCSs quando o chão de fábrica usa a versão baseada em cartão e o cartão representa proporcionalmente uma quantidade de trabalho de cada estação. O IL-POLC não reduz tanto o tempo total de atravessamento no chão de fábrica como os outros sistemas, mas não deteriora as outras medidas de desempenho. De acordo com os resultados, o experimento sugere o uso de IL-POLC em ambientes que tenha desafios para estimar as normas de carga de trabalho para a liberação centralizada. Pesquisas futuras são sugeridas para confirmar como a liberação de pedido adaptada reage com fatores externos, como disponibilidade de células, permutabilidade de mão de obra e teste em outros cenários como implementações práticas para confirmar esses resultados.

Palavras-chave: Controle de produção; Sistemas de controle por cartões; job shop; Carga indireta; POLCA.

ABSTRACT

Alternatives and complements to POLCA system, in QRM cells configuration: simulated experiments

The Paired-cell Overlapping Loops of Cards with Authorization (POLCA) is the Production Control System (PCS) that Quick Response Manufacturing (QRM) suggests for high variety and low volume. However, the suggested environment commonly describes the Make-to-order (MTO), which requires a PCS to control many routings of jobs, as the General Flow Shop (GFS) describes, and high variable lead times for each process. In those conditions, releasing orders by authorisation lists have presented some problems to deal with that time variability. One solution has been using alternatives PCSs as Control of Balance by Card-Based Navigation (COBACABANA). Another solution to that problem is that POLCA has incorporated other complements in the order release mechanism, as the Release-and-Flow POLCA (RF-POLCA), the Advanced Resource Planning and the COBA-POLCA. Nevertheless, those complements do not take advantage of the original concept of POLCA: the paired-cell overlapping loops. Therefore, this research proposes an order release mechanism based on the indirect load of a shop for POLCA, called IL-POLC, that uses the paired-cells loops to aggregate the indirect load. For that proposition, this research uses a systematic literature review to identify factors that influence how the PCSs work in the cited environment. Then, the research presents two simulation experiments to compare the IL-POLC proposed. The first experiment compares IL-POLC with the original POLCA and RF-POLCA, detailing specific factors for POLCA parametrisation. The second experiment compares IL-POLC with COBACABANA and COBA-POLCA. As a result of the literature review, this research presents a framework to describe how each factor influences the task of the PCSs. This framework was used to define factors in the experiments. The proposed IL-POLC shows that releasing jobs by indirect load conditions is better than the original order release of POLCA and RF-POLCA that uses the authorisation lists by release dates. IL-POLC solves the variability of lead times by the indirect load condition on the order release mechanism. According to the second experiment, IL-POLC is better than the other two PCSs when the shop uses the card-based version and a proportional representation of the card for the amount of work of each job. IL-POLC do not reduce the Shop Floor Throughput Time as the other systems, but it does not deteriorate the other performance measures. According to the results, the experiment suggests using IL-POLC in shops that it is challenging to estimate workload norms for the centralised release. Future research is suggested to confirm how the adapted order release reacts with external factors such as cell availability, labour interchangeability, and test in other scenarios as practical implementations to confirm these results.

Keywords: Production control; card-based system; job shop; Indirect load; POLCA

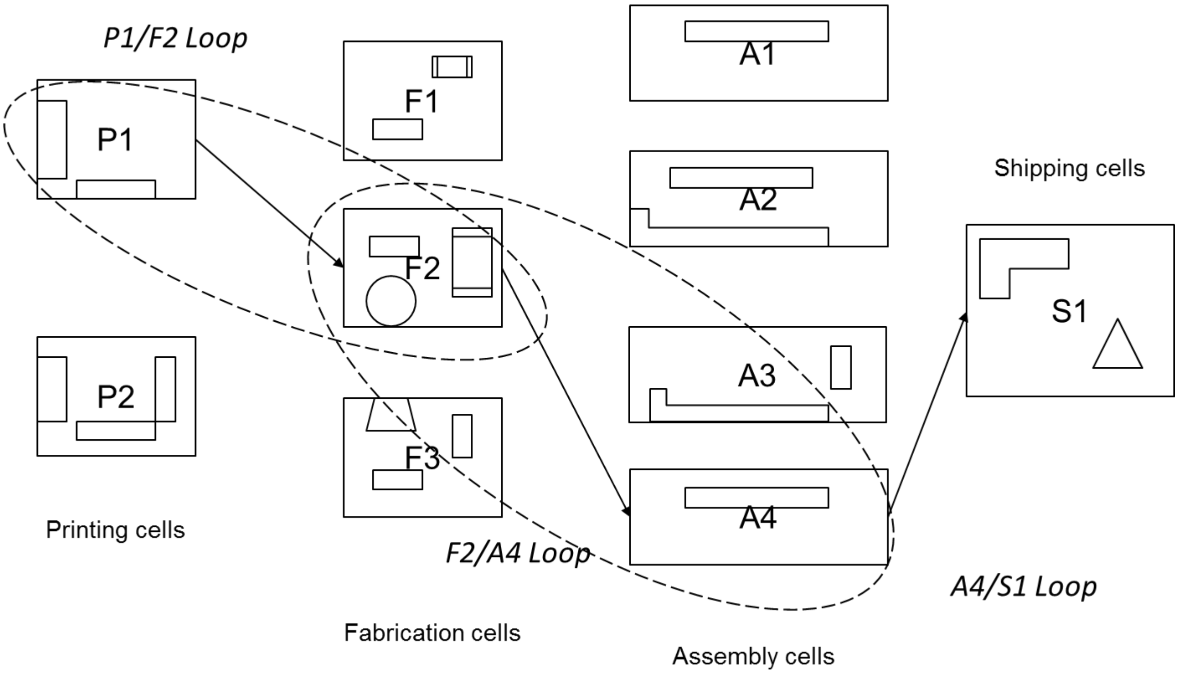
SUMMARY

1 INTRODUCTION

1.1 CONTEXT AND RESEARCH GAP

For Suri (1998), Quick Response Manufacturing (QRM) is an approach that focuses on the lead-time reduction and exploits the high variety of products as a competitive advantage. This approach suggests transforming functional layout to cellular-process layout to simplify the production flow to attend to the high variation and low volume demand. This cellular-process layout is commonly known as the “QRM cells”. On that transformation, Suri (2018) suggests avoiding re-entrant flows, or cycle flows, that complicates the job routing in the shop floor, suggesting the General Flow Shop (GFS). For this type of flow, Suri (1998) proposed the Paired-Cells Overlapping Loops of Cards with Authorization (POLCA) as the Production Control System (PCS) for that environment. POLCA reports successful implementations in real shops (KRISHNAMURTHY; SURI, 2009; SURI, 2018; VANDAELE et al., 2004) by designing the integration of push systems with the pull system (RIEZEBOS, 2010). Figure 1.1 shows a shop floor with a job routing that describes GFS. The job begins on any Printing cell, goes to any Fabrication cell, then to any Assembly cell and finishes at the Shipping cell. The job example of Figure 1.1 uses P1, F2, A4, S1. The paired cells for the first loop are P1 with F2, the second loop is F2 with A4, and the third loop is A4 with S1. These loops overlap because the F2 cell belongs to the first and second loops, as shown in Figure 1.1.

Figure 1.1 – Example of the paired-cells overlapping loops in the GFS



Source: Adapted from Suri (1998).

The integration of push systems with the pull system has allowed POLCA to substitute the original Kanban even for other high-variety approaches (POWELL; RIEZEBOS; STRANDHAGEN, 2013; SATOGLU; UCAN, 2015). Those results make POLCA very important to control production and support operations strategies in the high variety of products and routings. However, among Production Control Systems (PCSs), POLCA is not the only alternative system developed for alternatives for GFS. Some PCSs from other approaches proposed for high variety are Drum-Buffer-Rope (DBR) as PCS of the Theory of Constraints (GOLDRATT; FOX; GRASMAN, 1986); Constant Work-in-Process (CONWIP), proposed by Spearman, Woodruff and Hopp (1990); Control of Balance by Card-Based Navigation (COBACABANA) as a card-based PCS for Workload Control (WLC) approach (LAND, 2009), among others. Comparing those PCSs, Thürer, Stevenson and Protzman (2016b) present how each PCSs applies to the high-variety production. Bertolini, Romagnoli and Zammori (2017) establish which PCS is adequate for the production flow characteristics. Although the flow characteristics are the main factor for choosing the POLCA alternative, González-R, Framinan and Pierreval (2012) advises that many comparisons of the PCSs have different results because some scenarios could advantage one PCS over the other. Even though many PCSs have been identified in recent years (BAGNI et al., 2020), not all of them are designed for the general flow shop, and they present similar functions to the cited PCSs. This limitation on GFS raises the gap in which PCS is an alternative to POLCA for the high-variety. Nevertheless, their similar functions still include a gap in how an element can complement the POLCA to improve its performance.

Focused on the previous cited PCSs, some authors have compared different PCSs with POLCA, allowing an idea of which system to choose as an alternative for the GFS. For Kabadurmus (2009), POLCA has better performance, for the time in system and average WIP level, with unstable demand conditions than CONWIP, using batch sizes and arrival time distribution as variables. For Germs and Riezebos (2010), POLCA has better workload balancing capability than CONWIP since it minimizes lead time, considering the number of orders. Barros et al. (2016) also confirm those improvements of POLCA compared with CONWIP.

However, POLCA has also presented other disadvantages with PCSs. By the experiment that Lödding, Ya and Wiendahl (2003) presented, Decentralised WIP (DEWIP) outperforms POLCA in the Pure Job Shop (PJS) because POLCA presents heavy blockings on re-entrant flows. Germs and Riezebos (2010) exhibit that POLCA and m-CONWIP present similar results when the loops are coincident. Frazee and Standridge (2016) have reported that CONWIP has better performance than POLCA when the processing times present a low variability and the flow presents a mainstream of routings. Silva *et al.* (2017) report that the General Kanban System (GKS) outperforms POLCA when the GFS also has a more frequent routing because POLCA increases the waiting time of a job to seize available cards GKS. Then, there is a gap in how other PCS can substitute POLCA as an alternative in General Flow Shop, maintaining the QRM principles. When other PCSs outperformed POLCA, it is also an opportunity to complement POLCA.

POLCA presents some enhancements to the original system to improve it for the QRM Cells. Fernandes and Carmo-Silva (2006) proposed that the system releases the job if all the POLCA cards are available for that job, more than using only the first loop of cards. Thürer et al. (2017a) add a release trigger to avoid starvation on idle stations for the order release mechanism. They also proposed a mechanism that uses a combination of the card allocation rule (before assigning POLCA cards) and the dispatching rule (after assigning cards). After those improvements, there is still a gap in complementing POLCA adapting other elements from other approaches or PCSs (RIEZEBOS, 2018). For that author, there is a particular focus for the order release mechanism for POLCA because the original POLCA of Suri (1998) uses a release date to authorise each station working on a job, and Kingsman (2000) highlighted the importance of the configuration of the order release mechanism to improve performance to a PCS, in the MTO environment.

Focusing on the recent improvements for the order release for POLCA, the authorisation lists can use updated release dates when the shop runs an Advanced Resource Planning (ARP), as Vandaele et al. (2008) presented. Thürer et al. (2019a) show that using a combination of pool sequence rules is better than using fixed release dates to authorise a station to work in a job. Then, Carmo-Silva et al. (2020) proposed a centralised job release to improve job delivery performance. In the same line of centralised release, Thürer, Fernandes and Stevenson (2020) couple the COBACABANA system for the order release of POLCA. Even these improvements, these centralised releases do not take advantage of the paired-cells loops characteristic on POLCA for the order release when the lead times are not easily upgradeable. Because real shops have implemented POLCA (KRISHNAMURTHY; SURI, 2009; SURI, 2018; VANDAELE et al., 2004), it shows how the paired-cell loop has been essential to improve performance. Then, a gap arises in using that paired-cell concept for an order release mechanism for POLCA.

At the best of literature reviewed, this concept has been used for CONWIP to improve its performance by the variant of using many loops, known as m-CONWIP (HUANG et al., 2015, 2016, 2017). The proposed concept is the Indirect Load POLC (IL-POLC). For this system, order release is centralised at an initial pool, with the difference to aggregate only the indirect load using the same paired-cells loops of POLCA cards. The order does not use the release date for the authorisation list in each station for the proposed system, as Suri (1998) proposed. Land (2006) reports that release data (or the time limit) has several detriments of the Delivery Time Mean (DTM) when the release date is fixed and processing times are very variables, as used in the GFS. Suri (2018) proposes the Release and Flow POLCA (RF-POLCA) when the release date is hard to estimate for stations, but he also gives several limitations of that system for the GFS. IL-POLC uses the indirect load condition to release the job by the central concept that Kanet (1988) proposed for the job shop. This research focuses only on card-based systems because those control systems require implementation of them. However, the card-based systems are practical solutions for limited budgets, commonly in small and medium enterprises in natural MTO environments.

The two main questions that drive this research are which PCS can substitute or complement POLCA in QRM cells environment? Moreover, how to use that paired-cell concept for an order release mechanism for POLCA to complement it? Explicitly using the IL-POLC proposed. Those are important issues to solve for POLCA practitioners since that PCS has reported good results (SURI, 2018). The practical contribution is how to use the indirect load condition for the order release task of POLCA.

1.2 OBJECTIVE

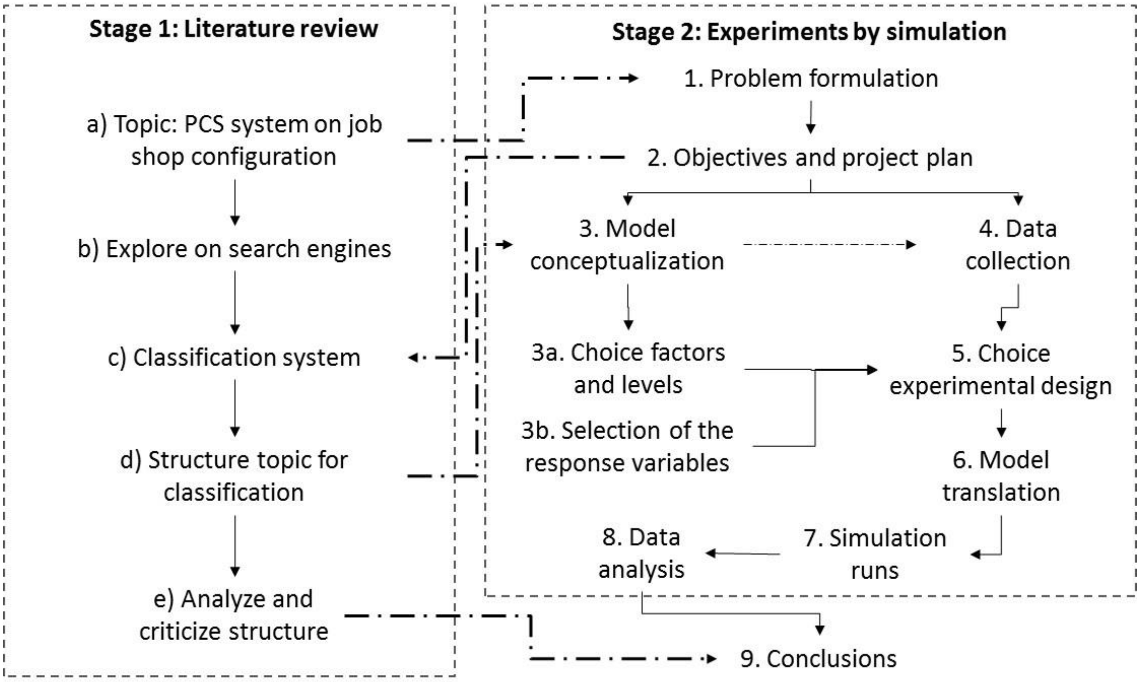
As POLCA is not the only PCS recommended to use in the general flow shop, there is a gap in which PCS can be used, considering the principles of the QRM approach. As POLCA also presented some limitations in the order release mechanism when the high variability on lead times is present, the second gap is how to use a complement from other PCS compatible with the QRM approach. For those gaps, the main objective of this research is to identify possible alternatives and possible complements of other PCSs for adapting to POLCA in the general flow shop. This research tests a proposed complement found in the PCSs identified and tests if some alternatives are suitable in terms of performance. The specific objectives for this research are:

1. Identify factors that allow comparing the performance of other PCSs to POLCA in the General Flow Shop context;
2. Propose the Indirect Load POLC (order release proposed for POLCA) and compare performance with original POLCA and RF-POLCA;
3. Test the proposed complement of POLCA comparing with other PCS proposed for a similar QRM cell environment.

1.3 RESEARCH METHOD

This research has two stages detailed in Figure 1.2: i) identification of factors of PCS for job shop configuration that evidence impact on performance and ii) proposing and testing the complements for POLCA with other PCSs. The first stage uses a systemic literature review to classify the factors studied in published comparisons and analyse previous PCSs results to achieve the first specific objective. This classification describes objectives reached by simulations reported in the literature. The second stage uses the simulation experiment to define the impact of the proposed complement (as intervention) in the general flow shop (as context) using the MTO performance measures (output). This research uses an experiment based on theoretical data to reduce the effect of a particular configuration from accurate data. This second stage uses two experiments: i) to compare and refine the complement proposed to the original POLCA and RF-POLCA to achieve the second specific objective, and ii) to compare this complement of POLCA with other suitable PCSs identified to accomplish the third specific objective. The first experiment (in chapter 4) focus on POLCA's factors to establish a parametrisation of the IL-POLC. That chapter targets the POLCA's practitioners that already had implemented one version of POLCA. The second experiment (in chapter 5) compares IL-POLC with other suitable PCSs found for the QRM cell environment: COBACABANA and COBA-POLCA. Chapter 5 targets how the paired-cell loop for order release performs against the centralised single station loops.

Figure 1.2 – Research method details



Source: Adapted from Denyer and Tranfield (2009), and Banks et al. (2005)

The literature review aims to identify PCSs that can control production flow between the QRM Cells. This research stage uses some steps proposed by Denyer and Tranfield (2009):

a) Topic selection and subject definition;
b) Explore publications and databases of publications;

c) Create an organization and classification of each publication;
d) Classify documents and make a structure for the topic;
e) Analyse and criticize obtained structure;
f) Show results and future research.

For the comparisons, this research uses simulation experiments that consider factors used for PCSs in the general flow shop to represent the QRM cells scenario. Data collection in this research uses a Discrete-Event Simulation (DES) because it is a valuable tool for comparing relations of a system in controlled scenarios (BERENDS; ROMME, 1999; GONZÁLEZ-R; FRAMINAN; PIERREVAL, 2012). This simulation approach is best suited to compare different systems' performance, as throughput time, lead time, inventory level, tardiness, among others (JEON; KIM, 2016). The second stage of the research use the Coleman and Montgomery (1993) steps and the guideline of simulation projects (BANKS et al., 2005), as suggested by some authors (JOHNSON et al., 2008; LAW, 2018).

Figure 1.2 describes the relations with a line-dotted arrow for each step between the two stages. Precedent tasks are in a continuous line. The topic of the Literature Review helps to define the problem approached by simulation. As the goal is to evaluate the complements and the PCSs differences in performance, papers' classification describes job shop configuration factors. The model conceptualization depends on the structure obtained from reviewed articles, as data used for simulation inputs. For model translation, Flexsim software is used. Experimental scenarios depend on the model conceptualization (as the shop's characteristics for comparison) and the structure obtained for the complements proposed. This structure considers the order as inputs, experimental factors for controllable and uncontrollable variables and the shop performance as output measures. Subsequently to the simulation runs, the analyses of simulation results are compared with literature review results for conclusions. For data analysing, this research uses the Analyse of Variance (ANOVA) (MONTGOMERY, 2009) and performance curves of results by its effectiveness in interpreting results (OLHAGER; PERSSON, 2008). As a simulation, this research considers some specific experiment design to compare the complements, as Kleijnen (2015) advises for a robust conclusion.

1.4 RESEARCH STRUCTURE

This document presents a structure by research articles. Some sections repeat information between the chapters. The structure is the following:

Chapter 1 – Introduction: This chapter shows the gap for this research, the contextualization for PCS, the research questions, selected research method and structure.

Chapter 2 – Production Control System (definition): This chapter presents some PCSs suggested for the GFS that could be used for QRM cells configuration; also, the chapter presents definitions and some parameters to operate each PCS.

Chapter 3 – Literature Review: The review proposes a comparison for PCSs suggested for QRM cells configuration. The chapter also includes the simulated shop characteristics in literature to describe the factors that influence performance. This chapter is already published (GÓMEZ PAREDES et al., 2020).

Chapter 4 – The Indirect Load POLC as a new production control system for the QRM: This chapter presents the proposed complement for POLCA: an order release based on the indirect load for the authorization element. The chapter shows a comparison of the IL-POLC with the original POLCA and a Release and Flow (RF-POLCA). The experiment considers some factors that influence how to set the IL-POLC for the GFS in the MTO environment.

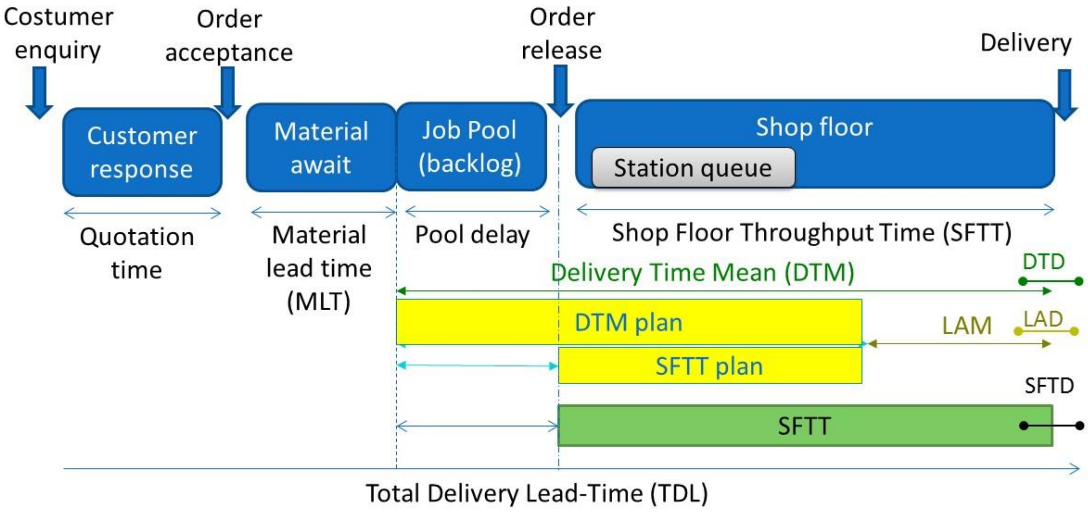
Chapter 5 – Indirect Load POLC (IL-POLC) or COBACABANA PCS for GFS? an assessment by simulation. This chapter compares the proposed Indirect Load POLC with the COBACABANA system and the combination of paired-cells control with centralized loops for the indirect load (COBA-POLCA). This chapter presents a simulation experiment to describe how these loops make a difference for performance in the studied context.

2 PRODUCTION CONTROL SYSTEM (PCS) FOR THE MAKE-TO-ORDER IN GENERAL FLOW SHOP

The Production Control System (PCS) is a “mechanism that regulates and controls the flow of materials along with the system of routes between the places where work is done on materials” (BURBIDGE, 1990). In the manufacturing control model proposed by Lödding (2011), the PCSs considers three tasks: a) release jobs to the shop, b) control the capacity across stations, and c) sequencing the jobs. According to how each PCS manages these tasks, the PCS influences the shop performance. This performance can rely on strategies adopted to increase productivity, reduce cost or reduce production lead time (KARRER, 2012).

According to Stevenson, Hendry, and Kingsman (2005), the requirements for the PCS in the Make-to-Order Industry (MTO) environment are the inclusion of the customer enquiry stage, consideration of the job entry and job release events for due date adherence, ability to manage a high variety of products, ability to manage variable job routings, and the applicability to Small and Medium Enterprises (SME). For the inclusion of the enquiry stage, the job entry and the job release, Kingsman (2000) present the main components of the Total Delivery Lead-time (TDL): i) Quotation time; ii) Material lead time (MLT), and iii) Delivery Time Mean (DTM). According to that author, controlling TDL by MLT and DTM gives the company the right image to the customers. As the company defines the promised delivery date after the client’s negotiation, the client does not perceive the quotation time in the TDL. As Porter et al. (1999) suggested, the PCS focus on controlling DTM and MLT since both are waiting time for the customer. As Hayes et al. (2005) described, the TDL is a remarkable performance for the MTO environment. The PCS estimates the DTM previous as a planned measure (DTM plan) by estimating the Shop Floor Throughput Time planned (SFTT plan). When a job has an additional waiting time in the station queue, the real Shop Floor Throughput Time (SFTT) becomes different from than SFTT plan, creating the Lateness Mean (LAM). As those differences can be randomized, the variation of lateness is measured by Lateness Deviation (LAD). That differences also generates the Shop Floor Throughput Deviation (SFTD) that also produces the Delivery Time Deviation (DTD). Figure 2.1 explains how the relationship of the cited measures with their respective events.

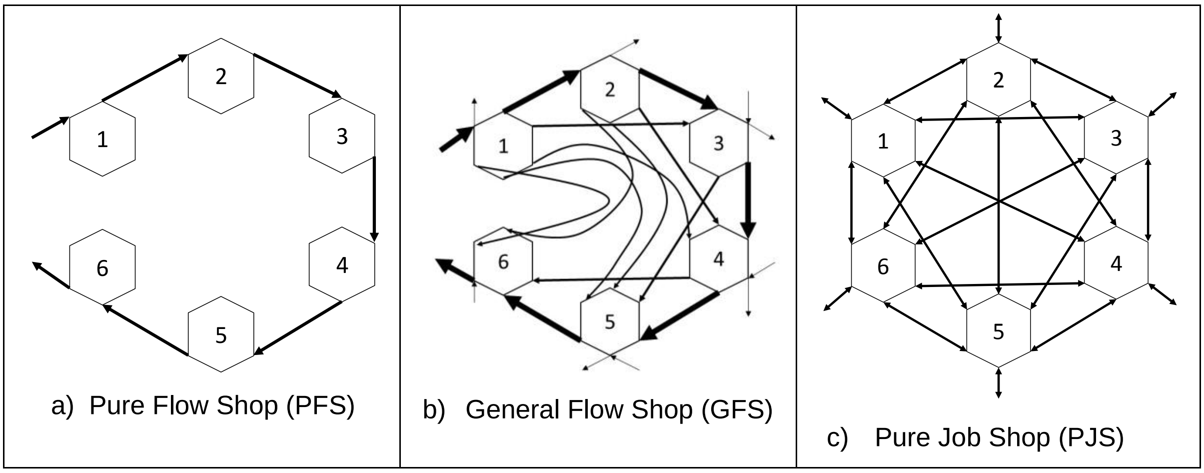
Figure 2.1 – Internal performance measure definition



Source: Adapted from Kingsman (2000) and Lödding (2013, cap. 2)

To understand the ability to manage a high variety of products and variable job routings, Stevenson, Hendry, and Kingsman (2005) defines the shop configuration. They also highlight that the shop configuration is the major determining factor of PCSs applicability. Figure 2.2 shows the most common definitions of shop configuration by type of flow, as Oosterman, Land and Gaalman (2000) described. The Pure Flow Shop (PFS) considers that jobs have the same sequence of stations (Figure 2.2 a). The General Flow Shop (GFS) has the same directions for stations, but the jobs may visit a subset of the stations (Figure 2.2 b). The Pure Job Shop (PJS) allows a variable routing length and random or undirected routing (Figure 2.2 c) that also describe the re-entrant flows for the stations.

Figure 2.2 – Shop configuration by routing products



a) Pure Flow Shop (PFS)

b) General Flow Shop (GFS)

c) Pure Job Shop (PJS)

Source: Adapted from Oosterman, Land and Gaalman (2000)

For the applicability to Small and Medium Enterprises (SME), Stevenson, Hendry and Kingsman (2005) explain that PCSs need to reach accessibility to companies with limited resources and constrained budgets. This factor is essential because the SMEs are usually MTO environment. Some PCSs are card-based because they use the principles of the original Kanban that Sugimori et al. (1977) presented: a visual card of production information that releases jobs by card availability, controls capacity by blocking the number of jobs in the cell and sequence the jobs by priority in a simple board.

Following the requirements presented for the PCSs in the MTO environment, Stevenson, Hendry and Kingsman (2005) present as suitable PCSs: Workload Control (WLC), Enterprise Resource Planning (ERP – defined as push production or MRP), and TOC (Theory of Constraints). According to these authors, the PCSs support and enhance those production controls. Thürer, Stevenson and Protzman (2016b) suggest POLCA for directed and straightforward routing (as GFS) and COBACABANA for high-variety routing as a PJS, as the card-based system of WLC. Conforming to Bertolini *et al.* (2017), DBR is convenient in multiple flows with a stable and identifiable bottleneck. POLCA is suitable in various flows with stable routings, but shifting bottleneck and WLC is proper for changeable routings and a random product mix. Table 2.1 summarises the PCSs chosen for this research due to their application in a job shop. The following sections describe the PCSs presented in that table.

Table 2.1 – Framework of PCS suitable by flow and routing

PCS	Flow characteristic	Routing variety
POLCA (QRM)	Mixed streams and high bottleneck shifting	Product mix as family routing stable
CONWIP	A single loop that contains all considered routings	Linear independent value stream suggested
COBACABANA (WLC)	Mixed streams and high bottleneck shifting	Unstable family and random routing
DBR (TOC)	Mixed streams	Stable bottleneck, product mix independent (or breakdowns)

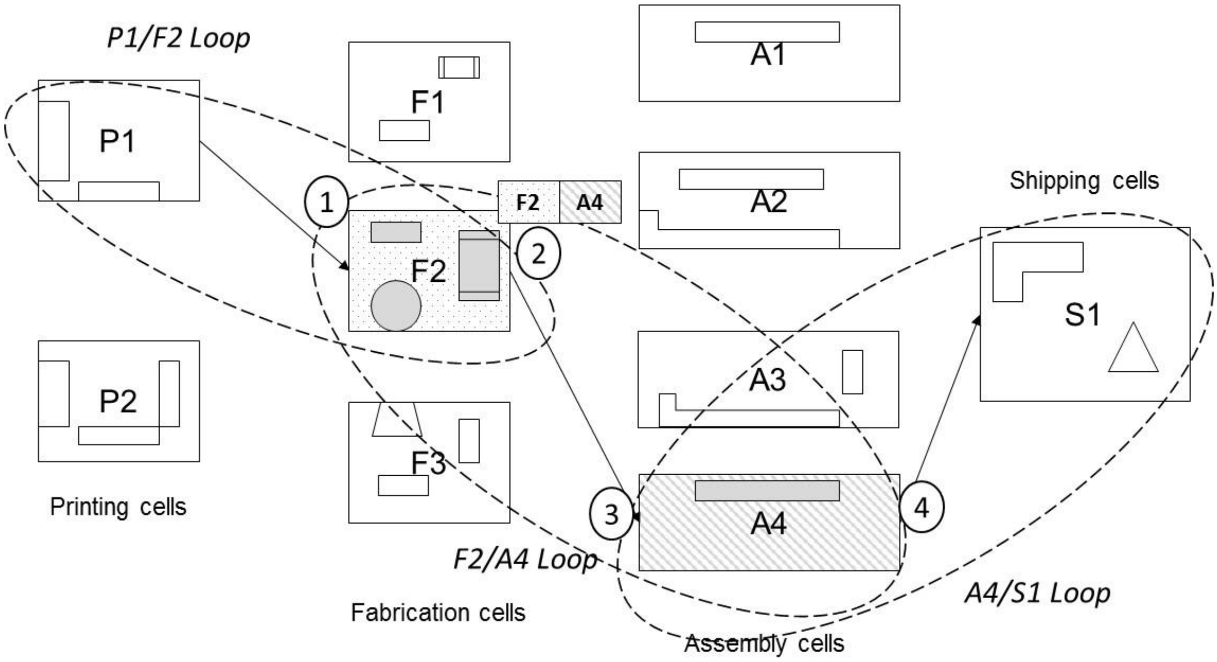
Source: Adapted from literature (BERTOLINI; ROMAGNOLI; ZAMMORI, 2017; STEVENSON; HENDRY; KINGSMAN, 2005; THÜRER; STEVENSON; PROTZMAN, 2016b)

2.1 POLCA

The Paired-cell Overlapping Loops of Cards with Authorization (POLCA) was defined as the Quick Response Manufacturing (QRM) PCS tool for implementing the hybrid concept for pull and push production to reduce the lead time (SURI, 1998). This PCS has been designed for cells configuration considering flow between cells for High-Variety, Low Volume and Custom (HVLVC) production environment (PIEFFERS, 2005). POLCA refers to three main concepts of control: a) Paired-cells: using two cells linked by flow material; b) Overlapping loops of cards: excepting the first and last cell of the flow, a cell has at least one upstream loop and one downstream loop; c) Authorization: a release date for each cell estimated by a High-Level Material Requirement Planning (HL/MRP), material availability for that order and downstream card available.

Figure 2.3 represents how the flow for a POLCA card is in a specific loop. For a particular order, the defined routing is for cells P1 to F2, then to A4 and ends on S1. The order will proceed through the card loops with the pair of cells: P1/F2, F2/A4 and A4/S1. For the specific loop F2/A4, the POLCA system works as follows (steps are marked inside Figure 2.3): the order arrives at the cell F2 (1). An operator checks the HL/MRP for authorisation, material availability on F2, and a POLCA card (in the example: an F2/A4) for the evaluated order. If the three conditions are checked, the order goes through the F2 cell; otherwise, the order waits until the three conditions are fulfilled. The card F2/A4 is attached to the order, and both are sent to cell A4 (2). In the A4, the order is checked for the three conditions but with the destination loop for A4/S1 (3). As the three conditions are satisfied, the order goes through A4. Until the A4 finishes the order, the planner releases the card F2/A4 and sends it to F2. This sending also releases capacity for the loop F2/A4 in F2 cell (4). The three conditions may stop an order: i) for a release authorisation that avoids earliness WIP; ii) for material availability, and iii) for capacity in the next loop of job routing. This last condition allows processing other orders in F2 for other loops.

Figure 2.3 – POLCA card flow for one loop



Source: Adapted from Suri (1998) and Suri and Krishnamurthy (2003)

The POLCA system parameters are (KRISHNAMURTHY; SURI, 2009): a) the definition of the loops, b) the estimation of release authorisations, c) the quantum of work that a POLCA card represents, d) the card design and procedure documentation, and e) the number of cards of each loop. The loop definition depends entirely on all possible routings between the considered cells for control. The POLCA loops design tries to balance workload at later stages rather than balancing workload at the release moment (ZIENGS; RIEZEBOS; GERMS, 2012). This balancing is possible when limiting WIP in two consecutive stages decreases. For those authors, a reduction in throughput time is most considerable when the first stage does not limit the WIP. A difference for this parameter is how to group the station as cells, as shown in Suri (2018, p. 252).

The estimation of release authorisation data uses the order due dates and planned lead times for each cell to develop a list of authorised orders. The original POLCA uses the HL/MRP, considering that the release date is for the entire cell and not for each machine (this concept evolved on the Demand-Driven MRP (MICLO et al., 2018; PTAK; SMITH, 2016)). There are some variations on this parameter. The Generic POLCA - G-POLCA releases the order if all the loops have the card available for that order (FERNANDES; CARMO-SILVA, 2006). POLCA Starvation Avoidance (POLCA-SA) authorises an order releasing to a starving cell even if the limit load is exceeded (THÜRER et al., 2017a). POLC-A is to use the card-based element with a dispatching floor because the authorisation element reduces performance (THÜRER et al., 2019a); and Release-and-Flow POLCA (RF-POLCA) that authorisation for the first cell of order routing implies the authorisation for the whole order routing (SURI, 2018, p. 94).

The quantum of work that a POLCA card represents has different definitions in the literature. Suri (1998) suggests an order for a quantum unit after estimating a batch size that minimizes the lead time or a standardized workload estimation. For Vandaele *et al*. (2008), quantum is calculated on the amount of workload for each order, defining the Load-Based POLCA (LB-POLCA). The design of cards and procedure documentation controls how the POLCA cards circulate on loops and how the system is executed on the shop floor. Even the original system uses paper-based cards; an electronic version has also been applied as E-POLCA (VANDAELE et al., 2004).

The estimation of the number of cards has different views. Suri (1998) uses a simple formula based on Little’s Law. For this system, the number of cards on the system may influence lead time performance (RIEZEBOS, 2006). Vandaele *et al*. (2008) use the accumulated load in the loop to estimate the number of quanta allowed, as the number of cards. Riezebos (2010) modifies that simple formula adding the queue effect inside the loop. Luan, Jia, and Kong (2013) propose a mathematical model to minimize the number of cards in circulation constrained to the flow required on the shop floor. Thürer, Fernandes, Carmo-Silva, *et al*. (2017a) found a significant impact on the number of cards for tested flows.

Besides, POLCA presents some empirical applications in the literature. The POLCA requires four phases for implementing (SURI; KRISHNAMURTHY, 2003): (i) pre-POLCA assessments; (ii) design of POLCA system; (iii) launch of POLCA implementation and; (iv) post-implementation. The application of POLCA helps to improve a lead time reduction in high-variety manufacturing (KRISHNAMURTHY; SURI, 2009). In POLCA implementation, Riezebos (2010) proposes a quick scan to identify how this system matches as a solution. The author also presents an application on a small enterprise, a unit of analysis with high variation product and routing, reducing lead-time. POLCA has been used for controlling flow between different entities in a supply chain (SEVERINO; GODINHO FILHO, 2019). Further reading for POLCA implementation and modifications are on Chinet and Godinho Filho (2014) and Suri (2018).

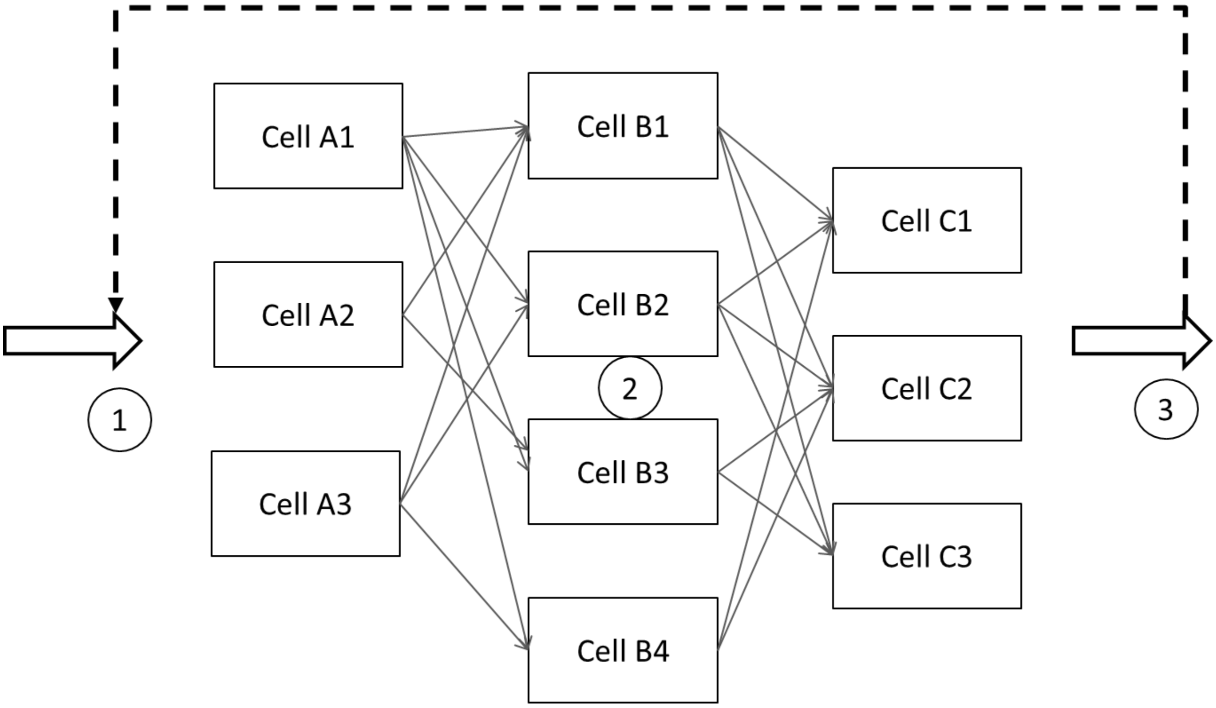
2.2 CONWIP

The system Constant Work-in-Progress (CONWIP) was developed as an alternative for Kanban in a flow shop with high-variety products (SPEARMAN; WOODRUFF; HOPP, 1990). The system merges the pull with the push production, intending to control many different parts. The pull phase limits the amount of WIP between input and output; for the push phase, flow is free inside the input and output. Hopp and Spearman (2004) initially defined CONWIP as a total pull control system because it was referred to as the work-in-progress limit. In its original design, two assumptions are implicit for the use of the CONWIP system: i) the production line consists of a single routing line, to which all parts flow into; ii) WIP can be reasonably measured in units because the jobs are similar. Thus, some authors have suggested CONWIP as one of the best production control strategy (HOPP; SPEARMAN; WOODRUFF, 1990) (BRIEN; JAFARI; WEN, 2006).

CONWIP is a suitable alternative to enhance lean principle implementation in a job shop environment because this PCS considers a high-variety, even routing (SLOMP; BOKHORST; GERMS, 2009). Some modifications reported in the literature for job shop flow are the multiple-card counting in the flow (FRAMINAN; RUIZ-USANO; LEISTEN, 2000), as a closed network queuing for controlling WIP mix (RYAN; FRED CHOOBINEH, 2003), among others. CONWIP has been examined in a nested control system for inter-cells flow control (used as push control) and intra-cell control (used as pull control). The system has reported good results for reducing throughput time inter cells (LI, 2010). Besides, Korugan and Gupta (2014) present an adaptive CONWIP system that uses a queuing network to represent the cellular flow. Then, CONWIP can control the flow within the QRM cells and between cells since the loop can control a line with two stations.

Kabadurmus (2009) describes how CONWIP controls the inter-cellular flow For job shop flows, as presented in Figure 2.4. The following steps represent how the whole loop works in the CONWIP system. The order arrives at the system, and the planner checks if a card is available to release (1). As the order receives the card, it goes through the shop following its routing (2). CONWIP considers all possible routings since they are inside the loop, including re-entrant flows. When the order finishes its routing, the planner withdraws the attached card and release capacity to other arrivals (3).

Figure 2.4 – CONWIP system control in a job shop



Source: Kabadurmus (2009)

According to Framinan, González and Ruiz-Usano (2003), the decisions to make for establishing the CONWIP system are the followings:

a) Production quota (or production rate);
b) The WIP limit for the loop;
c) Capacity Shortage Trigger;
d) Backlog list forecasting (including for MTO systems and also release method);
e) Number of cards in the system;
f) Jobs sequencing rules.

The production quota is established as the target production quantity (SPEARMAN; WOODRUFF; HOPP, 1990), also known as the effective demand for the loop. The production quota is considered a strategic decision for the demand rate. For those authors, the WIP limit is the maximum amount of work ahead in the period. Hopp and Roof (1998) propose a confidence interval for the WIP limit using a steady-state simulation. This maximum amount of work depends on the released workload. The capacity shortage trigger is a function of the actual production up to some time *t* that indicates if additional capacity must be used. This function depends on the production quota and the current state in order to unblock the line.

For Framinan *et al*. (2003), the forecasting backlog list contains the number of orders that managers can anticipate fulfilling demand. According to these authors, forecasting the backlog is suitable for MTS because of backlog results from Master Planning Scheduling (or similar techniques). Furthermore, for the Make-to-order environment, this issue is suitable to sequence jobs. Therefore, this research uses the sequence of job decision since it is proper for the job shop environment.

CONWIP presents different ways to calculate the number of cards. Herer and Masin (1997) propose a mathematical model to estimate the number of CONWIP cards considering the backlog order problem. In addition, CONWIP may use a Lagrangian relaxation of stations capacity constraint to adapt the shop’s general limitations to control inventory on a job shop (BRIEN; JAFARI; WEN, 2006). Dalalah and Al-Araidah (2010) present a non-linear programming model to optimize CONWIP settings for online balancing work in the queue, mainly on the number of cards used. Braglia *et al*. (2011) reveal an algorithm to estimate card number considering bottleneck effects for optimizing performance system, maximizing throughput and minimizing WIP level. Besides, linear programming is presented to optimize inventory levels on system performance (HELBER; SCHIMMELPFENG; STOLLETZ, 2011). González-R, Framinan, and Ruiz-Usano (2011) use the response surface method to establish the number of cards. Belisário and Pierreval (2015) present a genetic algorithm for estimating how many cards are necessary for the CONWIP system. There are more details for estimating the number of cards on Framinan *et al*. (2003), and Prakash and Chin (2014). The CONWIP card could represent the order workload as a solution for high processing-time variation (QI; SIVAKUMAR; GERSHWIN, 2009; THÜRER et al., 2019b), called as Load-Based CONWIP (LB-CONWIP). In that version, the workload norm substitutes the number of cards.

The first considered rule for defining a job sequence that Spearman *et al*. (1990) established was First-In-First-Out (FIFO). However, there are some rules tested for sequencing as Short Processing Time (SPT), Shortest Remaining Processing Time (SRPT), Operation Due Date (ODD), Shortest Total Work Content (STWK), among other classical sequence rules

(PRAKASH; CHIN, 2014; THÜRER et al., 2017b). One paper proposes that the capacity slack in the number of jobs contemplated in the direct load has better performance in the random routing job shop (THÜRER et al., 2017b). For the LB-CONWIP version, the Capacity Slack rule (CS) has reported better results than the other rules (THÜRER et al., 2019b). Those authors suggest the shop load as the accounting workload approach for that results.

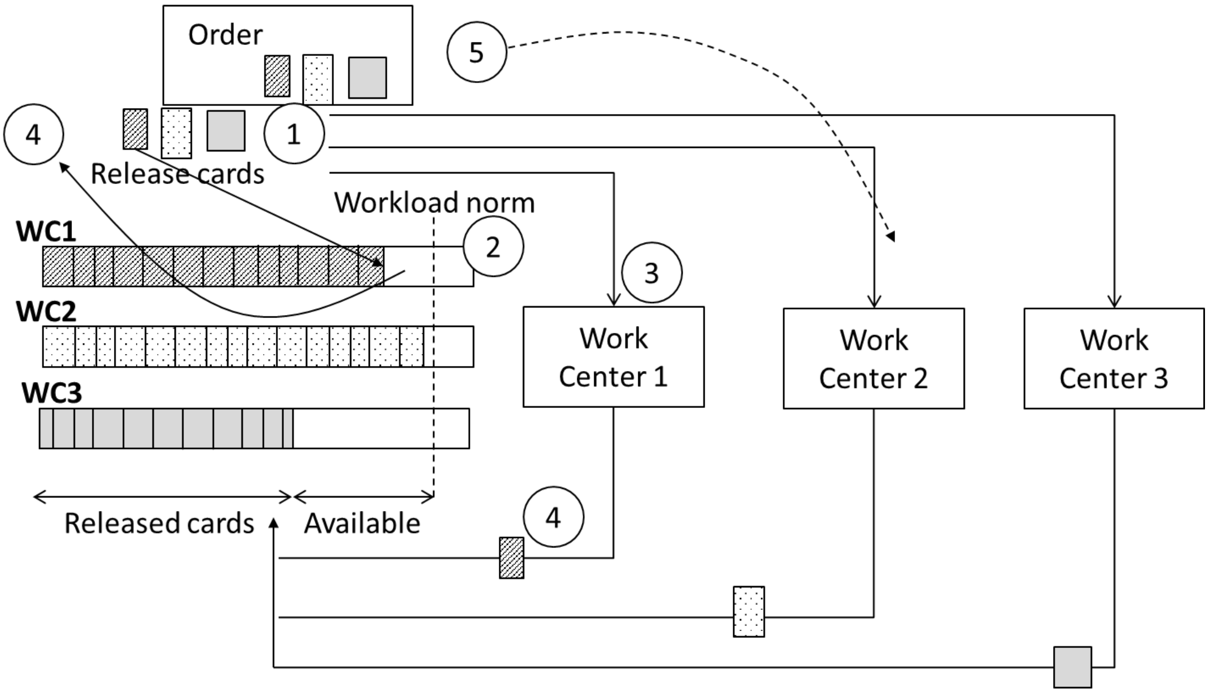
The CONWIP has several implementations, as listed in Jaegler *et al.* (2018). Those authors found that even this system has been widely spread in MTS, further research is MTO environment with job shop environments since the system has several modifications presented in Prakash and Chin (2014).

2.3 COBACABANA

Control of Balance by Card-Based Navigation (COBACABANA) was proposed as a production control system for a complex job shop using Workload Control (WLC) principles (LAND, 2009). It transforms the WLC concepts to use with simple cards as visual management. This card-based system can use different cards to represent time processing variety in assigned load to each station (THÜRER; LAND; STEVENSON, 2014). This system can also work with two kinds of cards to estimate due dates for an order, depending on current workload: a) customer enquiry management and b) order release control (THÜRER; STEVENSON; PROTZMAN, 2016b).

COBACABANA uses card loops between the central release function that precedes the shop floor and each work centre (LAND, 2009). The card represents the amount of direct load of each job on the respective station. The direct load is the amount of load of the order released currently waiting in the station’s queue, and the indirect load is the load for the downstream station, according to the order’s routing. The aggregate load is the direct load plus the indirect load. The system uses a central board to summarize the direct load released for all the shop floor stations. Each station has one loop to the central release (board), allowing any routing for orders, as the Pure Job Shop. The Workload norm limits the aggregated load for each station. Figure 2.5 shows the shop floor loop and the system’s main steps for the system, as appointed in the literature (THÜRER; LAND; STEVENSON, 2014; THÜRER; STEVENSON; PROTZMAN, 2016a). The planner chooses the card size according to the amount of planned work of the station's job, based on processing time. To release the order, the planner attaches the right number of operation cards of each station used to the order, according to the job routing (1). The planner verifies the workload status, and he can only release the order if the aggregate workload does not exceed the Workload norm (2). Subsequently, the planner releases the order to the routed station, and the station processes the assigned operation for that order (3). As a work centre completes its corresponding operation, the operation card returns to the central planner, and the planner withdraws the release card of that operation from the central board (4). Then, the planner executes steps two to four until all operations are complete (5). For the example showed in Figure 2.5, the planner will not release an order that contains work centre 2 in its routing because that station has reached its workload norm.

Figure 2.5 – COBACABANA card loops description



Source: Thürer *et al*. (2014)

According to Land (2006), the critical factors for implementing workload parameters are: a) workload calculation method, b) workload norms, c) planned throughput times, d) release period length and e) time limit. The two last items belong to the Order Release strategies. For the cited author, as the planned throughput times have a limited impact on configuring the Workload control parameters, the other four are briefly described for the COBACABANA system.

The workload calculation method refers to the level of aggregation of workload measure in the dimensions used by Bergamaschi *et al*. (1997). Stevenson and Hendry (2006) separate how this level modifies the planning decision of release based on the indirect load (or load in transit). The options cited by these authors are a) Upstream, downstream and load on hand; b) Corrected load method; c) Upstream and load on hand; d) load-oriented manufacturing control; and e) Bottleneck load-oriented release, including LUMS approach. For COBACABANA to take advantage of a standardized workload norm, Thürer, Land, and Stevenson (2014) suggest the corrected load method to perform better. However, those authors also suggest that a fixed size of three or five is right enough as a proper workload amount to use COBACABANA effectively.

Some authors have studied the effects of the Workload norm. For Cigolini and Portioli-Staudacher (2002), the robust norm for different performance is workload balancing. However, this norm does not have a significant difference in some performance measures. Other norms depend on those authors' shop configuration, and Fredendall, Ojha and Wayne Patterson (2010) confirmed it. These authors also found that the aggregation method used for buffer limit (norm) significantly influences the performance, even in bottleneck presence. Thürer, Silva, and Stevenson (2011) confirm that the workload norm depends on the workload aggregation method. The corrected aggregate load allows one workload norm for all work centres. If the classical aggregate load approach is used, each station's workload norm needs modification because the station's indirect load influences the norm. For unbalanced shops, no single workload norm strategy performs best for measures delivery time, throughput time, percentage of tardy jobs and standard deviation of job lateness (FERNANDES; LAND; CARMO-SILVA, 2014). For those unbalanced shops, the machine interchangeability has a significant influence on the routing decision.

