UNIVERSIDAD DE CONCEPCIÓN-CHILE FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

Proposición de una arquitectura holónica basada en productos inteligentes para el aumento de la flexibilidad en planificación de la producción y logística.

por

Patricio Antonio Sáez Bustos

Profesor Guía:

Ph.D. Sebastián Godoy Medel

Profesor Co-Guía:

Ph.D. Carlos Herrera López

Tesis presentada a la

DIRECCIÓN DE POSTGRADO UNIVERSIDAD DE CONCEPCIÓN

Para optar al grado de

DOCTOR EN INGENIERÍA INDUSTRIAL

RESUMEN

Proposición de una arquitectura holónica basada en productos inteligentes para el aumento de la flexibilidad en planificación de la producción y logística.

Patricio Sáez Bustos

Marzo 2023

En la era de la industria 4.0 los problemas de la planificación y el control de la producción son muy relevantes en los sistemas de manufactura y su solución es esencial para la optimización de los procesos en la cadena de suministro y la logística. Para resolver estos problemas se han utilizado modelos óptimos (programación matemática), heurísticas, metaheurísticas, simulaciones y metodologías asociadas a herramientas de machine learning. Sin embargo, los resultados de modelos óptimos son demandantes computacionalmente y su aplicación en ambientes productivos reales es impracticable. Alternativamente métodos de aproximación a resultados óptimos (como heurísticas), permitirían direccionar el problema a sistema de control dinámicos y en tiempo real.

En la presente tesis doctoral se aborda la programación de talleres de trabajo tipo job shop y la programación del plan maestro de producción. En la programación de talleres de trabajo se secuencian las operaciones con el fin de optimizar alguna medida de interés, para nuestro caso la minimización del makespan. La programación del plan maestro de producción es uno de los problemas de mayor relevancia en la industria moderna ya que busca identificar cuanto y cuando se debe producir. Para este caso se busca obtener planes más estables, evitando fenómenos perjudiciales como la nerviosidad. Estas programaciones (de taller y de plan maestro) son altamente susceptibles a perturbaciones, haciendo necesario generar alternativas que nos permitan amortiguar el deterioro global de las soluciones.

Los trabajos que componen esta tesis doctoral introducen estrategias de planificación de la producción usando los paradigmas de los sistemas holónicos, sistemas multiagentes, sistemas controlados por el producto y productos inteligentes para incrementar la flexibilidad de los procesos. Los modelos propuestos se basan en arquitecturas holónicas con diferentes tipos de inteligencia embebida en los recursos lo que permite proveerlos de capacidad de decisión para el flujo de su propio proceso. Esta capacidad de decisión a parte de optimizar los procesos permite a los recursos adaptarse a cambios o perturbaciones internas o externas al sistema, generando sistemas más agiles y con capacidad de adaptación.

Particularmente, se generaron modelos de inteligencia artificial con agentes representando virtualmente a productos inteligentes bajo una perspectiva de los sistemas controlados por el producto. En una primera instancia se trabaja sobre configuraciones de taller estándar sujetas a perturbaciones en los tiempos de proceso. Bajo estas condiciones el modelo propuesto reduce hasta en un 10.95% los tiempos de finalización post perturbación. El estudio se continuó analizando la resolución del clásico problema del Job Shop Scheduling Problem en diferentes escalas y bajo una perspectiva descentralizada. Las funciones de inteligencias implementadas fueron heurísticas ampliamente utilizadas en la literatura; shifting bottleneck heuristic y algoritmo evolutivo. La medida de desempeño evaluada es el makespan bajo condiciones óptimas en cortos periodos de ejecución. Las metodologías propuestas obtienen resultados cercanos al optimo en problemas de baja escala. En mayores escalas y en cortos tiempos de ejecución, la metodología propuesta obtiene mejores resultados que modelos óptimos incrementando la capacidad de respuesta con tiempo computacional similar. Por último, al trabajar con el plan maestro de producción es posible reducir la nerviosidad presente en el sistema sin un incremento sustantivo en el costo de producción. Es así como se obtiene una disminución del 11.42% en la nerviosidad con un incremento del 2.39% de los costos totales.

Como trabajo futuro se considera explorar instancias de problemas a escala real, incorporando nuevas perturbaciones y mejorando la toma de decisiones a través de una función de inteligencia más robusta. Además, para tener una medida real de flexibilidad es que se debe realizar una revisión bibliográfica para proponer una medida cuantitativa para su evaluación. Junto con esto, es necesario generar agentes con funciones de inteligencia más complejas asociadas a herramientas de inteligencia artificial que nos entreguen mayor conocimiento del proceso optimizando no solo la nerviosidad del sistema.