There are several rules for Order Release strategies, as appointed by Bergamaschi *et al*. (1997). Two order release strategies are highlighted for COBACABANA for the release period length: LUMS COR (Lancaster University Management School Corrected Order Release) and Continuous. The LUMS COR uses a corrected form of the aggregate load for downstream load. The aggregated load is divided by the station position in downstream, after release station (OOSTERMAN; LAND; GAALMAN, 2000). This strategy uses a periodic release, considering the workload norm, and a conditioned continuous release. This conditioned release is the Starvation Avoidance (SA) that injects a job onto the shop floor, even exceeding the workload norm if a station is starving (load amount becomes zero to that station) (THÜRER et al., 2012). As addressed by those authors, it has been considered as one of the best solutions for release orders for Pure Job Shop and General Flow Shop. The Continuous release has similar results for other workload strategies (FERNANDES; CARMO-SILVA, 2011). The continuous release intends to shorten delivery times by monitoring continuously a release for order and not for periodic time releases. For Thürer *et al*. (2014), continuous-release depends on the workload norm levels to be useful, but new approaches are better than classical approaches for pure job shop. Other parts of the release strategy, as the sequence rule, have a substantial impact on the performance, according to Thürer *et al*. (2015a).

The time limit is the time horizon considered for release to limit and choose jobs with a start date in that period (WIENDAHL, 1995). According to Land (2006), the time limit may improve timing performance to restrict the set of jobs considered for release to the more urgent ones. For this author, using the time limit reduces opportunities for balancing workload in the shop. This time limit has been considered infinite for continued release (FERNANDES; CARMO-SILVA, 2011) and unbalanced job shops (FERNANDES; LAND; CARMO-SILVA, 2014). There is no real definition of time limit standards because they must be estimated in the studied job conditions.

Thürer, Stevenson and Protzman (2016b) suggest COBACABANA for PJS and GFS (the QRM cells environment). This system also can work as the order release of POLCA, as shown in COBA-POLCA (THÜRER; FERNANDES; STEVENSON, 2020). As COBACABANA reports at least one case of application (BRAGLIA; MARRAZZINI; PADELLINI, 2020), this system is suitable for the QRM cells environment.

2.4 DRUM-BUFFER-ROPE (DBR)

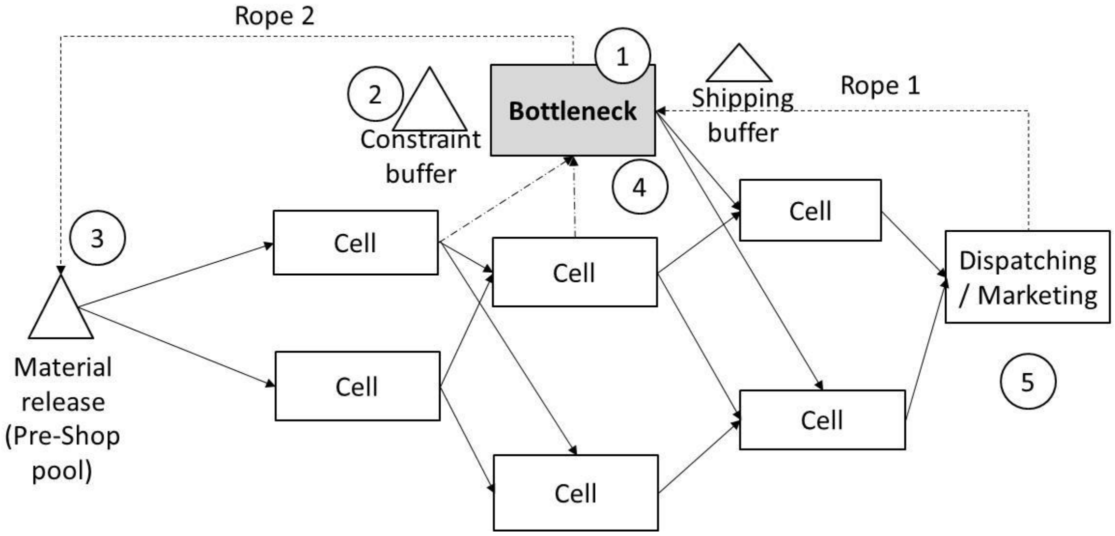
The Drum-Buffer-Rope system (DBR) is the production control system of the Theory of Constraints (TOC) proposed by Goldtratt and Cox (1984). The system uses a characteristic of Optimized Production Technology (OPT) scheduling (RONEN; STARR, 1990). For scheduling DBR, three steps are suggested (SCHRAGENHEIM; RONEN, 1990): a) schedule the identified constraint, exploiting it with most profitable products; b) determine the buffer sizes, defining buffer as the time for bottleneck working; c) derive the materials release schedule according to steps (a) and (b).

DBR has recognized improvements in high-variety environments (WU; MORRIS; GORDON, 1994). As the DBR method, Bottleneck input control is useful in a high complex routing, as typical in a job shop (ENNS; COSTA, 2002). In a general way, Chakravorty (2001) suggests DBR for job shops, especially for directed flows. Riezebos, Korte, and Land (2003) combine DBR with the workload concepts to add the capability to detect bottlenecks.

DBR has been implemented in a job shop environment, resulting in significant improvements (DARLINGTON et al., 2015). These authors suggest DBR for the job shop environment (defined as jobbing flow) because of its results. Considering the comparison of DBR with Workload order release (THÜRER et al., 2017c), DBR outperforms WLC when there is a strong bottleneck in the shop.

DBR uses the five steps of TOC for scheduling the constraint. As presented in Figure 2.6, these five steps are (CHAKRAVORTY; ATWATER, 2005): (1) Identify the bottleneck (system constraints). Then, (2) exploit the constraint (rope 1). This step has the following procedures: preparing the bottleneck (drum) schedule and determining the constraint buffer in time, aiming for the bottleneck to keep working (BHARDWAJ; GUPTA; KANDA, 2010). After that, (3) subordinate all other decisions to step (2). According to the last authors, it implies determining the material release schedule into the system (rope 2). Also, it needs to ensure that the bottleneck works strictly to the prepared schedule. The non-bottleneck stations work in FIFO scheduling. The next step (4) is elevating the constraint. This elevation requires estimating the bottleneck’s protective capacity to smooth marketing variations (in rope 1). As the last step, if the constraints have broken, go back to step (1), but do not let inertia become the system constraint.

Figure 2.6 – DBR mechanism in the job shop



Source: Adapted from Watson, Blackstone, and Gardine (2007), Chakravorty and Atwater (2005), and Thürer *et al*. (2017c).

As appointed by Schragenheim and Ronen (1990), the main parameters for the design of a DBR system are a) the scheduling rule or method for bottleneck; b) the size of the buffers on the bottleneck, even for constraint – input or shipping – output; and c) the scheduling or release of free products (those that do not pass through bottleneck). The single rule suggested for scheduling is a profit ratio using contribution margin over processing time on the bottleneck, in descending order until reach the active constraint, whether marketing or capacity constraint (FREDENDALL; LEA, 1997). For Gonzalez-R, Framinan, and Ruiz-Usano (2010), the best dispatching rules for DBR in a general flow shop are FIFO, SI (Shortest Imminent processing time), EDD (Earliest Due Date) and NSUT (No setup time).

For Schragenheim and Ronen (1990), the constraint buffer size is determined by initial simulation runs to protect the lead time upstream the constraint of undesirable variations on the system. They also explain a standard rule for the buffer size: three times the bottleneck's lead-time, depending on shop conditions. This rule has been used for applications of DBR in job shop environments as a finite forward downstream (CHAKRAVORTY; ATWATER, 2005; DARLINGTON et al., 2015; GOLMOHAMMADI, 2015). However, this buffer size significantly affects shop performance (CHAKRAVORTY, 2001; FREDENDALL; OJHA; WAYNE PATTERSON, 2010; THÜRER et al., 2017c).

After the bottleneck, the shipping buffer (or space buffer) protects the variation of constraint downstream until dispatching (or until assembly operation). This estimation depends on process variation and the due date estimation as a direct function of the two factors (DARLINGTON et al., 2015; SIMONS; SIMPSON, 1997). The scheduling of free products strongly impacts shop performance (CHAKRAVORTY; ATWATER, 2005).

Related literature gives only some general indications for free products scheduling (GOLMOHAMMADI, 2015). This scheduling's impact is more potent for this author when setup times are not negligible, and the sequence of operations is complex. For example, in a complex job-shop system, multiple setups are preferred to maximize throughput gain with small batches opposite to order preemption (favouring non-free products) because the batch size significantly impacts performance measures (GOLMOHAMMADI, 2015).

2.5 OTHERS

There are other PCS suggested for high-variety and job shop environment as the Synchro MRP and DEWIP. The Synchro MRP is defined as hybrid pull/push production and is designed for product variants and high setup times (BERTOLINI et al., 2013). That author suggests Synchro MRP reduces change-over times, and when the shop has an identifiable mainstream flow.

Decentralised Work-in-Progress (DEWIP) is a station-based control using workload concepts created for random routed job shop (LÖDDING; YU; WIENDAHL, 2003). This PCS controls a workload limit in each station on the shop floor. As described in Lödding (2013, cap. 24), this mechanism is the same mechanism to control stations' workload, as used in the COBACABANA system. The main difference is that DEWIP does not use a central board for the planner to control the shop floor. Other parameters, such as WIP limit (norm) and indirect load aggregation, are the same as COBACABANA. Other PCSs developed in recent years are available in the publication of Bagni et al. (2020).

2.6 FINAL CONSIDERATIONS

The potential PCSs identified as suitable for the QRM Cells environment are POLCA, CONWIP, COBACABANA and DBR. Even they have been created as a solution for different shop configuration, they have enough concepts to be implanted as production control for job shop with predefined routings. This research focuses on the cited PCSs only because other variants of PCSs use them as a base.

3 FACTORS FOR CHOOSING PRODUCTION CONTROL SYSTEMS IN MAKE-TO-ORDER SHOPS: A SYSTEMATIC LITERATURE REVIEW

Production control systems (PCSs) control the flow of jobs in a production system. The selection of a suitable PCS in the context of make-to-order (MTO) is challenging, due to the characteristics of MTO businesses and the number of parameters or factors that comprise a PCS. The literature that compares PCSs in the MTO context reported contradictory results. In fact, there is a gap in the literature concerning which factors or parameters explain a PCS performance. This paper presents an analysis of comparative studies on PCS in the MTO context, using a systematic literature review, to reveal which control factors and manufacturing conditions influence a PCS performance. The analysis concentrates on studies that use simulation to assess the performance of PCSs. Our results indicate that the main difference in PCSs performance is the design of the control loops. Other important factors that must be considered in the choice of a PCS are the order release mechanism, the workload aggregation approach, and the workload estimation method used on control loops. A framework for choosing a suitable PCS for MTO companies is presented, considering these factors.

Keywords: Production control; flow shop; job shop; systematic literature review; make-to-order.

3.1 INTRODUCTION

Production Control Systems (PCSs) control the flow of jobs in a production system (GRAVES; KONOPKA; JOHN MILNE, 1995). Consequently, a broad set of PCSs has emerged. According to Stevenson, Hendry and Kingsman (2005), existing PCSs require some adaptations for the make-to-order (MTO) sector, namely: a) inclusion of the job entry and the job release stages, b) ability to deal with non-repeat production (customised products), c) ability to deal with variable job routings and d) applicability to small and medium enterprises (SME). Those authors identified the following PCSs as being suitable for MTO sector: Workload Control (WLC), Constant Work-in-Process (CONWIP), Enterprise Resource Planning (ERP), Paired-cell Overlapping Loops of Cards with Authorisation (POLCA), and the Theory of Constraints (TOC). While Drum-Buffer-Rope (DBR) was proposed by Goldratt and Cox (1984) in the context TOC, POLCA was proposed by Suri (1998) in the context of Quick Response

Manufacturing (QRM). In the context of WLC, Lödding, Yu, and Wiendahl (2003) proposed Decentralized Work-in-Process (DEWIP), and Land (2009) proposed the Control of Balance by Card-Based Navigation (COBACABANA) to use that approach.

Although there are several studies comparing different PCSs (FERNANDES et al., 2017a; NIEHUES et al., 2016; THÜRER et al., 2017c), the findings of these studies are often inconclusive and sometimes even contradictory. Examples of contradicting conclusions found in the literature include the following: POLCA is better than CONWIP (BARROS et al., 2016), CONWIP is better than POLCA (FRAZEE; STANDRIDGE, 2016), the single loop is better than many loops (IP et al., 2007), and many loops for CONWIP is better than a single loop (HUANG et al., 2017). There are also several reviews of PCSs (BAGNI et al., 2020; FERNANDES; GODINHO FILHO, 2011; GERAGHTY; HEAVEY, 2005; SATO; KHOJASTEH-GHAMARI, 2012). However, these reviews lack utility for the MTO sector, because they focus on a single PCS and its modifications (CHINET; GODINHO FILHO, 2014; JAEGLER et al., 2018; PRAKASH; CHIN, 2014) and do not include performance comparisons between PCSs discussed or lack an explanation of the relationships between comparison conditions (BERTOLINI; ROMAGNOLI; ZAMMORI, 2017; GONZÁLEZ-R; FRAMINAN; PIERREVAL, 2012; THÜRER; STEVENSON; PROTZMAN, 2016b)

Apart from contradicting results, the large number of parameters of a PCS and the required shop floor configuration makes the task of selecting a suitable PCS difficult (KARRER, 2012). This difficulty has led to the development of agents to adapt PCSs using reinforcement learning (KUHNLE et al., 2020). However, this development is still at the initial stage, and the challenge is still unsolved for situations where parameters interact between themselves, as typically for most PCS. In these situations, it is important to implement existing knowledge on parameter and system interactions into the agent system. It is essential to avoid that the PCS deteriorate the shop performance by an unknown interaction of parameters or that managers misunderstand how the PCS can help or hinder the realisation of desired performance. This research gap, which is addressed in our study, has also become essential to develop cyber-physical systems in complex flows, as recently pointed out by Huang, Chen and Khojasteh (2020).

In response to the challenges that MTO organisations face in choosing a PCS, this study provides a systematic review of the literature on simulation studies. The focus of this study is to explore the impact of factors on shop performance and to clarify some potential effects of the selection. The findings of this study are summarised in a framework to help production managers to identify the main factors and their interactions. This paper proposes a framework using the attributes of the MTO sector to helps managers to choose a PCS for that environment. Also, some research gaps are identified, which provides important venues for future research.

The remainder of this article is structured as follows. Section 3.2 details the research method used, and Section 3.3 briefly describes the PCSs studied and their mechanisms. Section 3.4 then presents the results of the review, and Section 5 presents an extended discussion of the results, including performance comparison. Finally, Section 3.5 summarises the main conclusions and contributions of the article.

3.2 RESEARCH METHOD

This study started by asking:

RQ1: Which job and shop characteristics were used to model and compare the PCSs in simulation studies?

RQ2: What are the relationships between the conditions used for comparing the PCSs? These conditions include the experimental factors, the shop configurations, and the performance measures used in the experiments.

RQ3: What are the main results and conclusions of these studies?

This paper conducts a systematic literature review of studies that assess PCS performance to answer these research questions. This literature review follows the three-stages process suggested by Tranfield, Denyer and Smart (2003): i) plan the review, ii) conduct the review; and iii) report the results. The first two stages are detailed in the subsections below. Results are presented in Section 4. However, first we present in Table 1, a list of acronyms and terminology used in this paper.

Table 3.1 - List of acronyms used in this chapter

Acronyms	Definition
ATKS	Adapted Toyota Kanban System
COBACABANA	Control of Balance by Card-Based Navigation
CONWIP	Constant Work-in-Process
DBR	Drum-Buffer-Rope
DEWIP	Decentralised Work-in-Process
DRC	Dual-resource constraint
ERP	Enterprise Resource Planning
GFS	General flow shop
GJS	General job shop
GKS	General Kanban system
HPP	Hybrid push-pull
LB	Load-based
MRP II	Manufacturing Requirement Planning System
MTBF	Mean-time between failures
MTO	Make-to-order
MTTR	Mean-time to repair
OB	Order based
PCS	Production Control System
PFS	Pure flow shop
PJS	Pure job shop
POLC	Paired-cell Overlapping Loops of Cards
POLCA	Paired-cell Overlapping Loops of Cards with Authorisation
SME	Small and medium enterprises
SR	Shift Release
TOC	Theory of Constraints
WIP	Work-in-progress
WIPCtrl	Work-in-process load Control
WLC	Workload Control

3.2.1 Stage 1: Planning

The scientific databases used to search for articles were Scopus, Web of Science and Compendex. The criterion for search engine selection was the degree of the match to sample references found in previous literature reviews of the selected PCS (FRAMINAN; GONZÁLEZ; RUIZ-USANO, 2003; SEVERINO; GODINHO FILHO, 2019) that were related to the comparison of the PCSs. For the initial phase, this study selected the PCSs suggested by Stevenson, Hendry and Kingsman (2005) for MTO engaged in non-repeatable production. According to the initial results some terms were not included in the string. For example, WLC has a broader definition than a PCS, and it is considered within COBACABANA and DEWIP; ERP refers more to a software system than to a Production Control System (POWELL, 2013). The search string did not include the Kanban system because the original Kanban was not suitable for the studied environment. This study does not exclude variations of the Kanban system when those variations are studied with the selected PCSs. Lage Jr. and Godinho Filho (2010) present a list of the variations of Kanban that can be used in the studied environment. Besides, the PCSs included in the search strings are considered as a base for the selected systems (THÜRER; STEVENSON; PROTZMAN, 2016b).

In this study, we focus on the general flow shop (GFS), the general job shop (GJS), and the pure job shop (PJS) configurations because these describe routing variations commonly found in MTO companies. According to Stevenson, Hendry and Kingsman (2005), in a pure flow shop (PFS), all orders follow the same sequence of stations. In a GFS, orders may visit only a subset of the stations, but they have the same flow sequence between any two stations. That is, there is a dominant flow. In a PJS, orders have a variable routing length and a random or undirected routing sequence. For the system studied, we focus on the use of priority rules by its simplicity to use. This study excludes scheduling models because it can be infeasible by the computational effort required when shops are using those PCSs (a broader review in scheduling for job shops is available on Türkyılmaz et al. (2020). To guarantee a reasonable quality of research, only peer-reviewed articles and conference proceedings were considered. We focused on studies that used simulation, which is the main methodology used by researchers for comparing PCSs due to the difficulty of MTO production. Other research methods were considered when they described in detail the effect of factors on performance. The search was not limited to any specific years of publication or subject areas.

3.2.2 Stage 2: Conducting

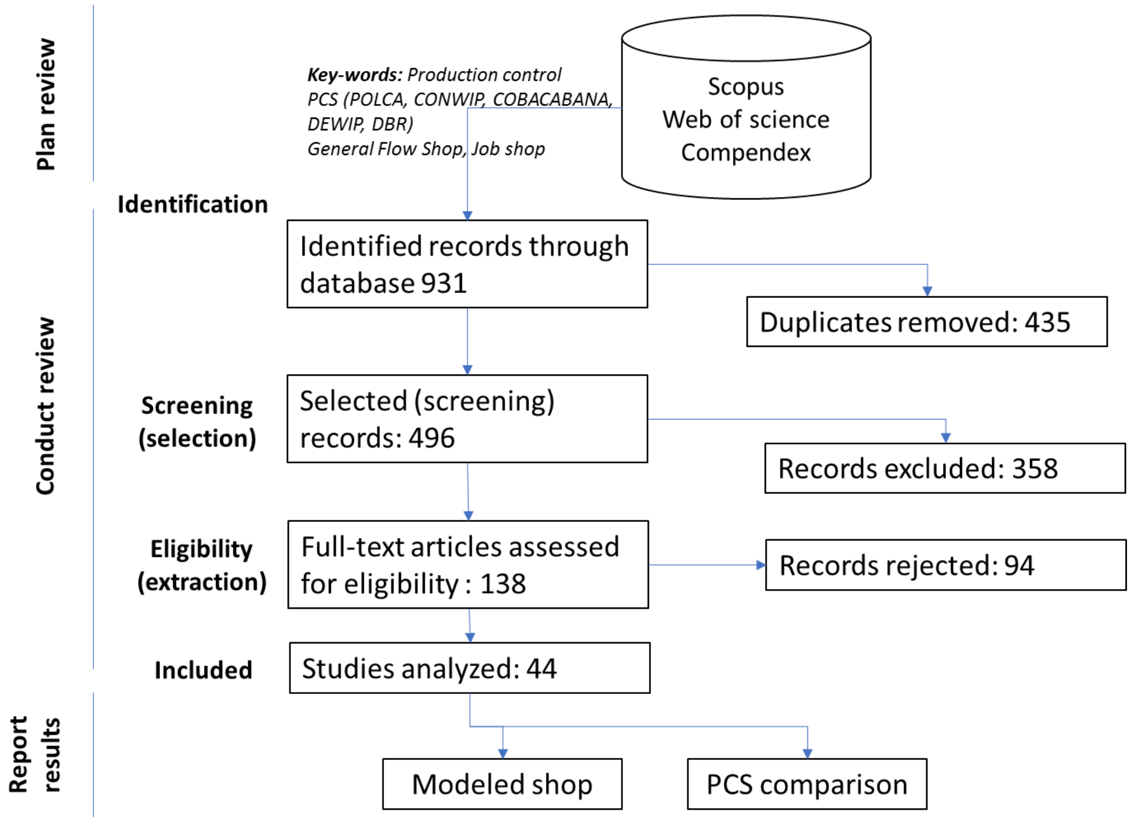
The search phase identified a total of 864 articles. The search was executed in March 2018 (updated in September 2020), and search strings were defined as follows: "(POLCA OR COBACABANA OR CONWIP OR DBR OR DEWIP) AND (JOB SHOP OR GENERAL FLOW SHOP) AND (PRODUCTION CONTROL)". The search was restricted to the title, abstract and keywords of publications.

In the selection phase, 435 out of the 931 articles were duplicate, yielding 496 unique papers. From these papers, 358 were rejected using the following exclusion criteria (more than one criterion can apply to a single paper):

a) The paper describes a pure flow shop: 175 articles. Examples include (BETTERTON; COX, 2009; HUANG; WANG; IP, 1998; YANG, 2000);
b) The paper does not include a simulation result or detailed results for an empirical case as a simulated experiment: 74 articles. Examples include (AGLAN; DURMUSOGLU, 2015; HOOSE; CONSALTER; DURÁN, 2016; MASIN; PRABHU, 2009);
c) The paper refers to a scheduling method and focuses on solving an analytical model: 61 articles. Examples include (AJORLOU; SHAMS, 2013; AL-TAHAT; RAWABDEH, 2008; LEE; PIRAMUTHU; TSAI, 1997);
d) The paper does not present a comparison with one of the PCS of interest: 31 articles. Examples include (GUO et al., 2010; KHOJASTEH-GHAMARI, 2012; MEHRSAI; KARIMI; SCHOLZ-REITER, 2013).

The extraction phase consisted of a careful reading of the 138 accepted papers. Among those papers, 94 were rejected because they presented a PFS production environment or did not describe a comparison of the PCS. The final number of papers included in this review was 44. During the extraction phase, reported data were collected in two forms: i) characteristics of the simulated shop and ii) general classification of the PCSs compared.

Figure 3.1 - The steps of the literature review performed



The studied characteristics of the simulated shop, suggested by Kiran (1998) and Henrich, Land and Gaalman (2004) as the most common variables, are the following:

a) Order arrivals (arrival intensity, inter-arrival time variability, and the source of data);
b) Processing and setup times (processing time lumpiness and variability, setup time consideration and the source of data);
c) Machines (number of stations);
d) Job routings (routing length and length variability, sequence variability);
e) Shop load factors (utilization of the system or station);
f) Due dates (due dates tightness and variability allowed, as a mechanism to establish it);
g) Priority rules: order release rule for sequencing in the pool and dispatching rule for sequencing in workstations after pool release (detailed in section 3.3.2).

The general classification of the PCS comparison uses the following dimensions reported by Karrer (2012):

a) *PCS*: the PCS considered in the study;

b) Shop configuration: describes the type of shop considered in the study;
c) Performance measures for PCS comparison: describes the performance measures used in the studies for comparing PCS;
d) Experimental factor (or factor levels): denotes the input variables considered in the model;
e) Highlighted conclusions: summarise the main conclusion for the study;
f) Analysis technique: describes how the difference of the PCSs is tested or presented.

3.3 THEORETICAL BACKGROUND: THE PCS STUDIED

This section briefly describes the studied PCSs and the priority rules used within these systems. The PCSs selected are POLCA, COBACABANA, DEWIP, CONWIP and DBR as identified in the previous section, as they are appointed to be suitable for the MTO environment.

3.3.1 The general structure of the PCSs studied

POLCA (SURI, 1998) is a hybrid pull-push concept to reduce the job throughput time throughout the manufacturing cells. POLCA uses production authorisation cards to limit WIP on the shop floor. POLCA cards act as capacity signals for production in the loop established at every two successive cells in the routing of jobs. POLCA blocks an order when there are no cards are available at the upstream cell of the loop, so the current cell chooses the next order authorised by a planning system, as a high-level Material Requirement Planning (MRP) system. The main parameters for POLCA concerning production authorisation cards are the card *quantum* and the number of cards per control loop. The *quantum* is the amount of workload that a card represents (e.g., an order unit or an amount of processing time).

COBACABANA (LAND, 2009) is also a card-based PCS to execute the WLC concept. This system uses a loop from each station (or cell) to a central release, which calculates the amount of load designed for each station. The card represents the workload (processing time) of the order in the selected station. COBACABANA blocks an order if the amount of released load exceeds the workload norms established for any of the stations in the routing of the order. According to Land (2006), the critical factors for Workload Control (and also for

COBACABANA) are the workload aggregation, the workload estimation method, the order release mechanism, and the time limit. The time limit restricts the set of jobs available for release from the pre-shop pool to the most urgent ones. Another system developed in the context of WLC is the DEWIP (LÖDDING; YU; WIENDAHL, 2003). This system controls decentralised loops between the stations to limit the workload at each station. A central system (i.e., Material Requirement Planning system) schedules the orders. Based on the load level of consecutive stations, the next order of the backlog is considered to avoid overload. The parameters for this system are the WIP limit at each station, the separation between direct and indirect WIP, and the relative position number (an average) of the station in the flow, according to the desired lead time for orders.

CONWIP (SPEARMAN; WOODRUFF; HOPP, 1990)was proposed as an alternative to Kanban for a wide variety of products. The system uses a generic loop that involves the entire shop floor. A generic loop may consider any possible routing inside the loop. The system limits the amount of WIP by allowing a maximum amount of load in the loop, which is represented by the number of CONWIP cards available. According to Framinan, González and Ruiz-Usano (2003), the parameters for this system are i) the production quota or rate; ii) the maximum amount of WIP; iii) a capacity shortage trigger for additional capacity if needed; iv) a pool sequence rule in the backlog list, and v) the card *quantum*.

DBR (GOLDRATT; COX, 1984) is a PCS derived from TOC that uses the concept of a drum for the constraint station. The order release depends on the schedule at the constraint. This PCS uses a rope (i.e., a control loop) between the drum and the pre-shop pool to control the amount of released work to the drum. This PCS requires a buffer to avoid drum stops. This system limits the amount of workload in the loop to control the release of orders to the system. This system also has many routings between the upstream loops (previous station before the drum) and downstream loops (subsequent stations after the drum) (DARLINGTON et al., 2015; WATSON; BLACKSTONE; GARDINER, 2007). According to Schragenheim and Ronen (1990), the main parameters for this system are i) the scheduling method for the constraint (c), ii) the limit of WIP in the control loop, and iii) the release mechanism for non-bottleneck stations

3.3.2 Priority rules for the PCS studied

Priority rules may be used at order release and dispatching. Priority rules at dispatching may interact with priority rules at order release (also known as the pool sequencing rules). While dispatching rules defines the sequence by which jobs are processed at machines, pool sequencing rules define the sequences by which jobs (orders) are considered for release to the shop floor each time order release is triggered. Table 3.2 details the priority rules and the order release mechanism used in the reviewed studies.

Table 3.2 – Priority rules and methods used in the studies reviewed

Type	Priority rule	Description	How the rule operates
Dispatching rule	CRT (CR)	Critical Ratio	Select the job on the ratio of time remaining of the due date and time processing remaining
	CRE	Critical Ratio expiration	Select the job with the lowest ratio of the expiration date (product lifespan) and time processing remaining.
	CS	Capacity Slack	Select the job with the lowest capacity slack ratio. This ratio is the slack that considers the aggregate load contribution to the station over the difference between the load norm and the current load of the station
	EDD (EODD)	Earliest Operation Due Date	Select the job with the earliest due date estimated for each operation in the job
	EPST	Earliest Planned Start Time (Urgency of the job or Slack sequence)	Select the job with minimum allowance time to start the operation. Starting time is estimated as the due date minus the time processing remaining
	FIFO (FCFS)	First-In-First-Out	Select the job that has arrived earliest
	LLM	Least Loaded Machine	Allocate job which next operation is the least workload measured in each machine considering queue and process
	LQ	Longest queue	Select the job that its next station has the longest queue available at a given instant in the input bins
	LSUT	Least Setup Time	Select the job with Minimum Set-up Time of job in that station (dispatching rule)
	MNJ	Minimum Number of Jobs	Select first the job that next process has the minimum number of jobs in the queue and process for each machine
	SRPT	Shortest Remaining Processing Time	Select the job with the least slack per remaining process time
	SPT	Shortest Processing Time	Select the job with the lowest processing time in the queue
	WINQ	Work in Next Queue	Select the job that will go on to its next operation where the machine has the least work
	XWINQ	Extended Work in Next Queue	Select the job that will go on for its next operation to the queue with the least work, both present and expected
Pool sequencing rule	CScor	Capacity Slack CORrected	Select the job with the lowest capacity slack ratio. This slack considers the corrected aggregate-load contribution (by task sequence) to the station, the difference in the load norm and current load of the station and the remaining operations of the job
	CSjdir	Capacity Slack Jobs Direct load	Select the job with the lowest capacity slack. This slack is the ratio of the job load (in the number of jobs) and the differences

Type	Priority rule	Description	How the rule operates
			of the current number of jobs and maximum number of jobs allowed in that station (jobs direct load norm), considering the operations remaining on average
	ERD	Earliest Release Date	Select the job with the earliest date that considers the order's confirmation and material available for the job
	IMR	Immediate Release (as Push)	Release a job immediately to the shop floor. This rule is a push system lacking a pre-shop pool
	LSK/RO	Least Slack per Remaining Operation	The minimum value of time allowance for the planned start time for the operation
	MERD	Modified Earliest Release Date	Select the job with the lowest release date: a negative release date (urgent job) by SPT rule or a positive release date by ERD rule
	MODCS	Modified Capacity Slack	Select the job with the earliest release date. For negative release dates (urgent jobs), use ERD to sort jobs; and for positive allowance on the release date, use the CS rule
	MPST	Modified Planned Start Time	Select the job with the lowest value of the maximum of the earliest planned finish time and earliest possible finish time for each job in the station queue
	MRP	Material Requirement Planning (as PRD)	The release date for a job is scheduled in MRP, but the rule considered is PRD
	PRD	Planned Release Date	Release a job on the date when the order should be released from the pre-shop pool if it is to be delivered on time
	STWK	Shortest Total Work Content	Release the job with the minimum sum of its processing times
Release method	C-BSA	Continuous workload Balancing and Starvation Avoidance	Release a job as C-SA, but the job could violate any indirect load norm
	CR	Continuous Release	Release a job at any time, under a condition or event (arrive, operation end or a lower bound of workload is reached in a station)
	C-SA (CR+)	Continuous Release with Starvation Avoidance	Release a job at any time or event, considering that, if a workstation starves, then release the job at that workstation as the first task in routing
	DBN	Dynamic Bottleneck Control	Continuous release if the load of the bottleneck is under the norm. The bottleneck is established dynamically in short-run during system operation
	FBN	Fixed Bottleneck Control	Continuous release if the load of the bottleneck is under the norm. Only one bottleneck is established after long-run
	LUMS COR	Lancaster University Management School Corrected Order Release	Jobs are released at periodic (fixed) times but can be released at any moment if a workstation starves (as in C-SA). The aggregation load method considers the number of steps remaining for the indirect load (or future load in that workstation)
	LUMS OR	Lancaster University Management School Order Release	Jobs are released at periodic times considering the direct load to the next station and the indirect load to the remaining stations. Also known as the Aggregate Load-oriented approach of WLC (ALOWLC)

Source: Authors based on the reviewed papers

3.4 RESULTS

3.4.1 Modelled Job and Shop Characteristics

Table B.1 (in Appendix B) contains the characteristics of the simulated shops extracted from the papers of interest. System behaviour primarily depends on demand variability and processing variability. To model demand variability, comparisons commonly use the exponential distribution to model inter-arrival times, using variants such as the Poisson distribution for arrival rates (25 papers out of the 44). Other order arrival distributions used are empirical distributions (estimated from real data without being fit to a specific function), the normal distribution, and the uniform distribution. In 17 articles, the arrival rate in the model estimates the utilisation level of 90% in all stations or the bottleneck modelled. To model processing variability, the exponential distribution and the Erlang distribution with a shape parameter (k=2) are the most used (18 of the 44 papers, for both distributions). Other distributions used to model processing variability are empirical (9 articles), deterministic, exponential, normal, gamma, lognormal and uniform. The processing variation was considered one of the most significant variables that influenced system performance. The most frequently considered form of setup was sequence-independent, with setup times being included in the operation processing times (31 articles). For other kinds of setups, the normal distribution was considered.

For most of the simulation studies found in the literature, authors use the simulation model based on Melnyk and Ragatz (1989). Several authors have used this model as a baseline to compare results with the same shop model (BARROS et al., 2016; CARMO-SILVA; FERNANDES, 2017; FERNANDES et al., 2017a, 2017b; SILVA et al., 2017; THÜRER et al., 2017a, 2017c). The model refers to a job shop with six machines, using a random product mix, with the exponential distributed inter-arrivals times, Erlang distributed for processing times, and sequence-independent setup times. Therefore, eight of the papers have modelled shops with six workstations, with the variable number of operations per job between one and six, each at a single station. The number of machines considered in studies based on empirical data is usually greater than on studies based on hypothetical data and ranges between 1 to 73 groups. Modelled job routings typically refer to pre-established routes (22 out of 44), followed by random routings (16 papers). The flow typically considered in paper with empirical data is linear.

Finally, there were two due date estimation methods typically used: i) the uniform allowance to represent an external (or exogenous) due date setting (17 out of 44 papers), and ii) the total work content to represent the internal due date setting (3 out of 44 papers). The due date estimation method has a significant impact on reliability performance. These estimations were defined only when reliability was considered as a performance measure, while 18 papers did not use this type of performance measure.

3.4.2 Conditions of the PCSs' comparisons

Table 3.3 shows the meaning of the acronyms of the performance measures used in the reviewed studies, and Table B.2 (in Appendix B) shows the general classification of the PCSs compared. Underlined elements indicate that a level or factor was reported as having a significant difference (when the comparison reports statistical test) or a large difference. Non-underlined elements indicate the other factors tested that did not present an important effect for that comparison

Table 3.3 – Performance measures for PCS comparison

	Logistic Performance	**Logistic Costs**
Order (External)	DTM – Delivery time mean DTD –Delivery time deviation DR – Delivery reliability SL – Service level	PR – Price ANR – Average net revenue TC – Total Cost RI – Robustness Index
Shop floor internal	Shop Floor Throughput Time (SFTT) TTM – Throughput time mean TR – Throughput rate CTM – Cycle time mean CTD – Cycle time deviation Due Date Deviation and Reliability LAM – Lateness, mean LAD – Lateness, standard deviation TDM – Tardiness, mean TDP – Tardiness, percentage BNS – Bottleneck Shiftiness MS – Makespan WQM – Wait time queue mean	CD – Costs of delays (tardiness of an order) BL – Backlog number of orders WIP – Inventory UT – Utilization LQM – Length queue, mean

Source: Adapted from Lödding (2013, cap. 2), according to measures found in the reviewed papers

In Table 4, the rows show the experimental factors used in studies comparing PCSs, and the columns show the performance measures considered. Each cell describes the frequency of using one factor with a performance measure. The number of papers using each factor or performance measure is in italic, at the last column and row, respectively. Trends in the comparison studies become visible in this Table. We see that the most frequently tested performance measures are time-based (TTM in 26/44, DTM in 21/44), followed by WIP (21/44), and reliability (TDP in 12/44, LAD in 12/44). Those performance measures are the most frequently used in MTO. Concerning experimental factors beyond the PCSs, the dispatching rules (20/44) and the WIP level (limited by WL norms in 17/44 and limited by the number of cards in 14/44) are the most explored factors. For dispatching rules and pool sequencing rules, there are different reported results because some rules are not aligned between them (e.g., ERD, SPT, CS when the type of pool sequence rule does not match with the dispatching rule). The factors with numerical levels such as WL Norm, number of cards, processing time, and card size have been described only as factor effects, and the studied papers do not report the mean of each level. Besides, some studies did not perform statistical tests to evaluate the significance of a reported difference.

Table 3.4 shows the experimental factors used in the comparisons studied (on the rows) and the performance measures considered for the experiments (on the columns). The intersection of the row with columns describes the frequency of the factor that has been tested with the respective performance. There are some trends in comparisons, focusing on how each experimental factor has been tested with a specific performance measure. From Table 3.4, the most used performance indicators are time-based (TTM in 22/38, DTM in 18/38), then WIP (18/38), and then some reliability (TDP in 10/38, LAD in 9/38). This is explained in the context of the MTO and the layout studied because those indicators have been mostly used in that environment. Concerning experimental factors beyond the PCSs, the Dispatching rule (16/38) and the WIP level limitation (WL Norm in 16/38 and Number of cards in 13/38) have been the most explored factors. For dispatching rules and order release rules, there are different results since they are compared with different rules. The factors with numerical levels as WL Norm, number of cards, processing time, card size, among other factors, has been described only as a factor effect, and the mean of each level has not been reported in the studied papers. In addition, some studies have not considered a formal test to identify the significance level of the reported difference.

Table 3.4 - Frequency of performance measures crossed with experimental factors in the reviewed papers

Performance Measure

Experimental factor	TTM	DTM	WIP	TDP	LAD	TDM	TR	SL	LAM	TC	CTM	UT	WQM	CTD	DDR	TTD	MS	SLP	Freq.
PCS	14	13	13	5	7	4	8	4	2	2	2	2	2	1	1	1		1	*31*
Dispatching rule	9	9	3	8	5	6	1	1	2	1	1	1	1	1			1		*20*
WL Norm	6	8		6	6	7	3		1				1			1			*17*
Number of cards	7	6	1	3	4	2	1			1			1		1				*14*
Order Release	2	4	2	2	3	3	3			1									*8*
Shop type	3	3	1	4	1	4						1					1		*5*
Card size	2	2	3	2	1	1			1	2									*6*
Processing time	4	3	3				2					1						1	*12*
Aggregation WLC	1	2	1	2	2	2	1		1										*5*
Loop policies	4		4				1			1		1							*7*
Bottleneck	2	2	1	1		1	1												*5*
Batch size	2	1	1				1		1		1			1					*5*
Lot splitting	2	2		1	2										1				*2*
Routing	1	1			1	1	1						1			1			*2*
Starvation avoidance	1	1	1	1	1				1										*1*
Order arrival	1	1	2					2											*4*
Card allocation	1	1	1	1	1				1										*2*
MTTR			1					1	1		1			1					*3*
Level of BOM	1		2				1	1											*4*
Protective capacity	1	1			1		1												*2*
Allocation policy			2					2											*2*
Output level									1		1			1					*1*
MTBF			1					1											*3*
Setup			1					1											*2*
Case empirical	1		1																*1*
Frequency	*26*	*21*	*21*	*12*	*12*	*10*	*11*	*4*	*4*	*4*	*2*	*3*	*3*	*1*	*1*	*1*	*1*	*1*	

Source: Authors

The reviewed papers describe some conceptual relationships in the studied factors with the performance measures. The main relationships are exhibited in Figure 3.2, highlighting studied factors. The shop type describes the routing variations that the modelled shop has. This routing variation defines the length of the routing and the number of times that the order passes in a different (or same) machine. This variation creates different possibilities to have loops of control that influences how each PCS is designed to limit the WIP level. The amount of workload represented in the card size (quantum) affects how the WIP is estimated, as obviously, defines how the WIP is aggregated in the constraint of the PCS. This WIP limitation difference is the core difference for the PCS because it affects the order release and the dispatching rule mechanism. As already known, the order release uses the pool delay to reduce the TTM value, avoiding queuing time. The dispatching rule manages to reduce the specific order queue time. The due date setting may affect how the order-release mechanism and the dispatching rule react to the urgency of the job. When the bottleneck is considered as a factor in the experiment, the effect is represented by increasing the processing time. In addition, the maintenance of the machine, considered by the mean time between failures (MTBF) and the meantime to repair (MTTR), augment the throughput time in the station. The increment by the bottleneck or the maintenance effect has a strong impact on the utilization level of the modelled stations. The utilization level defines the WIP level, as predicted in Little's Law. Some results of the PCS comparisons are explained by these relationships.

3.4.3 The results of the comparisons

In the comparisons identified from the reviewed studies, 21 papers used CONWIP or a modification; 18 papers used POLCA or a variant; 9 papers used COBACABANA or a WLC variant; 4 papers used a DBR and TOC control system, and 2 papers used DEWIP. The following subsections detail how these systems have been compared.

3.4.3.1 The results of comparisons within the same PCSs

For POLCA-based systems, three factors have been proposed as modifications of the original system: the pool sequence rule, the dispatching rule and the WIP represented by the cards. The proposals for order release are generic POLCA (G-POLCA), POLCA Starvation Avoidance (POLCA-SA), the POLC system combined with other mechanisms of authorisation (POLC-A), and the use of CScor as a release rule. G-POLCA releases an order when all the stations in the routing of the evaluated job are available instead of using only release by loops. This system of G-POLCA outperforms POLCA and MRP (push system) when compared in a GFS (FERNANDES; CARMO-SILVA, 2006). POLCA-SA releases an order to the shop floor when the first station is starving, regardless of the workload limits in the other stations (THÜRER et al., 2017a). In this modification, the best-performing pool sequence rule was different card allocation (CS) using the modified earliest release date rule (MERD). POLC-A proposes a single release and other dispatching rules to prioritise the jobs on stations instead of using the high-level authorisation, as the original POLCA proposes (THÜRER et al., 2019a). The CScor rule has been reported as a robust rule for the load-based version (CARMO-SILVA; FERNANDES, 2018) when the order release rule is the same for G-POLCA. Adding a modification for the dispatching rule, the POLCA long queue rule (POLCA-LQ) outperformed the POLCA system by reducing the TTM, when they are tested in an unbalanced shop (BRAGLIA; CASTELLANO; FROSOLINI, 2015). Load-based POLCA (LB-POLCA) uses the processing time of each station as the amount of WIP represented by each card, as reported by Fernandes et al. (2017b). The workload accounting method used in LB-POLCA influences the performance of the card system. The best accounting method tested for the LB-POLCA was to consider the load of the destination cell in the loop (or second station approach).

COBACABANA is particularly influenced by the workload accounting approach and the workload norms, as reported by Thürer, Land and Stevenson (2014). They reported a trade-off between using the corrected approach and using an adaptive workload norm at each station. In the corrected approach, the workload norm could be suited the same for all stations; but for the uncorrected approach, the workload norm needs an adjustment for each station. This is because the corrected approach considers the position of the station in the routing to compute the indirect load for a job. As tested by Thürer et al. (2017c), the level of workload norm has a strong influence on performance, especially in DTM and TTM. COBACABANA also shows a high interaction with dispatching rules. As stated by Fernandes et al. (2017a), order release may create the SPT effect, which releases later larger load jobs so as not to exceed the workload norm established and prioritises smaller jobs. As proposed by Neuner and Haeussler (2020), the control can be established for all the workstations or the bottleneck. Both approaches produce similar results in a limited shop.

The original CONWIP has been modified using different loop designs to improve its effectiveness, with contrasting results being reported. In two comparisons (GOLANY; DAR-EL; ZEEV, 1999; IP et al., 2007), using a single loop for CONWIP has outperformed using multiple loops. Even though these studies are for the GFS, they use a few routing options since they have empirical data from products. These results are contrasted in other comparisons (HUANG et al., 2015, 2016, 2017), in which multiple CONWIP loops (m-CONWIP) has better results than using a single loop. For these authors, the design of the loop has a significant difference in performance when high routing options are represented in the shop model, including a high variation in processing time. According to those authors, this design depends on each shop configuration and the estimation of utilisation for each loop. Other CONWIP comparisons report some improvements for the main system. Bokhorst and Slomp (2010) add rhythm control in the output to react to the Takt-time and use the FIFO rule. This result improves delivery time and utilisation compared to simple CONWIP. The Base-stock CONWIP outperforms other CONWIP modifications and other PCSs, such as Kanban (ONYEOCHA, 2015; ONYEOCHA; KHOURY; GERAGHTY, 2015). For these scenarios, the mean time between failures (MTBF) and the meantime to repair (MTTR) have been examined in the shop model as interruptions for processing availability. For Khojasteh and Sato (2015), CONWIP shows results similar to Kanban and outperforms the Base-stock system, considering the WIP level as the primary performance measure. These authors advise that there is no analytical difference in the PCS and that superiority depends on how each system is parameterised. For Olaitan et al. (2019), the loop design can consider human resources as cross-trained teams, using CT loops. This kind of loops reduces more DTM than a single loop. Huang et al. (2020) proposed the Path-Based Bottleneck (PBB) and the Capacity Slack on the single loop. For those authors, this implies the use of EODD and SPT rules inside the loops for balancing workload and improve performance. Bertolini et al. (2020) suggest the use of the rule Work in Next Queue (WINQ) for a single loop. This rule improves how the dispatching task within the loop of control.

For DBR, the accounting of the buffer based on load contribution (here referred to as LB DBR) has reported better results than the original system, as stated by Rezaei et al. (2011). According to Thürer and Stevenson (2018), the pool sequence rules for sequencing orders in the backlog and the dispatching rules exert strong influences on DBR performance in GFS. They proposed two modified rules to be used in the bottleneck that helps speed up urgent jobs (or jobs that the planned release time has passed) to improve performance: the MODPRD (for order release) and MODPST (for dispatching rule).

3.4.3.2 The results of comparison across different PCSs

Among comparisons, POLCA has reported better performance than CONWIP when they are tested in GFS (BARROS et al., 2016; BRAGLIA; CASTELLANO; FROSOLINI, 2014; GERMS; RIEZEBOS, 2010). These authors used an exponential distribution to represent variations in processing times. Although the results indicate that POLCA outperformed CONWIP, Frazee and Standridge (2016) reported that CONWIP has better performance than

POLCA when a deterministic or normal distribution represents the processing time. The control loops designed for POLCA created a mainstream flow between the loops (like a PFS), but CONWIP used a single loop in that comparison. According to Germs and Riezebos (2010), POLCA and m-CONWIP had similar performance results when the shop had low routing options between the loops. In that study, the POLCA loops and m-CONWIP loops considered the same stations, so both systems reflected the same control in the shop. According to Lödding, Ya and Wiendahl (2003), in line with loop control differences, DEWIP outperforms POLCA in a PJS because POLCA becomes blocked when the flow is bidirectional, which is explained by the design of the POLCA loop. Besides, as reported by Silva *et al*. (2017), the general Kanban system (GKS) outperforms POLCA in a directed flow system with three stations. This superior performance is due to the time a job has to wait before seizing an available card in POLCA. The Adapted Toyota Kanban System (ATKS) is better than POLCA but is worse than the GKS, according to those authors. For these comparisons, the main factor of difference between POLCA and other PCSs was the design of the paired-cell loops. Bong, Chong and How (2018) report that the Utilisation Based (UB) system performs better than POLCA. For this UB, the load norm suggested is 75% of the capacity of each station, a value based on the utilisation principle of QRM.

Bertolini, Romagnoli and Zammori (2015), reported a different result. According to them, the push system outperforms CONWIP and WLC for the DTM when infinite inventory buffers are considered for the shop, and the variability of utilisation is high. In their report, the push system reduces the time in the pre-shop pool, and it increases the TTM compared with the PCSs. CONWIP and WLC reduce the total waiting time on the shop floor and create a more predictable throughput time to the detriment of the pre-shop pool time when the job entry and job release stages are considered from the MTO context. This result agrees with the comparison of GKS with POLCA (SILVA et al., 2017) and GKS with CONWIP (GOMES et al., 2016) because GKS behave like a push system, as noted by Chang and Yih (1994). They did not consider the material waste effects when a job is released, which was previously recognised as a benefit of PCS (OHNO, 1988). As reported by Gomes et al. (2016), GKS outperforms CONWIP because CONWIP does not have a load balancing capability over the stations in a GFS. The lack of this capability increases the queue length in highly loaded stations, which increases the waiting time for some jobs.

The COBACABANA system has been recognised for balancing workload on the shop floor. According to Thürer et al. (2019c), COBACABANA outperforms Kanban because the centralised function of card acquisition allows for balancing the job load in the shop. This outperformance is not so evident when the Kanban present a job shop variant (similar to the GKS), as reported by González-R et al. (2018), modelling the expiration date for products. When Kanban for job shop uses the Critical Ratio (an especial version for expirable products), it has similar results than WLC (using discrete values to be comparable to a COBACABANA). This capability is also reported as the leading cause of the superior performance of COBACABANA when it is compared with the DBR system (THÜRER et al., 2017c) or when it uses an order release similar to that of DBR (CARMO-SILVA; FERNANDES, 2017). COBA-POLCA outperforms the single-use of COBACABANA or POLCA (THÜRER; FERNANDES; STEVENSON, 2020). COBA-POLCA exploits the workload balance of the order release of COBACABANA and the reduction of the fluctuation of the direct load of POLCA.

The DBR system has outperformed other systems when the environment has a powerful effect on the bottleneck. This result is consistent with the comparison with CONWIP (GILLAND, 2002) and COBACABANA (THÜRER et al., 2017c). Also, the superior performance is consistent with the suggestion by Bertolini et al. (2017) to use DBR. A variant of DBR was proposed by Rezaei et al. (2011) using WLC to schedule the rope on that system (as LB-DBR). This variant outperforms individual release systems, such as SA and Continuous Release. According to Carmo-Silva and Fernandes (2017), the counterpart of DBR is the lack of load balancing at the non-bottleneck stations so that the system can overload those stations; thus, this effect increases the total delivery time, causing the WLC to have better performance than DBR.

The other PCSs found in the literature are WIPLOAD control (QI; SIVAKUMAR; GERSHWIN, 2008), the Dual Constraint Resource Hybrid Push-Pull (SALUM; ARAZ, 2009) and Push-Kanban (MÜLLER; TOLUJEW; KIENZLE, 2014). The WIPLOAD control (WIPCtrl) uses an estimation of the remaining processing time in stations to limit the total load on the system. This system outperforms CONWIP and Shift release (SR), recognising that the release control is the most influential factor in the shop performance. This study does not consider the improvement on CONWIP with the Capacity Slack Corrected (CScor) for sequencing the pool. According to Thürer et al. (2017b), this backlog rule helps to balance the workload in the shop.

For the dual constraint resource (DRC) systems, the station is considered by two constraints that limit the system: labour (human availability) and machine (time-machine availability). The DRC Hybrid Push-Pull (HPP) proposed outperforms the corresponding version of DRC-Kanban (as a pure pull system) (SALUM; ARAZ, 2009). In a DRC system, performance depends on the magnitude of the effect of labour or machine as a constraint effect to have different results.

Push-Kanban uses robust scheduling from heuristics (as a push modal) and a Kanban time (as a pull capacity modal), with a decentralized WIP control (MÜLLER; TOLUJEW; KIENZLE, 2014). This system uses some concepts of DEWIP (LÖDDING; YU; WIENDAHL, 2003) to operate, but it is not so focused on Workload Control. For this system, the WIP limit has a strong effect on shop performance, so it is needed to parameterize these limits based on shop variables. This system outperforms MRP II.

In the GFS, some strategies could also make groups of technology (also known as cell manufacturing) into separate family flows and improve productiveness. This situation could create a nested system, such as external inter-cell flow control and internal flow cell control, as suggested by Thürer, Stevenson and Protzman (2016b). For those systems, Klausnitzer, Neufeld and Buscher (2017) found that release rules and sequencing rules have a strong impact on performance, using heuristics to schedule jobs. For the PCS, there are only two proposals identified in the literature: a) using POLCA for routing control and Generic Kanban for balancing load in the shop, as suggested by Olaitan et al. (2017); and using COBACABANA for external flow and Kanban inside the cells to control the local WIP, as suggested by Thürer, Stevenson and Protzman (2016a). This concept is still in development for this scenario of the job shop, and the question of how the performance of the nested control system could be improved is still open.

3.4.3.3 Summary

Table 3.5 presents the main results of comparisons between PCSs. The base of the comparison for the PCS is on the row. The other PCS compared is in the column. When the studied PCS (row) present a better performance, the result has a positive signal (“+”). When the studied PCS presents a similar performance, the result appears with an equal sign (“=”). When the PCS has worse performance than the other PCS, in the last column, the result shows a negative sign (“- “). Cells with the same PCS on the row and column (shown in italic) present parameters and design factors that influence performance for that specific PCS. These topics are the highlighted results for the extracted papers in the literature review. For the other PCSs, some details are omitted to focus on the selected systems.

Table 3.5 – Summarized comparison of PCSs for the job shop

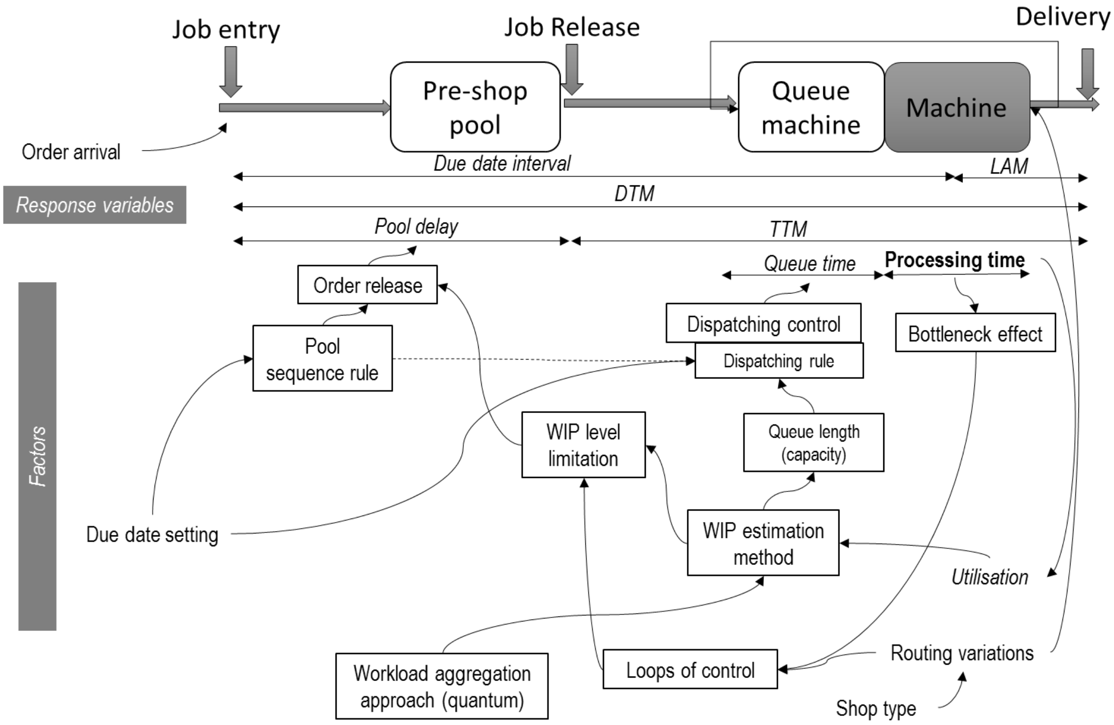
	POLCA	**COBACABANA**	**CONWIP**	**DBR**	**Other**
POLCA	*Improvements on the original POLCA:* • *Starvation Avoidance using rules: MERD, CS;* • *G-POLCA using CScor and EODD;* • *Use Longest Queue as dispatching rule-LQ;* • *Load Base, using the 2nd station approach for load accounting* • *POLC, using release allowance authorisation*	"-" COBA-POLCA: when combined with COBACABANA.	+ in a GFS with higher subset options in routing, by independent loops = M-CONWIP, in low routing options, and both systems coincide in the loop design		+ Push system (MRP) in GFS "-"UB in a GFS
COBACABANA	- Combine the order release control of COBACABANA and the direct load control of POLCA	*Parameters influencing the performance of the original COBACABANA:* • *Release mechanism and dispatching rule* • *Workload norms* • *The accounting workload approach*	= when considers order release as same as dispatching rule - WIP limit is tight on the shop floor	+ In GFS and GJS, because it balances the load in the shop	+ Kanban system in a GFS = for Kanban JS considering expiration date - Push system in unstable utilisation, depending on the load limitation
CONWIP	+ when processing times are deterministic or Normally distributed + when the flow has an identifiable mainstream (almost PFS with few routing options)	= when considers order release as same as dispatching rule + WIP limit is tightness in GFS, PJS	*Differences in the CONWIP system:* • *The design of the loop (simple for almost PFS and multiple for GFS and GJS), CT for resources.* • *Utilisation level considered on the designed loop* • *BK-CONWIP outperforms other modifications* • *WINQ rule suggested*	+ in a GFS with a mainstream identifiable (almost PFS)	-GKS by the lack of balancing capability in stations -/= Push system in unstable utilisation depending on the card limitation + As BK-CONWIP S-KAP outperforms GKCS and ECKS depending on the routing variation = than Kanban considering the WIP level at the same TR (superiority depends on the parameterisation)
DBR		+very strong effect on the bottleneck by process time; performance depends on the bottleneck schedule -DBR overload non-bottleneck stations as a secondary effect	+when there is a substantial effect on the bottleneck = when the CONWIP loop design coincides with the DBR loop	*Improvements for DBR* • *Scheduling rules (using MODPRD and MODPST)* • *LB-DBR has better results in high variation*	
Other	+GKS, when utilisation makes push + ATKS, like push system + DEWIP in a pure job shop by POLCA's blocking		+GKS by balancing capability +WIPCtrl when the flow has a mainstream		+DRC-HPP vs Kanban depending if labour or machine is the primary constraint for the system +DEWIP vs MRP (push) by balancing load in the shop; +Push-Kanban than MRP (as a push system)

"+" PCS on the row has better performance than selected on the column
"-"PCS on the row has worse performance than the respective column
"=" There is not a significant difference between the compared PCSs
• Comparison of the same system

Source: authors

The comparisons between the PCSs show some relationships between factors and performance indicators. Figure 3.2 shows the relationships on the framework of the MTO decision stages that considers the job entry, job release and delivery. For an initial decision, the shop type (GFS, GJS, or PJS) describes the routing variations for jobs. The level of variation of routings depends on the products, as available resources and kind of process. The relationship of PCSs with that level of variation is the design of the loop of control. Then, workload aggregation reflects how to estimate the quantum of a job, as the amount of work for each job. This quantum also represents the card size used when the PCS is card-based. The workload aggregation helps to define the WIP estimation method. The control loop and the WIP estimation method influence the WIP level limitation (the number of cards per loop or the workload norm used) used by the PCS. On the dispatching control tasks, the WIP estimation method helps to define the capacity (or the queue length allowed) by each station. The limit of WIP level depends on the order release mechanism, in how the order release balances the trade-off of the pool delay and the TTM. For that balance, the pool sequence rule prioritises which job release to the shop floor. Depending on the selected rule, the job due date can help with this prioritisation. That rule creates a relationship on the selection of the dispatching rule because the effectiveness of the pool sequence can be different from the queue sequence on each machine. The capacity of each queue (designed on loops) limits the subset of jobs available for dispatching rules. According to the reviewed papers, the dispatching rule must be aligned with the pool sequence rule (dotted line in Figure 3.2; in other words, both of them must use similar elements to prioritise jobs. The bottleneck effect must be considered because it affects the utilisation level of the stations. If this factor has a very strong effect, then it might require a different WIP level limitation or a unique loop control.

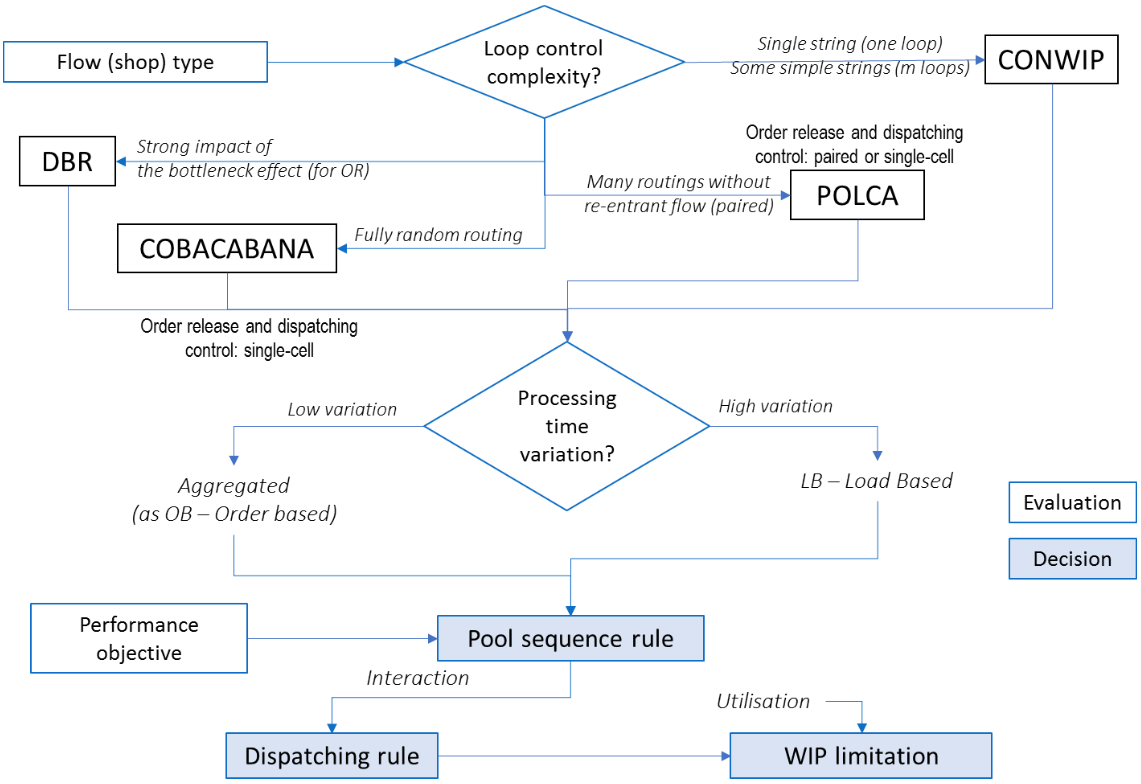
Figure 3.2 – Factors relationships found in the reviewed studies



Source: Authors

The comparisons also reveal how some factors are essential for choosing a PCS. The relationships found in the reviewed papers suggest evaluating the routing of jobs in the shop, the variance of the processing time, and the impact of the bottleneck effect. For routing of jobs, the choice of a PCS is based on the design of the loop control. CONWIP, with a single loop, is suitable when there are very few variations in job routing since the grouped stations in the loop allow to use of the order release to control dispatching tasks as a continuous flow. The m-CONWIP is suitable when there are some alternatives on routing as simple strings and the stations on the string do not need a dispatching control. When the routing has many options, the order release must consider the evaluation of paired-cell or single-cells to release the job. If the flow does not have loopbacks, the authorisation by each pair of stations, as in POLCA, is preferred. COBACABANA is suitable for an entirely random routing by its single-cell loop, but it implies using a capacity control for individual stations as ATKS or Kanban variations. If the bottleneck effect has a strong impact, then the focus needs to be on controlling that effect by considering a unique loop control, such as DBR controls. The level of variance of the processing times contemplated in the same loop helps to choose the approach of the workload aggregation. Results suggest a load-based approach for a high level of variance and an aggregated measure (such as an order-based or fixed quantum) for a low level of variance. The choice of the pool sequence rule and the dispatching rule depends more on the objective of the measured performance and the level of other indicators. As reported, the utilisation level (more specifically, the shape distribution) affects how the chosen WIP limitation affects the performance. In low utilisation levels, without material lead time constraints, the push system (or not selecting a PCS) could be considered. Figure 3.3 summarises the decisions and evaluations according to considerations reported by the reviewed studies.

Figure 3.3 – Factors related to choosing a PCS



Source: authors

3.5 CONCLUSION

This research aims to help managers in selecting a PCS and on its configuration to operate in the MTO sector. We conducted a systematic literature review on simulation studies comparing PCSs in this context. A proposed framework summarises the main results according to the effects reported in the reviewed papers. The selection of a PCS is a major decision that must consider the job routings on the shop floor, as the main difference between PCSs is explained by the control loops implemented. If the bottleneck effect is strong, then a differentiated control loop that includes the bottleneck is recommended. However, control loops design is also related to other decisions factors that affect how the selected PCS perform. Namely, the pool sequencing rule, priority rules, the workload aggregation approach and the workload estimation method.

Order release concerning pool sequencing rule has the most decisive impact on performance, and it has a strong interaction with dispatching rules. Both are related, and they may have a stronger impact than the control loop itself because these rules control the trade-off between the pre-shop pool delay and waiting times on the shop floor. According to the reviewed studies, if the pool sequencing rule considers priority based on the due date, the dispatching rule must be based on a due date-oriented rule. When these are not of the same type, performance is negatively affected. Priority rules have a substantial impact when due dates are external defined (i.e., by the customer). Since the modelled shops consider a non-repeatable and MTO production, authors usually adopt exponential inter-arrival times and Erlang or exponential distributed processing times. These distributions significantly increase variability, as reported in the reviewed studies.

The workload account approach used and what the card represents influence how the order release performs. When the PCS is card-based, the reviewed papers present two main versions of that approach: either it considers the amount of processing time (also known as load-based, LB), or it uses the number of jobs or orders (order-based, OB). The reviewed studies suggest OB when there is a low variation between the jobs processing times and LB for the high variation. The LB representation has improved the card-based version of the PCSs (POLCA, CONWIP and DBR).

3.5.1 Managerial implications

The proposed framework supports managers in two ways. First, for managers that seek to implement a new PCS, it provides guidance on which PCS offers the best fit. Second, for managers that have already implemented a PCS, it guides how to adjust a PCS in response to environmental change, which can be quite common in high variety MTO environments. The relationship of factors and the framework proposed in this study helps managers analyse how to decide changes for a PCS. For example, our framework suggests that changes in jobs routing may imply changes in control loops, which should trigger an analysis of other parameters based on the new loop unit. Similarly, an increase in the variation of the processing time affects the quantum representation decision, either order-base or load-base. The framework consequently reduces the decision space for managers, limiting attention to the factors affected by an environmental change.

3.5.2 Implications for literature: proposing research questions

Results of this research are limited. These limitations are: the shop configurations studied, the factors modelled simulation experiments and other publications not listed in the databases. Some questions are not yet fully answered in the literature on PCSs, at least in their comparisons. For the modelled shops, simulation models do not consider delays caused by materials; but this external factor can interact with the order release and impact the performance. External factors as the supplier effect are also not included in results. This factor can also change how the PCSs react to shop performance, but many of the studied papers do not contemplate those external factors. In the case of the priority rules, as the pool sequence rule and dispatching rule, the reviewed documents do not establish if one rule is robust for all the PCSs.

We also identified three topics for research questions based on the main results, as shown in Table 3.5: a) PCSs that are not being compared in the literature; and b) the effects of PCS refinement and parameters. For PCSs that have not been reached, the comparison of POLCA with DBR is remaining. For DBR comparison, an emerging question is how POLCA reacts to a strong bottleneck effect if the main difference is only the loop control. COBACABANA and CONWIP also do not have an exact comparison if they are using LB-CONWIP and variants for their loops. This LB version is essential in MTO environments because the job routing variability defines the possible loop controls. A critical issue in refinement is the consideration of setup on cards, as Lin et al. (2019) stated. Future research could also be done to consider setup constraints.

The refinement of each PCS also has some research questions. For POLCA, how could the best modification of GPOLCA, using a load-based approach, be used with the releasing rules and sequencing rules? For CONWIP, it remains to study how the load-based version could be used for job shop control, considering the pool sequence rules proposed. It could also improve how WIPCtrl helps with those rules. Considering that DBR has outperformed other

PCSs under strong bottleneck effects, would it be useful to activate a mechanism to change the rules in the other PCSs to control the bottleneck station when it has a critical utilisation level, even in temporary situations? These combinations of factors may help improve how to understand the actions of the PCSs to the MTO environment.

4 INDIRECT LOAD POLC (IL-POLC): A NEW PRODUCTION CONTROL SYSTEM FOR QUICK RESPONSE MANUFACTURING

POLCA (Paired-cell Overlapping Loops of Cards with Authorisation) is a production control system suggested on the Quick Response Manufacturing approach. It is a visual control that uses three conditions to release the job: limiting the number of jobs in a loop of two subsequent cells, evaluating the material available for the job, and releasing the job by authorisation date for each cell. Previous literature proposed to use only the first date to release the job (known as RF-POLCA) to simplify the release mechanism. Besides, RF-POLCA has not been compared to the original POLCA release mechanism (authorisation list by date), considering the General Flow Shop (GFS), and there is a gap if that simplification could enhance shop performance. This research proposes a new production control system based on POLCA that uses the indirect load as an order release mechanism, defined as Indirect Load POLC (IL-POLC). We used a simulation experiment to compare the three order release mechanisms, using a literature model to represent the General Flow Shop. Results suggest that IL-POLC outperforms the original POLCA and RF-POLCA, especially on Delivery Time Mean, independently of the dispatching rule and quantum representation used. A critical implication of this research to managers in practice is to consider the indirect load norm of POLCA to control flow.

4.1 INTRODUCTION

The POLCA (Paired-cell Overlapping Loops of Cards with Authorisation) system was idealised to control the production flow after some efforts to reduce lead-time, using the Quick Response Manufacturing (QRM) approach, as presented by Suri (1998). As a result, the flow is simplified in some applications of those efforts, resulting in a General Flow Shop (GFS) (KRISHNAMURTHY; SURI, 2009; SURI, 2018). The General Flow Shop (GFS) describes jobs with the same directions for stations, but the jobs may visit a subset of the stations (from one to all the stations in the shop). For this context, the POLCA system controls the flow of jobs combining releases by authorisation dates, the number of jobs being processed in a loop, and material availability. This combination has been designed for low-volume, high-variety in a GFS environment.

Suri (1998) defines that POLCA releases a job using the authorisations lists controlled by a cell-configured Material Requirement Planning (MRP). This authorisation uses an ideal release date to begin a job without the risk of missing the due date (SURI, 2018). This estimation needs to consider the throughput time for each station in the job routing. This authorisation element is appointed very useful since that lead-time estimation predicts the throughput time for the cell (RIEZEBOS, 2018). However, this authorisation element decreases the performance for high variability on process time because jobs wait until the authorised release date. For this problem, Vandaele et al. (2008) propose Advanced Resource Planning (ARP) to estimate lead times and release dates considering the shop conditions. This estimation uses approximations of queuing networks that require some complexity to update the lead-time values. According to Thürer *et al*. (2019a), using a different order release rule is a simple solution that can solve this problem. This situation arises the gap for other modifications in the POLCA order release. As Land (2006) reports, the authorisation by release dates (defined as the time limit) increases the Delivery Time Mean (DTM) for a job in the GFS. As a practical solution for this gap, Suri (2018) proposes the Release and Flow POLCA (RF-POLCA), which releases the job for the first loop with the authorisation date, and the following loops consider the job as authorised to be processed. This RF-POLCA presents several limitations for the environment studied: few stations on routings considered, low variation of processing times, and FIFO for dispatching rule. Because these conditions are inadequate for the MTO, there is a gap to solve the authorisation list problem for the original POLCA.

In contrast with those order release mechanisms, the Workload Control (WLC) approach uses the aggregate load, as the combination of the direct load (jobs in the queue of the cell) and indirect load (jobs released to the shop floor that are next to arrive at cell's line) (LAND; GAALMAN, 1998). For this approach, the indirect load significantly affects the pure job shop (as entirely random routing) and the general flow shop (OOSTERMAN; LAND; GAALMAN, 2000). This concept was used in the COBACABANA (Control of Balance by Card-Based Navigation) system, proposed by Land (2009).

In this chapter, we aim to use ideas of the indirect load from WLC to solve the authorisation date of the POLCA system when the shop does not have an ARP to estimate updated lead times and release dates. Besides, there is a gap concerning the combined use of direct and indirect load between cells within the POLCA system. Some previous studies aim the related questions. Fernandes *et al*. (2017b) have used direct and indirect load concept inside the paired-cell loops. Carmo-Silva et al. (2020) have tested how a centralised release mechanism improves POLCA performance, but that mechanism considers individual stations loops for aggregating the indirect load. Riezebos (2018) appoints the gap in how the indirect load can be used on POLCA to improve performance. This research proposes a new Production

Control mechanism, named Indirect Load POLC (IL-POLC), to adapt the workload control concept to the POLCA system, based on the loops of control of the original POLCA but authorised by the indirect load condition and not by release date. Kanet (1988) suggest the shop conditions for order release when there is a high variance of the processing time. This paper studies the effect of some parameters that IL-POLC needs to operate. This research compares the original POLCA, the RF-POLCA and the proposed IL-POLC using simulation experiments for attending this goal.

The remainder of this paper is structured as follows. POLCA order release mechanisms are described in section 4.2. The simulation model and the performed experiments are shown in section 4.3. Results are exhibited in section 4.4, and discussion is presented in section 4.5. Finally, section 4.6 shows conclusions, with managerial implications and future research.

4.2 POLCA ORDER RELEASE MECHANISMS

POLCA, as proposed in Suri (1998), is the production control element that combines the pull and push system. The pull system is made by controlling cards of loops of two paired and consecutive stations. These cards limit the amount of direct workload of each loop. Each card identifies the origin and the destination cell. The cards indicate that the loop has the available capacity to work on the job. The push system is in the order release mechanism of POLCA. This original POLCA using the MRP authorisation date is described in section 4.2.1, the RF-POLCA is exhibited in section 4.2.2, and the proposed Indirect Load POLC (IL-POLC) is presented in section 4.2.3.

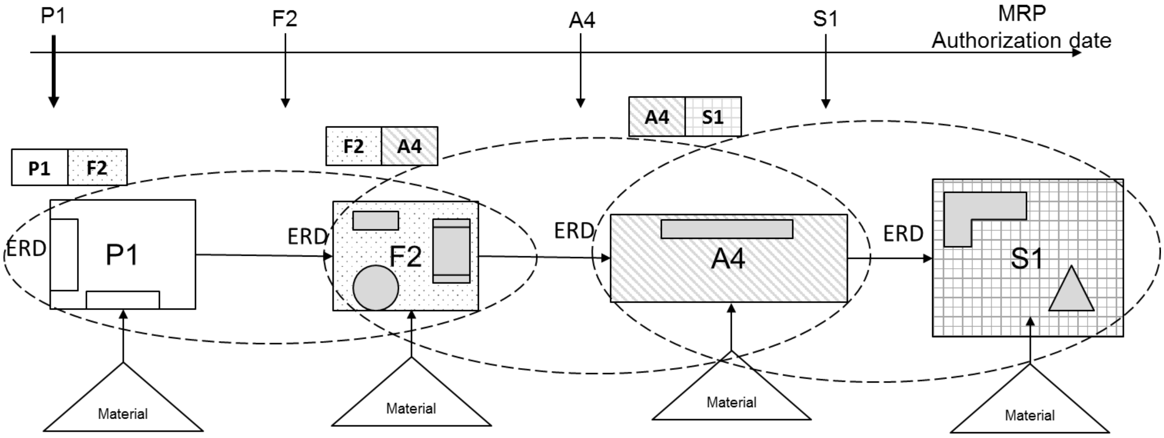
4.2.1 The original POLCA order release

Following the example of POLCA in Suri and Krishnamurthy (2003), the job follows the routing: P1, F2, A4 and S1. When the job arrives at each cell (e.g. at P1), three conditions must be satisfied: a) the card P1/F2 must be available; b) the job is released or authorised to be processed on P1; c) the material for the job in P1 is available. As the job begins on P1, the card P1/F2 is attached. Once the job is complete at P1, it follows to F2 according to the routing, where the operator evaluates the three conditions. The card P1/F2 is still attached to the job.

When the station F2 finishes the job, the P1/F2 is released to P1 to signal capacity availability. Figure 4.1 summarises this process.

The original POLCA releases the job using the release dates estimated in a High-Level (H/L) MRP that reports authorisation lists. This condition of POLCA is satisfied when the job reaches the release date for the respective station. This date is estimated based on the station throughput time and each station's due date considering the routing order. This station throughput time is commonly fixed for the MRP. For each station, the most common is that the MRP prioritises jobs using Earliest Due Date (EODD), limiting the possible selection to the subset of authorised jobs. According to Suri (2018), this release by authorisation lists allows avoiding working earlier in a job and giving resources necessary for other urgent orders.

Figure 4.1 – The original POLCA control system



Source: adapted from Suri and Krishnamurthy (2003)

4.2.2 The RF-POLCA Order Release Mechanism

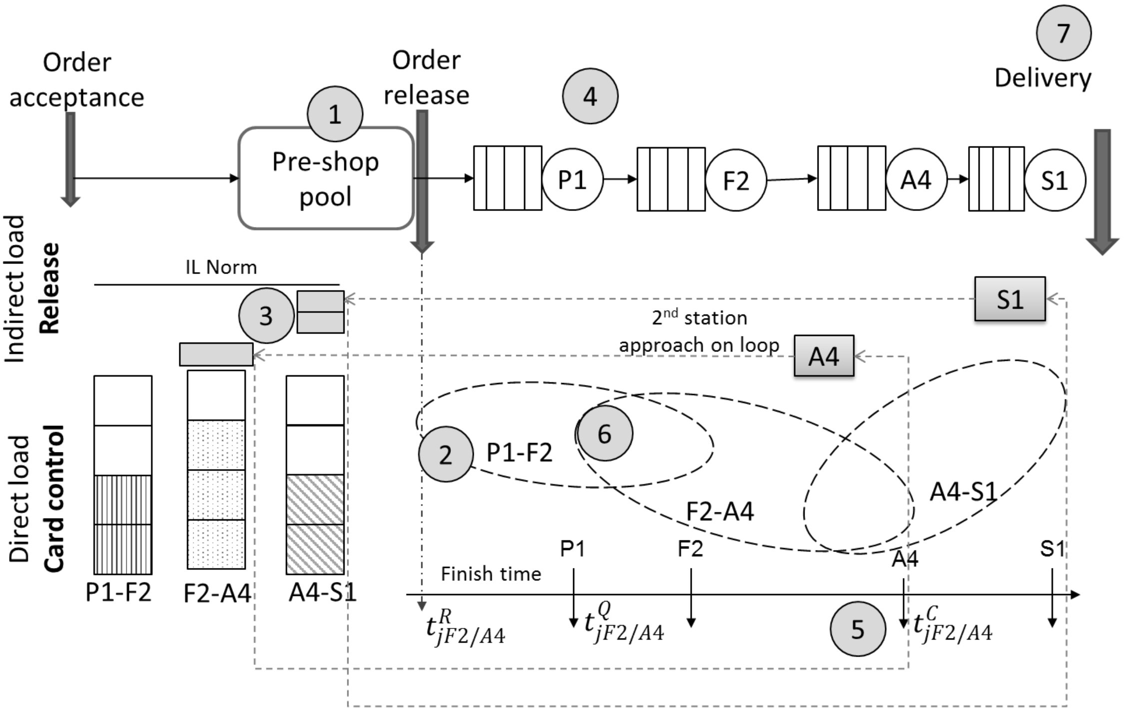
Instead of using the authorisation date for each station, the Release and Flow authorisation (RF-POLCA) authorises the job routing stations using the date estimated for the first station (SURI, 2018). The example shown in Figure 4.1 shows that this unique authorisation date is P1 on the MRP authorisation date. The dispatching rule suggested by that author for each queue is First-In-First-Out (FIFO) because the backlog list for each cell must be exact in which job is next. This order release assumes that the due date for operation is no longer necessary since the authorisation date contemplates the throughput time for all the stations in the job routing.

Suri (2018) does not recommend the RF-POLCA when: a) there are long chains of loops, more than three loops; b) there is a high variability of workload in the cell; c) the cell has many intersections with other loops and d) schedule and due dates change very often after releasing the job. In RF-POLCA, the job continues the flow without a due date for each station. This release mechanism increases the risk of wrong prioritisation because the jobs follow FIFO as a dispatching rule. When this due date is not nearly exact, or there is high variability in real throughput time, releasing jobs by authorisation lists decreases the performance (THÜRER et al., 2019a). Furthermore, for long chains or many loops involved in the flow, this wrong estimation creates the excessive building-up of the WIP effect downstream.

4.2.3 The proposed Indirect Load POLC

The Indirect Load POLC (IL-POLC) order release uses the indirect load amount for each loop instead of releasing by an authorisation date. The indirect load (or the upstream load) in a station is the amount of a load of released jobs in a queue of upstream stations (LAND; GAALMAN, 1998). The direct load is the amount of load actually in the queue of the station. The aggregate load of a station is the load released until the station evaluated. This concept is the sum of the indirect load and the direct load. IL-POLC releases a job to the shop floor when the indirect load of each loop does not exceed the load norm. This indirect load does not consider the first loop at the release moment because the first loop is considered as the direct load. The direct load in the job routine is controlled by POLCA cards, as in the original system. Figure 4.2 summarises how to apply IL-POLC to the example of the original POLCA presented in Suri and Krishnamurthy (2003). The job arrives in the pre-shop pool (1). When the system triggers the order release, the releaser evaluates if there are available cards for the first loop (2) and if the indirect load of the following loops does not exceed the load norm (3). If there are no available cards or the job exceeds the indirect load norm, the releaser returns the job to the pre-shop pool and evaluates the next job in the backlog list of the pre-shop pool. If both conditions, card available for the first loop and indirect load norm not exceeded, the releaser sends the job to the first station in the job routing, releasing the job in the shop floor (4). For example, the first station is P1, that the first loop is P1/F2. When the station finishes the process for that job, e.g. the station P1, the indirect load of the following loop (the loop F2/A4) is updated and discounted for that loop (5). This discount is because the loop F2/A4 become a direct load when the job finishes P1, and the job can continue if there are cards available for F2/A4 loop (6). As the job continues the routing, steps (5) and step (6) repeat the evaluation every time that the following station of the job routing finishes the processing for that job. For the example, steps (5) update the loop A4/S1 when the station F2 finishes the job, and (6) evaluate available cards for the A4 station. The update step (5) and the evaluation step (6) are repeated considering the followings loops until the last station of the job routing finishes the job (7), and the job can be delivered.

Figure 4.2 – Order Release proposed for IL-POLC



Source: authors

For the approach representation of the indirect load, IL-POLC uses the second station approach. This approach considers the indirect load of the paired cells by the amount of load of the destination cell within the loop. This approach is also used similarly in the DEWIP system for local workload estimation (LÖDDING; YU; WIENDAHL, 2003), with the difference that IL-POLC uses a centralised release. Fernandes *et al*. (2017b) reported this approach as the best load accounting approach for loops.

Three main definitions should be made within IL-POLC: (a) the strategy used to measure how each loop contributes to the load of the jobs released in the shop floor (refereed as load accounting method); (b) the release period length for the order release method; and (c) the dispatching rule for each station.

For load accounting methods for GFS, Oosterman, Land and Gaalman (2000) present the classical aggregation load approach (known as method B) and the corrected aggregation load approach (named as corrected B or B’). Method B aggregates the load contribution for stations of a job *j* from the release date (t_{js}^{R}) until the departure date of the station s (t_{js}^{C}). In this method, the indirect load at the station *s* is considered from the release date until the arrival date at that station. As shown in Figure 4.2, the account for the loop F2/A4 considers job load as an indirect load from release ($t_{jF2/A4}^{R}$) until the job arrives at the loop F2/A4 ($t_{jF2/A4}^{Q}$). After that moment, the account considers job load as direct load until the second station of the loop, e.g. A4 station, finishes the job ($t_{jF2/A4}^{C}$). This concept of method B has been previously used in G-POLCA (FERNANDES; CARMO-SILVA, 2006). The corrected method B computes the load contribution on the indirect load proportionally by the time remaining to the paired-cell loop s, as follows: $(t_{js}^{C} - t_{js}^{Q})/(t_{js}^{C} - t_{js}^{R})$. This value is equal to p_{js}/n_{js} when the throughput times are equal for all the stations in the routing job, being the load of the job *j* at the loop *s*, (p_{js}), and the position of the loop *s* in the routing of job *j* (n_{js}).

Concerning the release period length, the original POLCA system considers periodic timing to avoid high nervousness level by the frequency of releasing (RIEZEBOS, 2018). When the General Flow Shop has not a gateway station, the periodic timing is suggested (THÜRER; STEVENSON; PROTZMAN, 2016a). The continuous release methods are discarded to avoid high nervousness, even when they have better performance using similar order release methods (FERNANDES et al., 2017a). We suggest that IL-POLC use periodic timing for period release length, following the original POLCA system. In this factor, Land (2006) explains that a shorter period allows a low average of pool delay, as the time that a job waits to be released after arrival on the pre-shop pool. Still, a more extended period increases the possibility of stable release because machines can finish larger jobs.

Regarding the dispatching rule for each station, we suggest using Earliest Operation Due Date (EODD). The job with the earliest due date for that operation is prioritised first. This rule is suggested by Suri (1998) for the original POLCA. Lödding and Piontek (2017) showed that this rule is useful for scheduling reliability.

4.3 RESEARCH METHOD: EXPERIMENT BY SIMULATION

The environment in the model is a general flow shop, a typical shop in a make-to-order production system. This production environment considers a high-variety, low-volume production environment. We used discrete-event simulation, once it is a suggested tool for evaluating the performance exploring the conditions considered (TAKO; ROBINSON, 2012), especially at the shop floor level. The modelled shop and job are presented in the following subsection. Also, the parameters of the three studied production control systems are detailed. Finally, the experimental design is displayed with performance measures.

4.3.1 Model shop and job characteristics

The model proposed for comparison is based on Oosterman, Land and Gaalman (2000) because it allows validation with other experiments presented in the literature concerning POLCA and COBACABANA (BARROS et al., 2016; THÜRER et al., 2017a, 2017c; THÜRER; STEVENSON; PROTZMAN, 2015). As presented in the most common applications of POLCA (SURI, 2018), the model represents a General Flow Shop for job routing. This model does not consider the pure job shop because previous studies do not recommend the POLCA system for such an environment due to the gridlock effect (HARROD; KANET, 2013; LÖDDING; YU; WIENDAHL, 2003).

We implemented the simulation model using FlexSim simulation software. The model considers that job routings, processing times, inter-arrival times and due dates are random. The shop contains six stations with the same capacity. The routing length varies from two to six cells, considering a uniform variation. All stations are considered with an equal probability of being visited, and no-re-entrant flow is allowed. Appendix C shows detail of the simulation model.

As in other studies presented in the literature, the processing time follows a truncated Erlang with shape parameter two with a maximum value of 4 and a mean of 1 time unit. Set-up is considered sequence-independent, and its time is part of the processing time. The inter-arrival time follows an exponential distribution with 0,738 time-units, estimating a 90% utilisation level in the represented cells. As we deal with a make-to-order environment, due dates are set exogenously between 33 to 55 time-units. Table 4.1 summarises the modelled shop and job characteristics, and Table 4.2 presents the job routing percentage resultant from the model for each path for validation results of the model.

Table 4.1 – Shop configuration for the simulation

Variable	Review observed*
Routing variability	Random for general flow shop
Work centres	Six single station
Utilisation level	Estimated in 90% for each cell
Job routing	General Flow Shop using uniform distribution [2,6] for routing length, non-re-entrant flow.
Due dates estimation	Constant order allowance (N) for external setting limits, using the uniform distribution from 33 to 55 time-units
Order Arrivals	Random data using exponential distribution (arrival rate depending on utilisation level from processing time mean): estimated in 0.738 time-unit/item
Processing and set-up times	Random data using Erlang distribution (shape parameter: 2, mean 1 and max 4). Independent sequence for set-up times (neglected at simulation estimation)

*Based on Oosterman, Land and Gaalman (2000)

Table 4.2 – Routing matrix of the GFS modelled

From / To	Station 2	Station 3	Station 4	Station 5	Station 6
Station 1	0.604	0.149	0.063	0.030	0.017
Station 2		0.601	0.150	0.061	0.028
Station 3			0.604	0.144	0.061
Station 4				0.607	0.152
Station 5					0.600

4.3.2 Parameters and variables of the comparison proposed

We consider the following parameters for the three systems' comparisons: a) the quantum of work that a POLCA card represents and its limits, and b) release date estimation.

For the quantum (the amount of work represented by each card), we selected two levels: one-third or one-fifth part of the maximum processing time, to use 3 or 5 cards max per job in each loop. The quantum limit is the maximum number of cards per loop. For that, we used 8, 10, 12 and 14, as a subset used by Thürer *et al*. (2017a). We also included an infinite value for the quantum limit for other comparisons. This experiment fixes the quantum limits because the practical suggestion of Suri (2018) allows adjusting values according to the performance experienced when the PCS is already working. This comparison does not consider algorithms to dynamically control this quantum limit, as Renna, Magrino and Zaffina (2013), and Luan, Jia, and Jong (2013) proposed due to the unpractical complexity of small shops.

The release date estimation uses the throughput time allowance (the time expected for a job pass through each station) and the number of operations of each job. We performed a set of simulations without restriction on release to define the throughput time allowance. Thus, we set the time allowance as 18 time-units of throughput time for each station.

The model assumes that raw materials are always available for each cell's jobs, and all the information necessary for routing is known. The model neglects the transportation time between cells and the information delay of the system. Previous studies of POLCA (BRAGLIA; CASTELLANO; FROSOLINI, 2014; FERNANDES; CARMO-SILVA, 2006; HARROD; KANET, 2013; LÖDDING; YU; WIENDAHL, 2003) have used these assumptions.

For comparison, the model considers the following variables: a) release period length, b) order release mechanism, c) the indirect load norm and accounting method, and d) dispatching rule.

As the shop represents a GFS, then the model uses a periodic release. The model compares four different levels of the release period length (3, 4, 5 and 6 time-units), according to previous comparisons (LAND, 2006). These values allow comparing if a short period allows pool delay reduction or if a long period allows load balancing in releasing.

The main factor in comparing the three systems is the order release mechanism. The original POLCA release the jobs according to the authorisation lists of each station; RF-POLCA uses the release date only for the first station of job routing, and IL-POLC evaluates the indirect load as a condition to release a job. This condition depends on the indirect load norm and the load accounting method. We use the same quantum limit levels (8, 10, 12 and 14) as previous comparisons of PCSs (OOSTERMAN; LAND; GAALMAN, 2000; THÜRER; LAND; STEVENSON, 2014). For those values of quantum limit and indirect load norm, the difference is expected for two behaviours that Ziengs, Riezebos, Germs (2012) describe: when the order release mechanism limits the jobs or the quantum limit regulates the job flow. We consider the classical load aggregation approach and the corrected load aggregation approach for the indirect load accounting method, as Oosterman, Land and Gaalman (2000) proposed. We add the indirect load condition to the original POLCA and RF-POLCA to test this factor's effect in an extensive comparison.

According to the system’s configuration, the tested dispatching rules are EODD for original POLCA and IL-POLC; and FIFO for RF-POLCA and IL-POLC. For the EODD rule, each station’s due date considers 1.1 time-units before the due date of the following station. This value represents 10% plus the processing time mean for each station, to classify as an urgent job or non-urgent job.

4.3.3 Experimental design and performance measures

The experimental factors are a) the order release mechanism compared (the original POLCA, RF-POLCA, IL-POLC); b) the release period length (3, 4, 5, 6 time-units); c) the quantum card representation (3, 5 cards maximum per operation); and d) the number of cards as the quantum limit of direct load (8, 10, 12, 14 and infinite cards). For those factors, we ran a full factorial design. For the order release mechanism, we test two other additional sets. This first additional set considers the immediate release without the authorisation element for the original POLCA and RF-POLCA. The second set aims to verify the indirect load release condition, using the original POLCA and RF-POLCA with indirect load norm besides the authorisation element. In IL-POLC, the experiment uses the same values of the number of cards for the workload norms expressed in time units. For these norms, this time-unit represents the work content of the job load for that loop. Table 4.3 summarises the parameters for the model and the variables studied.

Table 4.3 – Parameters for the PCS comparison

Factor\ system	Original POLCA	RF-POLCA	IL-POLC
Release period length	3, 4, 5, 6 time-units		
Cards used for quantum representation	3 to 5 cards maximum per operation		
Quantum limit on the loop (quantity of Polca cards)	8, 10, 12, 14 and infinite (as push comparison)		
Order release mechanism	1. Release by date for each station 2. Immediate release	1. Release by date for the first station 2. Immediate release	Release using an indirect load norm condition
Dispatching rule	EODD	FIFO	FIFO/EODD
Indirect load norm	1. Infinite level 2. Workload norms	1. Infinite level 2. Workload norms	Workload norms (8, 10, 12 and 14 time-units as load limit)
The Indirect load accounting method	Not considered	Not considered	Classical load aggregation approach (B) and corrected load aggregation approach (B')

Source: Author

Following similar studies, the simulation setting was 13,000 time-units of run length and a warm-up period of 3,000 unit-times to reduce the model's initialisation bias. With those parameters, we verified that the proposed model obtained a stable result in the simulation.

The performance measures considered in this simulation are

(a) the Delivery Time Mean (DTM, or the Total Throughput Time) – the completion date minus the arrival date of a job;

(b) the Shop Floor Throughput Time (SFTT) – the completion date minus the release date of a job;

(c) the Standard Deviation of Lateness (LAD) – the standard deviation of the completion time minus the job due date, considering an early delivery (negative values) and tardy delivery (positive values)

(d) the Percentage Tardy (TDP) – the proportion of jobs completed after its due date;

(e) the Mean Tardiness (TDM) – the mean of the completion time minus the job's due date, when the job is tardy.

4.4 RESULTS

To better interpret simulation results and fair comparison of the systems, we exhibited data in performance curves, also known as operating characteristics curves (OLHAGER; PERSSON, 2008). Other comparisons of PCS (e. g., Carmo-Silva *et al*. (2020), and Thürer *et al*. (2017a)) have used these performance curves. When using performance curves, each point in the curve represents the mean of each performance indicator. To test the significance of the difference between each mean, we used the Analysis of Variance (ANOVA), detailed in Appendix C. The dispatching rule and indirect load are considered a nested factor of the order release mechanism in that analysis.

For the performance measure considered, all factors and their interactions were considered significant, at the 95% significance level. The number of cards used to represent the quantum in orders has the highest F-ratio level for all performance measures, so the experimented levels have an enormous difference. The quantum limit presents the second-highest value of F-ratio that indicates a different behaviour of the system for the tested values. The order release mechanism the third-highest value supporting the nested factor's difference. In the nested factor, the dispatching rule presents peculiar differences. When the scenario does not limit the system and represents the push system, the difference of the dispatching rule is 0.17% for DTM, 0,18% for SFTT, and already 30% for LAD and TDP. However, when the original POLCA and RF-POLCA uses the same conditions, with eight cards as the quantum limit, the difference is 413.33% for DTM and 457.77% for SFTT. For the IL-POLC, the main difference is for DTM, 321.03%, because the SFTT keeps similar values (difference of 12.90%). Because of those differences, we organised results by dispatching rule to focus on the order release mechanism and the indirect load norm, only for IL-POLC.

The following subsections present the performance curves organised by dispatching rules to aid in comparing the order release mechanism. Section 4.4.1 shows the effect of different quantum representation used in the cards to separate the difference expected for the curves. In addition, section 4.4.2 exhibits the impact of the release period length on the system. Section 4.4.3 exposes the effect of the indirect load condition for release without the authorisation element for the three systems. Section 4.4.4 presents if IL-POLC may substitute the systems or just a complement for the original POLCA and RF-POLCA.

4.4.1 The effect of the quantum representation

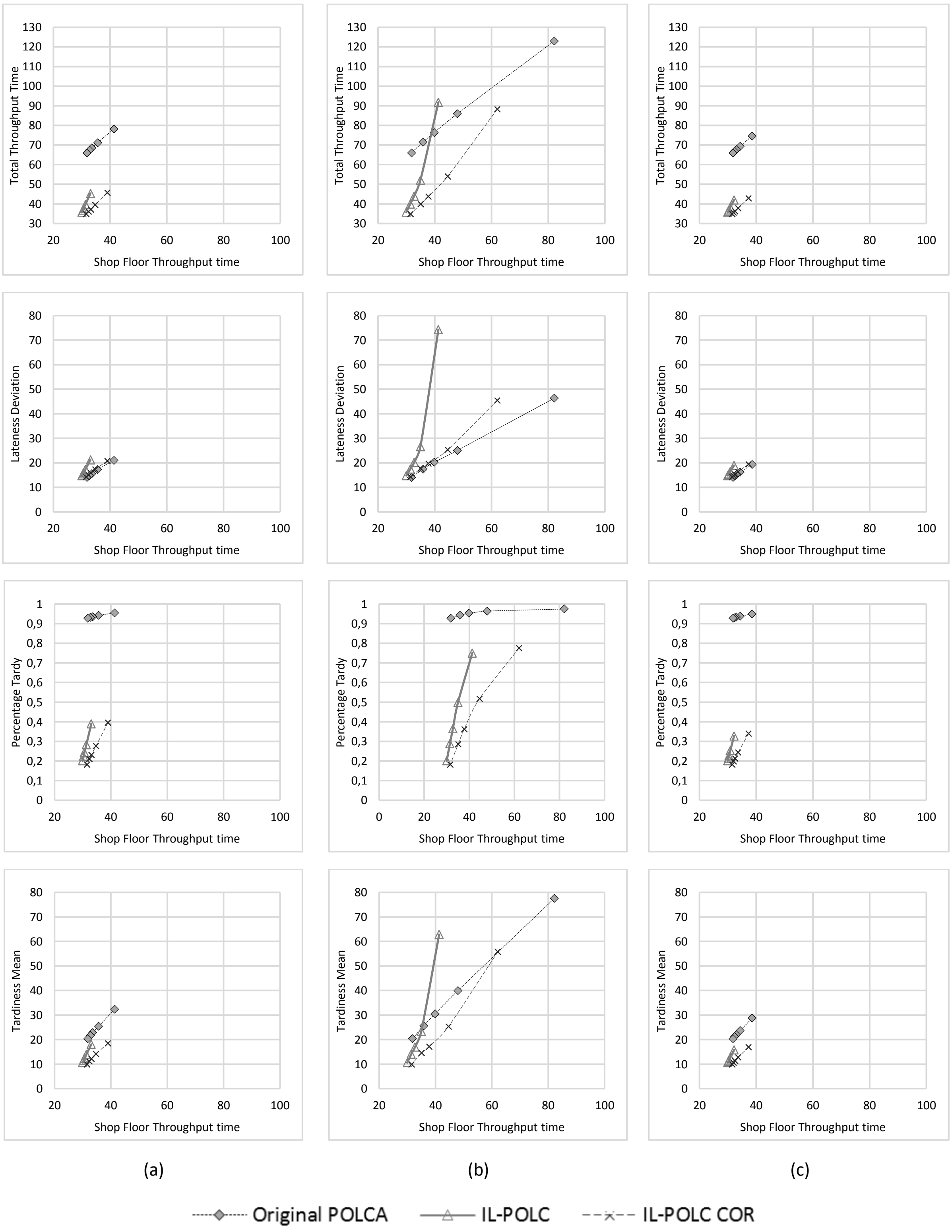
This section performance curves contrast three quantum representations: a) using three cards, b) using five cards, and c) using the load-based form to represent the quantum accurately. For the curves presented, the fixed factors are: release period length is five time-units, and indirect load norm is 14 time-units for IL-POLC. Each point plotted on each curve in both figures represents one level of the quantum limit, as the number of cards for the loop (8, 10, 12, 14 and infinite); the position to the extreme left is the highest level of the quantum limit. Figure 4.3 shows the performance curve for EODD as the dispatching rule, and Figure 4.4 presents the curves for the FIFO rule.

Figure 4.3 compares the performance curve, using the EODD rule of a) original POLCA, b) IL-POLC using classical aggregating load method, and c) IL-POLC using the corrected aggregating load method (IL-POLC COR). Regardless of the quantum representation, IL-POLC outperforms the original POLCA and the IL-POLC COR once it reduces the Total Throughput Time (approximately 45.0% for both aggregations of IL-POLC), Percentage Tardy (a mean of 71.5% reduction) and Shop Floor Throughput Time (mean of 10.8% for classical approach, and mean of 2.3% for the corrected approach of IL-POLC). The original POLCA shows higher values than IL-POLC and IL-POLC COR of Total Throughput Time and the Percentage Tardy by the effect of the authorised date for processing. When the PCS uses five cards for quantum representation with the same quantum limit, the three systems deteriorate the performance measures. This detriment does not occur when using three cards or load-based quantum representation.

Figure 4.4 compares the performance curve, using the FIFO rule of a) RF-POLCA, b) IL-POLC using classical aggregation load method (or method B), and c) IL-POLC using the corrected method B (IL-POLC COR). Regardless of the quantum representation, The IL-POLC and IL-POLC COR outperform better than the RF-POLCA by reducing the Shop Floor Throughput Time (a mean of 32.8% for classical approach and 47.4% for the corrected approach of IL-POLC) and Percentage Tardy (approximately 58% for both). A negative aspect of IL-POLC and IL-POLC COR is the detriment of Mean Tardiness (a maximum value of 22.6%) compared with RF-POLCA (a maximum value of 10.8%), considering the maximum level of tightness tested. When the quantum representation uses five cards, the IL-POLC and IL-POLC COR present great harm of Total Throughput Time (approximately 176% and 146%, respectively), Lateness Deviation (326% and 250%) compared with RF-POLCA (121% for

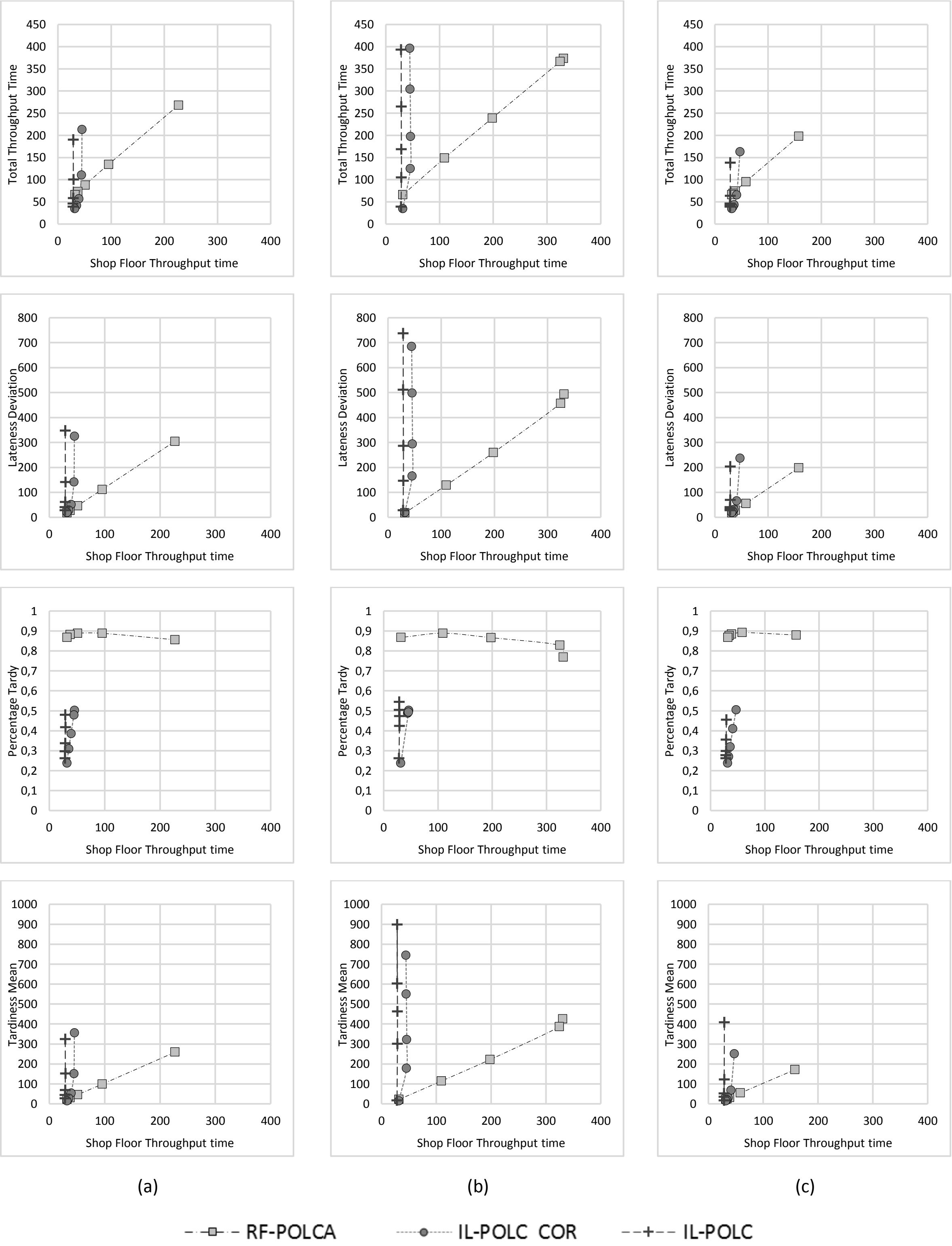
DTM, 301% for LAD). For the FIFO rule, although the IL-POLC and IL-POLC COR reduce the percentage of tardy (60% and 21%, respectively), the tardy jobs have higher tardiness than using RF-POLCA (360% and 434%).

Figure 4.3 – Performance curve of the order release mechanism using EODD: a) 3 cards for operation, b) 5 cards per operation, c) exact load representation



(a) (b) (c)

Figure 4.4 – Performance curve of the order release mechanism using FIFO: a) 3 cards for operation, b) 5 cards per operation, c) exact load representation



According to these results, representing quantum with three cards is a good approximation for the load-based version. Therefore, all the results following present quantum by three cards.