Y justo cuando la oruga pensó que era su fin…

se transformó en mariposa…

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

GLOSARIO

- **FMS:** Sistema flexible de manufactura
- **HMS:** Sistema holónico de manufactura
- **ABM:** Modelo basado en agentes
- **MAS:** Sistema multiagente
- **PDS:** Sistema controlado por el producto
- **JSSP:** Problema de programación de talleres (Job Shop Scheduling Problem)
- **SBH:** Heurística cuello de botella
- **LPT:** Regla de secuenciación por tiempo de procesamiento más largo
- **SPT:** Regla de secuenciación por tiempo de procesamiento más corto
- **IP:** Programación entera
- **HMS-MAS:** Sistema Holónicos Multiagente
- **Nervousness:** Nerviosidad en un sistema productivo
- **PPC:** Planificación y control de la producción
- **Netlogo:** Software de simulación basado en agentes
- **Makespan:** Tiempo de finalización de la producción
- **EA:** Algoritmo evolutivo
- **PDS-EA:** Sistema controlado por el producto con algoritmo evolutivo
- **GAP:** Análisis de brecha de conocimiento
- **MPS:** Programación maestra de producción
- **RFID:** Identificación por radiofrecuencia

CAPÍTULO 1. INTRODUCCIÓN

En el contexto de la industria moderna, se tienen ambientes altamente competitivos que evolucionan rápidamente de acuerdo con la disponibilidad de nuevas tecnologías y la globalización de los procesos. Esto ha empujado a las empresas a ser cada vez más reactivas, innovadoras y ágiles. Sin embargo, al mismo tiempo que existe una evolución en el desarrollo de los sistemas, los procesos de la industria se ven afectados por nuevos problemas que inciden directamente en los mercados mundiales. Esto enfoca las investigaciones científicas en el mejoramiento continuo del rendimiento y la calidad de los sistemas de producción.

En la actualidad los sistemas de manufactura buscan la forma de reducir la rigidez imperante, incorporando tecnologías que permitan flexibilizar los procesos productivos. Esta rigidez se ve reflejada en la dificultad que tienen los sistemas para adaptarse a cambios en las programaciones y a la poca agilidad para afrontar perturbaciones. Para subsanar este problema se proponen sistemas más flexibles con capacidad de reacción, sin interrumpir la línea de producción y manteniendo un desempeño satisfactorio (Gräßler & Pöhler, 2017).

Las empresas modernas proponen la utilización de sistemas de fabricación flexibles (FMS) para seguir siendo competitivas en el mercado. Aunque si bien, los FMS existen desde hace décadas, siguen siendo una de las soluciones fundamentales para que un sistema resista los requisitos cambiantes del mercado, reaccionando eficientemente ante perturbaciones internas y externas (Sáez & Herrera, 2021). Los FMS basan sus capacidades en una alta conectividad entre los recursos, permitiendo una toma de decisión descentralizada en diferentes puntos del proceso (Cardin et al., 2018).

Esta alta conectividad y la incorporación de tecnologías de la información entregan a los sistemas productivos una nueva forma de organización y de control, monitorizando, analizando y automatizando los procesos de manufactura. Este tipo de manufactura denominada inteligente genera nuevos paradigmas basados en la cooperación de recursos autónomos, capaces de catalogar, procesar y analizar la información obtenida (Muhuri et al., 2019).

Dentro de este contexto, en esta tesis se propone la implementación de estos nuevos paradigmas, particularmente los Sistemas Holónicos de Manufactura (HMS). Los HMS se han convertido en parte central de los modelos de control descentralizado en la industria, ya que han mostrado ser una eficiente manera de llevar los conceptos teóricos a la práctica. Los HMS se basan en la comunicación y la cooperación de los recursos presentes en el sistema dando una representación virtual a cada componente físico, generando modelos mucho más cerca de los sistemas naturales y alejándose de ser simples estructuras físicas automatizadas, sino más bien, entidades capaces de autoorganizarse autónomamente (mezclando el mundo físico y virtual), para resolver problemas antieconómicos e ineficiencias (Mcfarlane et al., 2002).

La aplicación de sistemas de control basados en HMS permite la adaptación a las condiciones cambiantes del mercado, generando una representación del mundo real y operando en los procesos a través de procedimientos que actúan de forma remota. Para esto, la simulación asume un papel crucial visualizando el comportamiento de las soluciones durante la fase de diseño y antes de su implementación real.