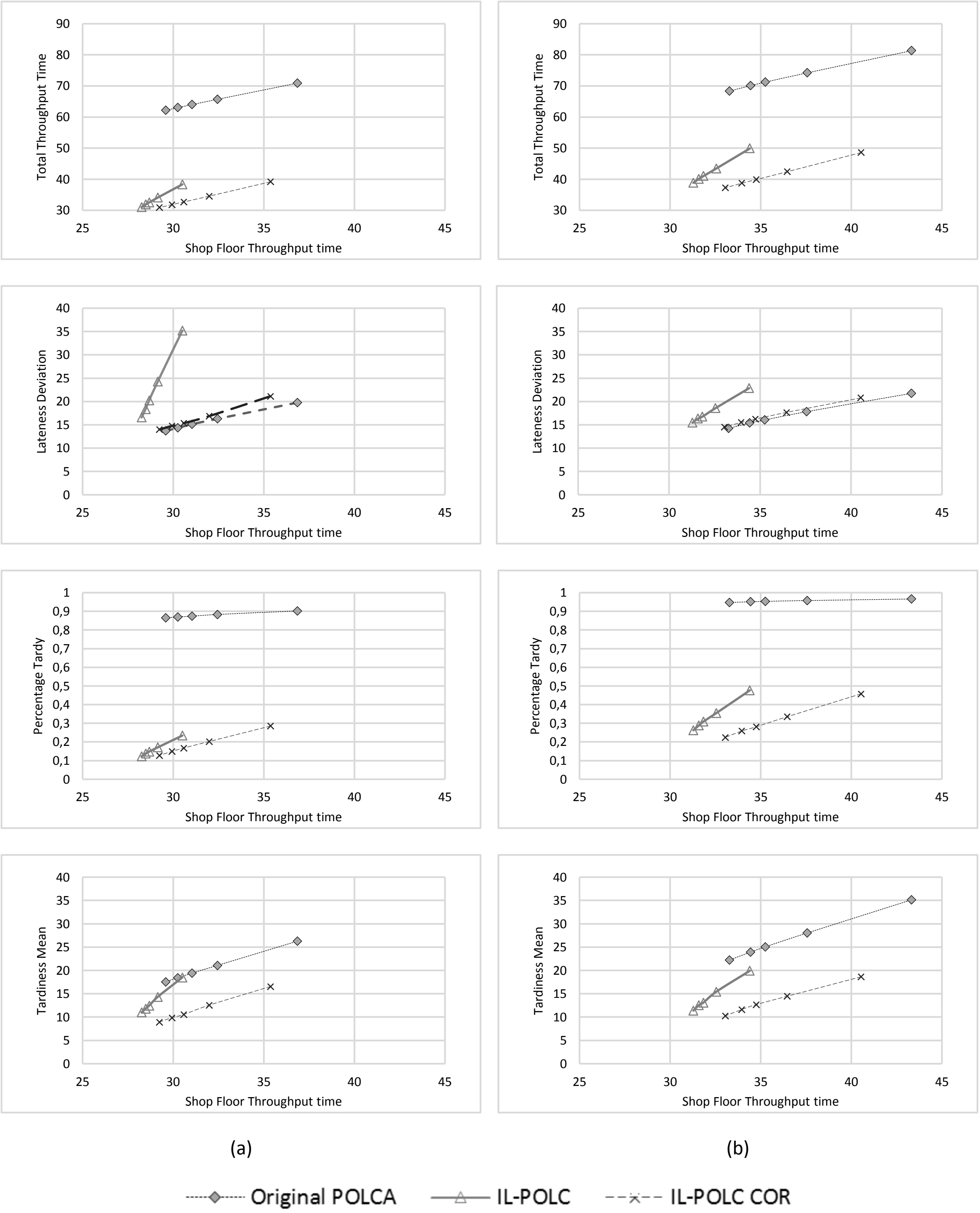
4.4.2 The effect of the release period length

This section shows how the release period length affects the systems. Results present the extreme values experimented: a) 3 time-units, and b) 6 time-units. To aid the comparison, we present the curves using three cards for quantum representation and using 14 for indirect load limit for IL-POLC and IL-POLC COR. Figure 4.5 exhibits the results for EODD as dispatching rule, and Figure 4.6 exhibits the results for the FIFO rule.

For the EODD rule (Figure 4.5), IL-POLC presents a better Total Throughput Time (reduction of 48%) and Shop Floor Throughput time (reduction of 9.1%) than IL-POLC COR and the original POLCA. As presented in section 4.1, the IL-POLC and IL-POLC COR reduce the Percentage Tardy (approximately 80%) compared with the original POLCA. The negative aspect for the IL-POLC is the increase of Lateness deviation (42% for the classical approach of IL-POLC) when the release period length is 3 time-units. For this measure, IL-POLC COR has a similar trend to the original POLCA. According to results, increasing the release period length decreases the performance (an overall average of 24%), independently of the order release mechanism used.

For the FIFO rule (Figure 4.6), IL-POLC performs better in Total Throughput Time (39.9%) and Shop Floor Throughput Time (27.1%) than the IL-POLC COR and RF-POLCA as shown in section 4.1. Increasing the release length also increases the Total Throughput Time and the Shop Floor Throughput Time (an overall mean of 25%). This increase affects more RF-POLCA than IL-POLC and IL-POLC COR. As exhibited for the EODD rule, incrementing the release period length decreases the performance of the measures used.

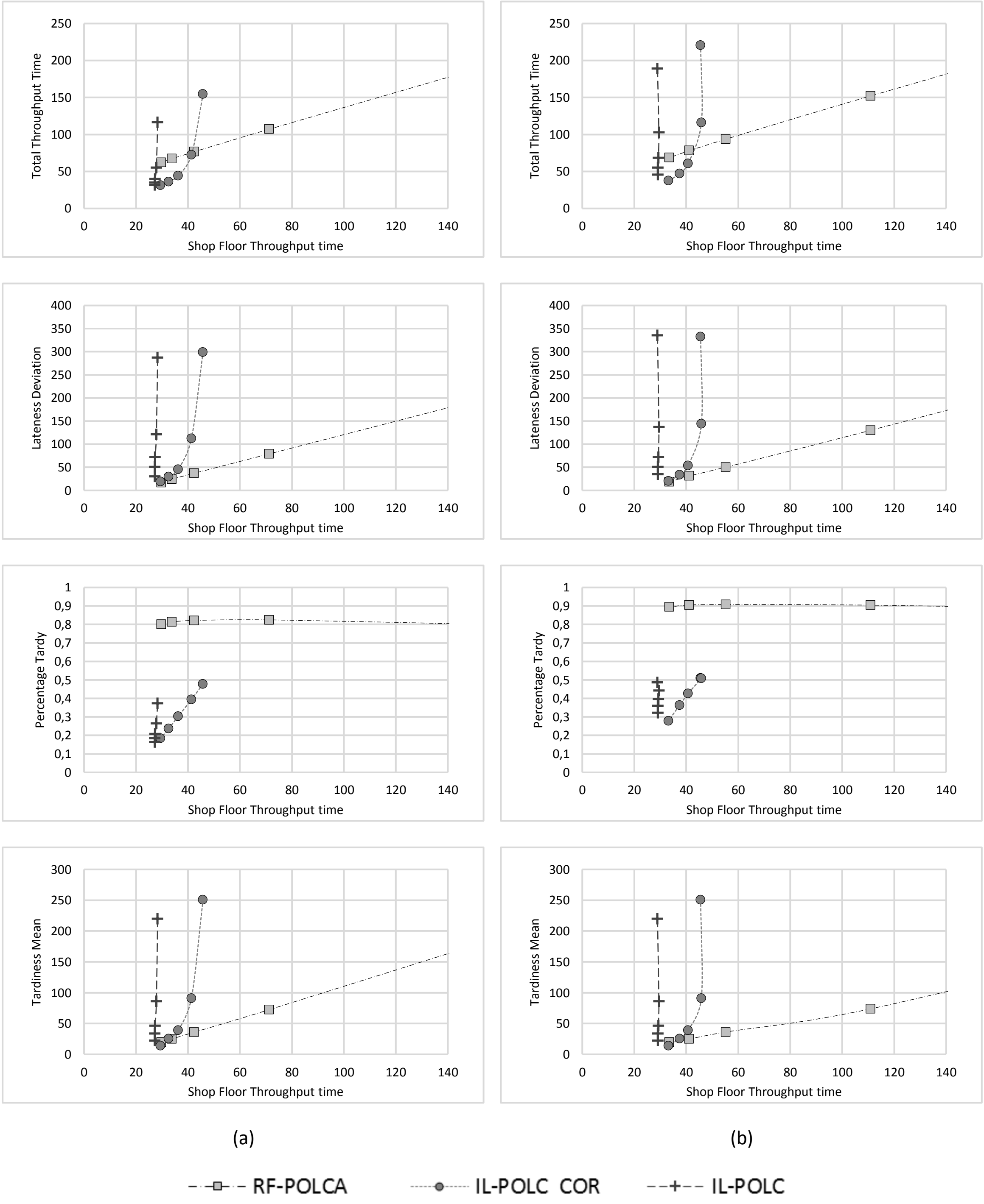
Figure 4.5 – Performance curve with three cards max per operation, using EODD considering: a) Release length of a) 3 time-units, and b) 6 time-units



(a)

(b)

Figure 4.6 – Performance curve with 3 cards max per operation, using FIFO considering Release Length of a) 3 time-units, and b) 6 time-units



(a)

(b)

To evaluate the results’ robustness, we also performed the same experiments for the other three different levels of indirect load (8, 10 and 12). As a result, the curves presented a similar behaviour to those levels.

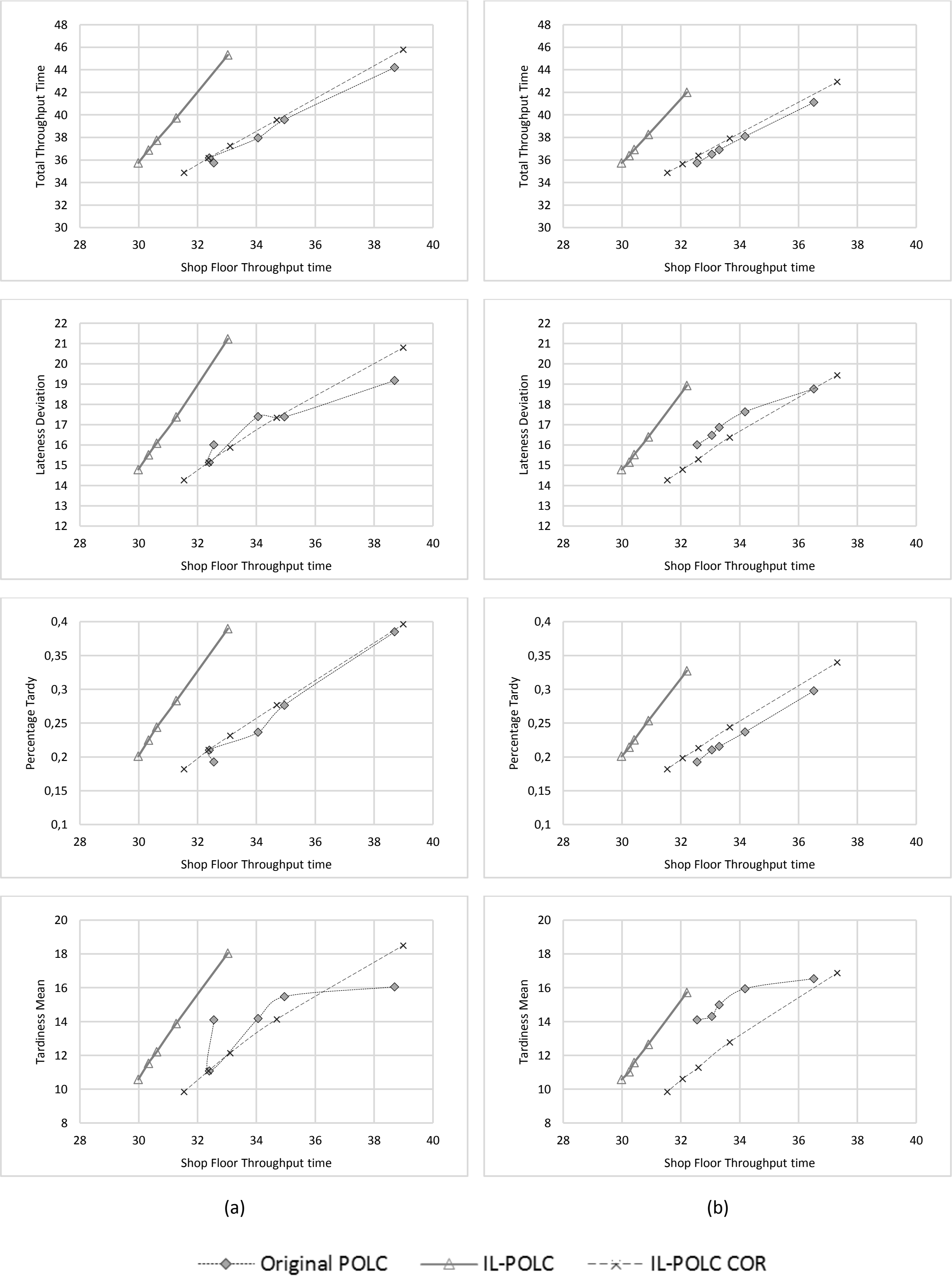
4.4.3 The effect of the indirect load condition for order release instead authorisation lists

This section shows results comparing the original POLC and RF-POLC (without the authorisation element) with IL-POLC. For this comparison, POLC and RF-POLC are the equivalents of the original POLCA and RF-POLCA, but in this section, they do not use release dates by authorisation lists because they consider that a job is previously authorised. For this section, the experiment focuses on estimating the effect of the indirect load condition (in IL-POLC) compared to the system without that condition (POLC and RF-POLC). These results follow the pattern of previous results for the quantum representation. The use of five cards per operation level is not shown in this section because it has the same detriment presented in Figure 4.3 and 4.4, in section 4.4.1. Each point plotted on each curve in both figures represents one level of the quantum norm limit (8, 10, 12, 14 and infinite); the position to the extreme left is the higher level of the quantum norm. Figure 4.7 exhibits the curves for the EODD rule, and Figure 4.8 displays the curves for the FIFO rule.

The difference presented in Figure 4.7 of the IL-POLC and original POLC is how the curves are nearer than section 4.4.1. The behaviour of the IL-POLC COR is very similar to the original POLC. IL-POLC outperforms both compared systems for EODD rule, as presented in previous sections (similar values in DTM, but 9.9% reduction for classical approach, and 1.2% reduction for corrected approach in SFTT). As expected, removing the authorisation element of the original POLCA for this model improves the performance shop. For Tardiness mean, original POLC shows a different pattern for the highest quantum limit values, using cards for representation. This reference is observable in the exact representation.

In Figure 4.8, results confirm the effect of the quantum representation when using the FIFO rule. The curve of RF-POLC using cards is significantly harmed compared with the exact load representation (564% increment of SFTT for 3 cards, and less than 1% for exact load representation). The behaviour presented by RF-POLC is the same as the IL-POLC for the exact load representation. For both quantum representations shown, the IL POLC COR presents the worse performance for the FIFO rule.

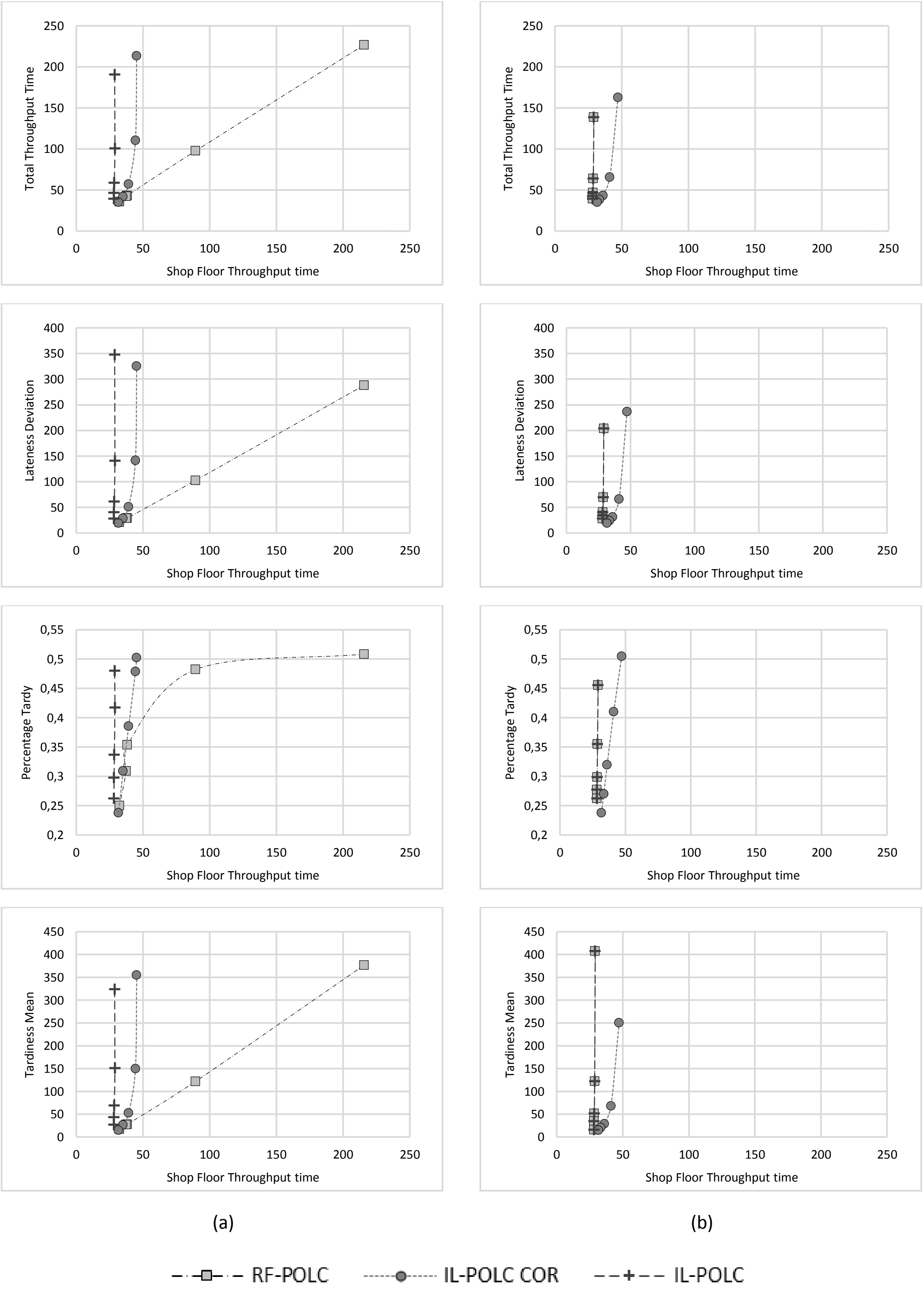
Figure 4.7 – Results using EODD without authorisation element: a) 3 cards for operation, b) exact load representation



(a)

(b)

Figure 4.8 – Results using FIFO without the authorisation element: a) 3 cards for operation, b) exact load representation



(a)

(b)

Figure 4.9 and Figure 4.10 present the curves of the original POLC and RF-POLC (without authorisation element), respectively, compared with IL-POLC to contrast the effect of the release length period. Those figures only show the quantum representation by three cards because the other representations have similar conclusions. The difference in curve pattern by dispatching rule is the same for section 4.4.2. For both figures, the release period length does not modify the pattern, and it moves the curve to the right. Then, its effect only increases the value of all the performance uniformly.

Figure 4.9 shows that the release length period does not create other difference for the original POLC and IL-POLC COR for the EODD rule. DTM present similar values (less than 1.0% of difference), but IL-POLC reduces the SFTT (7.8% of reduction with 3 as period length to 12% with 6 as period length). The two curves continue the same behaviour presented in previous sections. For the model tested, the release lower timing has better results for all the measured performance.

Figure 4.10 shows that the release period length scales the curves for the extreme values of quantum limit using the FIFO rule. The three systems have a better performance with a lower value of release period length because the highest value of period length increases DTM by 51.1%, as an overall mean for all scenarios. As results in previous sections, the quantum limit influences RF-POLC behaviour. Comparing the systems, IL-POLC still presents better performance (34.8% reduction in SFTT) than IL-POLC COR (17.5% reduction in SFTT) and RF-POLC.

Figure 4.9 – Results of 3 cards max per operation, using EODD considering release length period of a) 3 time-units, and b) 6 time-units

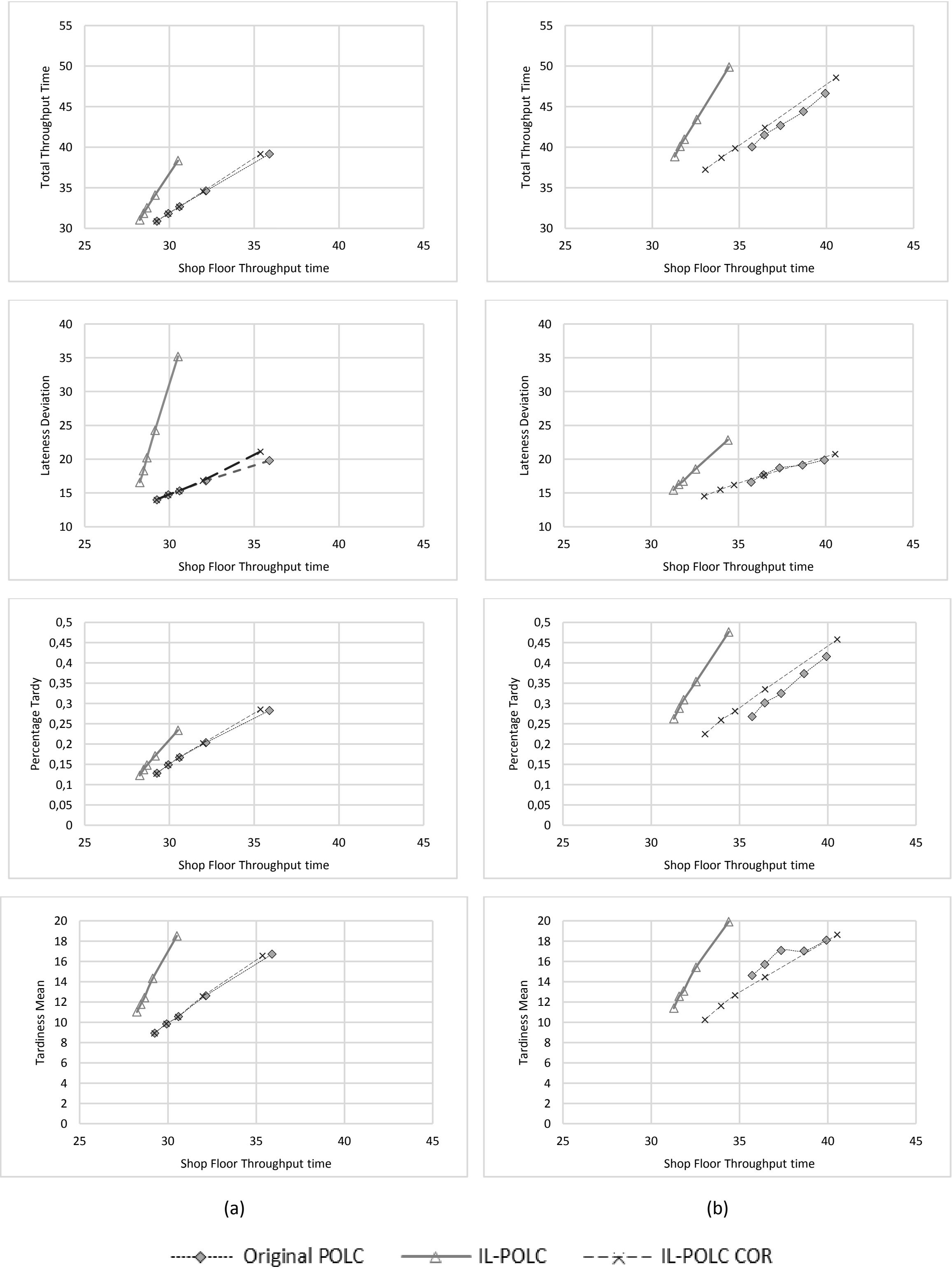
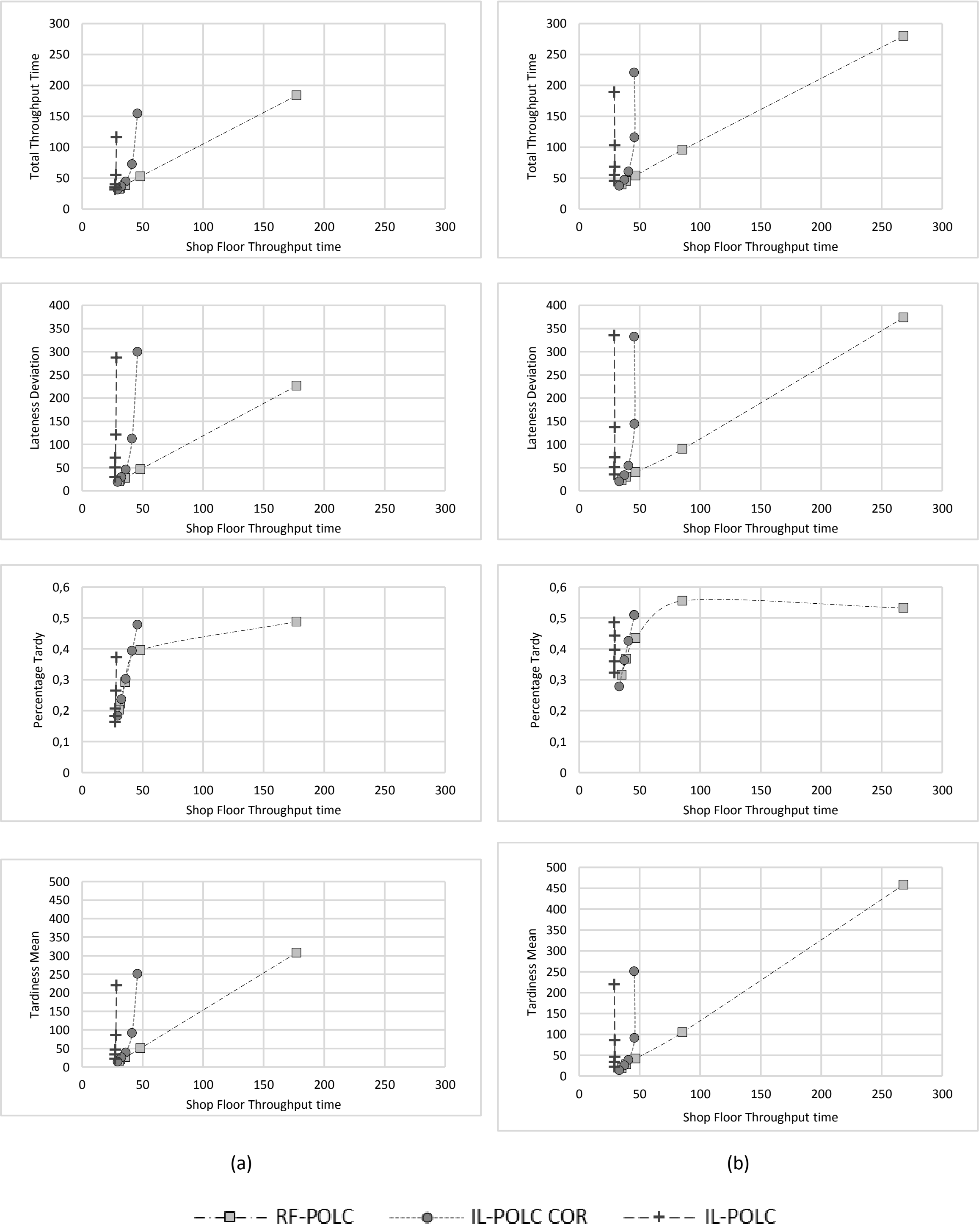


Figure 4.10 – Results of 3 cards max per operation, using FIFO considering release length period of a) 3 time-units, and b) 6 time-units



(a)

(b)

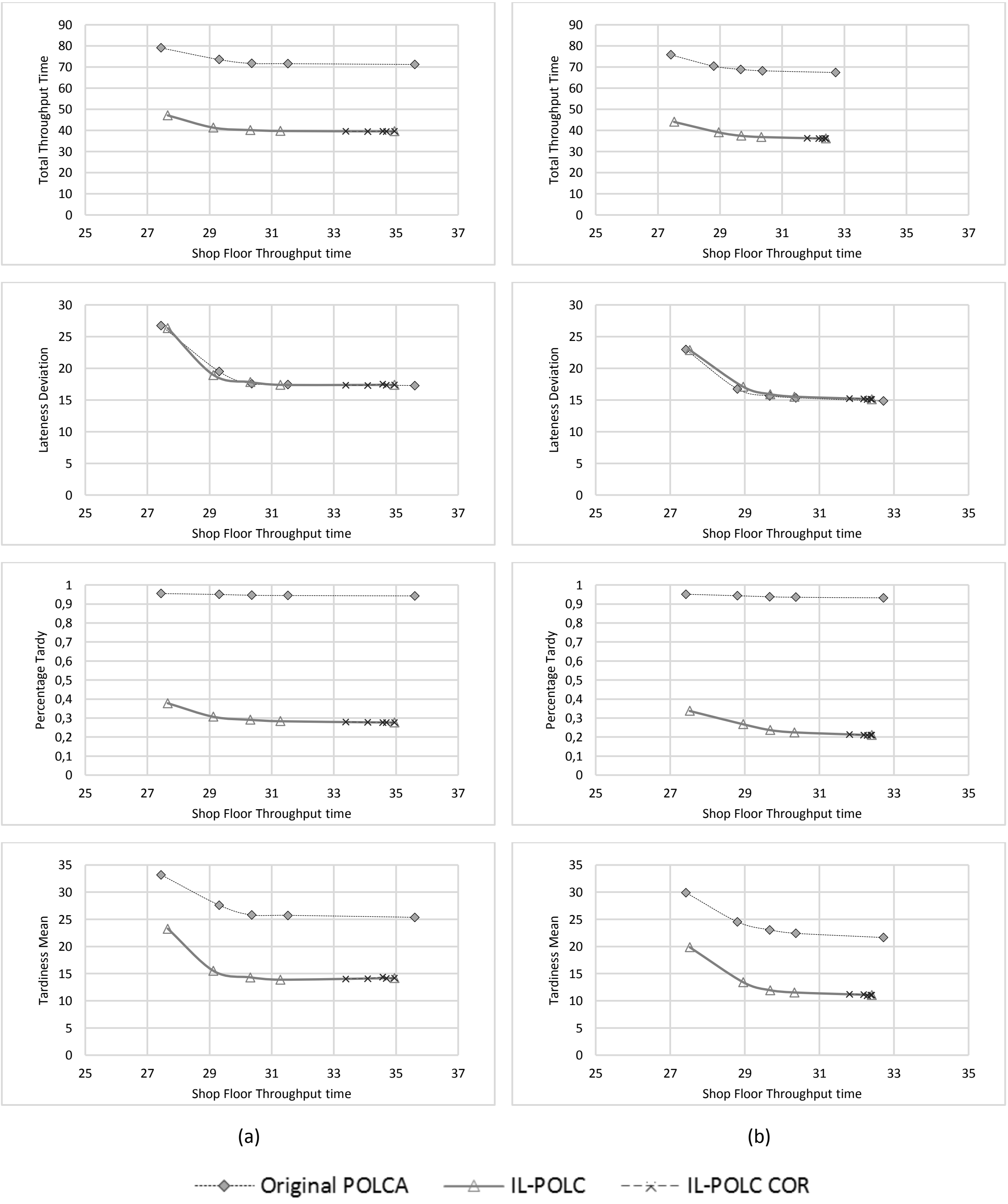
4.4.4 IL-POLC or indirect load adapted to the Original POLCA and RF-POLCA?

We show the second additional set of results to understand the indirect load norm's behaviour in the order release mechanisms. In this set, the original POLCA and the RF-POLCA add the indirect load condition to the authorisation lists by release date. They consider both conditions to release a job to the shop floor, the release date for the job is reached, and the job workload does not exceed the indirect load norm. The IL-POLC only uses the indirect load condition for releasing mechanism. This section exhibits only the performance curves with a release length of 5 time units and three cards for quantum representation to simplify the comparison. Other values present similar behaviour as compared with the shown values. This section divides results by dispatching rule contrasting only two quantum limit levels: a) 10 Polca cards and b) 14 Polca cards of the complete set (8, 10, 12, 14 and infinite). Each point of the curve represents each level of indirect load experimented. The extreme-right point on each curve represents the system without an indirect load norm. Figure 4.11 presents the EODD, and Figure 4.12 shows the curves using FIFO as a dispatching rule.

For the curves with EODD in Figure 4.11, the indirect load condition reduces the SFTT on the original POLCA and IL-POLC. However, the performance of IL-POLC, concerning DTM (43.4 % reduction), TDP (67.6% reduction) and TDM (41,6% reduction), is better than original POLCA. For IL-POLC COR, this reduction is not evident as for the other two PCSs. This reduction is higher using ten as a quantum limit rather than using 14. For Lateness deviation, the indirect load affects the IL-POLC and original POLCA similarly (Difference is 1.3% for classical approach and 10% for corrected approach). For the lowest value experimented of indirect load norm (8), there is a detriment to the other performance measures for all the tested PCSs (DTM present 45%, TDP, 73%, and TDM 44.9%). Even in this situation, IL-POLC has the best performance among the PCSs tested.

When the systems use the FIFO rule (Figure 4.12), the indirect load considerably reduces the Shop Floor Throughput Time on the three systems. This reduction is much more noticeable for 10 as a quantum limit than 14 for the three systems. The indirect load trade-off is the detriment in DTM (79.7%), LAD (159%) and TDM (222%) for the RF-POLCA and IL-POLC when using the lowest indirect load norm. For IL-POLC COR, this detriment is not noticeable (11.0% for DTM), but its reduction (1.2% of reduction) of SFTT is not as good as IL-POLC (45.5% of reduction). Therefore, for this rule, results suggest using the IL-POLC with a high indirect load norm. The critical point is that the reduction created by the indirect load norm depends on the quantum limit.

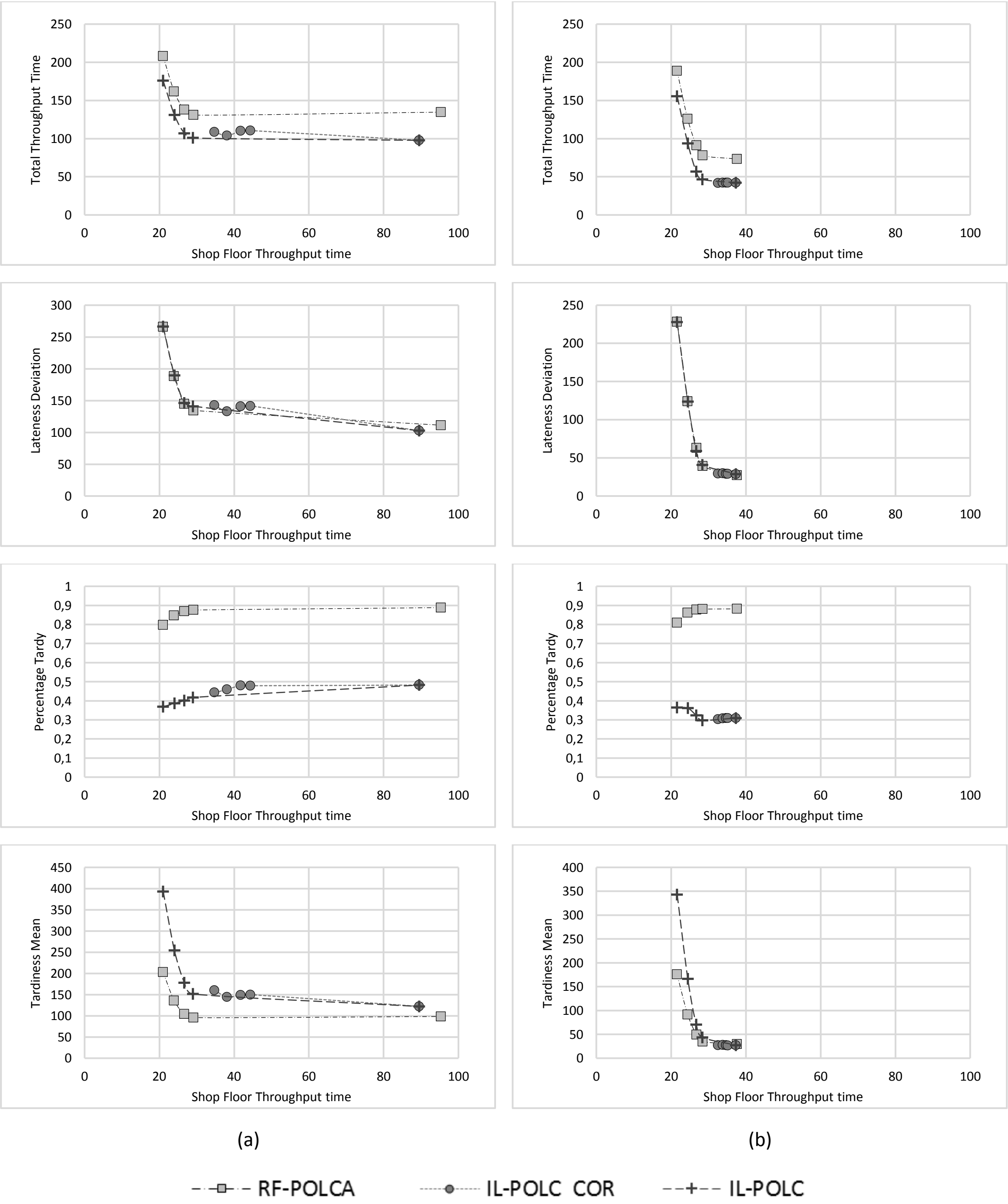
Figure 4.11 – Difference of the indirect load norm with 3 cards max per operation, using EODD, release length of 5 considering: a) 10 Polca cards on loop, and b) 14 Polca cards on loop



(a)

(b)

Figure 4.12 – Difference of the indirect load norm with 3 cards max per operation, using FIFO, release length of 5 considering: a) 10 Polca cards on loop, and b) 14 Polca cards on loop



4.5 DISCUSSION

The performance of the three order-release mechanisms studied, original POLCA and RF-POLCA, with IL-POLC, is strongly influenced by the tested dispatching rule, specifically EODD and FIFO. In the scenarios presented, FIFO has worse performance than the EODD rule. Results suggest that the FIFO rule creates considerable damage to the performance shop in the tested model because the rule does not prioritise the jobs well. Even Suri (2018) recommended the FIFO rule to create a logical order of jobs for workers in RF-POLCA. In addition, results suggest avoiding the FIFO rule for the GFS with the high variability. Results suggest using the EODD rule when there is a possibility to choose the dispatching rule among the two rules tested, as Lödding and Piontek (2017) appointed for that rule.

In EODD, the order release mechanisms that use authorisation date, original POLCA and RF-POLCA, are outperformed by IL-POLC concerning the Total Throughput Time, Percentage Tardy and Tardiness Mean. This result matches Thürer *et al.* (2019a) results, suggesting using the POLC system without the authorisation element to release a job. For the GFS represented in the model, the authorisation date reduces the jobs available to be processed. Still, this condition blocks jobs with few stations in the routing and a considerable time allowance of the due date. Land (2006) has also reported this result, and he suggests using this limit only when earliness is very costly. It is crucial to notice that even IL-POLC requires that the material for all stations is available. Therefore, managers should authorise by release date only if they want to control that material availability, as Suri (2018) indicated.

Results also show that the advantage of IL-POLC over the original POLCA and RF-POLCA is higher for low and intermediate values of indirect norm once using high values does not limit the load on the shop floor. It is essential to notice that using too tight indirect load values is also not appropriate for IL-POLC. Once the PCS releases only small jobs (jobs with short processing times in all its routing), blocking big jobs, this will cause an increase of the Tardiness Mean and the Lateness Deviation, an effect described by Carmo-Silva and Fernandes (2018). The behaviour of curves in sections 4.4.1, 4.4.2, and 4.4.3 indicates that the three systems are limiting jobs after the release moment, as Ziengs, Riezebos and Germs (2012) explained. However, even in these situations, Original POLCA and RF-POLCA do not present better DTM and SFTT than IL-POLC.

Our results indicate that the indirect load norm's value can strongly affect performance, as Ziengs, Riezebos and Germs (2012) explain, limiting the jobs at the release moment. As Thürer, Ma and Stevenson (2020) identified, a tightened load norm can negatively affect performance because the norm can block big jobs on the pool and release only small jobs to the shop floor. Those blocked jobs wait for the release, and they increase the mean of the Total Throughput Time. Fernandes et al. (2017a) define this as the Short Processing Time (SPT) effect, i.e. low values of workload norm prioritise small jobs like using SPT as a dispatching rule. As Thürer, Land and Stevenson (2014) recommended, our results suggest beginning with a large value of indirect load norm and gradually reducing the norm level until performance presents a detriment.

Concerning the load aggregation method for the indirect load, results suggest using the station load method and not use the corrected version. As the indirect load norm limits each loop separately, that load norm only affects flow between the origin and destination cell for that loop, as Suri (2018) designed for the original loop. The norm releases more jobs for the corrected version than the station load method because the correction reduces each job's contribution in each loop. Although Oosteman, Land and Gaalman (2000) suggested the corrected version for the load aggregation method in GFS, results indicate that using the classical aggregation load method is a better option. This choice is because the aggregation method does not use each station to cumulate the job contribution, and it separates that contribution by paired-cell loops.

For the release period length, results suggest using a shorter release period length. This result matches with Land (2006) that shorter periods have better performance than more prolonged periods. Therefore, the reduction of pool delay has a more substantial impact on measured performance than the more extended-release period's balancing effect. This result also matches Suri's (2018) suggestion that the release period can be defined in shorter logical periods, as half of the work shift. As we tested with EODD and FIFO rules, this suggestion of a short period is independent of those two dispatching rules.

For quantum representation, our results suggest that using three cards maximum per operation is a reasonable estimation of the load-based representation, but using five cards maximum per operation may need a new estimation of the quantum limit. Our results using three cards are according to the suggestion of Thürer, Land and Stevenson (2014), used for COBACABANA. The main difference for their results is that the quantum represented in IL-POLC is proportional to the station's job processing time, as Suri (2018) suggested simplifying how to manage cards. Therefore, for other quantum representation values, as five cards per operation, the quantum limit should be estimated to avoid constraining the flow. Table 4.4 presents the summary results of each factor tested in the experiment.

Table 4.4 – Summary of results for each factor

Factor	Results
Release period length	Use the lower value possible
Cards used for quantum representation	Significant factor (suggested 3)
Quantum limit on the loop (quantity of Polca cards)	A significant factor that needs attention Use high values and reduce until the damage
Order release mechanism (nested)	Use the indirect load conditions (IL-POLC) than release date (original POLCA or RF-POLCA)
Dispatching rule	Different behaviour for each one. FIFO present worse performance that EODD
Indirect load norm	A significant factor that needs attention Use large values and reduce until the damage
The Indirect load accounting method	Classical approach (for IL-POLC)

4.6 CONCLUSION

POLCA has been used to control production flow, using three elements: limiting jobs on flow by pair-cells, controlling material available for jobs, and releasing jobs by authorisation date in each station. For a more accessible release in linear and straightforward flows, RF-POLCA is proposed, which uses only the authorisation date for the first station, but this system has not been tested previously. This research proposes a release mechanism using indirect load control to adapt this concept to the original POLCA: IL-POLC. This proposal emerges to improve the release mechanism of the original POLCA to improve its performance. This research presents an experimental design using the dispatching rule, release period length, quantum representation, and the order release mechanism tested as factors to test the RF-POLCA and the IL-POLC against the original POLCA. Data was collected by discrete-event simulation, using a model previously used for other comparisons of POLCA, which considers the General Flow Shop. The results show that IL-POLC outperforms original POLCA and RF-POLCA in the tested scenarios, independently of the dispatching rule used, the quantum limit, or release period length. For IL-POLC, the indirect load norm should not be significantly tightened to have the best results, but removing the authorisation element is necessary. The IL-POLC can also use the same quantum representation of the original POLCA cards because the quantum representation by finite cards, using proportional representation to the processing time, has similar results to the load-based version. For the load aggregation method, IL-POLC presents better results when it uses the classical load aggregation approach because the indirect load norm limits the load only for a paired-cell loop instead of aggregate all the flow for a station. For release period length, short periods have better results for IL-POLC.

4.6.1 Managerial implications

Quick Response Manufacturing has Total Throughput Time as the primary performance measure. Suri (1998) proposed the original POLCA as the Production Control System to reduce the DTM in a high-variety and low-volume production environment as the make-to-order production strategy. Suri (2018) also proposed RF-POLCA to simplify the authorisation conditions for releasing jobs. According to that author, both systems are adequate for reducing DTM, controlling WIP, and managing some HVLV environment problems. However, despite these benefits, our research showed that the proposed system, named IL-POLC, outperform those systems concerning DTM and Shop Floor Throughput Time, Percentage of tardy jobs, and mean tardiness of a job. Therefore, managers implementing QRM should use the IL-POLC instead of the original POLCA and RF-POLCA. For IL-POLC, the implementers need special attention to the level of the indirect load. If the indirect load norm is very tightened, the IL-POLC will not reduce the DTM effectively, and the shop could present detriment for the other performance measures. Nevertheless, this detriment is smaller compared with using the original POLCA or RF-POLCA.

Our results suggest using IL-POLC to use a high value of indirect load norm and then tight gradually until the Total Throughput Time is injured. The quantum limit (number of cards limiting flow) also should follow that suggestion. For the load aggregation method, our results suggest using the classical aggregation load approach. For the dispatching rule, results suggest avoiding using FIFO for high variability in GFS, independently the combination selected for the PCS.

4.6.2 Limitations and Future research

A major limitation is that these results are from controlled experiments and data collected by simulation. The model used represents extreme randomness, and other patterns could be researched. Other models can consider a strongly unbalanced shop or a different demand for the loops. For empirical researches, the indirect load could be represented by cards and define how to implant IL-POLC procedures in shops. A second major limitation is that the model considers one operator available for each machine. Future research could explore the effect of the labour–machine as Dual Resource Constraint if the authorisation date is used to release labour to other machines, as the operators help in other stations as appointed by Suri (2018). In an empirical case, other research can be done in applying this indirect load norm in RF-POLCA to corroborate our results.

5 INDIRECT LOAD POLC (IL-POLC) OR COBACABANA PRODUCTION CONTROL SYSTEM FOR GENERAL FLOW SHOP? AN ASSESSMENT BY SIMULATION

Production Control Systems (PCSs) have a challenging task in the make-to-order environment: defining the order release and control capacity. For that context, we propose the Indirect Load Paired-cell Overlapping Loops of Cards (IL-POLC) that uses indirect load in paired-cells for the order release task and the POLC cards for controlling capacity. The indirect load uses concepts from the Control of Balance by Card-Based Navigation (COBACABANA) and COBA-POLCA. To test if the proposal of IL-POLC for order release is useful, we compared the performance of the three systems. Using simulation, we assess the performance of the PCSs in a high variety, make-to-order shop, using a representative model of this context. Results suggest using the paired-cell loop for release when the workload norms are difficult to estimate on the single station loop, and the shop uses the card-based version of the system. The IL-POLC presents lower detriments than the other two systems in those conditions. This effect allows the IL-POLC in small shops with a similar loop of POLCA cards to release jobs in the shop. Future research may consider applications of this system to corroborate the simulated results.

5.1 INTRODUCTION

In the context of the make-to-order (MTO), the manufacturing of products requires the ability to process diverse order specifications and to respond to a high-variety demand (AMARO; HENDRY; KINGSMAN, 1999). Because of that ability, the design of the shop floor needs to process a wide variety of job routings, and commonly are small and medium enterprises (MUDA; HENDRY, 2002). To control manufacturing in this context, the Production Control System (PCS) has the challenge to manage a high-variety routing and low demand of different products (STEVENSON; HENDRY; KINGSMAN, 2005). A vital task of the PCS in that context is to avoid long lead times but keep a reasonable utilisation level of the shop to manage lead times in that environment (KINGSMAN; TATSIOPOULOS; HENDRY, 1989).

Two of the PCS's typical tasks for the cited context is to control the WIP by an order release and to control capacity in the flow (LÖDDING, 2011). For the task of capacity control in the flow, the following PCSs have focused on solving this task: the original Kanban (SUGIMORI et al., 1977) and the POLCA (Paired-cell Overlapping Loops of Cards with

Authorisation) system (KRISHNAMURTHY; SURI, 2009; SURI, 1998). In addition, some well-known PCSs had addressed the order release task as Constant WIP control (known as CONWIP, e.g., JAEGLER et al., 2018; SPEARMAN; WOODRUFF; HOPP, 1990); Drum-Buffer-Rope from the theory of constraints (BHARDWAJ; GUPTA; KANDA, 2010; GOLDRATT; COX, 1984); Decentralised WIP Control, known as DEWIP (LÖDDING; YU; WIENDAHL, 2003) and COBACABANA (Control of Balance by Card-Based Navigation). (LAND, 2009; THÜRER; LAND; STEVENSON, 2014). A further list of modern PCS is available in Bagni et al. (2020).

Among those many alternatives, two PCS are highlighted for being the foundation for other card-based PCS to control that kind of environment (THÜRER; STEVENSON; PROTZMAN, 2016b): the POLCA and the COBACABANA systems. These two PCS are more highlighted than the others because they have been designed for the MTO context. For POLCA, literature reports some successful implementations (KRISHNAMURTHY; SURI, 2009; SURI, 2018). POLCA has been compared with other PCSs, as CONWIP (BARROS et al., 2016; BRAGLIA; CASTELLANO; FROSOLINI, 2014; GERMS; RIEZEBOS, 2010), Kanban modifications (SILVA et al., 2017) and other systems (LÖDDING; YU; WIENDAHL, 2003) with satisfactory results. COBACABANA also reports comparisons with Kanban (THÜRER; STEVENSON; PROTZMAN, 2015), and it has reported a better performance by centralising the control decision (THÜRER et al., 2019c).

For the PCS task mentioned earlier, the critical difference of POLCA, among other PCSs, is the paired-cell loop to control capacity (SURI, 1998). That author argues that the paired-cell loops allow processing jobs with downstream capacity available instead of process jobs that will be blocked in the following stations. This loop can react on downstream brakes as machine failures, lack of material or labour, and important causes for evaluating the shop's performance (PRAKASH; FENG, 2011; QI; SIVAKUMAR; GERSHWIN, 2009). This loop is only evaluated for the original system for the next pair of cells using cells' local information. Several implementations of POLCA show the effectiveness of the paired-cell loop (KRISHNAMURTHY; SURI, 2009; POWELL; RIEZEBOS; STRANDHAGEN, 2013; SURI, 2018). Those applications demonstrate how the local information and local decisions help in another kind of practical advantages. Some of those, and not limited to, are quick feedback of problems, encourages ownership for team tasks, and self-management for problem-solving as Hyer, Brown, and Zimmerman (1999) state for cell manufacturing. Fernandes et al. (2021)

demonstrate how POLCA with that loop reduces total throughput time considering assembled products.

In the order release task, COBACABANA outstands their function because it controls the shop condition for all possible routings. This system uses centralised loops between the releaser and each workstation (LAND, 2009). For this PCS, the order release is the main feature of balancing the shop load (THÜRER; LAND; STEVENSON, 2014). This order release is a critical advantage for this system when an extreme bottleneck is not present (THÜRER et al., 2017c). This system also reports a practical application that presents the real benefit of the order release task (BRAGLIA; MARRAZZINI; PADELLINI, 2020). An appointed problem for this system is the lack of the capacity control task because it does not control the flow of released jobs (BRAGLIA; MARRAZZINI; PADELLINI, 2020; THÜRER; FERNANDES; STEVENSON, 2020).

For order release, the original POLCA uses authorisation lists for each station, commonly controlled in an MRP system, using backward programming with known lead-times (SURI, 1998). For the processing-time high variability, Vandaele et al. (2008) propose the Advanced Resource Planning to update the lead-time and release dates, considering the workload shop condition. However, this proposal requires some POLCA module implementations that are not easily available for small shops. That authorisation, with fixed dates, shows to harm the Total Throughput Time in the general flow shop (THÜRER et al., 2019a). Instead of that authorisation lists, Carmo-Silva et al. (2020) propose a centralised release to POLCA that uses centralised loops for each station to release a job-based in the shop condition. Although those authors improve POLCA with a centralised release, they do not use the indirect load's paired-cell loop. As the main proposal, this chapter presents the Indirect Load POLC (IL-POLC), exposed in this doctoral dissertation. This system uses the indirect load status to release an order to the shop, considering paired-cell loops in the indirect load to aggregate the workload and when the lead-times are not easily upgradeable. The system uses the POLCA cards (the POLC element) for the capacity control task. The IL-POLC has reported better results compared with the original POLCA and Release and Flow POLCA (chapter 4). As a proposed mechanism, the IL-POLC does not report comparisons with other PCSs.

A third system studied combines the two mentioned PCSs that Thürer, Fernandes, and Stevenson (2020) presented. They combine the order release of the COBACABANA with the paired-cell loop to control the POLCA system's capacity in the system COBA-POLCA. This combination reports better results than the use of the two isolated systems. According to this combination, the COBA-POLCA uses centralised and single-station aggregation loops to control for the order release, and it uses paired-cell loops to control capacity.

A question arises if the paired-cells loop for the indirect load on the order release task can exploit the main advantage of the original POLCA. This study then compares IL-POLC with COBACABANA (just the order release task) and COBA-POLCA (the COBACABANA order release task with POLCA capacity control task) to define the effect of using paired-cell loops to aggregate the indirect load. For this comparison, this study uses discrete-event simulation to evaluate these PCS's performance to gain insights to improve the PCS functions in the studied context. For practical implications, this gap would help to clarify which improvement could be more beneficial to implement.

The remainder of this chapter is structured as follows. First, section 5.2 presents the systems and their main elements, and section 5.3 describes the simulation model and the experiment used for comparison. Then, focused on results, section 5.4 shows the comparisons, and section 5.5 displays a discussion for results. Finally, section 5.6 presents the main conclusion of the three systems compared.

5.2 BACKGROUND FOR THE PCS

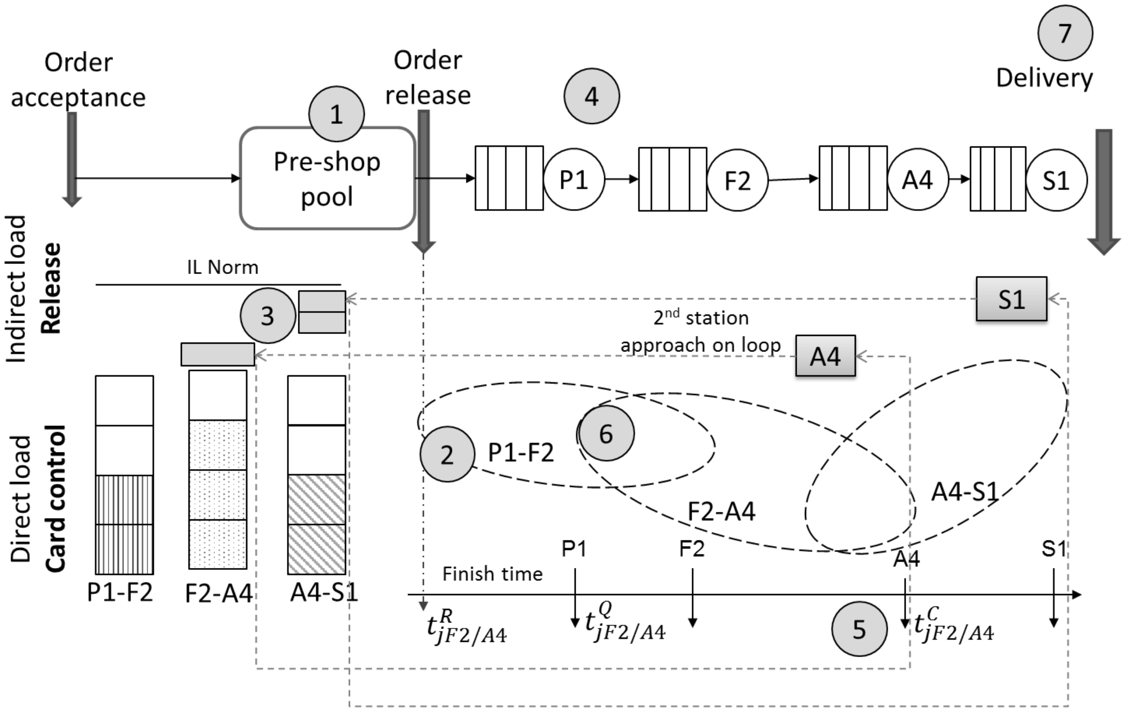
This section overviews the IL-POLC, COBACABANA, and COBA-POLCA systems to establish the comparison objectives, considering their order release and capacity control functions. IL-POLC, as a proposed variant of POLCA, is described in section 5.2.1. The COBACABANA is described in section 5.2.2, and the COBA-POLCA is presented in section 5.2.3. The three systems are presented for their use on the GFS.

5.2.1 Indirect Load (IL) POLC

The POLCA system links two consecutive cells in the routing of orders using a card's loops of controls. This card identifies the origin cell and destination cell (e.g., P1/F2 means the cell P1 to cell F2). The system created by Suri (1998) uses a release mechanism that authorises the processing after a date. The IL-POLC substitutes this authorisation date by a centralised release evaluating the indirect load norm to release a job. The tasks of IL-POLC are a) the control of the direct load on the flow (the control capacity task) and b) the indirect load release mechanism. The indirect load (or the upstream load) in a station is the amount of a load of released jobs in a queue of upstream stations (LAND; GAALMAN, 1998). The direct load is the amount of load actually in the queue of the station. The aggregate load of a station is the load released until the station evaluated. This concept is the sum of the indirect load and the direct load. IL-POLC releases a job to the shop floor when the indirect load of each loop does not exceed the load norm. This indirect load does not consider the first loop at the release moment because the first loop is considered as the direct load. The direct load in the job routine is controlled by POLCA cards, as in the original system.

Figure 5.1 summarises how to apply IL-POLC to the example of the original POLCA presented in Suri and Krishnamurthy (2003). First, the job arrives in the pre-shop pool (1). When the system triggers the order release, the releaser evaluates if there are available cards for the first loop (2) and if the indirect load of the following loops does not exceed the load norm (3). If there are no available cards or the job exceeds the indirect load norm, the releaser returns the job to the pre-shop pool and evaluates the next job in the backlog list of the pre-shop pool. If both conditions, card available for the first loop and indirect load norm not exceeded, the releaser sends the job to the first station in the job routing, releasing the job in the shop floor (4). For example, the first station is P1, that the first loop is P1/F2. When the station finishes that job, e.g. the station P1, the indirect load of the following loop (the loop F2/A4) is updated and discounted for that loop (5). This discount is because the loop F2/A4 become a direct load when the job finishes P1, and the job can continue if there are cards available for F2/A4 loop (6). As the job continues the routing, steps (5) and step (6) repeat the evaluation every time that the following station of the job routing finishes the processing for that job. For example, steps (5) update the loop A4/S1 when the station F2 finishes the job, and (6) evaluate available cards for the A4 station. The update step (5) and the evaluation step (6) are repeated considering the followings loops until the last station of the job routing finishes the job (7), and the job can be delivered. For the sequence task, a dispatching rule is suggested as a simple solution than complex algorithms to prioritize jobs in queues.

Figure 5.1 – The IL-POLC system



Source: authors

For the indirect load release mechanism, as a concept lent from WLC, the parameters define the indirect load loop and the load accounting method. The indirect load (or the upstream load) in a station is the amount of a load of released jobs in upstream stations' queue (LAND; GAALMAN, 1998). For IL-POLC, the loops consider the indirect load in the same loops of direct load used to control capacity. The loop considers only the destination cell's indirect load inside the upstream loop, known as the 2nd station approach. This loop design is a paired-cell because DEWIP uses a similar model for local workload estimation (LÖDDING; YU; WIENDAHL, 2003); and because Fernandes et al. (2017b) reported it as the best load accounting approach for loops. The mechanism releases the entrant job if this job does not overload the indirect load norm for all loops in the job routing. The shop uses a gateway to evaluate the jobs for each release period length. The use of that gateway for releasing jobs in POLCA has improved performance in centralised loops (FERNANDES et al., 2019). IL-POLC uses the same gateway to evaluate the load status in the shop. The difference is that the releaser does not evaluate the first loop of cards because the first station of that first loop assesses the job on the queue to assign cards.

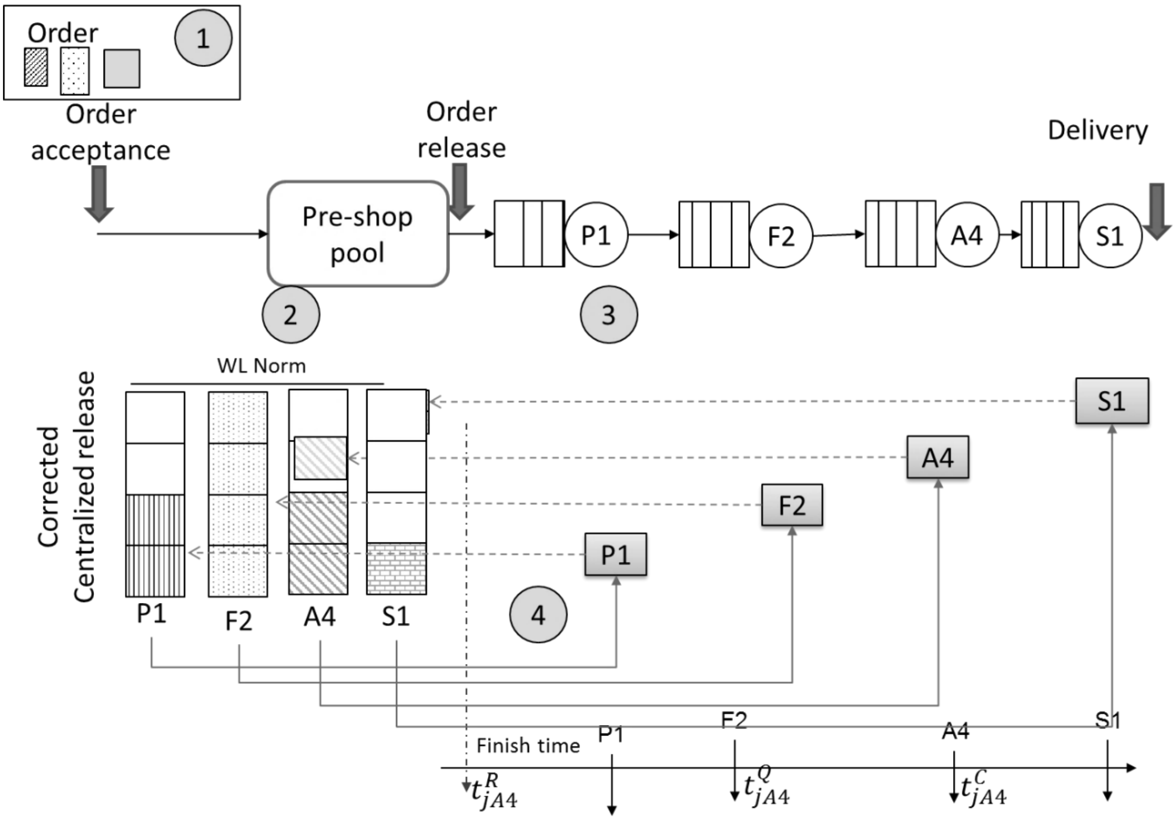
For load accounting methods for GFS, Oosterman, Land and Gaalman (2000) present the classical aggregation load approach (known as method B). Method B aggregates the load contribution for stations of a job j from the release date (t_{js}^{R}) until the departure date of the station s (t_{js}^{C}). In this method, the indirect load at the station s is considered from the release date until the arrival date at that station. As shown in Figure 5.1, the account for the loop F2/A4 considers job load as an indirect load from release ($t_{jF2/A4}^{R}$) until the job arrives at the loop F2/A4 ($t_{jF2/A4}^{Q}$). After that moment, the account considers job load as direct load until the second station of the loop, e.g. A4 station, finishes the job ($t_{jF2/A4}^{C}$). This concept of method B has been previously used in G-POLCA (FERNANDES; CARMO-SILVA, 2006). From the results of chapter 4, IL-POLC is better using the classical approach than the corrected approach for aggregation load. For this point, this comparison uses only the classical approach.

5.2.2 COBACABANA system

The COBACABANA system is the card-based PCS proposed by Land (2009) to operationalise the WLC approach. The approach WLC controls the jobs released to the shop to keep the planned lead times in a self-regulating way (BECHTE, 1994; HENDRY; KINGSMAN, 1991; WIENDAHL, 1995). In this approach, COBACABANA controls the amount of WIP in each station using a loop between the station and a central release. The main advantage highlighted for this system is the workload balance in each station. As the WLC application, it uses the concept of aggregate load (KINGSMAN; TATSIOPOULOS; HENDRY, 1989; LAND; GAALMAN, 1996), that considers the direct load (the amount of the load released currently waiting in the station's queue) and the indirect load (load for the upstream station, according to the order's routing).

In the refined version (THÜRER; LAND; STEVENSON, 2014), COBACABANA controls the direct load using two cards in a loop for a central release with the station: the release card in the central planner and the operation card. Figure 5.2 shows the steps in how this system works. When a job is accepted, it arrives at central planning. As the first step, the planner chooses the card size to represent the amount of work done in the station, based on processing time. Then, to release jobs, the planner attaches each station's operation cards to the job, according to the job routing (1). Next, the planner verifies the amount of aggregated workload released for each station on the planning board, including the new job (2). He can only release the job if the aggregate workload exceeds the Workload norm, but if it exceeds, he removes all the cards from the planning board for that job. Subsequently, the planner releases the order to the station depending on its routing. Then, the station processes the assigned operation for that order (3). As a work centre completes its corresponding transaction, the operation card returns to the central planner, and the planner withdraws the card released for that operation from the central board (4). Then, the planner executes steps two to four until all operations are complete (5). Figure 5.2 illustrates how this system works to control for the same example of POLCA. In this example, for that moment, if a job arrives requiring the F2 work centre, the planner will not release it because it will exceed the workload norm. For COBACABANA, the load accounting method preferred for the GFS is the corrected method, as described in the POLCA section.

Figure 5.2 – COBACABANA card loops mechanism for control

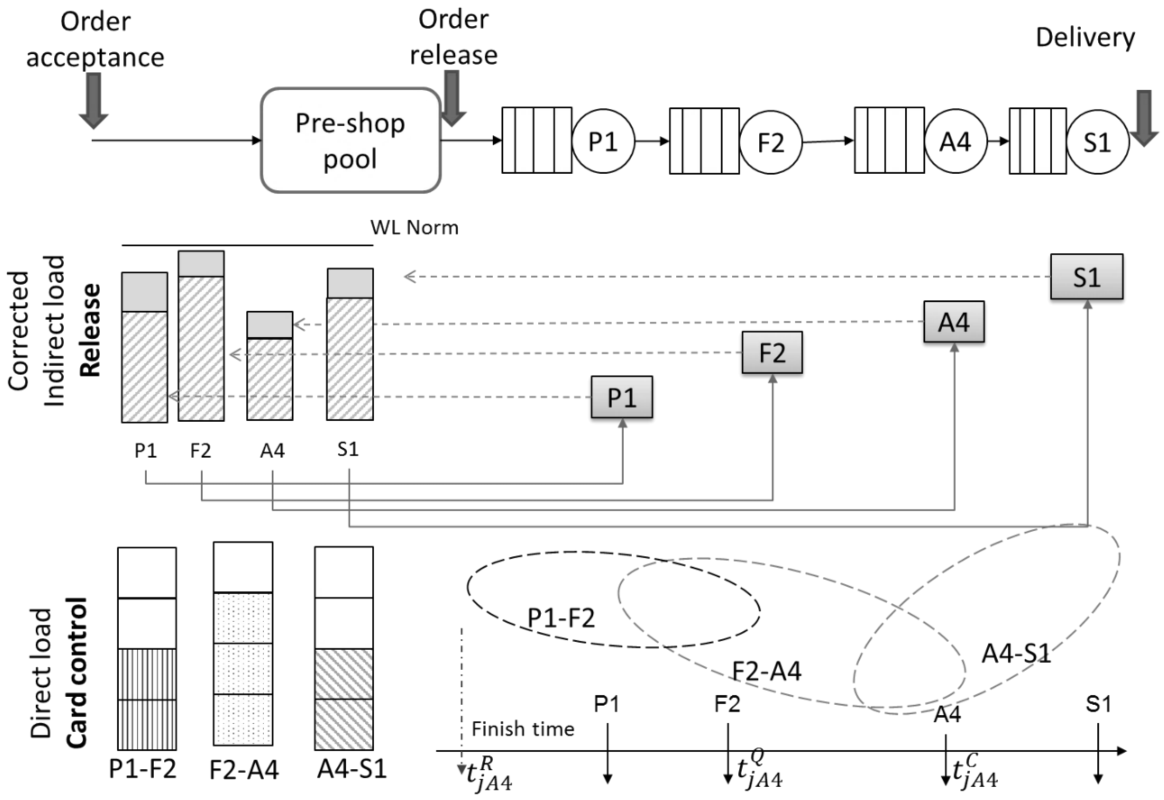


Source: adapted from Thürer *et al*. (2014)

5.2.3 COBA-POLCA

The COBA-POLCA combines the order release functions and capacity control (THÜRER; FERNANDES; STEVENSON, 2020). The order release uses the COBACABANA system. After the job arrived, it waits in the pre-shop pool to be released. The job is prioritised according to the pool sequence rule. To be released, the releaser evaluates the current contribution to the workload for each station. If the job's contribution exceeds the workload norm, the job is skipped, and the next job is evaluated; but if the contribution does not exceed, it is released. After the job is released, then the POLCA controls the capacity for the paired-cell loops. The POLCA does not use the authorisation date to process the station because this authorisation is done by releasing the order in the shop. Finally, the job arrives at the input queue of the station. In the station, the availability of POLCA cards for that job is evaluated. The job is processed if there are cards available for the loop with the following station in the job routing. If there are no cards available for that loop, the job waits on the queue station until getting a card. After the job is processed, the workload of loops is updated. Figure 5.3 shows that combination of the POLCA.

Figure 5.3 – COBA-POLCA mechanism of control



According to Thürer, Fernandes, and Stevenson (2020), the COBA-POLCA has better performance than the isolated systems COBACABANA and POLCA. For that system, the main advantage is using the best performance of the order release balancing the shop load and controlling the station's capacity by the paired-cell loop. For this study, the difference from IL-POLC to COBA-POLCA is in using the indirect load in paired-cells loops to use the same concept of capacity control to release a job in the shop. Table 5.1 summarises the tasks of each PCSs to review their differences.

Table 5.1 – Task summarise of the studied PCSs

System \ Task	Order release	Capacity control
IL-POLC	Release a job evaluating the indirect load of all loops in the routing for that job. The loops aggregate the load using the paired-cell account. Thus, each level of the indirect load contains a load of each pair of cells, using the 2nd station approach.	Limit the number of jobs between the paired-cell loops
COBACABANA	Release a job evaluating the level of the aggregate load (direct and indirect load) of the stations contemplated in that job's routing. Each level of aggregate load contains the load of every single station.	The job is no limited because the system focuses on order release.
COBA-POLCA	Release a job evaluating the level of the aggregate load (direct and indirect load) of the stations contemplated in that job's routing. Each level of aggregate load contains the load of every single station.	Limit the number of jobs between the paired-cell loops

5.3 RESEARCH METHOD: EXPERIMENTS BY SIMULATION

For evaluating the difference for both systems, we propose an experiment by simulation. For comparisons in the shop floor level, Jeon and Kim (2016) recommend the discrete-event simulation to measure how much each factor affects the performance. The model described in section 5.3.1 represents one MTO shop using the general flow shop, as the conditions aimed in this study. The parameters for the similarities in both PCS are described in section 5.3.2. Finally, section 5.3.3 presents the summary of the experiments, including the measure performance used for the comparison.

5.3.1 Model shop and job characteristics

The model used for representing the MTO environment in a general flow shop uses as base the model presented on Oosterman, Land and Gaalman (2000). That model allows comparing with other literature results for model validation used in further refinements of POLCA and COBACABANA (BARROS et al., 2016; THÜRER et al., 2017a, 2017c; THÜRER; STEVENSON; PROTZMAN, 2015). We implemented the simulation model using FlexSim simulation software. The shop contains six cells modelled as a single station, considering the same cells with a constant capacity. The routing length varies from two to six stations, considering a uniform variation to represent at least one loop of POLCA. The model estimate that a job has an equal probability of visiting all stations, but no-reentrant flow is allowed.

As the base model presented in other comparisons, the processing time follows a truncated Erlang with shape parameter two and a mean of 1 time-unit truncated in a maximum value of 4. Setup is considered sequence-independent, and its time is part of the processing time. The inter-arrival time follows an exponential distribution with an average of 0,738 time units, estimating a 90% utilisation level in the represented cells. As used in the make-to-order, due dates are set exogenously between 33 to 55 time units. As this model represents an ideal variation, it considers the typical variability for a small-medium shop.

The simulation model assumes that all jobs have material available, and the information necessary for routing and time processing is known at the arrival date of the job. Furthermore, the model neglects the transportation time between cells. These assumptions have been considered in previous studies of POLCA (BRAGLIA; CASTELLANO; FROSOLINI, 2014; FERNANDES; CARMO-SILVA, 2006; HARROD; KANET, 2013; LÖDDING; YU; WIENDAHL, 2003). Table 5.2 summarises job and shop characteristics for comparison, and Table 5.3 presents the job routing percentage resultant from the model for each path. Appendix C shows the detail of the code of the simulation model.

Table 5.2 – Shop configuration for the proposed model

Variable	Review observed*
Routing variability	Random for general flow shop
Cells (work centres)	Six similar cells, modelled as a single station
Utilisation level	Estimated in 90% for each cell
Job routing	General Flow Shop using uniform distribution [2,6] for routing length, non-reentrant flow.
Due dates estimation	Constant order allowance (N) for external setting limits, using the uniform distribution from 33 to 55 time-units
Order Arrivals	Random data using exponential distribution (arrival rate depending on utilisation level from processing time mean): estimated in 0.738 time-unit/item
Processing and setup times	Random data using Erlang distribution (shape parameter: 2, mean 1.0 and max 4.0). Independent sequence for setup times (neglected at simulation estimation)

*Based on Oosterman, Land and Gaalman (2000)

Table 5.3 – Routing matrix of the GFS modelled

From / To	Station 2	Station 3	Station 4	Station 5	Station 6
Station 1	0.604	0.149	0.063	0.030	0.017
Station 2		0.601	0.150	0.061	0.028
Station 3			0.604	0.144	0.061
Station 4				0.607	0.152
Station 5					0.600

5.3.2 Parameters of common factors for both PCSs

For the comparison, the following parameters are defined: i) the release period length for the pre-shop pool; ii) the load accounting method; iii) the quantum of a card, iv) the workload norm and the limit of the number of cards, and v) the dispatching rule

For this model, the order release considered is periodic, as POLCA systems suggested to avoid nervousness for releaser (SURI, 2018). The order release length was fixed in 5 time-units, as used in other comparisons (LAND, 2006). This comparison uses the aggregate load method for the loading accounting method, as Land and Gaalman (2000) indicate for general flow shop. IL-POLC considers the workload norm for the specific pair of stations, but there are as many loops as the shop routing allows for the GFS. The third of the maximum processing time (4 time-units) is used for both systems for the quantum card to consider three cards as the maximum required for a job in a station. Each card represents the same amount of job load (e.g., 1.33 time-units) as a proportional relationship. The load-based version of the systems is used only for comparison.

For limiting the WIP, the number of cards allowed for each loop of POLCA studied is 10, 12, 14 and 16, as used in previous research (THÜRER et al., 2017a). For indirect load in IL-POLC, the workload norm was considered the same value of each loop's number of cards. For the indirect load in COBACABANA, the workload norm is the limit of POLCA cards multiplied by 1.25 as a corrected factor for the indirect load, as contemplated in Thürer, Fernandes and Stevenson (2020). The limits are 12.5, 15, 17.5 and 20. For the limit of the number of cards for COBACABANA, it was calculated the next integer value of rounding the workload norm of IL-POLC. These values are 13, 15, 18 and 20. For each loop, the norms and limits are the same because the shop is considered balanced. The shop was considered unlimited (a very high value of limit and norm) to represent the Immediate Release (IMR) without the PCSs. This experiment fixes the norm values because Thürer, Silva and Stevenson (2011) suggest this a practical solution for establishing workload norms. They also advise that the dynamic adjustment of norms, as Belisário and Pierreval (2015) proposed, is a complex solution for managers in small shops.

The jobs are prioritised in each station queue using the Earliest Operation Due Date (EODD) for the dispatching rule. The EODD uses an estimation of the station throughput time to define a station due date. The job with the earliest due date for that station is processed first. The original POLCA proposed using this rule (SURI, 1998), and it has been tested as a robust rule for scheduling reliability (LÖDDING; PIONTEK, 2017). For this comparison, the station throughput time was estimated in 1.1 unit-times as a fixed value.

For this comparison, the factors to be compared are the three production control mechanisms: a) IL-POLC, b) COBACABANA, and c) COBA-POLCA, as described in section 5.2. Table 5.4 summarises the parameters defined for the experiment.

Table 5.4 – Parameters for the PCS' comparison

Parameter	**Studied levels**		
PCS studied	**IL-POLC**	**COBACABANA**	**COBA-POLCA**
Order release task	Indirect load condition using paired-cell loops	Aggregate load condition using single station loops	Aggregate load condition using single station loops
Capacity control task	Limit Jobs in the paired-cell loops	No job limitation	Limit Jobs in the paired-cell loops
Number of cards allowed for loop	8, 10, 12, 14 and inf.	13, 15, 18, 20 and inf	8, 10, 12, 14 and inf.
Workload norm	8, 10, 12, 14 and inf.	Considered in cards limit	12.5, 15, 17.5, 20 and inf.
Quantum card	Three cards maximum per job in each station (load-based for comparison)		
Dispatching rules	EODD (1.1 u.t for each station)		
Pool sequence rule	ERD, CS, MODCS (ERD | CS)		
Cell availability	100% and 90% of station 4		

Source: Authors

5.3.3 Experimental design and performance measures

This comparison's experimental factors are a) the PCS described (IL-POLC, COBACABANA and COBA-POLCA). The PCS's main difference is how they aggregate load for releasing task. The experiments contemplated the following scenarios to compare the three mechanisms: a) the pool sequencing rule and b) the cell availability.

The pool sequence rules considered are a) the Earliest Release Date (ERD); b) Capacity Slack (CS); and c) Modified Capacity Slack (MODCS) because previous studies of POLCA and COBACABANA used them. For ERD, the job with the earliest release date for that station is processed first. The original POLCA uses the ERD after the MRP authorises each job (SURI, 1998). For this study, the release date was considered as the first station of the EODD rule. The CS uses a slack ratio of the stations considered in the routing of the job. This ratio estimates the station's aggregate load contribution over the difference between the load norm and the current load of the station. The job with the lowest slack ratio is processed first. Previous comparisons of PCSs used this CS rule (FERNANDES et al., 2017a, 2017b; THÜRER et al., 2017a). The MODCS uses two rules for jobs: for urgent jobs (jobs that are late for their operation due date), they are prioritised by the EODD rule; for non-urgent jobs, they are prioritised by the CS rule. The urgent jobs are always prioritised over the non-urgent jobs. This combination was proposed as a useful rule (THÜRER et al., 2015b), and it has been used by other comparisons (CARMO-SILVA et al., 2020; THÜRER; FERNANDES; STEVENSON, 2020).

The first scenario is 100% of the availability of the cell, and the second scenario represents a 90% availability of station four, describing a machine failure. This factor is modelled based on the effect that Qi, Sivakumar and Gershwin (2009) reported. The model represents the time between fails and time to repair with exponential distributions, as Silva et al. (2017) use. The mean time between fails (MTBF) used is 1.000 time-units, and the meantime to repair (MTTR) is two release period (10 time-units). This cell availability attempts to model one of the problems cited by Suri (2018) as solved situations with the POLCA loop. This cell availability tries to represent a temporary bottleneck station in the routing to isolate the effect and not to affect uniformly. The model represented defines Station 4 as the middle stream with upstream and downstream loops available.

The simulation scenarios were collected using 13,000 time-units of run length, using a warm-up period of 3,000 unit-times to reduce the initialisation bias of the model. These simulation parameters are used in other simulation comparisons, and the proposed model was verified to obtain a stable result in the simulation.

The performance measures considered in this simulation are: the Delivery Time Mean (DTM, or the Total Throughput Time)– the mean of the delivery date minus the arrival time for a job; the Shop Floor Throughput Time Mean (SFTT) – the mean of the delivery date minus the release date to the shop; the Percentage Tardy (TDP) – the proportion of jobs are delivered after its due date; the Tardiness Mean (TDM) – the mean of time-delayed of delivery of a job when it is completed after its due date; the Standard Deviation of Lateness (LAD) – the deviation of the delivery time minus the job due date, considering an early delivery (negative values) and tardy delivery (positive values). These measures are wildly used in PCSs designed for the MTO environment (HENDRY; KINGSMAN, 1989), and they are used as a standard comparison for literature (chapter 3).

5.4 RESULTS

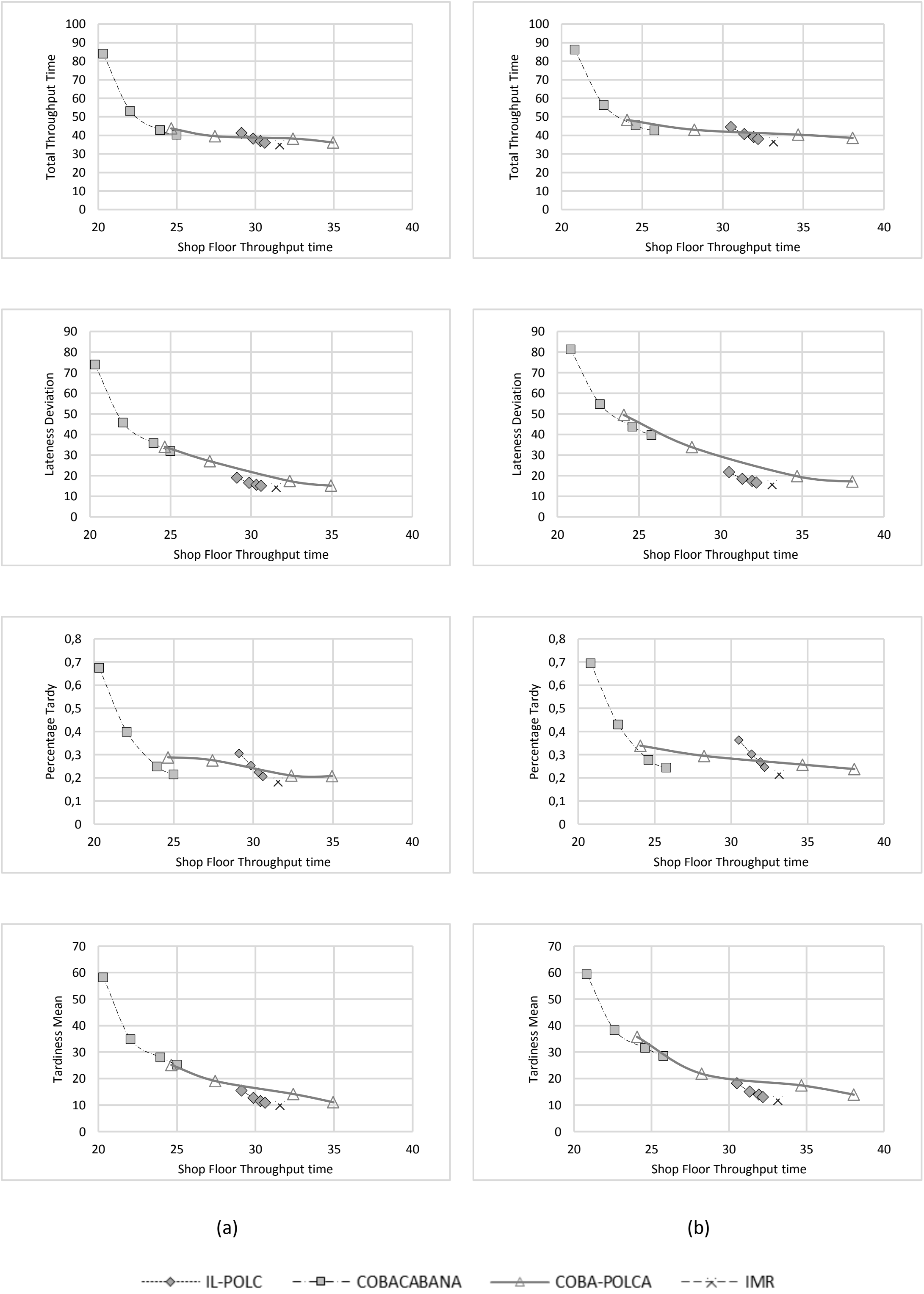
For a better interpretation of results, data collected is exhibited using the performance curves, as suggested by Olhager and Persson (2008). Previous comparisons of the PCSs (CARMO-SILVA et al., 2020; THÜRER; FERNANDES; STEVENSON, 2020) have used these curves. Each point on the curve represents the mean of each performance measure. We used the Analysis of Variance (ANOVA) as a statistical analysis to test the significance of the difference of each plotted point (tables in Appendix D). Results verify that the PCSs studied significantly affect the selected performance measures, as other factors studied. The following subsections present results showing a) the card-based version and b) the load-based version. In each subsection, we show results by the pool sequence rule. On each rule, the effect of cell availability is contrasted for each PCS.

5.4.1 Performance curves for each pool sequence rule, using card-based version

Results are presented in the following order: the ERD rule, the CS rule and the MODCS rule. Figure 5.4 shows results for the ERD rule. As expected, the curves are dislocated to the right, incrementing the measures (5.7%, an overall average). However, the behaviour of cell availability of 100% (Figure 5.4 a) against 90% of station 4 (Figure 5.4 b) is not different for the PCSs. The other PCS outperforms IL-POLC. For Shop Floor Throughput Time (SFTT), COBACABANA presents higher reductions (34.5%) than COBA-POLCA (14.4%), but in the percentage tardy, COBACABANA has higher values (111.4%) than COBA-POLCA (35.2%). As expected, COBACABANA and COBA-POLCA are much better than use the Immediate Release on SFTT (27.7% and 5.4% of reduction, respectively). IL-POLC reduces the SFTT (5.0%) without deteriorating the other performances. The effect on Delivery Time Mean (DTM) is an increment for each PCS, but COBACABANA has a larger increment (57.7%) than COBA-POLCA (13.1%) and IL-POLC (9.3%). One crucial point is for COBACABANA to use the workload norm as 13. For that point, the percentage tardy is the highest-value (270.7% of increment vs the push system), explained that that value of norm is too tight for the studied configuration.

Figure 5.5 shows the results for the CS rule. For that rule, IL-POLC presents a lower reduction of the SFTT (7.6%) compared with COBACABANA (overall mean 29.5%) and COBA-POLCA (14.6% of reduction) but it does not harm the other measures, as in the previous rule. Figure 5.6 shows the results using the MODCS rule. As previous results, IL-POLC has a similar effect of reduction. For MODCS, IL-POLC presents similar results to the tardy jobs as the CS rule because many jobs become urgent jobs, and they are prioritised with the CS rule.

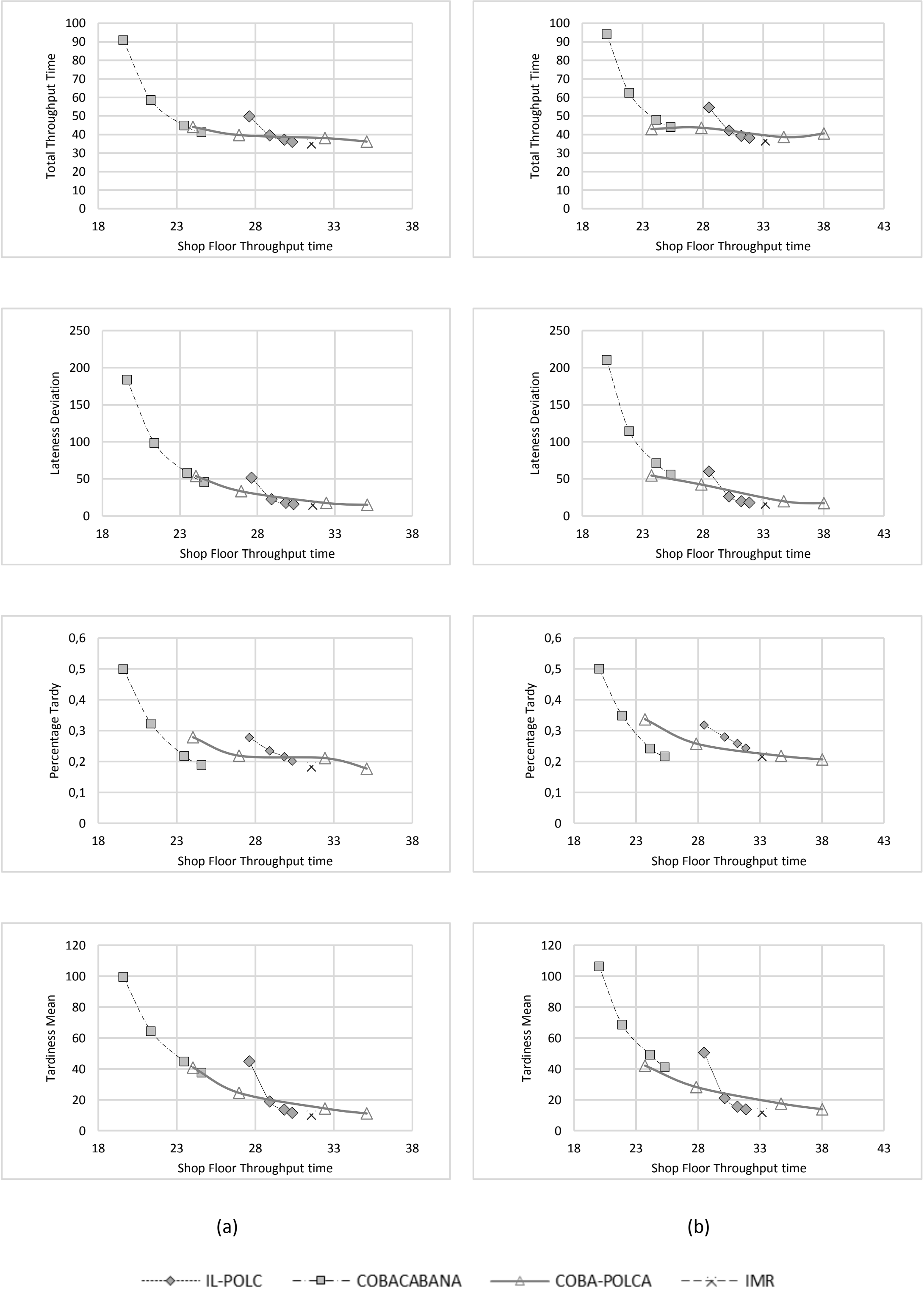
Figure 5.4 – Performance curve using ERD rule comparing cell availability: a) 100%; b) 90% of station 4



(a)

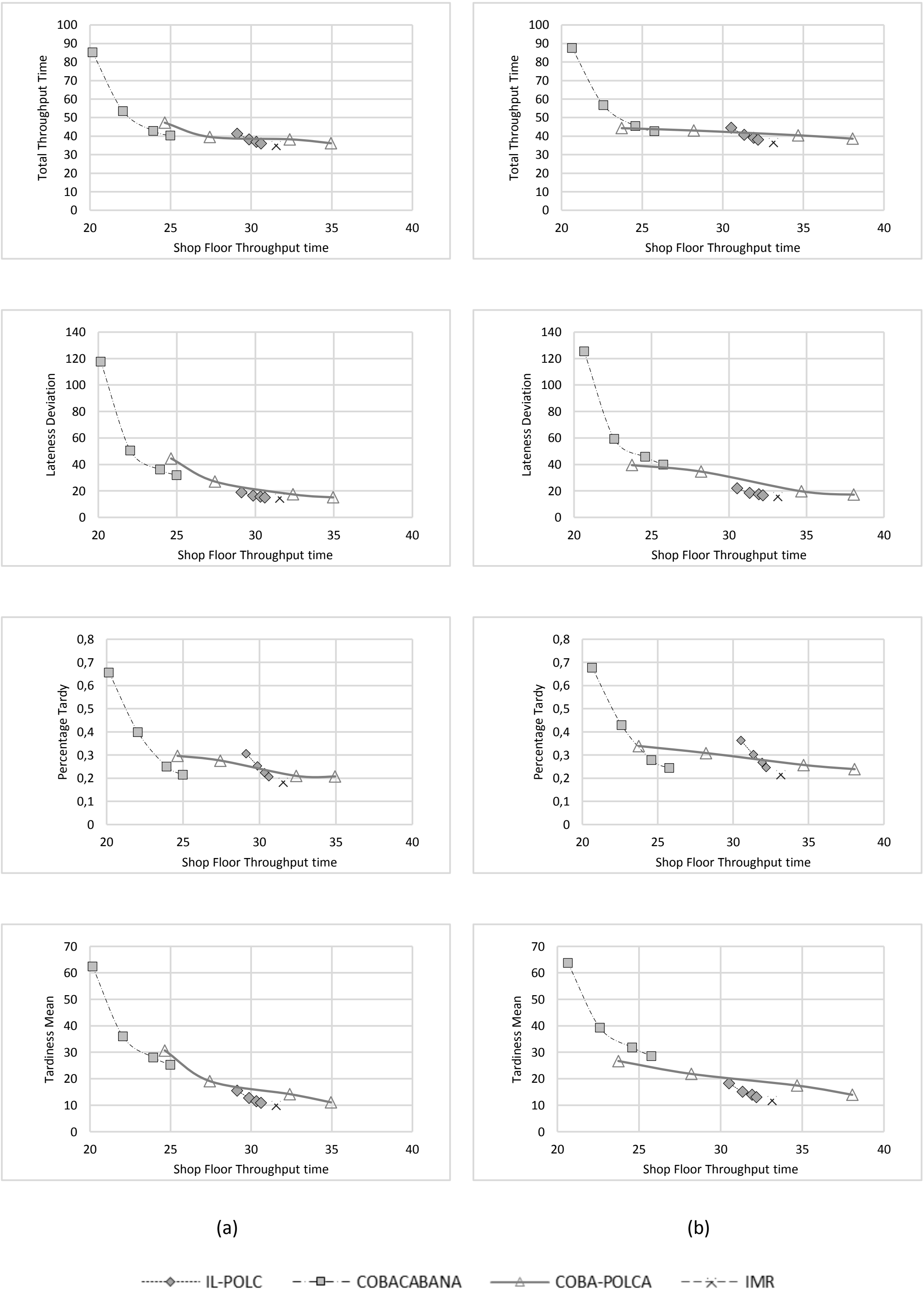
(b)

Figure 5.5 – Performance curve using CS rule comparing cell availability: a) 100%; b) 90% of station 4



(a) (b)

Figure 5.6 – Performance curve using MODCS rule comparing cell availability: a) 100%; b) 90% of station 4

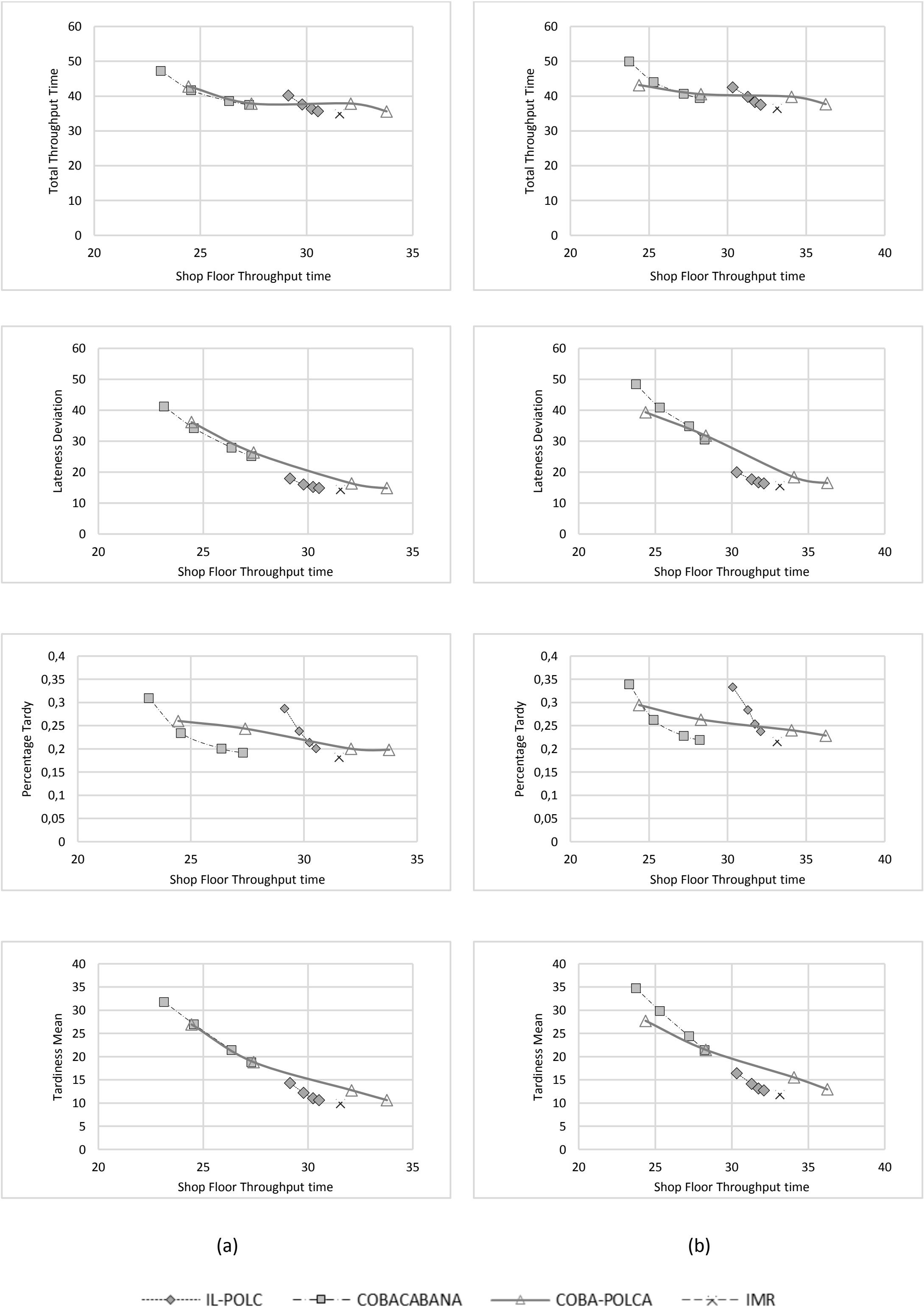


(a) (b)

5.4.2 Performance curves for each pool sequence rule, using load-based version

The following curves show results for the load-based version of the studied PCSs. Figure 5.7 shows using the ERD rule. The difference shown on the systems is similar when the systems are using cards. The cell availability also increases the measures (5.5%, as an overall average). COBACABANA reduce the SFTT (19.7%) more than IL-POLC (5.2%) and COBA-POLCA (6.8%). For the DTM, IL-POLC has a little detriment (7.5%) for reducing the SFTT. COBACABANA also presents the DTM disadvantage (increases 18.3% as average), but the load-based version does not show the same injury levels as the card-based. COBA-POLCA reduces the DTM (17.8% average) when it uses the lower values of limit, but it increases the DTM (4.3%) more than the push system for the larger values. For the other measures, COBACABANA reduces the SFTT more than the other systems, but it shows a detriment on LAD (124.8%) and TDM (150.8%). IL-POLC presents a lower reduction of SFTT than the other systems, but it shows the lower harm of the measures (12% increment for LAD and 22.1% for TDM). COBA-POLCA reduces the DTM for the lower values of limits, but it harms the other measures. Figure 5.8 presents curves using the CS rule, and Figure 5.9 presents using the MODCS rule. The effects presented for the ERD rule are similar for both rules. The behaviour of the curves of MODCS is very similar to the CS more than the ERD rule.

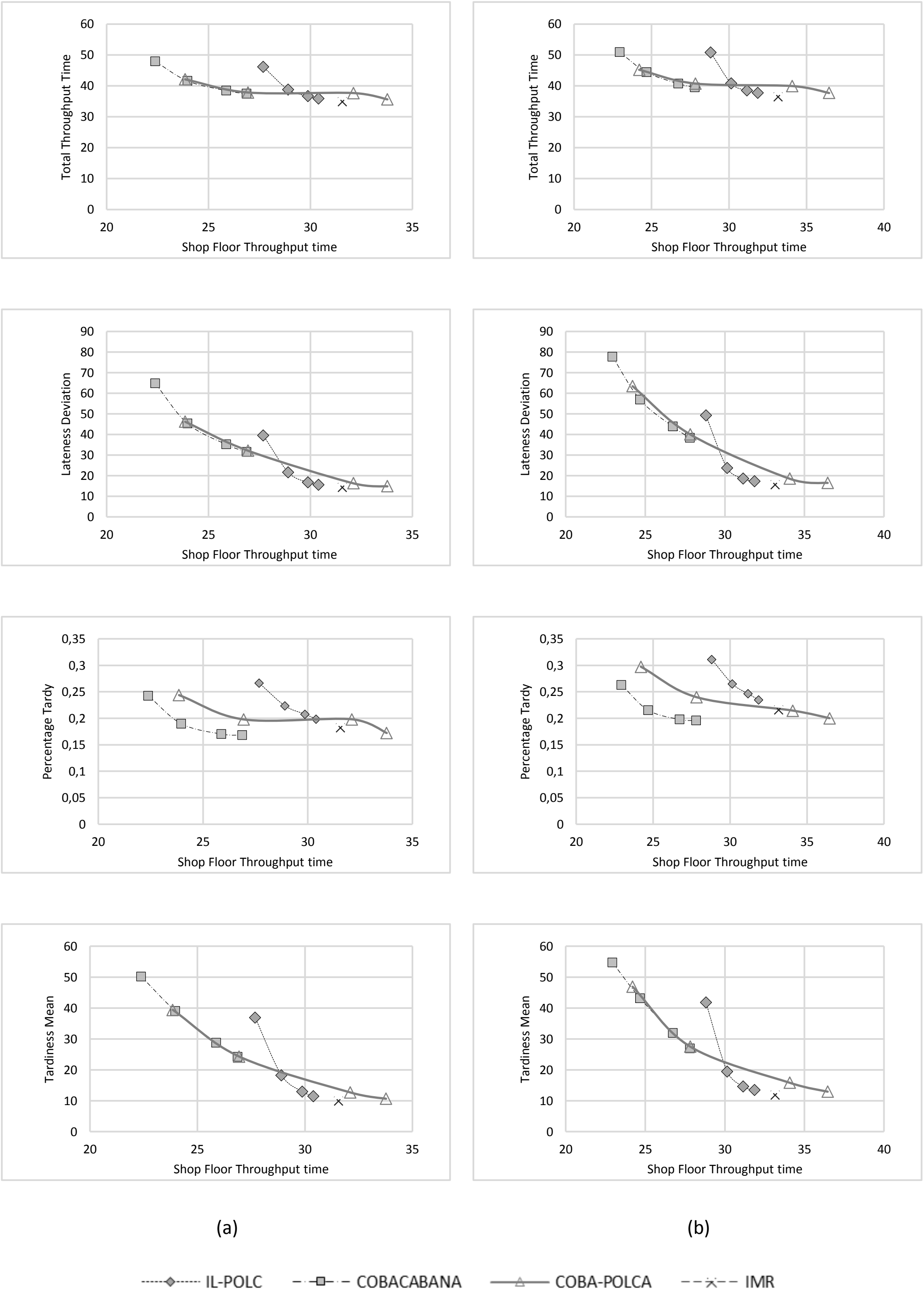
Figure 5.7 – Performance curve of load-based version using ERD rule comparing cell availability: a) 100%; b) 90% of station 4



(a)

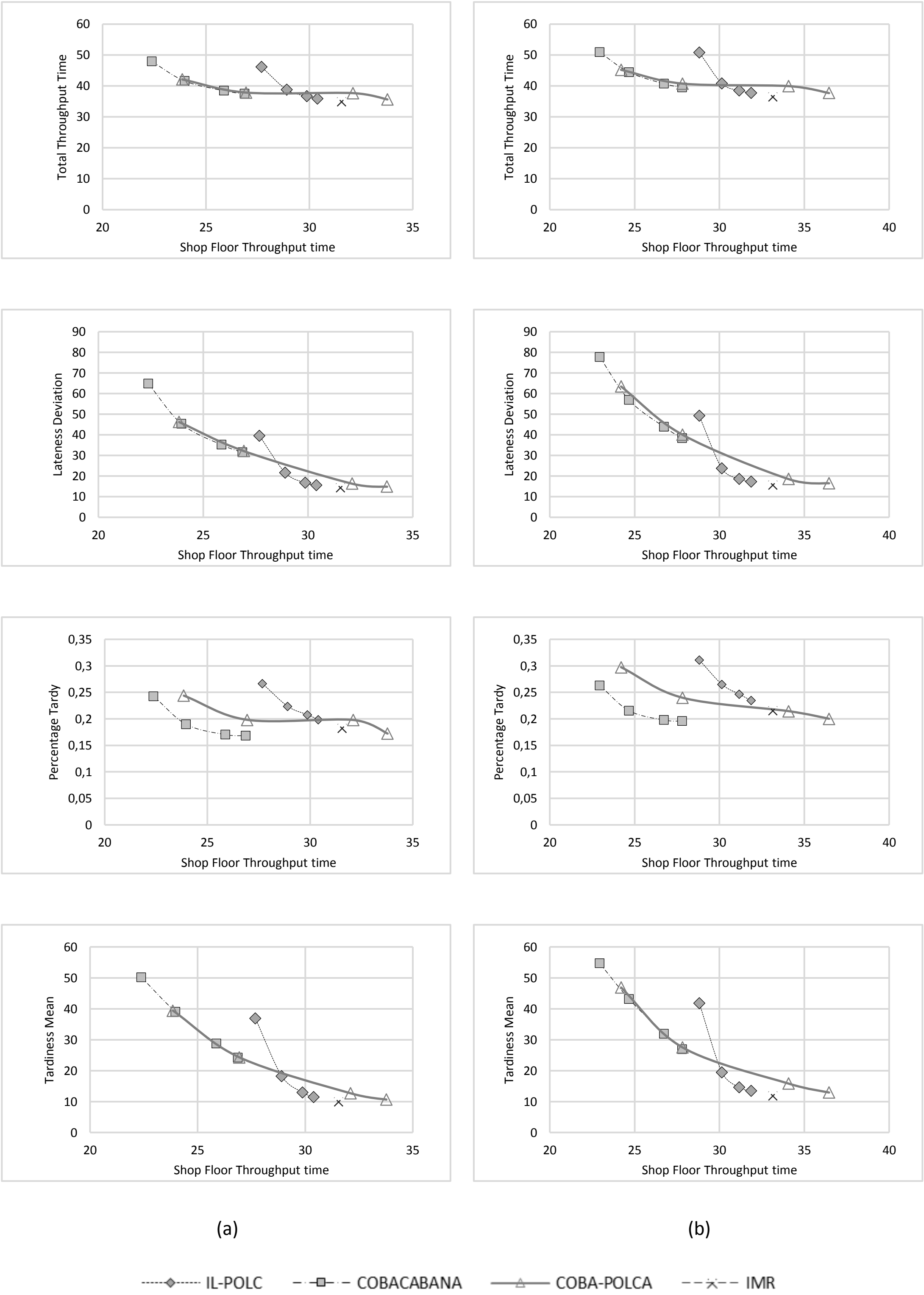
(b)

Figure 5.8 – Performance curve of load-based version using CS rule comparing cell availability: a) 100%; b) 90% of station 4



(a) (b)

Figure 5.9 – Performance curve of load-based version using MODCS rule comparing cell availability: a) 100%; b) 90% of station 4



(a) (b)

5.4.3 Discussion

For the three systems, results present a significant difference in the compared PCSs to control. They present better results than the push system, but they compensate for managing the lead time, as Melnyk and Ragatz (1989) present. IL-POLC presents the lowest detriment of the Delivery Time Mean (DTM) but has a lower reduction of the Shop Floor Throughput Time (SFTT), especially for card-based representation. COBACABANA has the most considerable reduction for SFTT, but it shows the most considerable detriment of the other performance measures. COBA-POLCA depends on its quantum limits and the releasing norms to present the reduction of the SFTT. The results highlight that the centralised release with a paired-cell to control capacity works better than using only a centralised release, as Thürer, Fernandes, and Stevenson (2020) stated. IL-POLC and COBA-POLCA do not present the higher detriment of COBACABANA in lower limits because the capacity control avoids the DTM increase. This effect is less noticeable in the load-based version of the systems, but it still has a difference. The difference between IL-POLC and COBA-POLCA is how they aggregate the indirect load.

For the difference of loop design, results suggest that the single station loop for indirect load for order release reduces the SFTT, but it harms other performance measures. The indirect paired-cell loop (IL-POLC) has the lowest injury. For the load-based comparison, this difference is less sensitive to the single station loop and the loss presented is lower than the card-based representation. Still, the paired-cell loop presents less harm to the DTM than the single station loop because the paired-cell does not limit the station and limits only the specified path on the loop as Suri (1998) designed. The injury observed of the single station loop matches with Ziengs, Riezebos, and Germs (2012) presented that the lowest norm tested values reduce SFTT so much that the pool delay time increases and reduces the balance of those measures. This effect explains the disadvantage of COBACABANA and COBA-POLCA in respect of IL-POLC when the card-based is used. This effect also matches with Carmo Silva et al. (2020) for the centralised release of POLCA. The difference observed between IL-POLC and the centralised release is how much the quantum limit (direct and indirect load) defines the performance of the PCSs. As a summarized comparison, COBACABANA may reduce the SFTT, but it damages the other measures. COBA-POLCA has some reduction of the SFTT, but it depends on the quantum limit. IL-POLC has a conservative strategy that does not reduce the SFTT as well as the other two PCSs, but it does not deteriorate the other measures. Depending on the mannter that the managers prioritize each measure, then they can select the PCS for their shop.

For the card representation used in the model, results suggest that IL-POLC present a similar behaviour comparing card-based (using only 3 cards maximum per job) and load-based version (aggregate precisely the amount of a load of each job) experimented, as Suri (2018) expected. COBA-POLCA does not present a practical difference. On that point, COBACABANA is more sensitive to the quantum representation than the other two systems. Because each card represents the same amount of job load in COBACABANA, as Braglia, Marrazzini, and Padellini (2020) implemented, these results differ from what Thürer, Land, and Stevenson (2014) stated for the card representation. Those authors used some corrections in how each card represent a job load, depending on the maximum number of cards per jobs. Results highlight the difficulty of using COBACABANA with this proportional representation of quantum because it may harm how that system reacts to the shop. As expected for that system, the load-based version did not show those detriments levels in the card-based version. This result requires that COBACABANA and COBA-POLCA need a correction for using cards to aggregate the indirect load for releasing task. IL-POLC can use cards to aggregate the indirect load for this factor, using the same amount of job load without presenting detriments. This point is easier to implement because each card represents the same job load, and releasing can use cards also.

These results suggest that the compared PCSs do not present a practical difference for the scenarios representing machine breakdowns. The control flow and the release mechanism's dominance are similar when the cell availability is 100% and 90% for the tested scenarios. As Suri (1998, 2018) and Prakash and Feng (2011) appointed, cell availability interferes with the shop performance. These results show similar curves for both cell availability levels because it is very tiny for creating other performance curve behaviour. This result suggests other researches for defining how this factor can influence performance. Table 5.5 summarizes results for each factor considered in this experiment.

Table 5.5 -Summary of the comparison of the three PCSs

Parameter	**Studied levels**		
PCS studied	**IL-POLC**	**COBACABANA**	**COBA-POLCA**
Quantum card	Similar behaviour for both representations	Card-based affect strongly for each norm compared to load-based	Similar behaviour for both representations (different values)
Number of cards allowed for loop (and workload norm)	It does not reduce SFTT It does not harm other performances measures	Card-based: reduce the SFTT but harms other performance measures Load based: reduce SFTT without a high detriment of others	Low values of norm reduce the SFTT, but high values detriment SFTT
Highlighted suggestion	Use for proportional card representation when the detriment has a strong effect	Do not use for proportional representation of quantum	Use for higher SFTT reductions and detriment effect is low (valuable set for norms)
Pool sequence rule	Each rule has different behaviour but similar conclusions for PCSs		
Cell availability	Significant effect but not a practical difference for PCSs (uniform effect)		

5.5 CONCLUSIONS

The use of control in the order release and capacity is an effective way to increase performance. The design of the job release mechanism and the capacity control define its relevance to improving performance in that context. For design alternatives, two card-based PCSs highlighted by their design: POLCA, which uses a paired-cell to control capacity, and COBACABANA that uses centralised loops to each station to release jobs to the shop. Since the paired-cells have been used to solve downstream breaks in capacity control, this research proposes a new design that uses paired-cells to job release and capacity control, named IL-POLC. As reported by literature, combining those two previous systems, COBA-POLCA, also reports outperformance than original systems. This research compares the IL-POLC with COBACABANA and COBA-POLCA to know if the paired-cell loop to release job is effective. Tested by a simulation experiment, results suggest that IL-POLC is less sensitive to the quantum limit (and workload norm) than the other two systems, primarily when cards represent the same amount of job load. IL-POLC presents this effect because both limits only affect the specific path between the two cells.

The IL-POLC paired-cell for job release does not substantially reduce the Shop Floor Throughput Time, although it does not deteriorate the other performance measures for tight norms. The single station loop outperforms the paired-cell loop for order release for high values of workload norm. This advantage of the single station loop is very noticeable if the shop uses the system's load-version for many norm levels. The cell availability tested does not create a practical difference in the behaviour of the PCSs compared. The three tested sequence rules do not present a different behaviour for the PCSs, and even the rules present different performance.

5.5.1 Practical implications

Results suggest that POLCA practitioners should use the paired-cell loop to release jobs (IL-POLC) when they have difficulty estimating workload norms, monitoring functions, quick feedback in that decision, and using a card-based version of the system. The IL-POLC can use a simple rule of three for quantum representation for that card-based version, simplifying the aggregated load's estimation across stations that the single station loops need. These conditions are very important for practitioners in small shops with low technology investments. For the IL-POLC, practitioners can add another set of POLC cards to release jobs to the shop, based on the original POLCA cards. One piece of advice is that the paired-cell loops to aggregate the indirect load will not improve performance, but it will not present a high injury. As every PCS, IL-POLC can present detriment if the norm is excessively tightened.

If the practitioners have more resources to estimate each station's load aggregation, results suggest using the centralised loop that aggregates a single cell load (or station) to release jobs based on indirect load conditions. Using that condition and reducing the SFTT is more relevant than the other measures effect. This experiment suggests COBA-POLCA as the PCS for that environment. COBACABANA is suggested for a large reduction of SFTT when the other measures are not so relevant. For that single station loops, practitioners must be cautious in choosing a norm. A very high value of the norm to release jobs could detriment the performance, instead of improving it. In addition, if POLCA is used for capacity control, jobs could delay more than expected of a push system. Since the estimation of the centralised loop limits may not be the same as that of the POLCA loops, practitioners should review if those limits worsen performance. Those limits are critical because they change how the system takes advantage of performance. For the initial parametrisation for the implementation, the quantum limit must begin with a loose value and reduce that limit until it harms the shop measures at an undesired level.

5.5.2 Limitations and Future research

This answer for the paired-cell loop effect is limited because this comparison only uses one setting for the routing characteristics, as a balanced shop with similar processing times for cells. Future research may consider a real data scenario to test if the paired-cells loops to order release are useful as the capacity control's paired-cells loops. Another limitation is how the model represents the routing variation and other resource allocations. The model assumes dedicated resources for the stations, and it does not consider other interchangeable resources, as material and labour, as an example. As Suri (1998) claims, paired-cell loops help identify when this interchangeability of activities needs to occur, and future research can use other models to test how the paired cells help on that interchangeability of activities. For estimating limits, future research can be developed to calculate the centralised limits when the capacity control of POLCA is used. This estimation is an essential issue because those levels can worse performance. Other research could also explore how to use the combined system of COBA-POLCA to discover the practical implications of both cards systems working.

6 CONCLUSIONS

This research aims to find Production Control Systems (PCSs) that can substitute Paired-cell Overlapping Loops of Cards with Authorization (POLCA) or complement it. For the PCSs, this study considers the Make-to-order (MTO) environment and the General Flow Shop (GFS) as the typical description flow for that environment. This thesis uses a systematic literature review to identify possible PCSs substitutes. For the POLCA complement, this research proposes an order release system to control the indirect load by paired-cell loops (named IL-POLC). Two experiments by simulation test this proposition with the original POLCA and Release-and-Flow POLCA to understand how the IL-POLC works and compare with the possible substitutes for the original POLCA: COBACABANA and COBA-POLCA.

According to the reviewed literature, the most suitable PCSs are the COBACABANA, the card-based system to implement the WLC concept, and the COBA-POLCA, the original POLCA using the COBACABANA as the order release mechanism. Other PCSs can handle different situations according to shop conditions. The literature review findings for this study suggest choosing the PCSs matching the loop's design and the routing variation possible. For this choice, the pool sequence rule's interaction with the dispatching rule affects more the performance than the loop design. Furthermore, this interaction affects how accurately the cards represent the workload of the jobs in the system. Results suggest using the load-based version (exact addition) for the processing time high variation and card-based (or order-based) for low variation of the processing time.

Based on the literature review results, this thesis proposed the Indirect Load condition to release a job in a shop to the POLCA system, named IL-POLC. This proposition substitutes the authorisation lists based on the release date that the original POLCA uses for the order release task. IL-POLC uses the same concept of the Paired-cell overlapping loops of cards (POLC element) for the control capacity task. IL-POLC suggests using dispatching rules for the sequencing task because those rules are more accessible to implement and understand than a complex algorithm that can schedule jobs. Some factors are necessary to implement the IL-POLC, as the period release length, the maximum number of cards representing a quantum limit, the indirect load norm for the order release mechanism, and the accounting method for the indirect load.

This thesis uses two experiments to test the proposed IL-POLC. The first experiment compares the IL-POLC with the original POLCA and RF-POLCA to test the difference in the order release mechanism proposed against the known mechanisms. The second experiment tested the proposed IL-POLC with two suitable PCSs useful in the MTO environment: COBACABANA and COBA-POLCA. For both experiments, the performance measures represent how the MTO environment defines a shop performance. In each experiment, this research uses discrete-event simulation to estimate the performances measures used in the MTO (Delivery Time Mean, Shop Floor Throughput Time, Lateness Deviation, Percentage of tardy jobs, and Tardiness mean) using the main events of the MTO flow (order acceptance, order release, and delivery). Thus, the model represents a typical small shop with high variability of job routings and processing times as a common scenario in the studied environment. In addition, the model uses data from a shop used in literature to compare results with other PCSs' comparison.

For the first experiment, to test the proposed indirect load POLC (IL-POLC), this complement has better results than the original POLCA and RF-POLCA. The order release using the indirect load condition can substitute the original mechanism of POLCA, using authorisations lists based on release date) in the MTO conditions. The indirect load norm limits the number of jobs released to the shop and avoids an overloaded shop station. The level of the indirect load norm affects significantly how this IL-POLC improves the system. Findings suggest using a high norm first and then reduce until the norm harms the shop floor throughput time. Based on the performance curves, results propose to use the classical load aggregation approach instead of the corrected approach. For this system, results also suggest that a fixed number of cards maximum per loop can represent the quantum of a job (the workload of a job without deteriorating the estimation. For the maximum number of cards per job, results indicate that using three cards is a good representation; but the quantum limit and indirect load norm need an adjustment if that number changes. This result allows using a card system without needing a sophisticated system to estimate the indirect load, as in important results for the small shops. Findings also recommend using shorter release period lengths to improve the performance, using a periodic release. For the dispatching rules, results suggest using EODD than FIFO because FIFO presents a high detriment of performance. EODD can balance job release without having some blockings on indirect load or card availability.

The second experiment compares the proposed IL-POLC with COBACABANA and COBA-POLCA to test the paired-cell loop's effectiveness to aggregate the indirect load in the order release task. Conforming to simulations results, IL-POLC depends less on the workload norm to present detriment in the performance than the other systems when the shop uses the card-based version, and it uses a proportional rule to represent the quantum card. IL-POLCA does not reduce the Shop Floor Throughput Time as the other two PCSs, but it does not damage the other performance measures. The quantum limit and the indirect load norm do not affect how the PCS improves the shop performance for this system. For the two systems with a single station loop, when they use the proportional rule to represent a quantum card, the workload norms are critical in avoiding the detriment caused by tightened limits. COBACABANA shows a high sensitivity by the norm value, at the point that a tight value harms all performances measure if it uses a tightened value. For COBA-POLCA, the capacity control task of the POLCA cards smoothes that detriment. Thus, COBA-POLCA presents a balanced performance, reducing the Shop Floor Throughput Time without deteriorating the other performances. From these experiment's results, COBA-POLCA is a good option for shop performance, but this PCS requires two different system cards, with a different meaning: one card system for the order release and another card system for the control capacity task. IL - POLC can use the exact meaning of the cards to both tasks, even using two card systems, simplifying the PCS. The breakdown machine, modelled as cell availability, does not differ in practical implications for the PCSs' behaviour in one station.

This doctoral research's original contribution is how POLCA can adapt elements from Workload Control's theory to improve its performance in the MTO environment. The original proposal for this research is the paired-cell loop in the indirect load for the order release mechanism of POLCA, called IL-POLC. This paired-cell loop to account for the indirect load is more effective than using only the capacity control of the original POLCA. Moreover, compared with COBACABANA and COBA-POLCA, IL-POLC is better in a conservative position to avoid detriments by tightened norms, using a unique and straightforward norm for the loops. Still, it is not as effective as using the centralised loops by each station for the order release proposed in COBACABANA and COBA-POLCA for the load-based version.

This research has limited results in the literature review and the proposed experiments. For the reviewed literature, simulation models do not represent some external factors as labours, materials and assemblies. These factors also play an essential role in PCSs implementation. In the experiments, the model used do not perform those factors either. That misrepresentation on the simulation model limits how the IL-POLC performs some advantages reported of employments of the original POLCA. For example, the paired-cell loop also allows a reallocation of resources between cells, and the simulation model does not characterise that element. Nevertheless, implementations of POLCA report some benefits for that function.

Another limitation is how IL-POLC use the 2nd cell approach on the indirect load representation. The original approach will also require other estimation of load norms.

For the experiments, the data collection uses steady-state simulation for the performance measures used. As the cell availability (for machine breaks), material delays, labour absence, due date changes, some external factors are not easily visible on that kind of simulations. They need other simulation analysis techniques to evaluate their effects. These factors need additional measures to assess how they affect the PCSs because their applications face those problems.

There are some gaps still uncovered. From alternatives to POLCA, other PCSs have not been compared to the studied PCSs that can substitute POLCA in specific environments. POLCA needs further research on how those external factors (cell availability, material delays, labour absence, due date changes) affect the practical solution from other PCSs in empirical applications. These factors also have a gap in how important they are for managers to choose a PCS. As this research concludes using centralised loops for accounting for the jobs' load, future research is needed to implement that centralised release in a real application. Those researches will also help design how other technologies, such as information systems, need to react to production control.

REFERENCES

AGLAN, C.; DURMUSOGLU, M. B. Lot-splitting approach of a hybrid manufacturing system under CONWIP production control: A mathematical model. **International Journal of Production Research**, v. 53, n. 5, p. 1561–1583, 2015.

AJORLOU, S.; SHAMS, I. Artificial bee colony algorithm for CONWIP production control system in a multi-product multi-machine manufacturing environment. **Journal of Intelligent Manufacturing**, v. 24, n. 6, p. 1145–1156, 2013.

AL-TAHAT, M. D.; RAWABDEH, I. A. Stochastic analysis and design of CONWIP controlled production systems. **Journal of Manufacturing Technology Management**, v. 19, n. 2, p. 253–273, 2008.

AMARO, G.; HENDRY, L.; KINGSMAN, B. Competitive advantage, customisation and a new taxonomy for non make-to-stock companies. **International Journal of Operations and Production Management**, v. 19, n. 4, p. 349–371, 1999.

AZIZ, M. H.; BOHEZ, E. L. J.; PISUCHPEN, R.; PARNICHKUN, M. Petri Net model of repetitive push manufacturing with Polca to minimise value-added WIP. **International Journal of Production Research**, v. 51, n. 15, p. 4464–4483, 2013.

BAGNI, G.; GODINHO FILHO, M.; THÜRER, M.; STEVENSON, M. Systematic review and discussion of production control systems that emerged between 1999 and 2018. **Production Planning and Control**, v. 0, n. 0, p. 1–15, 2020.

BANKS, J.; CARSON, J. S.; NELSON, B. L.; NICOL, D. M. **Discrete-Event System Simulation**. 4th. ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2005.

BARROS, C.; SILVA, C.; MARTINS, S.; DIAS, L.; PEREIR, G.; FERNANDES, N. O.; CARMO-SILVA, S. Are card-based systems effective for make-to-order production? **Romanian Review Precision Mechanics, Optics and Mechatronics**, n. 49, p. 5–9, 2016.

BAYSAN, S.; KABADURMUS, O.; DURMUSOGLU, M. B. Economic Analysis of POLCA System Design via Simulation: A Multi-Cell Manufacturing Case Study. **37th International Conference on Computers and Industrial Engineering, October**, p. 20–23, 2007.

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

BELISÁRIO, L. S.; PIERREVAL, H. Using genetic programming and simulation to learn how to dynamically adapt the number of cards in reactive pull systems. **Expert Systems with Applications**, v. 42, n. 6, p. 3129–3141, 2015.

BERENDS, P.; ROMME, G. Simulation as a research tool in management studies. **European Management Journal**, v. 17, n. 6, p. 576–583, dic. 1999.

BERGAMASCHI, D.; CIGOLINI, R.; PERONA, M.; PORTIOLI, A. Order review and release strategies in a job shop environment: A review and a classification. **International Journal of**

Production Research, v. 35, n. 2, p. 399–420, 1997.

BERTOLINI, M.; BRAGLIA, M.; ROMAGNOLI, G.; ZAMMORI, F. Extending value stream mapping: the synchro-MRP case. **International Journal of Production Research**, v. 51, n. 18, p. 5499–5519, 2013.

BERTOLINI, M.; BRAGLIA, M.; FROSOLINI, M.; MARRAZZINI, L. Work In Next Queue CONWIP. **Computers and Industrial Engineering**, v. 143, n. March, p. 106437, 2020.

BERTOLINI, M.; ROMAGNOLI, G.; ZAMMORI, F. Simulation of two hybrid production planning and control systems: A comparative analysis. **Proceedings of 2015 International Conference on Industrial Engineering and Systems Management, IEEE IESM 2015**. 2015

BERTOLINI, M.; ROMAGNOLI, G.; ZAMMORI, F. 2MTO, a new mapping tool to achieve lean benefits in high-variety low-volume job shops. **Production Planning and Control**, v. 28, n. 5, p. 444–458, 2017.

BETTERTON, C. E.; COX, J. F. Espoused drum-buffer-rope flow control in serial lines: A comparative study of simulation models. **International Journal of Production Economics**, v. 117, n. 1, p. 66–79, 2009.

BHARDWAJ, A.; GUPTA, A.; KANDA, A. Drum-buffer-rope: The technique to plan and control the production using theory of constraints. **World Academy of Science, Engineering and Technology**, v. 70, p. 103–106, 2010.

BOKHORST, J. A. C.; SLOMP, J. Lean Production Control at a High-Variety, Low-Volume Parts Manufacturer. **Interfaces**, v. 40, n. 4, p. 303–312, ago. 2010.

BONG, C. S.; CHONG, K. E.; HOW, W. C. Job shop material control based on the principles of quick response manufacturing. **Journal of Advanced Manufacturing Technology**, v. 12, n. 1 Special Issue 2, p. 15–30, 2018.

BRAGLIA, M.; FROSOLINI, M.; GABBRIELLI, R.; ZAMMORI, F. CONWIP card setting in a flow-shop system with a batch production machine. **International Journal of Industrial Engineering Computations**, v. 2, n. 1, p. 1–18, 2011.

BRAGLIA, M.; CASTELLANO, D.; FROSOLINI, M. Optimization of POLCA-controlled production systems with a simulation-driven genetic algorithm. **The International Journal of Advanced Manufacturing Technology**, v. 70, n. 1–4, p. 385–395, 2014.

BRAGLIA, M.; CASTELLANO, D.; FROSOLINI, M. A study on the importance of selection rules within unbalanced MTO POLCA-controlled production systems. **International Journal of Industrial and Systems Engineering**, v. 20, n. 4, p. 457–468, 2015.

BRAGLIA, M.; MARRAZZINI, L.; PADELLINI, L. COBACANA: a real industrial application in a job shop system. **Production Planning & Control**, p. 1–17, 2020.

BRIEN, J. F. O.; JAFARI, F.; WEN, J. T. An Effective Method to Reduce Inventory in Job Shops. **IEEE Transactions on Robotics and Automation**, v. 22, n. 1, p. 160–167, 2006.

BURBIDGE, J. L. Production control: a universal conceptual framework. **Production**

Planning & Control, v. 1, n. 1, p. 3–16, 1990.

CARMO-SILVA, S.; FERNANDES, N.; THÜRER, M.; FERREIRA, L. P. Extending the POLCA production control system with centralized job release. **Production**, v. 30, 2020.

CARMO-SILVA, S.; FERNANDES, N. O. Bottleneck-oriented order release: An assessment by simulation. En: ROCHA, Á. et al. (Eds.). . **Recent Advances in Information Systems and Technologies**. Lisboa: Springer International Publishing, 2017. v. 2p. 406–415.

CARMO-SILVA, S.; FERNANDES, N. O. Generic POLCA: An Assessment of the Pool Sequencing Decision for Job Release. En: ROCHA, Á. et al. (Eds.). . **Trends and Advances in Information Systems and Technologies. WorldCIST'18 2018. Advances in Intelligent Systems and Computing**. Naples: Springer International Publishing, 2018. v. 745p. 697–705.

CHAKRAVORTY, S. S. An evaluation of the DBR control mechanism in a job shop environment. **Omega**, v. 29, n. 4, p. 335–342, 2001.

CHAKRAVORTY, S. S.; ATWATER, J. B. The impact of free goods on the performance of drum-buffer-rope scheduling systems. **International Journal of Production Economics**, v. 95, n. 3, p. 347–357, 2005.

CHANG, T. M.; YIH, Y. Generic kanban systems for dynamic environments. **International Journal of Production Research**, v. 32, n. 4, p. 889–902, 1994.

CHINET, F. S.; GODINHO FILHO, M. Sistema POLCA: revisão, classificação e análise da literatura. **Gestão & Produção**, v. 21, n. 3, p. 532–542, 2014.

CHONG, K. E.; HOW, W. C.; BONG, C. S. Paired-cell overlapping loops of cards with authorization simulation in job shop environment. **International Journal of Mechanical and Mechatronics Engineering**, v. 15, n. 3, p. 68–73, 2015.

CIGOLINI, R.; PORTIOLI-STAUDACHER, A. An experimental investigation on workload limiting methods within ORR policies in a job shop environment. **Production Planning & Control**, v. 13, n. 7, p. 602–613, 2002.

COLEMAN, D. E.; MONTGOMERY, D. C. A systematic approach to planning for a designed industrial experiment. **Technometrics**, v. 35, n. 1, p. 1–12, 1993.

DALALAH, D.; AL-ARAIDAH, O. Dynamic decentralised balancing of CONWIP production systems. **International Journal of Production Research**, v. 48, n. 13, p. 3925–3941, 2010.

DARLINGTON, J.; FRANCIS, M.; FOUND, P.; THOMAS, A. Design and implementation of a Drum-Buffer-Rope pull-system. **Production Planning & Control**, v. 26, n. 6, p. 489–504, 2015.

DENYER, D.; TRANFIELD, D. Producing a Systematic Review. En: BUCHANA, D.; BRYMAN, A. (Eds.). . **The SAGE Handbook of Organizational Research Methods**. London: SAGE Publications Ltd, 2009. p. 671–689.

ENNS, S. T.; COSTA, M. P. The effectiveness of input control based on aggregate versus bottleneck work loads. **Production Planning & Control**, v. 13, n. March 2015, p. 614–624, 2002.

FERNANDES, F. C. F.; GODINHO FILHO, M. Production control systems: Literature review, classification, and insights regarding practical application. **African Journal of Business Management**, v. 5, n. 14, p. 5573–5582, 2011.

FERNANDES, N. O.; THÜRER, M.; SILVA, C.; CARMO-SILVA, S. Improving workload control order release: Incorporating a starvation avoidance trigger into continuous release. **International Journal of Production Economics**, v. 194, p. 181–189, 2017a.

FERNANDES, N. O.; THÜRER, M.; STEVENSON, M.; SILVA, S. C. Load-Based POLCA: An Assessment of the Load Accounting Approach. En: ROCHA Á., CORREIA A., ADELI H., REIS L., C. S. (Ed.). . **Recent Advances in Information Systems and Technologies**. London: Springer, 2017b. v. 571p. 397–405.

FERNANDES, N. O.; THÜRER, M.; FERREIRA, L. P.; CARMO-SILVA, S. POLCA: Centralised vs. Decentralised Job. **IFAC PapersOnLine**, v. 52, n. 13, p. 1427–1431, 2019.

FERNANDES, N. O.; THÜRER, M.; MIRZAEI, N.; PINTO, L.; FRANCISCO, J. G.; CARMO-SILVA, S. POLCA Control in Two-Stage Production Systems. **Procedia Manufacturing**, v. 51, n. 2019, p. 1491–1496, 2021.

FERNANDES, N. O.; CARMO-SILVA, S. Generic POLCA—A production and materials flow control mechanism for quick response manufacturing. **International Journal of Production Economics**, v. 104, n. 1, p. 74–84, nov. 2006.

FERNANDES, N. O.; CARMO-SILVA, S. Workload control under continuous order release. **International Journal of Production Economics**, v. 131, n. 1, p. 257–262, 2011.

FERNANDES, N. O.; LAND, M. J.; CARMO-SILVA, S. Workload control in unbalanced job shops. **International Journal of Production Research**, v. 52, n. 3, p. 679–690, 2014.

FRAMINAN, J. M.; GONZÁLEZ, P. L.; RUIZ-USANO, R. The CONWIP production control system: Review and research issues. **Production Planning & Control**, v. 14, n. 3, p. 255–265, 2003.

FRAMINAN, J. M.; RUIZ-USANO, R.; LEISTEN, R. Input control and dispatching rules in a dynamic CONWIP flow-shop. **International Journal of Production Research**, v. 38, n. 18, p. 4589–4598, 2000.

FRAZEE, T.; STANDRIDGE, C. Conwip versus POLCA: A comparative analysis in a high-mix, low-volume (HMLV) manufacturing environment with batch processing. **Journal of Industrial Engineering and Management**, v. 9, n. 2, p. 432–449, 2016.

FREDENDALL, L. D.; LEA, B. R. Improving the product mix heuristic in the theory of constraints. **International Journal of Production Research**, v. 35, n. 6, p. 1535–1544, 1997.

FREDENDALL, L. D.; OJHA, D.; WAYNE PATTERSON, J. Concerning the theory of workload control. **European Journal of Operational Research**, v. 201, n. 1, p. 99–111, 2010.

GERAGHTY, J.; HEAVEY, C. A review and comparison of hybrid and pull-type production control strategies. **OR Spectrum**, v. 27, n. 2–3, p. 435–457, 2005.

GERMS, R.; RIEZEBOS, J. Workload balancing capability of pull systems in MTO production. **International Journal of Production Research**, v. 48, n. 8, p. 2345–2360, 2010.

GILLAND, W. G. A simulation study comparing performance of CONWIP and bottleneck-based release rules. **Production Planning and Control**, v. 13, n. 2, p. 211–219, 2002.

GOLANY, B.; DAR-EL, E. M.; ZEEV, N. Controlling shop floor operations in a multi-family, multi-cell manufacturing environment through constant work-in-process. **IIE Transactions (Institute of Industrial Engineers)**, v. 31, n. 8, p. 771–781, 1999.

GOLDRATT, E. M.; COX, J. **The Goal**. Great Barrington, MA: North River Press, 1984.

GOLDRATT, E. M.; FOX, R. E.; GRASMAN, G. **The Race**. New York, NY: North River Press, 1986.

GOLMOHAMMADI, D. A study of scheduling under the theory of constraints. **International Journal of Production Economics**, v. 165, p. 38–50, 2015.

GOMES, C.; RIBEIRO, A.; FREITAS, J.; DIAS, L.; PEREIRA, G.; VIEIRA, A.; FERNANDES, N. O.; CARMO-SILVA, S. Improving Production Logistics Through Materials Flow Control and Lot Splitting. (A. Paias, M. Ruthmair, S. Voß, Eds.)**Computational Logistics. ICCL 2016. Lecture Notes in Computer Science**. Lisboa: Springer, Cham, 2016Disponível em: <http://link.springer.com/10.1007/978-3-319-44896-1_29>. Acesso em: 1 feb. 2018

GÓMEZ PAREDES, F. J.; GODINHO FILHO, M.; THÜRER, M.; FERNANDES, N. O.; JABBOUR, C. J. C. Factors for choosing production control systems in make-to-order shops: a systematic literature review. **Journal of Intelligent Manufacturing**, 2020.

GONZÁLEZ-R, P. L.; CALLE, M.; ANDRADE-PINEDA, J. L. Job shop management of products under internal lifespan and external due date. **International Journal of Production Research**, v. 56, n. 16, p. 5457–5474, 2018.

GONZÁLEZ-R, P. L.; FRAMINAN, J. M.; PIERREVAL, H. Token-based pull production control systems: An introductory overview. **Journal of Intelligent Manufacturing**, v. 23, n. 1, p. 5–22, 2012.

GONZALEZ-R, P. L.; FRAMINAN, J. M.; RUIZ-USANO, R. A multi-objective comparison of dispatching rules in a drum–buffer–rope production control system. **International Journal of Computer Integrated Manufacturing**, v. 23, n. 2, p. 155–167, 2010.

GONZÁLEZ-R, P. L.; FRAMINAN, J. M.; RUIZ-USANO, R. A response surface methodology for parameter setting in a dynamic Conwip production control system. **International Journal of Manufacturing Technology and Management**, v. 23, n. 1–2, p. 16–33, 2011.

GRAVES, R. J.; KONOPKA, J. M.; JOHN MILNE, R. Literature review of material flow control mechanisms. **Production Planning and Control**, v. 6, n. 5, p. 395–403, 1995.

GUO, R. S.; CHIANG, M. H.; LIN, H. W.; CHEN, J. Y. Forward echelon-based inventory monitoring in a semiconductor supply Chain. **IEEE Transactions on Semiconductor Manufacturing**. 2010

HARROD, S.; KANET, J. J. Applying work flow control in make-to-order job shops. **International Journal of Production Economics**, v. 143, n. 2, p. 620–626, jun. 2013.

HAYES, R.; PISANO, G.; UPTON, D.; WHEELWRIGHT, S. Operations Strategy-origins and new directions. En: **Operations, Strategy and Technology - Pursuing the competitive edge**. New Jersey: Wiley, 2005. p. 33–74.

HELBER, S.; SCHIMMELPFENG, K.; STOLLETZ, R. Setting Inventory Levels of Conwip Flow Lines via Linear Programming. **Business Research**, v. 4, n. 1, p. 98–115, 2011.

HENDRY, L. C.; KINGSMAN, B. G. Production planning systems and their applicability to make-to-order companies. **European Journal of Operational Research**, v. 40, n. 1, p. 1–15, may 1989.

HENDRY, L. C.; KINGSMAN, B. G. Job release: Part of a hierarchical system to manage manufacturing lead times in make-to-order companies. **Journal of the Operational Research Society**, v. 42, n. 10, p. 871–883, 1991.

HENRICH, P.; LAND, M.; GAALMAN, G. Exploring applicability of the workload control concept. **International Journal of Production Economics**, v. 90, n. 2, p. 187–198, 2004.

HERER, Y. T.; MASIN, M. Mathematical programming formulation of CONWIP based production lines; and relationships to MRP. **International Journal of Production Research**, v. 35, n. 4, p. 1067–1076, 1997.

HOOSE, A.; CONSALTER, L. A.; DURÁN, O. M. Implementación de un sistema híbrido tipo trabajo constante en progreso (CONWIP) para control de producción en una industria de implementos agrícolas. **Informacion Tecnologica**, v. 27, n. 2, p. 111–120, 2016.

HOPP, W. J.; ROOF, M. L. Setting WIP levels with statistical throughput control (STC) in CONWIP production lines. **International Journal of Production Research**, v. 36, n. 4, p. 867–882, 1998.

HOPP, W. J.; SPEARMAN, M. L. To Pull or Not to Pull: What Is the Question? **Manufacturing & Service Operations Management**, v. 6, n. 2, p. 133–148, abr. 2004.

HOPP, W. J.; SPEARMAN, M. L.; WOODRUFF, D. L. Practical Strategies for Lead Time Reduction. **Manufacturing Perspective**, v. 3, n. 2, p. 78–84, 1990.

HUANG, G.; CHEN, J.; WANG, X.; SHI, Y. A simulation study of CONWIP assembly with multi-loop in mass production, multi-products and low volume and OKP environments. **International Journal of Production Research**, v. 53, n. 14, p. 4160–4175, 2015.

HUANG, G.; CHEN, J.; WANG, X.; SHI, Y. An approach of designing CONWIP loop for assembly system in one-of-a-kind production environment. **International Journal of Computer Integrated Manufacturing**, v. 29, n. 7, p. 805–820, 2016.

HUANG, G.; CHEN, J.; WANG, X.; SHI, Y.; TIAN, H. From loop structure to policy-making: a CONWIP design framework for hybrid flow shop control in one-of-a-kind production environment. **International Journal of Production Research**, v. 55, n. 12, p. 3374–3391, 2017.

HUANG, G.; CHEN, J.; KHOJASTEH, Y. A cyber-physical system deployment based on pull strategies for one-of-a-kind production with limited resources. **Journal of Intelligent Manufacturing**, 2020.

HUANG, M.; WANG, D.; IP, W. H. A simulation and comparative study of the CONWIP. Kanban and MRP production control systems in a cold rolling plant. **Production Planning and Control**, v. 9, n. 8, p. 803–812, 1998.

HYER, N. L.; BROWN, K. A; ZIMMERMAN, S. A socio-technical systems approach to cell design: case study and analysis. **Journal of Operations Management**, v. 17, n. 2, p. 179–203, 1999.

IP, W. H.; HUANG, M.; YUNG, K. L.; WANG, D.; WANG, X. CONWIP based control of a lamp assembly production line. **Journal of Intelligent Manufacturing**, v. 18, n. 2, p. 261–271, 2007.

JAEGLER, Y.; JAEGLER, A.; BURLAT, P.; LAMOURI, S.; TRENTESAUX, D. The ConWip production control system: a systematic review and classification. **International Journal of Production Research**, v. 56, n. 17, p. 5736–5756, 2018.

JEON, S. M.; KIM, G. A survey of simulation modeling techniques in production planning and control (PPC). **Production Planning & Control**, v. 27, n. 5, p. 360–377, 2016.

JOHNSON, R. T.; MONTGOMERY, D. C.; JONES, B.; FOWLER, J. W. Comparing designs for computer simulation experiments. **Proceedings - Winter Simulation Conference**. Miami, FL, USA, USA: IEEE, 2008Disponível em: <https://ieeexplore.ieee.org/document/4736101>

KABADURMUS, O. A Comparative Study of POLCA and Generic CONWIP Production Control Systems in Erratic Demand Conditions. **IIE Annual Conference. Proceedings**, n. 1983, p. 1197, 2009.

KANET, J. J. Load-limited order release in job shop scheduling systems. **Journal of Operations Management**, v. 7, n. 3–4, p. 44–58, 1988.

KARRER, C. **Engineering Production Control Strategies**. Berlin: Springer-Verlag Berlin Heidelberg, 2012.

KHOJASTEH-GHAMARI, Y. Developing a framework for performance analysis of a production process controlled by Kanban and CONWIP. **Journal of Intelligent Manufacturing**, v. 23, n. 1, p. 61–71, 2012.

KHOJASTEH, Y.; SATO, R. Selection of a pull production control system in multi-stage production processes. **International Journal of Production Research**, v. 53, n. 14, p. 4363–4379, 2015.

KINGSMAN, B. G. Modelling input-output workload control for dynamic capacity planning in production planning systems. **International Journal of Production Economics**, v. 68, n. 1, p. 73–93, 2000.

KINGSMAN, B. G.; TATSIOPOULOS, I. P.; HENDRY, L. C. The structural methodology for managing manufacturing lead times in make-to-order companies. **European Journal of Operational Research**, v. 40, p. 196–209, 1989.

KIRAN, A. S. Simulation and Scheduling. En: BANKS, J. (Ed.). . **Handbook of Simulation**. Atlanta, Georgia: John Wiley and Sons, 1998. p. 677–717.

KLAUSNITZER, A.; NEUFELD, J. S.; BUSCHER, U. Scheduling dynamic job shop manufacturing cells with family setup times: A simulation study. **Logistics Research**, v. 10, n. 1, p. 1–18, 2017.

KLEIJNEN, J. P. C. **Design and Analysis of Simulation Experiments**. 2. ed. Tiburg, Netherlands: Springer International Publishing, 2015.

KORUGAN, A.; GUPTA, S. M. An adaptive CONWIP mechanism for hybrid production systems. **International Journal of Advanced Manufacturing Technology**, v. 74, n. 5–8, p. 715–727, 2014.

KRISHNAMURTHY, A.; SURI, R. Planning and implementing POLCA: a card-based control system for high variety or custom engineered products. **Production Planning & Control**, v. 20, n. 7, p. 596–610, 2009.

KUHNLE, A.; KAISER, J. P.; THEISS, F.; STRICKER, N.; LANZA, G. Designing an adaptive production control system using reinforcement learning. **Journal of Intelligent Manufacturing**, 2020.

LAGE JUNIOR, M.; GODINHO FILHO, M. Variations of the kanban system: Literature review and classification. **International Journal of Production Economics**, v. 125, n. 1, p. 13–21, may 2010.

LAND, M. Parameters and sensitivity in workload control. **International Journal of Production Economics**, v. 104, n. 2, p. 625–638, 2006.

LAND, M.; GAALMAN, G. Workload control concepts in job shops a critical assessment. **International Journal of Production Economics**, v. 46–47, p. 535–548, 1996.

LAND, M. J. Cobacabana (control of balance by card-based navigation): A card-based system for job shop control. **International Journal of Production Economics**, v. 117, n. 1, p. 97–103, 2009.

LAND, M. J.; GAALMAN, G. J. C. The performance of workload control concepts in job shops: Improving the release method. **International Journal of Production Economics**, v. 56–57, p. 347–364, 1998.

LAW, A. M. A tutorial on design of experiments for simulation modeling. **Proceedings - Winter Simulation Conference**, p. 550–564, 2018.

LEE, C.-Y.; PIRAMUTHU, S.; TSAI, Y.-K. Job shop scheduling with a genetic algorithm and machine learning. **International Journal of Production Research**, v. 35, n. 4, p. 1171–1191, 1997.

LI, J.-W. Simulation study of coordinating layout change and quality improvement for adapting job shop manufacturing to CONWIP control. **International Journal of Production Research**, v. 48, n. 3, p. 879–900, 2010.

LIN, P.; SHEN, L.; ZHAO, Z.; HUANG, G. Q. Graduation manufacturing system: synchronization with IoT-enabled smart tickets. **Journal of Intelligent Manufacturing**, v. 30, n. 8, p. 2885–2900, 2019.

LÖDDING, H. A manufacturing control model. **International Journal of Production Research**, v. 50, n. 22, p. 6311–6328, 2011.

LÖDDING, H. **Handbook of Manufacturing Control Fundamentals, description, configuration**. Hamburg, Germany: Springer-Verlag Berlin Heidelberg, 2013.

LÖDDING, H.; PIONTEK, A. The surprising effectiveness of earliest operation due-date sequencing. **Production Planning and Control**, v. 28, n. 5, p. 459–471, 2017.

LÖDDING, H.; YU, K.-W.; WIENDAHL, H.-P. Decentralized WIP-oriented manufacturing control (DEWIP). **Production Planning & Control**, v. 14, n. 1, p. 42–54, 2003.

LUAN, S.; JIA, G.; KONG, J. A Model Approach to POLCA System for Quick Response Manufacturing Formal Description and Mathematical Programming Formal description for POLCA system. **Journal of Computational Information Systems**, v. 3, n. 9, p. 1167–1174, 2013.

MASIN, M.; PRABHU, V. AWIP: A simulation-based feedback control algorithm for scalable design of self-regulating production control systems. **IIE Transactions (Institute of Industrial Engineers)**, v. 41, n. 2, p. 120–133, 2009.

MEHRSAI, A.; KARIMI, H. R.; SCHOLZ-REITER, B. Toward learning autonomous pallets by using fuzzy rules, applied in a Conwip system. **International Journal of Advanced Manufacturing Technology**, v. 64, n. 5–8, p. 1131–1150, 2013.

MELNYK, S. A.; RAGATZ, G. L. Order review/release: research issues and perspectives. **International Journal of Production Research**, v. 27, n. 7, p. 1081–1096, 1989.

MICLO, R.; LAURAS, M.; FONTANILI, F.; LAMOTHE, J.; MELNYK, S. A. Demand Driven MRP: assessment of a new approach to materials management. **International Journal of Production Research**, v. 7543, p. 1–16, 2018.

MONTGOMERY, D. C. **Design and Analysis of Experiments**. 7. ed. New Jersey: John Wiley and Sons, 2009.

MUDA, S.; HENDRY, L. Proposing a world-class manufacturing concept for the make-to-order sector. **International Journal of Production Research**, v. 40, n. 2, p. 353–373, ene. 2002.

MÜLLER, E.; TOLUJEW, J.; KIENZLE, F. Push-Kanban – a kanban-based production control concept for job shops. **Production Planning & Control**, v. 25, n. 5, p. 401–413, 2014.

NEUNER, P.; HAEUSSLER, S. Rule based workload control in semiconductor manufacturing revisited. **International Journal of Production Research**, v. 0, n. 0, p. 1–20, 2020.

NIEHUES, M.; SELLMAIER, P.; STEINHAEUSSER, T.; REINHART, G. Adaptive Job-Shop Control Using Resource Accounts. **Procedia CIRP**, v. 57, p. 351–356, 2016.

OHNO, T. **Toyota System Production: Beyond Large-Scale Production**. New York: Productivity Press, 1988.

OLAITAN, O.; FRAGAPANE, G.; ALFNES, E.; STRANDHAGEN, J. O. A Nested Configuration of POLCA and Generic Kanban in a High Product Mix Manufacturing System. En: LÖDDING, H.; RIEDEL, R.; THOBEN, K.-D. (Eds.). . **Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable**. Cham: Springer International Publishing, 2017. p. 518–526.

OLAITAN, O.; ALFNES, E.; VATN, J.; STRANDHAGEN, J. O. CONWIP implementation in a system with cross-trained teams. **International Journal of Production Research**, v. 57, n. 20, p. 6473–6486, 2019.

OLHAGER, J.; PERSSON, F. Using simulation-generated operating characteristics curves for manufacturing improvement. **IFIP International Federation for Information Processing**, v. 257, p. 195–204, 2008.

ONYEOCHA, C. E. Effect of product mix on multi-product pull control. **Simulation Modelling Practice and Theory**, v. 56, p. 16–35, 2015.

ONYEOCHA, C. E.; KHOURY, J.; GERAGHTY, J. A comparison of Kanban-Like control strategies in a multi-product manufacturing system under erratic demand. **Proceedings of the 2013 Winter Simulation Conference - Simulation: Making Decisions in a Complex World, WSC 2013**. 2013

ONYEOCHA, C. E.; KHOURY, J.; GERAGHTY, J. Robustness analysis of pull strategies in multi-product systems. **Journal of Industrial Engineering and Management**, v. 8, n. 4, p. 1125–1161, 2015.

OOSTERMAN, B.; LAND, M.; GAALMAN, G. Influence of shop characteristics on workload control. **International Journal of Production Economics**, v. 68, n. 1, p. 107–119, 2000.

PIEFFERS, J. **Let’s POLCA**. Groningen: Rijksuniversiteit Groningen, 2005.

PORTER, K.; LITTLE, D.; PECK, M.; ROLLINS, R. Manufacturing classifications: relationships with production control systems. **Integrated Manufacturing Systems**, v. 10, n. 4, p. 189–198, 1999.

POWELL, D. ERP systems in lean production: new insights from a review of lean and ERP literature. **International Journal of Operations & Production Management**, v. 33, n. 11, p. 1490–1510, 2013.

POWELL, D.; RIEZEBOS, J.; STRANDHAGEN, J. O. Lean production and ERP systems in small- and medium-sized enterprises: ERP support for pull production. **International Journal of Production Research**, v. 51, n. 2, p. 395–409, 15 ene. 2013.

PRAKASH, J.; CHIN, J. F. Modified CONWIP systems: A review and classification. **Production Planning and Control**, v. 26, n. 4, p. 296–307, 2014.

PRAKASH, J.; FENG, C. A comparison of push and pull production controls under machine breakdown. **International Journal of Business Science and Applied Management**, v. 6, n. 3, p. 58–70, 2011.

PTAK, C.; SMITH, C. **Demand Driven Material Requirements Planning (DDMRP)**. Connecticut: Industrial Press, Inc., 2016.

QI, C.; SIVAKUMAR, A. I.; GERSHWIN, S. B. Impact of production control and system factors in semiconductor wafer fabrication. **IEEE Transactions on Semiconductor Manufacturing**, v. 21, n. 3, p. 376–389, 2008.

QI, C.; SIVAKUMAR, A. I.; GERSHWIN, S. B. An efficient new job release control methodology. **International Journal of Production Research**, v. 47, n. 3, p. 703–731, 2009.

RENNA, P.; MAGRINO, L.; ZAFFINA, R. Dynamic card control strategy in pull manufacturing systems. **International Journal of Computer Integrated Manufacturing**, v. 26, n. 9, p. 881–894, 2013.

REZAEI, K.; EIVAZY, H.; REZAZADEH, A.; NAZARI-SHIRKOUHI, S. A Production Planning and Scheduling Model for Semiconductor Wafer Manufacturing Plants. **Proceedings of the 41st International Conference on Computers & Industrial Engineering**. 2011Disponível em: <http://sunedisonsilicon.com/index.php?view=Semiconductor-Wafers>

RIEZEBOS, J. Polca simulation of a unidirectional flow system. **Proceedings of the Third International Conference on Group Technology/Cellular Manufacturing**, 2006.

RIEZEBOS, J. Design of POLCA material control systems. **International Journal of Production Research**, v. 05, p. 1455–1477, 2010.

RIEZEBOS, J. POLCA: A State-of-the-Art Overview of Research Contributions. En: SURI, R. (Ed.). . **The practitioner’s guide to POLCA : the production control system for high-mix, low-volume and custom products**. Boca Raton, FL: CRC Press, 2018.

RIEZEBOS, J.; KORTE, G. J.; LAND, M. J. Improving a practical DBR buffering approach using Workload Control. **International Journal of Production Research**, v. 41, n. 4, p. 699–712, 2003.

RONEN, B.; STARR, M. K. Synchronized manufacturing as in OPT: from practice to theory. **Computers and Industrial Engineering**, v. 18, n. 4, p. 585–600, 1990.

RYAN, S. M.; FRED CHOOBINEH, F. Total WIP and WIP mix for a CONWIP controlled job shop. **IIE Transactions**, v. 35, n. 5, p. 405–418, 2003.

SALUM, L.; ARAZ, Ö. U. Using the when/where rules in dual resource constrained systems for a hybrid push-pull control. **International Journal of Production Research**, v. 47, n. 6, p.

1661–1677, 2009.

SATO, R.; KHOJASTEH-GHAMARI, Y. An integrated framework for card-based production control systems. **Journal of Intelligent Manufacturing**, v. 23, n. 3, p. 717–731, 2012.

SATOGLU, S. I.; UCAN, K. Redesigning the material supply system of the automotive suppliers based on lean principles and an application. **IEOM 2015 - 5th International Conference on Industrial Engineering and Operations Management, Proceeding**. IEEE, 2015

SCHRAGENHEIM, E.; RONEN, B. Drum-buffer-rope shop floor control. **Production and Inventory Management Journal**, v. 31, n. 3, p. 18–22, 1990.

SEVERINO, M. R.; GODINHO FILHO, M. POLCA system for supply chain management: simulation in the automotive industry. **Journal of Intelligent Manufacturing**, v. 30, n. 3, p. 1271–1289, 2019.

SILVA, C.; REIS, V.; MORAIS, A.; BRILENKOV, I.; VAZA, J.; PINHEIRO, T.; NEVES, M.; HENRIQUES, M.; VARELA, M. L.; PEREIRA, G.; et al. A comparison of production control systems in a flexible flow shop. **Procedia Manufacturing**, v. 13, p. 1090–1095, 2017.

SIMONS, J. V; SIMPSON, W. P. An Exposition of Multiple Constraint Scheduling As Implemented in the Goal System (Formerly Disastertm). **Production and Operations Management**, v. 6, n. 1, p. 3–22, 1997.

SLOMP, J.; BOKHORST, J. A. C.; GERMS, R. A lean production control system for high-variety/low-volume environments: a case study implementation. **Production Planning & Control**, v. 20, n. 7, p. 586–595, oct. 2009.

SPEARMAN, M.; WOODRUFF, D.; HOPP, W. CONWIP: a pull alternative to kanban. **International Journal of Production Research**, v. 28, n. 5, p. 879–894, 1990.

STEVENSON, M.; HENDRY, L. C. Aggregate load-oriented workload control: A review and a re-classification of a key approach. **International Journal of Production Economics**, v. 104, n. 2, p. 676–693, 2006.

STEVENSON, M.; HENDRY, L. C.; KINGSMAN, B. G. A review of production planning and control: The applicability of key concepts to the make-to-order industry. **International Journal of Production Research**, v. 43, n. 5, p. 869–898, 2005.

SUGIMORI, Y.; KUSUNOKI, K.; CHO, F.; UCHIKAWA, S. Toyota production system and kanban system materialization of just-in-time and respect-for-human system. **International Journal of Production Research**, v. 15, n. 6, p. 553–564, 1977.

SURI, R. **Quick response manufacturing: a companywide approach to reducing lead times**. Portland, Oregon: Productivity Press, 1998.

SURI, R. **The practitioner’s guide to POLCA : the production control system for high-mix, low-volume and custom products**. Boca Raton, FL: CRC Press, 2018.

SURI, R.; KRISHNAMURTHY, A. How to Plan and Implement POLCA–A Material Control

System for High Variety or Custom-Engineered Products. **Center for Quick Response Management, Technical Report**. 2003Disponível em: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:How+to+Plan+and+Implement+POLCA+:+A+Material+Control+System+for+High-Variety+or+Custom-Engineered+Products#0>. Acesso em: 9 may. 2014

TAKO, A. A.; ROBINSON, S. The application of discrete event simulation and system dynamics in the logistics and supply chain context. **Decision Support Systems**, v. 52, n. 4, p. 802–815, 2012.

THÜRER, M.; STEVENSON, M.; SILVA, C.; LAND, M. J.; FREDENDALL, L. D. Workload control and order release: A lean solution for make-to-order companies. **Production and Operations Management**, v. 21, n. 5, p. 939–953, 2012.

THÜRER, M.; QU, T.; STEVENSON, M.; MASCHEK, T.; GODINHO FILHO, M. Continuous workload control order release revisited: an assessment by simulation. **International Journal of Production Research**, v. 52, n. 22, p. 6664–6680, 2014.

THÜRER, M.; LAND, M. J.; STEVENSON, M.; FREDENDALL, L. D.; GODINHO FILHO, M. Concerning Workload Control and Order Release: The Pre-Shop Pool Sequencing Decision. **Production and Operations Management**, v. 24, n. 7, p. 1179–1192, 2015a.

THÜRER, M.; LAND, M. J.; STEVENSON, M.; FREDENDALL, L. D.; GODINHO FILHO, M. Concerning Workload Control and Order Release: The Pre-Shop Pool Sequencing Decision. **Production and Operations Management**, v. 24, n. 7, p. 1179–1192, 2015b.

THÜRER, M.; FERNANDES, N. O.; CARMO-SILVA, S.; STEVENSON, M. Improving performance in POLCA controlled high variety shops: An assessment by simulation. **Journal of Manufacturing Systems**, v. 44, p. 143–153, 2017a.

THÜRER, M.; FERNANDES, N. O.; STEVENSON, M.; QU, T. On the backlog-sequencing decision for extending the applicability of ConWIP to high-variety contexts: an assessment by simulation. **International Journal of Production Research**, v. 55, n. 16, p. 4695–4711, 2017b.

THÜRER, M.; STEVENSON, M.; SILVA, C.; QU, T. Drum-buffer-rope and workload control in High-variety flow and job shops with bottlenecks: An assessment by simulation. **International Journal of Production Economics**, v. 188, n. April, p. 116–127, 2017c.

THÜRER, M.; FERNANDES, N. O.; STEVENSON, M.; SILVA, C.; CARMO-SILVA, S. POLC-A: an assessment of POLCA's authorization element. **Journal of Intelligent Manufacturing**, v. 30, n. 6, p. 2435–2447, 2019a.

THÜRER, M.; FERNANDES, N. O.; ZIENGS, N.; STEVENSON, M. On the meaning of ConWIP cards: an assessment by simulation. **Journal of Industrial and Production Engineering**, v. 00, n. 00, p. 1–10, 2019b.

THÜRER, M.; FERNANDES, N. O.; STEVENSON, M.; QU, T.; LI, C. D. Centralised vs. decentralised control decision in card-based control systems: comparing kanban systems and COBACABANA. **International Journal of Production Research**, v. 57, n. 2, p. 322–337, 2019c.

THÜRER, M.; FERNANDES, N. O.; STEVENSON, M. Material Flow Control in High-Variety Make-to-Order Shops: Combining COBACABANA and POLCA. **Production & Operations Management**, 2020.

THÜRER, M.; LAND, M. J.; STEVENSON, M. Card-based workload control for job shops: Improving COBACABANA. **International Journal of Production Economics**, v. 147, n. PART A, p. 180–188, 2014.

THÜRER, M.; MA, L.; STEVENSON, M. Workload Control order release in general and pure flow shops with limited buffer size induced blocking : an assessment by simulation. **International Journal of Production Research**, v. 0, n. 0, p. 1–12, 2020.

THÜRER, M.; SILVA, C.; STEVENSON, M. Optimising workload norms : the influence of shop floor characteristics on setting workload norms for the workload control concept. **International Journal of Production Research**, v. 49, n. 4, p. 1151–1171, 2011.

THÜRER, M.; STEVENSON, M. On the beat of the drum: improving the flow shop performance of the Drum–Buffer–Rope scheduling mechanism. **International Journal of Production Research**, v. 56, n. 9, p. 3294–3305, 2018.

THÜRER, M.; STEVENSON, M.; PROTZMAN, C. **Card-based control systems for a lean work design : the fundamentals of kanban, ConWIP, POLCA, and COBACABANA**. Boca Raton, FL: CRC Press, 2016a.

THÜRER, M.; STEVENSON, M.; PROTZMAN, C. W. COBACABANA (Control of Balance by Card Based Navigation): An alternative to kanban in the pure flow shop? **International Journal of Production Economics**, v. 166, p. 143–151, 2015.

THÜRER, M.; STEVENSON, M.; PROTZMAN, C. W. Card-based production control: a review of the control mechanisms underpinning Kanban, ConWIP, POLCA and COBACABANA systems. **Production Planning and Control**, v. 27, n. 14, p. 1143–1157, 2016b.

TRANFIELD, D.; DENYER, D.; SMART, P. Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. **British Journal of Management**, v. 14, n. 3, p. 207–222, sep. 2003.

TÜRKYILMAZ, A.; ŞENVAR, Ö.; ÜNAL, İ.; BULKAN, S. A research survey: heuristic approaches for solving multi objective flexible job shop problems. **Journal of Intelligent Manufacturing**, 2020.

VANDAELE, N.; CLAERHOUT, D.; NIEUWENHUYSE, I. VAN; RONY, C. **E-POLCA to control multi-product, multi-machine job shops**AntwerpUA, , 2004. Disponível em: <https://core.ac.uk/download/pdf/6304342.pdf>

VANDAELE, N.; VAN NIEUWENHUYSE, I.; CLAERHOUT, D.; CREMMERY, R. Load-Based POLCA: An Integrated Material Control System for Multiproduct, Multimachine Job Shops. **Manufacturing & Service Operations Management**, v. 10, n. February 2015, p. 181–197, 2008.

WATSON, K. J.; BLACKSTONE, J. H.; GARDINER, S. C. The evolution of a management philosophy: The theory of constraints. **Journal of Operations Management**, v. 25, n. 2, p. 387–402, 2007.

WIENDAHL, H.-P. **Load-Oriented Manufacturing Control**. New York: Springer-Verlag Berlin Heidelberg, 1995.

WU, S.-Y.; MORRIS, J. S.; GORDON, T. M. A simulation analysis of the effectiveness Drum-Buffer-Rope scheduling in furniture. **Computers & Industrial Engineering**, v. 26, n. 4, p. 757–764, 1994.

YANG, K. K. Managing a flow line with Single-kanban, Dual-kanban or Conwip. **Production and Operations Management**, v. 9, n. 4, p. 349–366, 2000.

ZIENGS, N.; RIEZEBOS, J.; GERMS, R. Placement of effective work-in-progress limits in route-specific unit-based pull systems. **International Journal of Production Research**, v. 50, n. 16, p. 4358–4371, 2012.

APPENDIX A - RESULTS OF THESIS

GÓMEZ PAREDES, F. J.; GODINHO FILHO, M. Sistemas de control de producción para ambientes por encomienda (make-to-order). Congreso de Ingeniería y Arquitectura CONIA. San Salvador: UCA, 2018. Disponível em: http://conia.uca.edu.sv/memoria/

GÓMEZ PAREDES, F. J.; COSTA, L. B. M.; GODINHO FILHO, M. Enseñando sistemas de control de producción: juego de POLCA. Congreso de Ingeniería y Arquitectura CONIA. San Salvador: UCA, 2019. Disponível em: http://hdl.handle.net/11674/3814

GÓMEZ PAREDES, F. J.; GODINHO FILHO, M.; THÜRER, M.; FERNANDES, N. O.; et al. Factors for choosing production control systems in make-to-order shops: a systematic literature review. Journal of Intelligent Manufacturing, Springer US, 2020. Disponível em: https://doi.org/10.1007/s10845-020-01673-z

APPENDIX B - DETAILS OF REVIEWED STUDIES IN THE LITERATURE REVIEW

Table B.1 – Characteristics of the simulated shop in the reviewed papers

Authors/Year	**Order Arrivals**	**Processing and setup times**	**Machines**	**Job routings**	**Shop load factors**	**Due dates**	**Dispatching rule**	**Pool sequence rule**
Bertolini et al. (2020)	Random.Exponential for 90% utilization	Random.Erlang-2 truncated (mean 1, max 4)	7	Directed routing, randomly choose for stations	90%	Not information	WINQ, FIFO	WINQ, FIFO
Huang et al. (2020)	Random.Exponential (300 min)	Random Uniform [1, 1500] min	3	Random directed routing	87%, 92%	Arrival time + Total work content	FIFO, EODD, SPT	FIFO, EODD, SPT
Neuner and Haeussler (2020)	Random exponential (98 min)	Log normal (mean by each station according product mix)	11	Directed flow by product mix (re-entrant flow allowed)	90%	Arrival time + Uniform allowance [7868; 14,612]	FIFO	FIFO, EODD, CR
Thürer et al. (2020)	Order arrival: Random.Exponential (0.642)	Random.Erlang-2 truncated (mean 1, max 4)	6	Random routing; no re-entrant flows	90%	Arrival time + Uniform allowance [30,50]	ERD	ERD
Olaitan et al. (2019)	Historical data	Historical data	10	Product mix predefined routing (36)	Not shown	Not used	FIFO	FIFO
Thürer *et al*. (2019a)	Random.Exponential (1.111)	Random.Lognormal (mean: depends on each station to have the same utilization; cv² = 0.25, 0.5, 1)	7	Directed routing, randomly choose for stations	90%	Arrival time + Uniform allowance [75,100]	ERD, PST	ERD, PST
Bong et al. (2018)	Empirical data	Empirical data	Not reported	Not reported	80%, 85%, 90%	Not used	EODD	ERD
Carmo-Slva e Fernandes (2018)	Order arrival: Random.Exponential (0.648)	Random.Erlang-2 truncated (mean 1, max 4)	6	Homogeneous mix, predefined routings	90%	Arrival time + Uniform allowance [33,55]	EODD	ERD, STWK, CScor, CSjdir EF

Authors/Year	Order Arrivals	Processing and setup times	Machines	Job routings	Shop load factors	Due dates	Dispatching rule	Pool sequence rule
Gonzalez-R et al. (2018)	Random.Exponential (0.703)	Random.Erlang-2 truncated (mean 1, max 4)	8	Random routing; no re-entrant flows	80%	Arrival time + Uniform allowance [40,50]	FIFO, EODD, SRPT, CR	FIFO, EODD, SRPT, CR
Thürer and Stevenson (2018)	Random.Exponential (0.635)	Random.Erlang 2-Truncated * factor: (1 bottleneck, 0.85 non-bottleneck)	7	Random routing; no re-entrant flows	90% at bottleneck station	Arrival time + Constant allowance Uniform [28; 36]	FIFO, PRD, SPT, MODPRD (on bottleneck)	FIFO, PST, SPT, MODPST (on bottleneck)
Fernandes et al. (2017a)	Random.Exponential (0.648)	Random.Erlang-2 truncated (mean 1, max 4)	6	Random routing	90%	Arrival time + Constant allowance Uniform [28, 50]	FIFO; EODD	ERD, MODCS, Method (LUMS COR, CR, CR+, C-BSA)
Fernandes et al. (2017b)	Random.Exponential (0.648)	Random.Erlang-2 Truncated (Mean 1, max 4)	6	Random, directed flow	90%	Arrival time + Constant allowance Uniform [35, 55]	EODD	ERD, CS MODCS
Thürer *et al.* (2017a)	Random.Exponential (0.738);	Random.Erlang-2 Truncated (0.99 max 4); Independent sequence setup times.	6	Directed, random routing for Random.Uniform [2, 6]	90%	Arrival time + Uniform allowance [33,55]	ERD, SPT, MERD	ERD, CS, MODCS
Thürer *et al.* (2017c)	Random.Exponential(0.635)	Random.Erlang 2-Truncated * factor: (1 bottleneck, 0.95 moderate, 0.80 severe, 0.65 very severe on non-bottleneck); Independent sequence setup times	7	Random routing, no-entrant flows	90% at bottleneck station	Arrival time + Constant allowance Uniform [32,40]	PST, SPT, MPST	LUMS COR, CR, SA COR, DBR
Carmo-Silva and Fernandes (2017)	Random.Exponential(0.642)	Random.Erlang-2(1), max: 4;	6	Random routing	90%	Arrival time + Time allowance U(25,55)	PST	CR, FBC, DBC, IMR

Authors/Year	Order Arrivals	Processing and setup times	Machines	Job routings	Shop load factors	Due dates	Dispatching rule	Pool sequence rule
Silva et al. (2017)	Random. Exponential (0.648 h)	Random.Erlang-2(i), max: 4, i from a table for each product; MTBF = Random.Exponential (3000); MTTR (3)	3 stages of 3 operations each	Random routing.	90%	Arrival time + Time allowance U(10,30)	FIFO	MNJ; LLM
Barros *et al.* (2016)	Random.Exponential(0.647)	2-Erlang, mean = 0.4 and truncated at 1.6 time-units for operation; Set-up time Sequence independent	6	Operation per job: Discrete uniformly distributed [1, 6] operations Split jobs in U[1,4] smaller lots from the order quantity.	90%	Arrival + Uniform [33,55]	Job attribute	FIFO
Frazee and Standridge (2016)	Random.Exponential (1 day / 8 parts)	Not detailed	12	Product mix of 8 items with parts, operations grouped	Less than 50% with a bottleneck in 80%	Not used	Not information	Not information
Gomes et al. (2016)	Random.Exponential(0.667)	Random.Erlang-2(mean 0.4), max: 1.6	6	Random routing, uniform [1,6]	90%	Arrival time + Time allowance U[35,55]	FIFO	PRD
Huang *et al.* (2016)	Empirical distribution	Random.Normal(100, Var); Var: L = 5, M= 30, H = 60	12	Assembly routing	Not estimated	Arrival time + Uniform (Total Work content)	FIFO until loop	FIFO
Niehues et al. (2016)	Read from the empirical table as product mix (unknown data)	Read from the empirical table as product mix (unknown data)	1 - 5 machines	Random product mix	Unknown	Not declared	XWINQ, LSK/RO, LSUT	PRD
Bertolini, Romagnoli and Zammori (2015)	Random.Exponential for 90% utilization	Random.Erlang-2 truncated (mean 1, max 4)	6	Random routing, uniformly [1,6]	90%	Arrival time + 28 +Constant allowance Uniform [0, 30]	FIFO, EODD, PST, SPT	CR for LB and OB
Braglia, Castellano and Frosolini (2015)	Random.Exponential(1.6)	Random.Erlang(Mean, k=2) Using Mean: A=C =1, B=1.75, D=1.75;	4	3-product mix (60%, 30%, 10%) in predefined routing	Unbalanced stations, only	Not considered	FIFO; LQ-POLCA; WINQ	PRD

Authors/Year	Order Arrivals	Processing and setup times	Machines	Job routings	Shop load factors	Due dates	Dispatching rule	Pool sequence rule
Eng, Ching and Siong (2015)	Empirical data (not shown)	Lead-time for A = 2.55 days; B = C = 2 days.	3	2 Jobs in the paired loop	Not declared	Not used	FIFO	PRD
Huang *et al*. (2015)	Not declared	Independent setup times	6	Random routing	Upper bound WIP considered as the load response.	Not used	FIFO	Not information
Onyeocha (2015)	Product mix, empirical data for 6 weeks in 4 datasets	Empirical data by station, considering both scenarios, using deterministic; Setup times as Random.Normal by each station; MTBF: Exponential by Station; MTTR: Exponential by each station.	9	Product mix from the dataset in demand	Not considered	Not considered	Using product demand as a priority for stages	Not information
Khojasteh and Sato (2015)	Not information	Random.Exponential (S1=5, S2=10, S3=25)	7 in an assembly of 3 stages	Fixed	Not information	Not used	FIFO	Not information
Onyeocha, Khoury and Geraghty (2015)	Random.Normal(5.61,2.805) = A; Random.Normal(5.72, 0.572) = B; Erratic demand as table	Deterministic (1.5), Deterministic (3), MTBF Exponential (90 h), MTTR Exponential (10 h); Deterministic as empirical data; Setup times: Random.Normal(0.327, 0.109)	3 to 5 stations	Setting 2 products for each process as Linear flow	Used as a control variable	Not used	FIFO and family attribute priority, using demand information	FIFO
Braglia, Castellano and Frosolini, (2014)	Random.Exponential(1.6 min)	Random.Erlang-2 (A =1, B=1.75, C=1, D=1.5), Independent sequence setup time	4	3 Predefined routings, for a specific loop (P1, 60%; P2, 30%, P3, 10%)	Not information	Not used	LQ (MNJ)	PRD

Authors/Year	Order Arrivals	Processing and setup times	Machines	Job routings	Shop load factors	Due dates	Dispatching rule	Pool sequence rule
Müller, Tolujew and Kienzle (2014)	From cases, not presented	From cases, not presented	Unknown	Established by products	Unknown	Estimated on multivariable linear regression (unknown parameters)	EODD blocked by WIP	PRD
Thürer, Land and Stevenson (2014)	Random Exponential (0.648)	Random.Erlang-2 Truncated (1 max 4); Independent sequence setup times.	6	Random routing; no re-entrant flow	90%	Arrival time + Time allowance U[28,55]	EODD	PRD, ODD
Aziz *et al*. (2013)	Read from the table as demand	Deterministic times	3	Defined routes by each family product	Unknown	Not used	Not described	Not described
Onyeocha, Khoury and Geraghty (2013)	Product mix, empirical data for 6 weeks in 4 datasets	Reading for the empirical data table. Deterministic by each station; Setup times as Random.Normal by each station; MTBF: Exponential by Station; MTTR: Exponential by each station.	5	Routes from dataset in demand	Considered as Service Level	Not used	FIFO	PRD
Rezaei et al. (2011)	Poisson Arrival Rate of 5 lots per day, from six different order types	Product mix table for each type of order	10	Defined routes by product	Not used	Arrival time + Cycle Time * Random. Uniform [1,3]	CRT bottleneck, SPT in nonbottleneck, FIFO for ties	C-BSA, using CRT
Bokhorst and Slomp (2010)	Not information	Random.Gamma (Mean: 32, as Erlang-8 to Exponential (32))	7 with 5 workers	Randomly assigned in 4 steps for each job	Estimated near 100%	Not used.	EODD Manual for urgent jobs	CR, PR, ERD
Germs and Riezebos (2010)	Simultaneously, uniform (1, 10); Arrival rate estimated from utilization level, using Random Exponential.	Erlang-2 distributed, cv^2 = 0,5 A=1, B=C=2, D=E=F=G=4	7	4 types, 25% each, random	80%; 85%; 90%	Not used	FIFO	CR
Salum and Araz (2009)	Random.Exponential(each product has its mean in the table)	Random.Normal(each station has its own mean, SD= 0,15*Mean)	24	Using 10 routes for cells	Using as a performance variable	Arrival time + Total work content	FIFO, SPT, EODD, CRT Workers in EODD (LNQ, LWQ, SPT)	FIFO, SPT, EODD, CRT (matching with dispatching)

Authors/Year	Order Arrivals	Processing and setup times	Machines	Job routings	Shop load factors	Due dates	Dispatching rule	Pool sequence rule
Qi, Sivakumar and Gershwin (2008)	Historical data from orders	Historical data from orders; MTTR considered	511 grouped in 73 cells	Use defined routes for each product	Not considered	Arrival time + Constant order allowance	FIFO; EODD; CRT	PR for OB, LB
Ip *et al*. (2007)	Empirical distribution (shows each distribution with a probability graphic)	Table distribution for each of 3 families of products for 21 machines	21, in Assembly line	Fixed to 3 types of products	Not considered	Not considered	FIFO	FIFO
Baysan, Kabadurmus and Durmusoglu (2007)	Internal rate read from the table, using a Normal distribution for demand rate	Reading for a table for each process and product. Unknown distribution. Independent sequence setup time.	3	Using a 4-routes for each family product	Not used	Not used	FIFO	FIFO
Fernandes and Carmo-Silva (2006)	Exponential (0.5) to Exponential (0.95)	Exponential (1); Exponential (2) Setup times are negligible	3 workstations	Homogeneous mix, using routing by family	Not declared	Not considered	FIFO until load limit	IMR
Lödding, Ya and Wiendahl (2003)	Not information	Empirical data, not information	37	Random mix used	Variable from 100% above	Not information	FIFO (XWINQ when order exceed load limit)	FIFO
Gilland (2002)	Poisson arrival process = Random.Exponential	0.8 h/lot (Bottleneck 1.0 h/lot)	9	Flow defined by family products	Variable factor (90% - 98%)	Not used	Not information	PR, DBR (as PFB1, PFB2, PFBB), CONWIP
Golany, Dar-El and Zeev (1999)	Read from the table (Not detailed information)	Random.Uniform [0,100]	28	Predefined for 4 families	Not considered	Not used	Not information	Not information

Source: Authors

Table B.2 – General classification of PCS comparison in the reviewed papers

Authors/Year	PCS	Shop Configuration	Performance compared	Experimental factors	Highlighted conclusion for PCS	Analysis Technique
Bertolini et al. (2020)	m-CONWIP, CONWIP	General flow shop	DTM; TTM	Processing times variation (constant and random), dispatching rule (FIFO, WINQ, m-CONWIP)	The adoption of WINQ improves the single-loop CONWIP to have comparable results to m-CONWIP.	Direct comparison
Huang et al. (2020)	Path-Based Bottleneck (PBB), CONWIP, CONWIP Capacity Slack (CSC)	General flow shop	TDP; LAD	PCS in a form of pull control (PBB, CONWIP, CSC)	The PBB control has the best performance of PCSs compared. The Capacity Slack version also has similar results when it uses the EODD and SPT as dispatching rule.	Direct comparison
Neuner and Haeussler (2020)	WLC (COBACABANA order release rule: ConLoad, SA for Bottleneck)	General flow shop	TC; WIP	Order release model (SA, LUMS-COR, ConLOAD), Pool sequencing rule (FIFO, ERD, CR), Quantum limit (according combinations)	The LUMS COR adapted for bottleneck outperforms the SA and ConLOAD. For those rules, CR is the pool sequencing suggested. The LUMS COR used for bottleneck is as good as that rule using in all workstations.	Direct comparison
Thürer et al. (2020)	POLCA, COBACABANA, COBA-POLCA	General flow shop	DTM; TDP; TDM; LAD; TTM	PCS (POLCA, COBACABANA and COBA-POLCA); workload limit (5, 6.3, 7.8, 9.8, 12.2, 15.3 and inf); Limit of cards per loop (3, 4, 6, 10 and inf)	Both mechanisms are complementary because COBACABANA balance the workload released in the shop and POLCA reduces the fluctuation of the direct load. Tight control could reduce this benefit.	ANOVA at 95% significance level
Olaitan et al. (2019)	CONWIP -FIFO, CONWIP-CT	General flow shop	TTM, UT, WIP	PCS CONWIP at loop (FIFO or single loop, CT cross trained loop or m-CONWIP by resources)	The use of the Cross-Trained loops of CONWIP reduces the DTM. These loops have similar values on workload balance across the loops.	Confidence interval of 95%
Thürer *et al.* (2019a)	POLCA, POLC-A	General flow shop	DTM; TDP; TAM; LAD	POLCA system (POLCA with MRP Authorization and POLC) Shop structure (divergent and convergent flow); Processing time variability (cv² = 0.25, 0.5, 1); Allowance accuracy of dispatching rule estimation (-20%, -10%, 0%, +10%, +20%); Number of cards (10, 12, 14, 16, 18, 20 and infinite)	POLC outperforms POLCA by the authorization element of dependence in the MRP release date. Other release rules may be combined with POLC instead of MRP estimation for authorization release.	ANOVA block design with Scheffé multiple-comparison at the 95% significance level

Authors/Year	PCS	Shop Configuration	Performance compared	Experimental factors	Highlighted conclusion for PCS	Analysis Technique
Bong et al. (2018)	POLCA, Utilisation based (UB)	General flow shop	DTM, WIP, TR, TTM	PCS (POLCA, UB), Quantum limit (original, UB: 80%, 75%)	UB has better performance than POLCA in the studied model. The loop for UB is single for station.	Direct comparison
Carmo-Slva e Fernandes (2018)	GPOLCA	General flow shop	DTM; TDP; LAM; LAD	Workload measure (LB, OB); WIP capacity, limited cards (15, 20, 24, 29, 34, 39, infinity cards (or 6, 8, 10, 12, 14, 16, infinity hours))	CScor is a robust pool sequencing that allows the use of the workload measure for limiting in processing time (or LB) or jobs units (as original order Based – OB).	Direct comparison
Gonzalez-R et al. (2018)	WLC, Kanban JS	Pure and General flow shop	DTM; TDP; TDM; TTM; LAD; EP	PCSs (Kanban, WLC LUMS COR), dispatching rules (FIFO, EDD, SRPT, CRE)	WLC and Kanban JS works better with CRE rule than the other rules. Their difference is that WLC uses more the pre-shop pool waiting time than Kanban JS.	Comparison in Pareto frontier
Thürer and Stevenson (2018)	DBR – (MODPRD; SBPT)	General flow shop	DTM; TDP; TDM; TTM	Shop type (GFS, PFS); Backlog sequencing rules (FIFO, PRD, SBPT, MODPRD); dispatching rules (FIFO, PST, SPT, MODPST); buffer limit (9 - 20 jobs)	Order release rule (as backlog sequencing rule) and dispatching rule that focus on urgent jobs (risk of a tardy job) outperform original rules. The combination suggested in GFS is MODPRD with MODPST in the bottleneck station.	ANOVA and Scheffé post hoc analysis at 95% significance level
Carmo-Silva and Fernandes (2017)	Order release as PCS (WLC, FBC, DBC, IMR as push system for baseline reference)	General Flow Shop	DTM; TTM; LAD; TR	Release strategies (WLC, DBN, FBN, IMR); Protective Capacity (0%, 5%, 10%); Norm levels (8 levels)	WLC has better performance than FBC, DBC by balancing the function in workstations. Balancing the load across workstations is the main reason that WLC outperforms. The bottleneck position is also important to performance. The more downstream the BN is, the worse the performance is.	Direct comparison
Fernandes et al. (2017a)	WLC order releases (CR+, C-BSA, LUMS COR)	General flow shop	DTM; TDP; TDM; LAD	Release method (LUMS COR, CR, CR+, C-BSA); Pool sequence rule (PRD, MODCS); Dispatching rule (FIFO, ODD); Workload norm levels (6 levels)	C-BSA and CR+ (both integrate starvation avoidance) have better performance as a release method, but it has an interaction with the dispatching rule. In high workloads, the release methods create the SPT effect. Integrating a starvation avoidance trigger in CR improves TDM; TDD.	ANOVA and Scheffé post hoc analysis at the 95% significance level (Tukey comparison used but not shown)

Authors/Year	PCS	Shop Configuration	Performance compared	Experimental factors	Highlighted conclusion for PCS	Analysis Technique
Fernandes et al. (2017b)	LB-POLCA with WLC rules	General flow shop	DTM; TR; TDM; LAD	Load accounting approach (1st, 2nd, both, corrected); card acquisition rule (ERD, CS, MODCS); load norms (6 rules)	The load accounting approach based on the second station is the best performance. The corrected load approach from Workload Control literature needs corrections to be included in the POLCA system.	Direct comparison
Silva et al. (2017)	GKS, ATKS, POLCA	General flow shop	DTM; TTM; TR	PCS (ATKS, GKS, POLCA); Job allocation rules (Minimum jobs, least loaded machine); Number of cards (8 levels)	ATKS: outperforms POLCA but is worse than GKS. Priority rule does not have a significant effect on the performance of this study.	Direct comparison
Thürer *et al.* (2017a)	POLCA, POLCA-SA	General Flow Shop	DTM; LAM; TDP; LAD; TTM	Starvation Avoidance (SA); Mechanism (0, 1, 2, infinite cards); Card allocation rule; Dispatching rule; Blocking factor: Number of POLCA cards (8:16:2, infinite)	POLCA – SA (Starvation Avoidance) has better performance than simple POLCA even if the number of cards violates the limit. Modified ERD (MERD) and the capacity slack-based card allocation rule improve performance.	ANOVA and Scheffé post hoc analysis at 95% significance level
Thürer *et al.* (2017c)	DBR - Workload Control norms	General flow shop and Pure job shop	DTM; TTM; TDP; TDM	Norm/buffer level for Release Methods - Blocked factor; Release method; Dispatching rules; Bottleneck severity (moderate, severe, very severe); Shop type (pure job shop, general flow shop).	Workload control has better performance than DBR in balanced and severe bottlenecks, but DBR outperforms WLC if there is a severe bottleneck. All variables have significant main effects.	ANOVA block design with Scheffé multiple-comparison at the 95% significance level.
Barros *et al.* (2016)	CONWIP; POLCA	General flow shop	DTM; LAD; TDP; TTM	PCS; Number of cards; Lot splitting policies (3 levels)	POLCA outperforms CONWIP concerning throughput time and lateness. The split policy has no significant effect on the results.	Direct comparison
Frazee and Standridge (2016)	CONWIP; POLCA	General flow shop/ Pure Flow shop	TTM; WIP; TR	PCS compared (CONWIP; POLCA); Random presence (yes, no)	CONWIP (one loop) has better performance than POLCA in throughput and WIP level. POLCA´s loop used creates a linear flow. Both systems have similar throughput times. The utilization level is considered low compared to other studies.	Descriptive comparison

Authors/Year	PCS	Shop Configuration	Performance compared	Experimental factors	Highlighted conclusion for PCS	Analysis Technique
Gomes et al. (2016)	CONWIP; GKS	General Flow Shop	TTM, DTM, DDR, LAD	PCS (CONWIP, GKS; IMR as push system reference); Lot splitting policies (no split, split before and release sub lots, split in shop floor after release); Number of cards (5 levels)	GKS outperforms CONWIP in TR, LAD and DTM because it has load-balancing capability over workstations. Lot splitting policies have a positive impact on both systems. Splitting after the release point has better results as a splitting policy.	Direct comparison
Huang *et al.* (2016)	CONWIP; m-CONWIP	General flow shop (assembly stations)	TTM; WIP	Flow fluctuation level (variance at processing time); 8 types of loops of CONWIP	M-CONWIP outperforms CONWIP depending on the looping configuration. Delivery time accuracy is more important for OKP.	Graphic and descriptive analysis
Niehues et al. (2016)	DEWIP modification; MRP (as unlimited WIP)	Job shop	WIP	Accounting load approach to determine the WIP limit	Resource load accounting establishes different behaviours in limiting WIP. This helps to identify possible bottlenecks and downstream problem sequences.	Direct comparison
Bertolini, Romagnoli and Zammori (2015)	WLC; CONWIP; Push	General Flow Shop and Pure Job shop	TR; TDM; TDD; WQM; LAP; TTM; DTM; LQM	Type of control (Push, Shop Load WLC, CONWIP); Sorting probability (0%, 50%, 100%); Norm level (each control in unit); Dispatching rules	Push outperforms CONWIP and WLC in Delivery Time, mean and due dates, and even the waiting time is on the shop floor. HPPC reduces shop floor time because it allows a low level of WIP compared to a push system. The dispatching rule has a dominant effect.	ANOVA Nested factors for sorting probability, focusing on PPC (Only mean values presented)
Braglia, Castellano and Frosolini (2015)	POLCA FIFO; POLCA LQ	General flow shop	TTM; WQM	PCS POLCA type (Normal as FIFO, and Longest Queue as dispatching rule)	POLCA using the Longest Queue rule has better performance than the FIFO rule for throughput time, using an unbalanced shop.	Paired t, Interval confidence, 99% level
Eng, Ching and Siong (2015)	POLCA; MRP (Push system)	General flow shop	WIP; TTM; TR	Type of systems (current and POLCA);	POLCA has better performance than MRP as a current system in throughput time, throughput rate and WIP level.	T-test, 95% confidence interval
Huang *et al.* (2015)	CONWIP; m-CONWIP	General flow shop (assembly stations)	WIP; TTM; TR	8 loop design; Assembly proportion (Levels of BOM); Assembly batch	M-CONWIP has better performance than a single-loop CONWIP. To reduce deadlock phenomenon in assembly, dispatching rule is modified to consider the other assembling parts	Graphic analysis, descriptive statistics

Authors/Year	PCS	Shop Configuration	Performance compared	Experimental factors	Highlighted conclusion for PCS	Analysis Technique
Khojasteh and Sato (2015)	Kanban; CONWIP; Base-Stock	General flow shop	TR; WIP	PCS factors (Kanban, CONWIP, Base-Stock); Processing time mean level	CONWIP similar results Kanban for the multiple-stage loops tested, although this outperforming is considered as a variable for the analytical model.	Confidence interval at 95% level (comparison)
Onyeocha (2015)	General Kanban CS; Extended Kanban CS; Basestock Kanban CONWIP	General flow shop	WIP; SL	Production control systems (GKanban System, Extended Kanban system, Based Kanban-CONWIP); Allocation Policy (Dedicated, Shared); Scenarios (1 for 9 stages, 2 for 8 stages)	BK-CONWIP with S-KAP outperforms other systems. A pull system has poor performance in a high mix variation product. Product mix influences inventory level, which results in low service level and high WIP.	Nelson's Ranking and selection procedure (based on survival test)
Onyeocha, Khoury and Geraghty (2015)	HK-CONWIP; BK-CONWIP	General flow shop	WIP; SL (Service level)	Arrival time mean; Arrival time standard deviation; MTBF mean; MTTR mean; Changeover time mean; Changeover time standard deviation; Demand table for the week	BK-CONWIP outperforms HK-CONWIP. The S-KAP policy is better for simple and complex systems. These conclusions consider significant setup times as failure and repair time on stations.	Pareto frontier curvature analyses (using a GA optimization in tangent point of curvature); Stochastic dominance test under Latin Hypercube Sampling;
Braglia, Castellano and Frosolini, (2014)	POLCA, m-CONWIP	General flow shop	DTM; TTM;	PCS compared (POLCA, m-CONWIP); Nested factor: card number from GA	The results show that the POLCA outcomes are generally better than those of the m-CONWIP model in all cases (with the Longest queue dispatching rule). Shorter loops could explain that capability in POLCA.	Fitness function comparison, descriptive statistics from ranking
Müller, Tolujew and Kienzle (2014)	Push-Kanban; MRP II	General flow shop	TTM; WIP	3 cases (empirical) as a block; 2 years as a block; PCS selection	Push-Kanban has better performance than MRP II for some WIP limit levels used in the shop.	Graphical comparison
Thürer, Land and Stevenson (2014)	COBACABANA (classical, corrected)	General job shop	TTM; TDP; TDM	Workload estimation (classical and corrected); 9 levels of workload norm (4-12 time units); 6 cards size (1-5, exactly)	The corrected approach for WL estimation is recommended to adopt a standard norm for all stations. With 3 sizes of cards, processing time represented is accurate enough.	Graphical comparison

Authors/Year	PCS	Shop Configuration	Performance compared	Experimental factors	Highlighted conclusion for PCS	Analysis Technique
Aziz *et al*. (2013)	POLCA (unit-based); Push system (actual PR)	General flow shop	WIP; WQM; TC	PCS used (Actual PR, with POLCA, without POLCA)	POLCA presents better performance on profits than the actual PR by controlling WIP in shifting queues on bottleneck stations.	Graphical comparison of averages
Onyeocha, Khoury and Geraghty (2013)	General Kanban CS; Extended Kanban CS; Base-stock Kanban CONWIP	General flow shop	WIP; SL; BL	Production control systems (GKanban System, Extended Kanban system, Based Kanban-CONWIP) and Allocation Policies (Dedicated, Shared) giving 5 models; Demand variability (read from the table scenarios as a blocking factor)	S-KAP BK CONWIP outperforms Kanban system modifications in WIP and Service Level. The system is able to respond to the demand variability included quickly.	Nelson's Ranking and selection procedure (based on survival test)
Rezaei et al. (2011)	LB-DBR; SA-CR WLC	General job shop	DTM; DR; CTM	PPC (as Starvation Avoidance, Critical Ratio and WLC+DBR combined)	Integration of TOC and WLC (considered as LB-DBR) outperforms the individual release system. PERT is suggested for estimating the cycle time of each order, even in a dynamic workload.	Duncan's multiple range tests
Bokhorst and Slomp (2010)	CONWIP; CONWIP+FIFO; CONWIP+FIFO+TAKT	General job shop	DTM; UT; Status level percentage	Control mode (CONWIP; CONWIP + FIFO; CONWIP+FIFO+TAKT); Processing time variability SCV (0.125:1);	CONWIP/FIFO/TAKT has better performance than the other mode controls. This mode reduces WIP to reduce the delivery time mean, considering a stable 100% utilization in the system.	Confidence interval at the 95% level, validation with one scenario as implementation
Germs and Riezebos (2010)	POLCA; CONWIP; m-CONWIP	General job shop	DTM; TTM	Order Arrival (from Utilization); Batch size arrival; Processing time; PCS (selection)	POLCA and m-CONWIP have similar performance for delivery time and throughput time. CONWIP has worse performance because it cannot balance the workload in the shop.	Paired t-test, 95% significance level
Salum and Araz (2009)	DCR HPP, DCR-Kanban	General job shop	TTM, UT, TDM, WIP, TDP, MS	Priority dispatching rules; job shop layout; PCS	DCR Hybrid Push-Pull (HPP) is more suitable for a general job shop than DCR-Kanban. This comparison depends on labour or the machine constrains effect.	Benchmarking comparison
Qi, Sivakumar and Gershwin (2008)	WIPLCtrl; CONWIP; SR	General flow shop	CTM; CTD; LAM	ANOVA with paired t-test	WIP Load Control outperforms Shift release and CONWIP; Job release control can improve performance measures.	ANOVA with paired t-test

Authors/Year	PCS	Shop Configuration	Performance compared	Experimental factors	Highlighted conclusion for PCS	Analysis Technique
Ip *et al*. (2007)	Single-loop CONWIP; Multiple-loop CONWIP	General flow shop	TC	Direct comparison	Single CONWIP has better performance than multiloop using the cellular structure in the shop, for assembly parts	Direct comparison
Baysan, Kabadurmus and Durmusoglu (2007)	POLCA; push	General job shop	TC; DTM; WIP	Direct comparison	POLCA outperforms the push system, considering the economic evaluation.	Direct comparison
Fernandes and Carmo-Silva (2006)	G-POLCA; POLCA; MRP*	General flow shop	TR; TTM; WIP	Interval confidence based on t-Student	General POLCA outperforms POLCA and pushed production for throughput time and throughput rate. This result is robust for high product mix	Interval confidence based on t-Student
Lödding, Ya and Wiendahl (2003)	DEWIP; Load Oriented Order Release (LOOR); CONWIP; POLCA	Pure Job Shop	WIP; TTM; LAM	Confidence interval at the 95% level in a simulation logistic operating curve	DEWIP can be used to establish each WIP level individual for the station. LOOR was the best performing PCS in comparison by logistics curves. Other PCSs outperformed POLCA when the model is pure job shop.	Confidence interval at the 95% level in a simulation logistic operating curve
Gilland (2002)	DBR; CONWIP	General flow shop	WIP; TR	Confidence interval at 95% level DBR and CONWIP using Welch's confidence interval (same level)	DBR outperforms CONWIP when the bottleneck effect is significant. The difference between the systems diminishes when non-bottleneck stations are near the bottleneck utilization level (the bottleneck effect is not significant).	Confidence interval at 95% level DBR and CONWIP using Welch's confidence interval (same level)
Golany, Dar-El and Zeev (1999)	Single-loop CONWIP; Multiple-loop CONWIP	General Flow Shop	TTM; WIP	Gap deviation and graphical comparison	The single-loop CONWIP system enables better throughput than the multiloop CONWIP system. The m-CONWIP reduces the effect of delay in the bottleneck, but it does not improve general performance.	Gap deviation and graphical comparison

Source: Authors

APPENDIX C - SIMULATION MODEL

This appendix presents the main parts of the code of the simulation model. This model is in FlexScript, using FlexSim 18.2, academic version. First, Figure C.1. shows the event of how the simulation collects data to estimate the performance measures, according events. Figure C.2. show the items on FlexSim that represent the objects.

Figure C.1 – Simulation model conceptualization

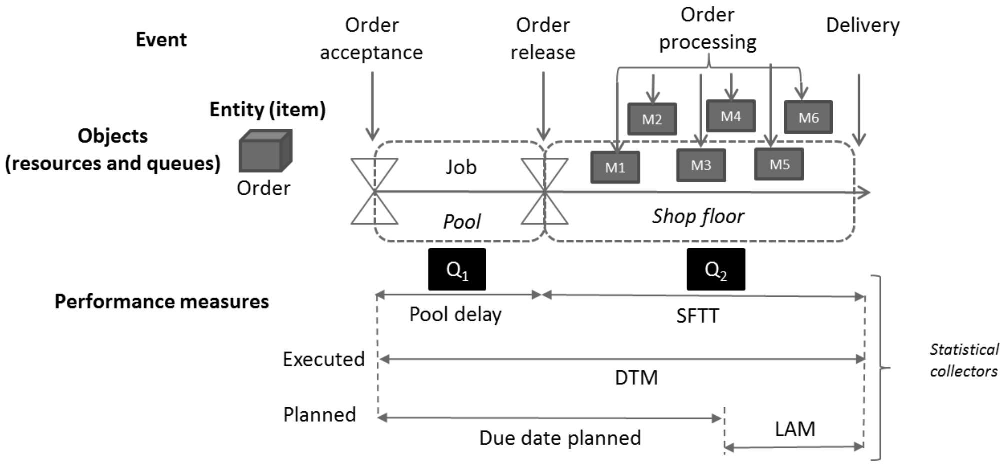


Figure C.2 – Simulation model view with objects and links

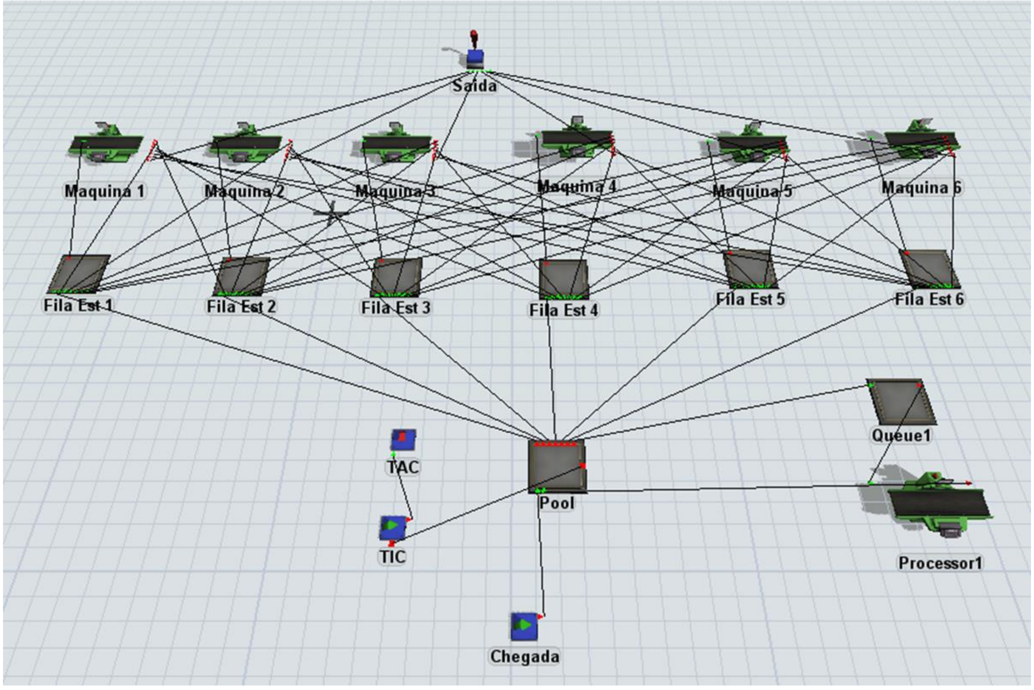


Figure C.3 shows the main flow of information decided for the simulation model for the POLCA systems. This flow includes some differences for each POLCA system, but those functionalities were deactivated by numbers of scenarios.

Figure C.3 – Process flow description for POLCAs system

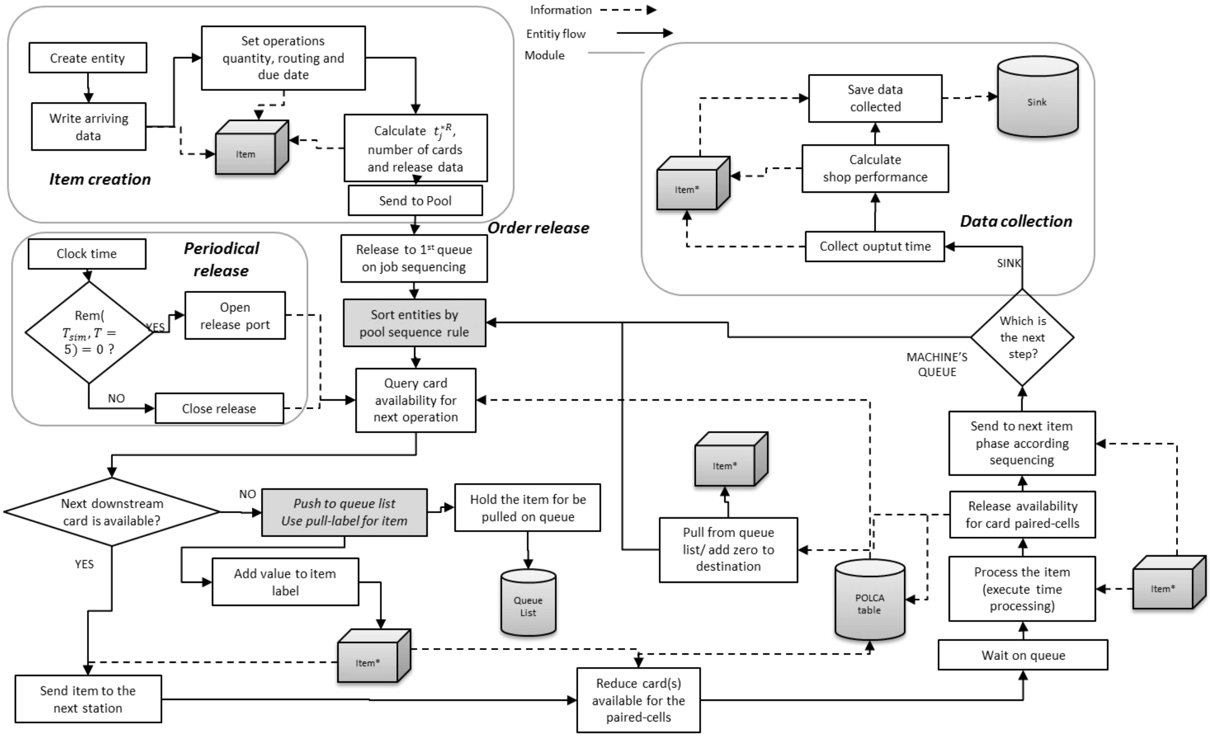
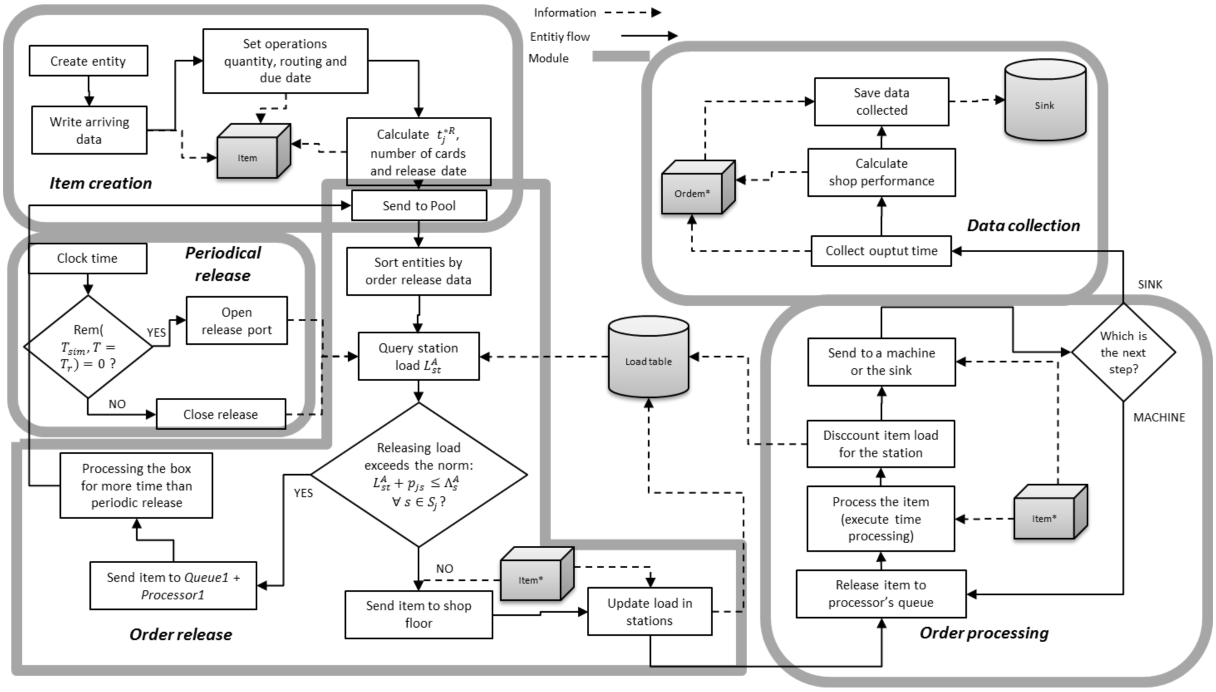


Figure C.4. shows the main flow process for the COBACABANA order release system. This logical process was used for COBACABANA and the model-based in another implementation of the Workload Control. For COBA-POLCA, both logical procedures were used embedded in the same model.

Figure C.4 – Process flow description for COBACABANA order release



The following lines are the FlexScript for the code. For this section, the number lines are omitted.

Source (Chegada)

Process triggering: On creation

```
Object item = param(1);
Object current = ownerobject(c);
int port = param(2);
int rownumber = param(2);  //row number of the schedule/sequence table
{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"numeroOperacoes"/**/;
Variant value = /** \nValue: *//***tag:value*//**/duniform(2,6)/**/; //De 2 a 6
operaÃ§Ãµes
involved.labels.assert(labelname).value = value;
item.labels["st-numop"].value = value;
} // ******* PickOption End ******* //

setlabel(item,"tempoChegada",time()); //adiciona uma label de tempo de chegada (dia
de chegada)

setlabel(item,"DueDate-box",duniform(35,60));//Calculando item's due date
setlabel(item,"diaEntrega",item.labels["DueDate-box"].value+time()); // adiciona
uma label com a data de entrega prevista
int OPS = getlabel(item,"numeroOperacoes"); // Adiciona o nÃºmero de operaÃ§Ãµes em
OPS

intarray OPED = makearray(6); //Cria um vetor OPED que irÃ¡ receber os valores de
OP1 a OP6
```

```
OPED.fill(7);

//Array for Operation Due Date ODD
intarray ODuDate = makearray(6); //Array for operation due date interval
intarray HDuDate = makearray(6); //Array for operation due date hour
intarray AthDate = makearray(6); //Array for Authorization date for station
double TTauth = Table("TableParameters")[5][1];
ODuDate.fill(getlabel(item,"DueDate-box")); //Filling with the due date
HDuDate.fill(getlabel(item,"DueDate-box")); //Filling with the due date
AthDate.fill(getlabel(item,"DueDate-box")-TTauth);//Filling with the due date minus
TTauth
for (int i = 1; i <= item.numeroOperacoes; i++)
{
ODuDate[i] = getlabel(item,"DueDate-box") - 1.1*(getlabel(item,"numeroOperacoes") -
i);
//DueDate - 1.1 * (num operations... last case is due date)
HDuDate[i] = getlabel(item,"tempoChegada") + ODuDate[i];
// Clock time + Due date for operation
switch (Table("TableParameters")[7][1]) //Auth date type
{
case 1:
AthDate[i] = HDuDate[i] - TTauth; //Auth by station
break;
case 2:
AthDate[i] = HDuDate[1] - TTauth; //Auth on First station for others
break;
default:
AthDate[i] = 0; //Auth for all station in time 0
break;
}//endswitch
} //endfor
item.OperDueDate = ODuDate; //Transfer for item
item.HODueDate = HDuDate; //Transfer for item
item.AuthDate = AthDate; //Transfer for item

//Comando OP1:

{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"OP1"/**/;
Variant value = /** \nValue: *//***tag:value*//**/duniform(1,6)/**/;

involved.labels.assert(labelname).value = value;
} // ******* PickOption End ******* //

OPS = (OPS -1); // Decrementa o nÃºmero de operaÃ§Ãµes

setlabel(item,"TP1",Math.min(erlang(0.0, 0.5, 2.0, 0),4)); //cria a tempo de
processamento para OP1

//Comando OP2:
if (OPS >0){

{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"OP2"/**/;
Variant value = /** \nValue: *//***tag:value*//**/duniform(1,6)/**/;

involved.labels.assert(labelname).value = value;
} // ******* PickOption End ******* //

int status = 0;
```

```
while(status==0){
if (getlabel(item,"OP2")==getlabel(item,"OP1")){
setlabel(item,"OP2",duniform(1,6));
}
else{
status = 1;
}
}//end while

OPS = (OPS -1); // Decrementa o nÃ°mero de operaÃ§Ãµes

setlabel(item,"TP2",Math.min(erlang(0.0, 0.5, 2.0, 0),4)); //cria a tempo de
processamento para OP2

}//end comando 2

if (OPS >0){ //Comando OP3

{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"OP3"/**/;
Variant value = /** \nValue: *//***tag:value*//**/duniform(1,6)/**/;

involved.labels.assert(labelname).value = value;
} // ******* PickOption End ******* //

int status = 0;

while(status==0){
if (getlabel(item,"OP3")==getlabel(item,"OP1") ||
(getlabel(item,"OP3")==getlabel(item,"OP2"))){
setlabel(item,"OP3",duniform(1,6));
}
else{
status = 1;
}
}//end while

OPS = (OPS -1); // Decrementa o nÃ°mero de operaÃ§Ãµes

setlabel(item,"TP3",Math.min(erlang(0.0, 0.5, 2.0, 0),4)); //cria a tempo de
processamento para OP3

}//end comando 3.


if (OPS > 0){ //Comando OP4

{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"OP4"/**/;
Variant value = /** \nValue: *//***tag:value*//**/duniform(1,6)/**/;

involved.labels.assert(labelname).value = value;
} // ******* PickOption End ******* //

int status = 0;

while(status==0){
```

```
if (getlabel(item,"OP4")==getlabel(item,"OP1") ||
(getlabel(item,"OP4")==getlabel(item,"OP2"))||(getlabel(item,"OP4")==getlabel(item,
"OP3"))){
setlabel(item,"OP4",duniform(1,6));
}
else{
status = 1;
}
}//end while

OPS = (OPS -1); // Decrementa o nÃºmero de operaÃ§Ãµes

setlabel(item,"TP4",Math.min(erlang(0.0, 0.5, 2.0, 0),4)); //cria a tempo de
processamento para OP4

} // end comando 4.

if (OPS > 0){//Comando OP5

{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"OP5"/**/;
Variant value = /** \nValue: *//***tag:value*//**/duniform(1,6)/**/;

involved.labels.assert(labelname).value = value;
} // ******* PickOption End ******* //

int status = 0;

while(status==0){
if (getlabel(item,"OP5")==getlabel(item,"OP1") ||
(getlabel(item,"OP5")==getlabel(item,"OP2"))||(getlabel(item,"OP5")==getlabel(item,
"OP3"))||(getlabel(item,"OP5")==getlabel(item,"OP4"))){
setlabel(item,"OP5",duniform(1,6));
}
else{
status = 1;
}
}//end while

OPS = (OPS -1); // Decrementa o nÃºmero de operaÃ§Ãµes

setlabel(item,"TP5",Math.min(erlang(0.0, 0.5, 2.0, 0),4)); //cria a tempo de
processamento para OP5

} //end comando 5.


if (OPS > 0){ //Comando OP6

{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"OP6"/**/;
Variant value = /** \nValue: *//***tag:value*//**/6/**/;

involved.labels.assert(labelname).value = value;
} // ******* PickOption End ******* //


int status = 0;

while(status==0){
```

```
if (getlabel(item,"OP6")==getlabel(item,"OP1") ||
(getlabel(item,"OP6")==getlabel(item,"OP2"))||(getlabel(item,"OP6")==getlabel(item,
"OP3"))||(getlabel(item,"OP6")==getlabel(item,"OP4"))||(getlabel(item,"OP6")==getla
bel(item,"OP4"))||(getlabel(item,"OP6")==getlabel(item,"OP5"))){
setlabel(item,"OP6",duniform(1,6));
}
else{
status = 1;
}
}//end while

OPS = (OPS -1); // Decrementa o nÃºmero de operaÃ§Ãµes


setlabel(item,"TP6",Math.min(erlang(0.0, 0.5, 2.0, 0),4)); //cria a tempo de
processamento para OP6

}//end comando 6.

//cria uma label de data de entrega prevista baseada no nÃºmero de operaÃ§Ãµes
multiplicado pela soma do dia de chegada com a distribuiÃ§Ã£o do tempo de
processamento
//setlabel(item,"tempoAtravessamento",((getlabel(item,"numeroOperacoes")*(erlang(0.
0, 0.5, 2.0, 0)))+(getlabel(item,"tempoChegada"))));

intarray TP = makearray(6);

TP[1] = getlabel(item,"TP1");
TP[2] = getlabel(item,"TP2");
TP[3] = getlabel(item,"TP3");
TP[4] = getlabel(item,"TP4");
TP[5] = getlabel(item,"TP5");
TP[6] = getlabel(item,"TP6");
item.TP = TP; //transferindo o vetor local para o item

OPED[1] = item.OP1; //transferindo labels para array OPED
OPED[2] = item.OP2;
OPED[3] = item.OP3;
OPED[4] = item.OP4;
OPED[5] = item.OP5;
OPED[6] = item.OP6;

for (int m = 1; m <= 6; m++)
{
if (OPED[m] == 0)
{
OPED[m] = 7; //7th value for saida port
}
}

//sorting OPED for GFS
int temp = 0;
for (int i = 1; i <= (6-1); i++)
{
for (int j = 1; j <= (6-1); j++)
{
if (OPED[j] > OPED[j+1])
{
temp = OPED[j+1];
OPED[j+1] = OPED[j];
OPED[j] = temp;
} //end if
} //endfor j
}// endfor i
/*for (int i = 1; i <= 6; i++)
{ //sorting the 0 values on OPED
```

```
if (OPED[i] == 0)
{ OPED.shift();
OPED.push(0);
} //endif
}//endfor
//com 1 estaÃ§Ã£o o vetor chega zerado */

for (int m = 1; m <= 6; m++)
{
if (OPED[m] == 7)
{
OPED[m] = 0; //returning the 0 value for operations without machine
}
}

item.opeseq1 = OPED;

// tempoChegada = soma prevista dos TP
setlabel(item,"tempoAtravessamento",((TP[1]+TP[2]+TP[3]+TP[4]+TP[5]+TP[6])+(getlabe
l(item,"tempoChegada"))));

//cria lebal data de liberaÃ§Ã£o planejada
setlabel(item,"DlPlanejada",(getlabel(item,"diaEntrega")-
getlabel(item,"tempoAtravessamento")));
double TPcum; //cumulando os tempos - para contribuiÃ§Ã£o relativa
int cardnum = Table("TableParameters")[9][1]; //retrieve the number of cards
int TPCardu = 0; //Local of each TPcard.
intarray CumTP = makearray(6);
intarray CardTPa = makearray(6);
intarray TPcor = makearray(6);
intarray CardTPcor = makearray(6);
int lcorrection = Table("TableParameters")[2][1]; //read correction type
for (int w = 1; w <= 6; w++)
{
TPcum = TPcum + TP[w];
CumTP[w]= TPcum;
if (lcorrection == 2)
{TPcor [w] = TP[w]/w; //correction for COBACABANA
}
else
{
TPcor [w] = TP[w]; //uncorrection for COBACABANA
}
if (cardnum != 0) //different than 0 is using cards
{//using ceil for round up the number of assigned cards
CardTPa[w] = Math.ceil(TP[w]/ (4/cardnum)); //estimate cards normal
CardTPcor[w] = Math.ceil(TPcor[w]/ (4/cardnum)); // estimate corrected cards COBA
}
else
{
CardTPcor[w] = TPcor[w]; //not using cards
}
}
item.Cumulate2TP = CumTP;
item.TPcor = TPcor;
item.CardTPCor = CardTPcor; //cor is for local array and TPCor is for item
if (Table("TableParameters")[9][1] != 0) //using or not cards
{
item.CardTP =  CardTPa; //transfering cards as integer number on item (cards are
used)
} else
{
item.CardTP =  TP; //transfering cards as integer number on item (cards are not
used)
}
```

```
//Comando que verifica se OPS Ã© zero, caso positivo finaliza
// Espera-se que caso OPS seja zero em algum valor antes de 6, todas as operaÃ§Ãµes
nÃ£o preenchidas recebam zero.
if (OPS == 0){
return 0;


}{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"tempoChegada"/**/;
Variant value = /** \nValue: *//***tag:value*//**/time()/**/;

involved.labels.assert(labelname).value = value;
} // ******* PickOption End ******* //
```

Releaser processor

Output: Send to port (custom code)

```
Object item = param(1);
Object current = ownerobject(c);

intarray carga = makearray(6);
intarray OP = makearray(6);
intarray TP = makearray(6);
intarray CardWLcurr = makearray(6);

int i;
int k;
int w=1; //var w para carga
int q; //var for verification > CardWLcurr
int boolean = 0;
int lcorrection = Table("TableParameters")[2][1];
double auxi = 0; //var for decimal load
int norma = gettablenum("Cargas",1,7);
int Cardnorm = gettablenum("Cargas",2,7);
double ilpolcnorm = Table("TableParameters")[1][1];
int v1;

carga[0] =0; // SÃ³ para nÃ£o dar erro quando a OP for 0
carga[1] = gettablenum("Cargas",1,1);
carga[2] = gettablenum("Cargas",1,2);
carga[3] = gettablenum("Cargas",1,3);
carga[4] = gettablenum("Cargas",1,4);
carga[5] = gettablenum("Cargas",1,5);
carga[6] = gettablenum("Cargas",1,6);

OP = item.opeseq1; //transferindo do item
/*
OP[1] = getlabel(item,"OP1");
OP[2] = getlabel(item,"OP2");
OP[3] = getlabel(item,"OP3");
OP[4] = getlabel(item,"OP4");
OP[5] = getlabel(item,"OP5");
OP[6] = getlabel(item,"OP6");
*/
TP = item.TP;
/*
TP[1] = getlabel(item,"TP1");
TP[2] = getlabel(item,"TP2");
TP[3] = getlabel(item,"TP3");
TP[4] = getlabel(item,"TP4");
```

```
TP[5] = getlabel(item,"TP5");
TP[6] = getlabel(item,"TP6");
*/
for(v1=1;v1<7;v1++)
{
CardWLcurr[v1] = gettablenum("Cargas",2,v1);
//reading cards COBA load
}

double dpolcnorm = Table("TableParameters")[6][1]; //limit for direct load

for (i=1;i<7;i++)
{
if (((carga[OP[i]])+TP[i]) > (norma*i) ||
Table("POLCATable")[item.opeseq1[1]][item.opeseq1[2]]+item.CardTP[item.opeseq1[2]]
> dpolcnorm )
{//Excede a norma estabelecida para estaÃ§Ã£o i
current.CargaCount = current.CargaCount + 1;
return 7; //Saida para Queue1+Processor1
msg("Excede Norma", "SerÃ¡ enviada a Processor pivote");
}
else { //else1
if (CardWLcurr[OP[i]] + item.CardTPCor[i]> (Cardnorm*i)) //more cards than allowed
{
current.CardNcount = current.CardNcount + 1;
current.PCardNc = current.CardNcount/current.stats.input.count;
return 7; //
}
else {
if ((item.labels["st-numop"].value>2 && i>2) && i<item.labels["st-numop"].value)
//more than 1 loop (or more than 2 stations) and less than std num of operations
{switch (lcorrection)
{case 1:
if ( Table("IndLoadPOLCA")[item.opeseq1[i-1]][item.opeseq1[i]] + TP[i] >
ilpolcnorm)
{ current.ILPolcacount = current.ILPolcacount + 1;
current.PILPolcaNc = current.ILPolcacount / current.stats.input.count;
return 7; } //IndirectLoad of POLCA - direct load
break;
case 2:
if ( Table("IndLoadPOLCA")[item.opeseq1[i-1]][item.opeseq1[i]] + 1/(i-2) >
ilpolcnorm)
{ return 7; } //IndirectLoad of POLCA - corrected estimated load
break;
default:
if ( Table("IndLoadPOLCA")[item.opeseq1[i-1]][item.opeseq1[i]] + TP[i] >
ilpolcnorm)
{ //msg("Saindo porta 7","Saindo Processor pelo default do else");
current.ILPolcacount = current.ILPolcacount + 1; //stop();
return 7; } //IndirectLoad of POLCA - direct load
break;
}//endswitch
}//endif3
else {
if ((i==6) || (OP[i+1]==0))//OP[i+1]==0 indica que hÃ¡ menos que 6 operaÃ§Ãµes na
ordem
{
for (w=1;w<=6;w++) //ex: numeroOperacoes = 6, i<7.
//for (w=1;w<=getlabel(item,"numeroOperacoes");w++) //ex: numeroOperacoes = 6, i<7.
{settablenum("Cargas",1,OP[w],(gettablenum("Cargas",1,OP[w]))+TP[w]);
//escreve na tabela cargas o valor correspondente de Q1
//Faz uma funÃ§Ã£o de incremento com o valor jÃ¡ existente em cargas
settablenum("Cargas",2,OP[w],(gettablenum("Cargas",2,OP[w]))+item.CardTPCor[w]);
//exceeding norm by using eval.
}//end for - update workload
for (k = 3; k<= item.labels["st-numop"].value ;k++)
```

```
{switch (lcorrection)
{case 1:
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]] = TP[k] +
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]];
break;
case 2:
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]] = 1/(k-1) +
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]];
break;
default:
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]] = TP[k] +
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]];
break;
}//endswitch
}//endfor Indirect Load POLCA
//msg("Sending port", "saindo do Pool");
return OP[1]; //1era porta do roteiro
}//end if for output port
}//end else2
}//end else
}//end else1
}//endfor ini
//
```

Input: Pull Strategy (modified code of Pull from port)

```
Object current = ownerobject(c);
/***popup:PullFromList:listType=Item*/
/***tag:description*//**Pull from ItemListPool*/
/***tag:showAdvanced*//**0*/

if (/** \nCondition: *//***tag:condition*//**/true/**/) {
int callbackCase = param(5);
#define CALLBACK_CASE 10293
treenode pulled = 0;
if (callbackCase != CALLBACK_CASE) {
string listName = /** \nList: *//***tag:listName*//**/"ItemListPool"/**/;
List list = List(listName);
string queryStr = /** \nQuery: *//***tag:query*//**/"WHERE Authorized > 0 AND
ModReleaseLength == 0  ORDER BY Urgentcondition,CS-col ASC"/**/;
switch (Table("TableParameters")[3][1]) {
case 2: //Using EODD
queryStr = model().find("Releaser").Release5rule;
break;
case 3: //Using MODCS for COBA
queryStr = model().find("Releaser").Release1rule;
break;
case 4: //using MODCS for POLCA
queryStr = model().find("Releaser").Release2rule;
break;
case 5: //using CS for POLCA
queryStr = model().find("Releaser").Release3rule;
break;
case 6: //using CS for COBA
queryStr = model().find("Releaser").Release4rule;
break;
default: //the text on the processor machine
queryStr = queryStr;
break;
}
Variant partitionId = /** \nPartition ID: *//***tag:partitionId*//**/0/**/;
int removeFromList = /** \nRemove From List: *//***tag:removeFromList*//**/1/**/;
int addToBackOrders = /** \nAdd To Back Orders:
*//***tag:addToBackOrders*//**/1/**/;
pulled = list.pull(queryStr, 1, addToBackOrders ? 1 : 0, current, partitionId,
(removeFromList ? 0 : LIST_DO_NOT_REMOVE) | LIST_RETURN_BACK_ORDER_IF_NOT_FULFILL);
```

```
if (isclasstype(pulled, "ListBackOrder"))
eventlisten(pulled, "OnFulfill", c, 0, LIST_ON_FULFILL_VALUE, 0, 0, 0,
CALLBACK_CASE, current);
else if (objectexists(pulled))
pullitem(current, pulled, 0, BYPASS_ALL);
} else {
pulled = param(1);
current = param(6);
pushitem(pulled, current, 0, 0);
}

}

return current.inObjects.length + 1;
```

Fila Est 1 (Queue processor for each machine)

Process triggering: On entry

```
/**Add etapa*/

Object item = param(1);
Object current = ownerobject(c);
int port = param(2);

setlabel(item,"Etapa",getlabel(item,"Etapa")+1);
// Na entrada da fila, soma o label "Etapa" do item em 1.

if (getlabel(item,"Etapa")==1)
{
doublearray TP = makearray(6);
intarray OP = makearray(6);
//for (int j=1;j++;j<7)
//{ OP[j]=0;}

TP[1] = getlabel(item,"TP1");
TP[2] = getlabel(item,"TP2");
TP[3] = getlabel(item,"TP3");
TP[4] = getlabel(item,"TP4");
TP[5] = getlabel(item,"TP5");
TP[6] = getlabel(item,"TP6");

OP[1] = getlabel(item,"OP1");
OP[2] = getlabel(item,"OP2");
OP[3] = getlabel(item,"OP3");
OP[4] = getlabel(item,"OP4");
OP[5] = getlabel(item,"OP5");
OP[6] = getlabel(item,"OP6");
}

//settablenum("Cargas",1,1,(gettablenum("Cargas",1,1)+TP[i]));
```

Machine 1 (Processor model)

Input: Pull strategy (Pull from ItemList1 custom code)

```
Object current = ownerobject(c);
/***popup:PullFromList:listType=Item*/
/***tag:description*//**Pull from ItemList1*/
/***tag:showAdvanced*//**1*/

if (/** \nCondition: *//***tag:condition*//**/true/**/) {
int callbackCase = param(5);
#define CALLBACK_CASE 10293
```

```
treenode pulled = 0;
if (callbackCase != CALLBACK_CASE) {
string listName = /** \nList: *//***tag:listName*//**/"ItemList1"/**/;
List list = List(listName);
string queryStr = /** \nQuery: *//***tag:query*//**/"WHERE Authorized > 0  AND
DPOLCAhab >= 0 ORDER BY DDate DESC"/**/;
switch (Table("TableParameters")[4][1]) {
case 1: //Using AgeInQueue as FIFO
queryStr = model().find("Processor1").Dispatching1;
break;
case 2: //using DDate as EODD
queryStr = model().find("Processor1").Dispatching2;
break;
default: //the text on the processor machine
queryStr = queryStr;
break;
}
Variant partitionId = /** \nPartition ID: *//***tag:partitionId*//**/0/**/;
int removeFromList = /** \nRemove From List: *//***tag:removeFromList*//**/1/**/;
int addToBackOrders = /** \nAdd To Back Orders:
*//***tag:addToBackOrders*//**/1/**/;
pulled = list.pull(queryStr, 1, addToBackOrders ? 1 : 0, current, partitionId,
(removeFromList ? 0 : LIST_DO_NOT_REMOVE) | LIST_RETURN_BACK_ORDER_IF_NOT_FULFILL);
if (isclasstype(pulled, "ListBackOrder"))
eventlisten(pulled, "OnFulfill", c, 0, LIST_ON_FULFILL_VALUE, 0, 0, 0,
CALLBACK_CASE);
else if (objectexists(pulled))
pullitem(current, pulled, 0, BYPASS_ALL);
} else {
pulled = param(1);
pushitem(pulled, current, 0, 0);
}

}

return current.inObjects.length + 1;
```

Output: Send to port (modified code)

```
Object item = param(1);
Object current = ownerobject(c);
/**First available*/

if (getlabel(item,"numeroOperacoes") == 0)
{
//AvaliaÃ§Ã£o item para Saida em Sink
return 7;
//Enviar na porta 7: O Sink
} else {
//Se caso hÃ¡ mais operaÃ§Ãµes
int enviar = getlabel(item,"proximaPorta");
//Ler a prÃ³xima porta
return enviar;
//Enviar na prÃ³xima porta
}
```

Triggering process: On entry

```
/**Custom Code*/
Object item = param(1);
Object current = ownerobject(c);
int port = param(2);

setlabel(item,"numeroOperacoes",getlabel(item,"numeroOperacoes")-1);
// Na entrada na estaÃ§Ã£o, diminui o label "numeroOperacoes" do item em 1.
```

```
if (item.opeseq1[item.Etapa+1] == 0 || item.numeroOperacoes == 0)
{
setlabel(item,"proximaPorta",7); //enviar a Saida
} else {
setlabel(item,"proximaPorta",item.opeseq1[item.Etapa+1]);
} //end ifelse

//direct load on POLCA Table
int w = getlabel(item,"Etapa"); //Last etapa will be equal to number of operations
int cardnum = Table("TableParameters")[9][1]; //retrieve the number of cards
if (w < item.labels["st-numop"].value)
{
if (cardnum != 0)
{
Table("POLCATable")[item.opeseq1[w]][item.opeseq1[w+1]] =
 Table("POLCATable")[item.opeseq1[w]][item.opeseq1[w+1]] + item.CardTP[w+1];
} else
{
Table("POLCATable")[item.opeseq1[w]][item.opeseq1[w+1]] =
 Table("POLCATable")[item.opeseq1[w]][item.opeseq1[w+1]] + item.TP[w+1];
}
}

int tempo = time();
if (tempo >= 0){ //condiciona o registro do histÃ³rico de carga ao tempo 3000.
Tempo de warm up


{ // ************* PickOption Start ************* //
/***popup:IncrementValue*/
/**Increment Value*/
treenode thenode = /** \nNode: */ /***tag:node*//**/current.labels["indice"]/**/;
double value = /** \nIncrement By: */ /***tag:value*//**/1/**/;
inc(thenode,value);
} // ******* PickOption End ******* //

//Coloca o valor da carga 1 na label da mÃ¡quina
setlabel(current,"Carga1",gettablenum("Cargas",1,1));

//carrega a global table de histÃ³rico
settablenum("Historico
Cargas",getlabel(current,"indice"),1,(gettablenum("Cargas",1,1)));
}
```

Triggering process: On exit

```
/**Subtract carga and update Indirect Load*/
Object item = param(1);
Object current = ownerobject(c);
int port = param(2);

intarray TP = makearray(6);
intarray CardTPCor = makearray(6);

TP = item.TP;
CardTPCor = item.CardTPCor;
int i = getlabel(item,"Etapa");// i Ã© atualizado com o valor da etapa

//Decrementa na tabela cargas o valor correspondente de TP1 que estÃ¡ saindo
settablenum("Cargas",1,1,(gettablenum("Cargas",1,1)-TP[i]));
settablenum("Cargas",2,1,(gettablenum("Cargas",2,1)-item.CardTPCor[i]));

if ((gettablenum("Cargas",1,1))<-0.01)
{ //validation for negative load
```

```
msg("CARGA NEGATIVA", "Verificar desconto da mÃ¡quina");
stop();
}
//Subtract the indirect load on exit.
int k;
int lcorrection = Table("TableParameters")[2][1];
for (k = item.Etapa + 2 ; k<= item.labels["st-numop"].value ;k++)
{if (k>=3)
{switch (lcorrection)
{case 1:
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]] =
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]] - TP[k];
break;
case 2:
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]] =
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]] - 1/(k-1);
break;
default:
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]] =
Table("IndLoadPOLCA")[item.opeseq1[k-1]][item.opeseq1[k]] - TP[k];
break;
}//endswitch
}//endif
}//endfor
//recarregar as cargas
int w; //adding the indirect load for the 2 next steps
for (w = item.Etapa + 3 ; w<= item.labels["st-numop"].value ;w++)
{switch (lcorrection)
{case 1:
Table("IndLoadPOLCA")[item.opeseq1[w-1]][item.opeseq1[w]] =
Table("IndLoadPOLCA")[item.opeseq1[w-1]][item.opeseq1[w]] + TP[w];
break;
case 2:
Table("IndLoadPOLCA")[item.opeseq1[w-1]][item.opeseq1[w]] =
Table("IndLoadPOLCA")[item.opeseq1[w-1]][item.opeseq1[w]] + 1/(w-1);
break;
default:
Table("IndLoadPOLCA")[item.opeseq1[w-1]][item.opeseq1[w]] =
Table("IndLoadPOLCA")[item.opeseq1[w-1]][item.opeseq1[w]] + TP[w];
break;
}//endswitch
        }//endfor
```

Sink (Saida)

Process triggering: On entry

```
Object item = param(1);
Object current = ownerobject(c);
int port = param(2);

{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/item/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"tempoSaidaTotal"/**/;
Variant value = /** \nValue: *//***tag:value*//**/time()/**/;

involved.labels.assert(labelname).value = value;
} // ******* PickOption End ******* //

settablenum("Cargas",1,8,(getlabel(item,"tempoSaidaTotal")-
(getlabel(item,"tempoChegada"))));
setlabel(item,"throughputTotal",gettablenum("Cargas",1,8));
{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
```

```
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/current/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"throughputTotal"/**/;
Variant value = /** \nValue: *//***tag:value*//**/gettablenum("Cargas",1,8)/**/;
involved.labels.assert(labelname).value = value;
current.labels["DTM-vs"].value = current.throughputTotal; //transfer label to
Tracked variable
} // ******* PickOption End ******* //
{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/current/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"throughputChao"/**/;
Variant value = /** \nValue: *//***tag:value*//**/gettablenum("Cargas",1,11)/**/;
involved.labels.assert(labelname).value = value;
current.labels["TTM-vs"].value = current.throughputChao; //transfer label to
Tracked variable
} // ******* PickOption End ******* //


settablenum("Cargas",1,11,(getlabel(item,"tempoSaidaTotal")-
(getlabel(item,"saidaPool"))));

//Lateness for LA = t_end - t_end-planned
setlabel(current,"Lateness-sink",getlabel(current,"throughputTotal")-
getlabel(item,"DueDate-box"));
//current.labels["Lateness-sink"].value = current.throughputTotal.value -
item.labels["DueDate-box"].value;
current.labels["Lateness-vs"].value = current.labels["Lateness-sink"].value;
//transfer label to Tracked Variable
current.sumLAM = current.sumLAM +  current.labels["Lateness-sink"].value; //using
for cumLAM
current.LAM2 = current.sumLAM / current.indice ;
double lax = current.labels["Lateness-vs"].value;
double slax = Math.pow(lax,2);
current.cumlax = current.cumlax + lax;
current.cumslax = current.cumslax + slax;
settablenum("Cargas",1,12,current.labels["Lateness-vs"].value);
//Estimating TDP and TDM data
if (current.labels["Lateness-sink"].value > 0 )
{ // Contagem para proporÃ§Ã£o
current.labels["TardinessP-sink"].value += 1;
//Transferindo valor
current.labels["Tardiness-sink"].value = current.labels["Lateness-sink"].value;
current.labels["Tardiness-vs"].value = current.labels["Tardiness-sink"].value;
//transfer label to Tracked Variable Tardiness
current.sumTDM = current.sumTDM + current.labels["Tardiness-sink"].value;
current.TDM2 = current.sumTDM / current.labels["TardinessP-sink"].value; //using
TDM2
}

settablenum("Cargas",1,13,current.labels["Tardiness-vs"].value);
int tempo = time();
if (tempo >= 0){ //condiciona o registro dos throughput total e chÃ£o de fÃ¡brica
ao tempo 3000.

{ // ************* PickOption Start ************* //
/***popup:IncrementValue*/
/**Increment Value*/
treenode thenode = /** \nNode: */ /***tag:node*//**/current.labels["indice"]/**/;
double value = /** \nIncrement By: */ /***tag:value*//**/1/**/;
inc(thenode,value);
current.labels["TDP-using"].value = current.labels["TardinessP-sink"].value/
current.indice;
} // ******* PickOption End ******* //
double n = current.indice;
```

```
current.Latedesvpad = Math.sqrt(((n*current.cumslax)-
Math.pow(current.cumlax,2))/(n*(n-1)));

//carrega a global table Throughput chÃ£o de fÃ¡brica
settablenum("Throughput",getlabel(current,"indice"),2,(getlabel(current,"throughput
Chao")));

//carrega a global table Throughput total
settablenum("Throughput",getlabel(current,"indice"),1,(getlabel(current,"throughput
Total")));
}{ // ************* PickOption Start ************* //
/***popup:SetLabel*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/current/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"DTM2"/**/;
Variant value = /** \nValue: *//***tag:value*//**/current.throughputTotal/**/;
current.sumDTM = value + current.sumDTM; //adding sum DTM
involved.labels.assert(labelname).value = current.sumDTM / current.indice;
} // ******* PickOption End ******* //
{ // ************* PickOption Start ************* //
/***popup:SetLabel*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/current/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"TTM2"/**/;
Variant value = /** \nValue: *//***tag:value*//**/current.throughputChao/**/;
current.sumTTM = value + current.sumTTM; //adding sum DTM
involved.labels.assert(labelname).value = current.sumTTM / current.indice;
} // ******* PickOption End ******* //
```

Process triggering: On reset

```
Object current = ownerobject(c);
{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/current/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"cumlax"/**/;
Variant value = /** \nValue: *//***tag:value*//**/0/**/;

involved.labels.assert(labelname).value = value;
} // ******* PickOption End ******* //
{ // ************* PickOption Start ************* //
/***popup:SetLabel:hasitem=0*/
/**Set Label*/
Object involved = /** \nObject: *//***tag:object*//**/current/**/;
string labelname = /** \nLabel: *//***tag:label*//**/"cumslax"/**/;
Variant value = /** \nValue: *//***tag:value*//**/0/**/;

involved.labels.assert(labelname).value = value;
} // ******* PickOption End ******* //
```

Table Parameters

Parameters table to change variables in the Experiment setting.
Each row has a specific parameter on the experiment

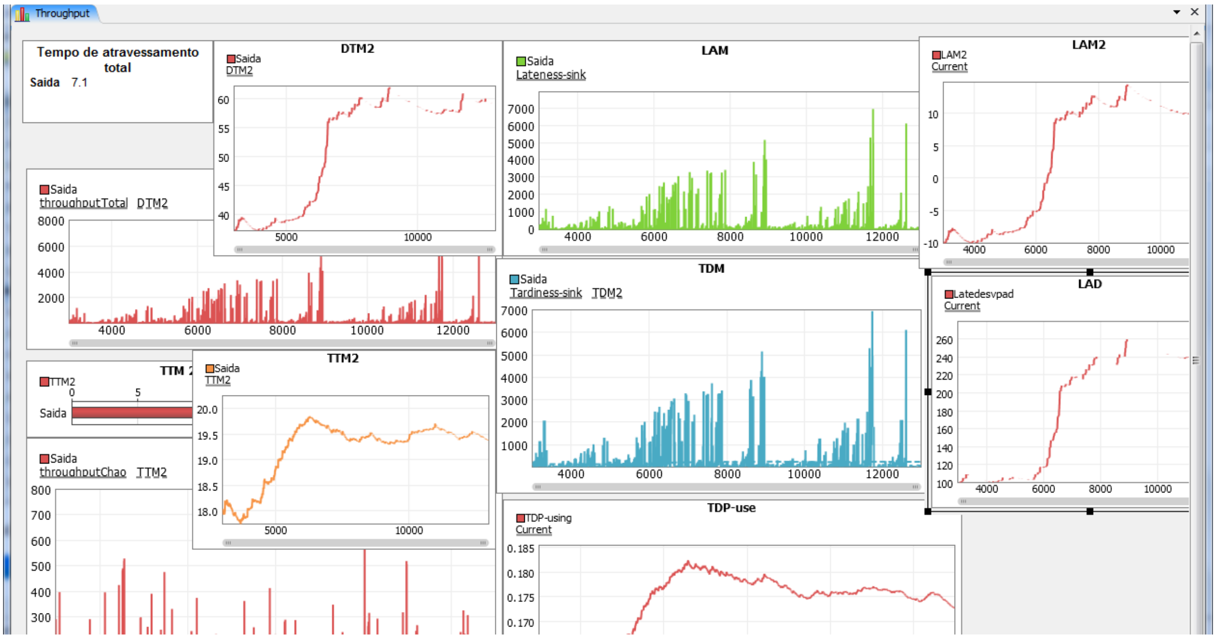
```
Col 1 Col 2
IL-POLCA norm250
Correction type10
Order Release Rule20
Dispatching rule20
ThroughputTime auth160
Polca Limit 120
Auth date type00
Release period length10
```

```
Number of cards 30
MTBF-value00
```

Validation and verification

For validation and verification of the model, several graphs were used to control the statistics collectors for the measured performance. As stated for the model, results were in the steady-state. Figure C.5 shows a sample for those graphs. For each graph, and each performance results, the push system in the model of Oosterman, Land, Gaalman (2000) was the reference value for validation. As those reference values are graphicals, the comparison was graphically to avoid errors in round those values.

Figure C.5 -Sample of graphs of steady-state simulation results



APPENDIX D - ANALYSIS OF VARIANCE OF CHAPTER 4

Table C.1 – Statistical analysis of the experimented factors (IL-POLC, Original POLCA, RF-POLCA)

Performance measure	**Experimental factor**	**Df**	**Sum of squares**	**Mean squares**	**F-ratio**	**P-value**
Delivery Time Mean (DTM)	Number of Cards (C)	3	1529394170,41	509798056,80	825,56	**0,00**
	Number of POLCA cards (P)	4	904377668,81	226094417,20	366,13	**0,00**
	Release Length (RL)	3	100271063,44	33423687,81	54,13	**0,00**
	C : P	8	850065480,71	106258185,09	172,07	**0,00**
	C : RL	6	121794359,74	20299059,96	32,87	**0,00**
	P : RL	12	266051533,14	22170961,09	35,90	**0,00**
	Indirect Load norm (I), Dispatching Rule (D), Order Release mech. (OR)	13	198257351,57	15250565,51	24,70	**0,00**
	C:P:RL	24	387974696,46	16165612,35	26,18	**0,00**
	I:D:OR:C	26	199397115,75	7669119,84	12,42	**0,00**
	I:D:OR:P	52	392174727,88	7541821,69	12,21	**0,00**
	I:D:OR:RL	39	101360216,70	2598979,92	4,21	**0,00**
	I:D:OR:C:P	104	679690195,84	6535482,65	10,58	**0,00**
	I:D:OR:C:RL	78	139938973,68	1794089,41	2,91	**0,00**
	I:D:OR:P:RL	156	254958509,71	1634349,42	2,65	**0,00**
	I:D:OR:C:P:RL	312	467088725,25	1497079,25	2,42	**0,00**
	Residuals	23160	14301777198,03	617520,60		

Cont. Table C.1 – Statistical analysis of the experimented factors (IL-POLC, Original POLCA, RF-POLCA)

Performance measure	Experimental factor	Df	Sum of squares	Mean squares	F-ratio	P-value
Shop Floor Throughput Time (SFTT)	Number of Cards (C)	3	76714690,00	25571563,33	1104,32	0,00
	Number of POLCA cards (P)	4	13435189,11	3358797,28	145,05	0,00
	Release Length (RL)	3	24226,06	8075,35	0,35	0,79
	C : P	8	5342048,51	667756,06	28,84	0,00
	C : RL	6	134096,72	22349,45	0,97	0,45
	P : RL	12	312476,32	26039,69	1,12	0,33
	Indirect Load norm (I):Dispatching Rule (D): Order Release mech. (OR)	13	111632116,73	8587085,90	370,84	0,00
	C:P:RL	24	379009,04	15792,04	0,68	0,87
	I:D:OR:C	26	65593382,36	2522822,40	108,95	0,00
	I:D:OR:P	52	80805132,99	1553944,87	67,11	0,00
	I:D:OR:RL	39	1238273,80	31750,61	1,37	0,06
	I:D:OR:C:P	104	41690910,22	400874,14	17,31	0,00
	I:D:OR:C:RL	78	2972949,43	38114,74	1,65	0,00
	I:D:OR:P:RL	156	4426104,16	28372,46	1,23	0,03
	I:D:OR:C:P:RL	312	10720069,59	34359,20	1,48	0,00
	Residuals	23160	536293514,61	23156,02		

Performance measure	Experimental factor	Df	Sum of squares	Mean squares	F-ratio	P-value
Lateness Deviation (LAD)	Number of Cards (C)	3	2100215359,83	700071786,61	68594,86	0,00
	Number of POLCA cards (P)	4	285213461,17	71303365,29	6986,49	0,00
	Release Length (RL)	3	11994145,30	3998048,43	391,74	0,00
	C : P	8	34892718,22	4361589,78	427,36	0,00
	C : RL	6	1251428,82	208571,47	20,44	0,00
	P : RL	12	3412921,77	284410,15	27,87	0,00
	Indirect Load norm (I):Dispatching Rule (D): Order Release mech. (OR)	13	435236702,48	33479746,34	3280,43	0,00
	C:P:RL	24	5092066,07	212169,42	20,79	0,00
	I:D:OR:C	26	43262350,27	1663936,55	163,04	0,00
	I:D:OR:P	52	71409378,52	1373257,28	134,56	0,00
	I:D:OR:RL	39	7484187,13	191902,23	18,80	0,00
	I:D:OR:C:P	104	22265728,52	214093,54	20,98	0,00
	I:D:OR:C:RL	78	1041240,13	13349,23	1,31	0,04
	I:D:OR:P:RL	156	2747202,21	17610,27	1,73	0,00
	I:D:OR:C:P:RL	312	6063238,28	19433,46	1,90	0,00
	Residuals	23160	236368459,89	10205,89		

Cont. Table C.1 – Statistical analysis of the experimented factors (IL-POLC, Original POLCA, RF-POLCA)

Performance measure	**Experimental factor**	**Df**	**Sum of squares**	**Mean squares**	**F-ratio**	**P-value**
Percentage of Tardy jobs (TDP)	Number of Cards (C)	3	1610,09	536,70	581874,55	0,00
	Number of POLCA cards (P)	4	51,95	12,99	14081,11	0,00
	Release Length (RL)	3	0,19	0,06	67,93	0,00
	C : P	8	4,82	0,60	653,53	0,00
	C : RL	6	0,08	0,01	14,88	0,00
	P : RL	12	0,26	0,02	23,06	0,00
	Indirect Load norm (I):Dispatching Rule (D): Order Release mech. (OR)	13	261,96	20,15	21846,76	0,00
	C:P:RL	24	0,04	0,00	1,93	0,00
	I:D:OR:C	26	7,50	0,29	312,56	0,00
	I:D:OR:P	52	28,41	0,55	592,23	0,00
	I:D:OR:RL	39	1,31	0,03	36,52	0,00
	I:D:OR:C:P	104	9,76	0,09	101,74	0,00
	I:D:OR:C:RL	78	0,05	0,00	0,67	0,99
	I:D:OR:P:RL	156	0,16	0,00	1,13	0,12
	I:D:OR:C:P:RL	312	0,07	0,00	0,26	1,00
	Residuals	23160	21,36	0,00		

Performance measure	**Experimental factor**	**Df**	**Sum of squares**	**Mean squares**	**F-ratio**	**P-value**
Tardiness Mean of a tardy job (TDM)	Number of Cards (C)	3	177913850,90	59304616,97	12976401,12	0,00
	Number of POLCA cards (P)	4	338976,81	84744,20	18542,82	0,00
	Release Length (RL)	3	108,03	36,01	7,88	0,00
	C : P	8	218286,03	27285,75	5970,38	0,00
	C : RL	6	209,98	35,00	7,66	0,00
	P : RL	12	358,38	29,87	6,53	0,00
	Indirect Load norm (I):Dispatching Rule (D): Order Release mech. (OR)	13	80280,57	6175,43	1351,24	0,00
	C:P:RL	24	1034,84	43,12	9,43	0,00
	I:D:OR:C	26	112455,74	4325,22	946,40	0,00
	I:D:OR:P	52	145320,61	2794,63	611,49	0,00
	I:D:OR:RL	39	637,23	16,34	3,58	0,00
	I:D:OR:C:P	104	219897,52	2114,40	462,65	0,00
	I:D:OR:C:RL	78	589,20	7,55	1,65	0,00
	I:D:OR:P:RL	156	1157,24	7,42	1,62	0,00
	I:D:OR:C:P:RL	312	2140,27	6,86	1,50	0,00
	Residuals	23160	105845,60	4,57		

APPENDIX E - ANALYSIS OF VARIANCE TABLES FOR CHAPTER 5

Table D.1 - Statistical analysis of the experimented factors (IL-POLC, COBACABANA and COBA-POLCA)

Performance measure	**Experimental factor**	**Df**	**Sum of squares**	**Mean squares**	**F-ratio**	**P-value**
Delivery Time Mean (DTM)	PCS (P)	4	9592873.69	2398218.42	47715.16	0.00
	Order Release Rule (OR)	2	1619.24	809.62	16.11	0.00
	Cell availability (CA)	1	6674.28	6674.28	132.79	0.00
	Quantum of card (Q)	1	29923.75	29923.75	595.37	0.00
	Limit (L)[1]	3	111956.38	37318.79	742.50	0.00
	P:OR	6	1678.09	279.68	5.56	0.00
	P:CA	3	184.17	61.39	1.22	0.30
	OR:CA	2	39.35	19.67	0.39	0.68
	P:Q	3	55622.91	18540.97	368.89	0.00
	OR:Q	2	281.02	140.51	2.80	0.06
	CA:	1	18.17	18.17	0.36	0.55
	P:L	6	91431.42	15238.57	303.19	0.00
	OR:L	6	2188.05	364.67	7.26	0.00
	CA:L	3	183.99	61.33	1.22	0.30
	Q:L	3	28869.02	9623.01	191.46	0.00
	P:OR:CA	6	48.83	8.14	0.16	0.99
	P:OR:Q	6	892.43	148.74	2.96	0.01
	P:CA:Q	3	11.22	3.74	0.07	0.97
	OR:CA:Q	2	18.14	9.07	0.18	0.83
	P:OR:L	12	1965.47	163.79	3.26	0.00
	P:CA:L	6	188.01	31.34	0.62	0.71
	OR:CA:L	6	78.15	13.03	0.26	0.96
	P:Q:L	6	44235.96	7372.66	146.69	0.00
	OR:Q:L	6	201.66	33.61	0.67	0.68
	CA:Q:L	3	2.47	0.82	0.02	1.00
	P:OR:CA:Q	6	61.52	10.25	0.20	0.98
	P:OR:CA:L	12	145.48	12.12	0.24	1.00
	P:OR:Q:L	12	324.64	27.05	0.54	0.89
	P:CA:Q:L	6	47.50	7.92	0.16	0.99
	OR:CA:Q:L	6	75.43	12.57	0.25	0.96
	P:OR:CA:Q:L	12	151.55	12.63	0.25	1.00
	Residuals	5244	263569.40	50.26		

1 Limit is the combination presented of number of cards allowed for loop and Workload norm.

Cont. Table D.1 - Statistical analysis of the experimented factors (IL-POLC, COBACABANA and COBA-POLCA)

Performance measure	**Experimental factor**	**Df**	**Sum of squares**	**Mean squares**	**F-ratio**	**P-value**
Shop Floor Throughput Time (SFTT)	PCS (P)	4	4625097.79	1156274.45	171129.01	0.00
	Order Release Rule (OR)	2	219.59	109.79	16.25	0.00
	Cell availability (CA)	1	2084.00	2084.00	308.43	0.00
	Quantum of card (Q)	1	427.41	427.41	63.26	0.00
	Limit (L)	3	9061.17	3020.39	447.02	0.00
	P:OR	6	134.94	22.49	3.33	0.00
	P:CA	3	157.48	52.49	7.77	0.00
	OR:CA	2	0.34	0.17	0.03	0.98
	P:Q	3	2110.30	703.43	104.11	0.00
	OR:Q	2	0.56	0.28	0.04	0.96
	CA:	1	0.27	0.27	0.04	0.84
	P:L	6	26520.37	4420.06	654.17	0.00
	OR:L	6	33.25	5.54	0.82	0.55
	CA:L	3	146.70	48.90	7.24	0.00
	Q:L	3	25.30	8.43	1.25	0.29
	P:OR:CA	6	5.05	0.84	0.12	0.99
	P:OR:Q	6	1.22	0.20	0.03	1.00
	P:CA:Q	3	2.27	0.76	0.11	0.95
	OR:CA:Q	2	0.22	0.11	0.02	0.98
	P:OR:L	12	77.44	6.45	0.96	0.49
	P:CA:L	6	370.47	61.75	9.14	0.00
	OR:CA:L	6	1.28	0.21	0.03	1.00
	P:Q:L	6	129.64	21.61	3.20	0.00
	OR:Q:L	6	0.20	0.03	0.01	1.00
	CA:Q:L	3	6.59	2.20	0.33	0.81
	P:OR:CA:Q	6	0.20	0.03	0.00	1.00
	P:OR:CA:L	12	2.93	0.24	0.04	1.00
	P:OR:Q:L	12	1.13	0.09	0.01	1.00
	P:CA:Q:L	6	9.68	1.61	0.24	0.96
	OR:CA:Q:L	6	0.99	0.17	0.02	1.00
	P:OR:CA:Q:L	12	0.60	0.05	0.01	1.00
	Residuals	5244	35432.35	6.76		

Cont. Table D.1 - Statistical analysis of the experimented factors (IL-POLC, COBACABANA and COBA-POLCA)

Performance measure	Experimental factor	Df	Sum of squares	Mean squares	F-ratio	P-value
Standard Deviation of Lateness (LAD)	PCS (P)	4	6552798.75	1638199.69	3272.34	0.00
	Order Release Rule (OR)	2	181202.84	90601.42	180.98	0.00
	Cell availability (CA)	1	28884.26	28884.26	57.70	0.00
	Quantum of card (Q)	1	121661.02	121661.02	243.02	0.00
	Limit (L)	3	355115.72	118371.91	236.45	0.00
	P:OR	6	165496.68	27582.78	55.10	0.00
	P:CA	3	13174.25	4391.42	8.77	0.00
	OR:CA	2	1943.25	971.63	1.94	0.14
	P:Q	3	260339.53	86779.84	173.34	0.00
	OR:Q	2	29615.98	14807.99	29.58	0.00
	CA:	1	196.08	196.08	0.39	0.53
	P:L	6	617572.64	102928.77	205.60	0.00
	OR:L	6	104807.29	17467.88	34.89	0.00
	CA:L	3	721.25	240.42	0.48	0.70
	Q:L	3	96822.34	32274.11	64.47	0.00
	P:OR:CA	6	1545.05	257.51	0.51	0.80
	P:OR:Q	6	72093.26	12015.54	24.00	0.00
	P:CA:Q	3	681.24	227.08	0.45	0.71
	OR:CA:Q	2	122.58	61.29	0.12	0.88
	P:OR:L	12	98230.06	8185.84	16.35	0.00
	P:CA:L	6	2747.80	457.97	0.91	0.48
	OR:CA:L	6	1404.79	234.13	0.47	0.83
	P:Q:L	6	160061.61	26676.94	53.29	0.00
	OR:Q:L	6	27924.72	4654.12	9.30	0.00
	CA:Q:L	3	141.23	47.08	0.09	0.96
	P:OR:CA:Q	6	1251.36	208.56	0.42	0.87
	P:OR:CA:L	12	1731.00	144.25	0.29	0.99
	P:OR:Q:L	12	41945.86	3495.49	6.98	0.00
	P:CA:Q:L	6	352.91	58.82	0.12	0.99
	OR:CA:Q:L	6	1078.16	179.69	0.36	0.91
	P:OR:CA:Q:L	12	2092.48	174.37	0.35	0.98
	Residuals	5244	2625254.39	500.62		

Cont. Table D.1 - Statistical analysis of the experimented factors (IL-POLC, COBACABANA and COBA-POLCA)

Performance measure	Experimental factor	Df	Sum of squares	Mean squares	F-ratio	P-value
Percentage Tardy (TDP)	PCS (P)	4	357.06	89.27	13077.35	0.00
	Order Release Rule (OR)	2	0.85	0.43	62.61	0.00
	Cell availability (CA)	1	1.61	1.61	236.11	0.00
	Quantum of card (Q)	1	2.79	2.79	408.73	0.00
	Limit (L)	3	12.69	4.23	619.85	0.00
	P:OR	6	0.60	0.10	14.59	0.00
	P:CA	3	0.06	0.02	2.79	0.04
	OR:CA	2	0.00	0.00	0.14	0.87
	P:Q	3	4.06	1.35	198.30	0.00
	OR:Q	2	0.05	0.02	3.61	0.03
	CA:	1	0.00	0.00	0.09	0.77
	P:L	6	4.29	0.71	104.65	0.00
	OR:L	6	0.29	0.05	7.00	0.00
	CA:L	3	0.03	0.01	1.57	0.19
	Q:L	3	2.06	0.69	100.83	0.00
	P:OR:CA	6	0.00	0.00	0.04	1.00
	P:OR:Q	6	0.06	0.01	1.53	0.16
	P:CA:Q	3	0.00	0.00	0.13	0.94
	OR:CA:Q	2	0.00	0.00	0.08	0.92
	P:OR:L	12	0.50	0.04	6.08	0.00
	P:CA:L	6	0.10	0.02	2.35	0.03
	OR:CA:L	6	0.00	0.00	0.06	1.00
	P:Q:L	6	2.79	0.47	68.14	0.00
	OR:Q:L	6	0.06	0.01	1.47	0.18
	CA:Q:L	3	0.00	0.00	0.01	1.00
	P:OR:CA:Q	6	0.00	0.00	0.01	1.00
	P:OR:CA:L	12	0.00	0.00	0.02	1.00
	P:OR:Q:L	12	0.09	0.01	1.05	0.40
	P:CA:Q:L	6	0.00	0.00	0.12	0.99
	OR:CA:Q:L	6	0.00	0.00	0.03	1.00
	P:OR:CA:Q:L	12	0.00	0.00	0.02	1.00
	Residuals	5244	35.80	0.01		

Cont. Table D.1 - Statistical analysis of the experimented factors (IL-POLC, COBACABANA and COBA-POLCA)

Performance measure	**Experimental factor**	**Df**	**Sum of squares**	**Mean squares**	**F-ratio**	**P-value**
Mean Tardiness (TDM)	PCS (P)	4	3260727.07	815181.77	7403.98	0.00
	Order Release Rule (OR)	2	87487.04	43743.52	397.31	0.00
	Cell availability (CA)	1	9603.19	9603.19	87.22	0.00
	Quantum of card (Q)	1	36618.12	36618.12	332.59	0.00
	Limit (L)	3	151870.66	50623.55	459.79	0.00
	P:OR	6	51134.72	8522.45	77.41	0.00
	P:CA	3	348.98	116.33	1.06	0.37
	OR:CA	2	188.01	94.01	0.85	0.43
	P:Q	3	70052.53	23350.84	212.09	0.00
	OR:Q	2	4683.31	2341.65	21.27	0.00
	CA:	1	20.49	20.49	0.19	0.67
	P:L	6	185016.63	30836.11	280.07	0.00
	OR:L	6	35987.84	5997.97	54.48	0.00
	CA:L	3	66.49	22.16	0.20	0.90
	Q:L	3	18359.77	6119.92	55.58	0.00
	P:OR:CA	6	224.05	37.34	0.34	0.92
	P:OR:Q	6	11474.73	1912.45	17.37	0.00
	P:CA:Q	3	6.79	2.26	0.02	1.00
	OR:CA:Q	2	59.42	29.71	0.27	0.76
	P:OR:L	12	34429.57	2869.13	26.06	0.00
	P:CA:L	6	75.70	12.62	0.11	0.99
	OR:CA:L	6	384.22	64.04	0.58	0.75
	P:Q:L	6	22262.04	3710.34	33.70	0.00
	OR:Q:L	6	2164.91	360.82	3.28	0.00
	CA:Q:L	3	7.10	2.37	0.02	1.00
	P:OR:CA:Q	6	257.91	42.98	0.39	0.89
	P:OR:CA:L	12	598.13	49.84	0.45	0.94
	P:OR:Q:L	12	2079.56	173.30	1.57	0.09
	P:CA:Q:L	6	35.22	5.87	0.05	1.00
	OR:CA:Q:L	6	339.22	56.54	0.51	0.80
	P:OR:CA:Q:L	12	607.09	50.59	0.46	0.94
	Residuals	5244	577366.70	110.10		