Para la representación de los procesos de manufactura se utiliza particularmente el modelado basado en agentes (ABM). Este tipo de modelo simula conductas de fenómenos complejos como comportamientos emergentes y de auto organización. Estas conductas entregan una gran capacidad de adaptación ante las perturbaciones con un alto grado de autonomía sin intervención externa al proceso. Estas características proporcionan automatización, modularidad y robustez, solucionando al menos el 25% de los problemas de fabricación (Barbosa & Leitao, 2011; Lee, 2008).

El modelo propuesto incorpora el paradigma Holónicos con la simulación basada en agentes (ABM) en un proceso productivo donde cada recurso tiene la capacidad de comunicarse y cooperar, permitiendo la posibilidad de que el sistema sea controlado por el producto (PDS). Un PDS hace evolucionar la visión hacia un sistema más interoperable e inteligente, en el que se define el producto como capaz de tomar decisiones e influir en su proceso llevando a la práctica los conceptos establecidos por los HMS (Herrera, 2011).

Los PDS sumados a productos inteligentes entregan autonomía y crean nuevas oportunidades de mejora en reactividad y agilidad, debido a la descentralización de la toma de decisión y la generación de soluciones en cortos periodos de tiempo. Los productos inteligentes y los PDS pueden coexistir de forma dinámica y temporal, o incluso ser parte de las coordinaciones en los diferentes niveles de la empresa (Herrera et al., 2010). El modelo propuesto muestra que la coordinación entre las entidades de un sistema productivo es factible y destaca la importancia de la reactividad en la toma de decisiones para la generación de ambientes productivos más estables.

1.1 Contribución de tesis

La presente tesis doctoral propone un modelo de Planificación y Control de la Producción (PDS) que utiliza Herramientas de Sistemas Multiagente (HMS) y Modelado Basado en Agentes (ABM) para mejorar la flexibilidad y optimización en el uso de recursos en sistemas productivos. La principal contribución de esta propuesta es mostrar que los PDS son una herramienta confiable para enfrentar perturbaciones en la planificación de la producción y minimizar la nerviosidad presente en un plan maestro de producción.

Los artículos que componen este manuscrito entregan un contexto de la aplicación del modelo propuesto. En los primeros modelos presentados se aplica un PDS a un sistema productivo con una arquitectura estándar evaluando las consecuencias de perturbaciones en los tiempos de producción. Continuamos el estudio aplicando el modelo propuesto a problemas clásicos de la literatura (Job Shop Scheduling Problem) con una función de inteligencia para la toma de decisiones basada en una heurística de descomposición. En el tercer artículo, se continua con el análisis a problemas clásicos en la literatura, pero se utiliza como función de inteligencia para la toma de decisiones un algoritmo evolutivo que permite mejorar las soluciones propuestas en cortos periodos de tiempo. Por último, se aplica el modelo PDS para la planificación y control de la producción en un caso de estudio sintético con 12 productos y un horizonte de planificación de 52 semanas, con el objetivo de reducir la nerviosidad del sistema. El detalle de los artículos es el siguiente:

(1) Sáez Bustos, P., Herrera López, C. (2021). Implementation of a Holonic Product-Based Platform for Increased Flexibility in Production Planning. In: Trentesaux, D., Borangiu, T., Leitão, P., Jimenez, JF., Montoya-Torres, J.R. (eds) Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future. SOHOMA 2021. Studies in Computational Intelligence, vol 987. Springer, Cham. https://doi.org/10.1007/978-3-030-80906-5_12.

(2) P. Sáez, C. Herrera, J.E. Pezoa, A product-driven system approach to generate fast solutions to the job shop scheduling problem., IFACPapersOnLine, Volume 55, Issue 10,2022, Pages 1930-1937, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2022.09.681.

(3) Sáez P, Herrera C, Booth C, Belmokhtar-Berraf S, Parada V (2023) A product-driven system with an evolutionary algorithm to increase flexibility in planning a job shop. PLoS ONE 18(2): e0281807. https://doi.org/10.1371/journal.pone.0281807

(4) Sáez, Patricio. Herrera, Carlos. Parada Victor. A product-driven system approach to reduce nervousness in master production Schedule. International Journal of Production Research. 2023

En el artículo **"Implementation of a Holonic Product-Based Platform for Increased Flexibility in Production Planning"** se propone una Plataforma HMS con una arquitectura altamente distribuida para la toma de decisiones en un sistema de producción estándar. En este artículo se analiza la performance en base al makespan como medida de desempeño en presencia de perturbaciones de demanda. Nuestro objetivo es mostrar la necesidad de proveer de flexibilidad en el proceso de toma de decisiones en los sistemas productivos. Los resultados de este articulo muestran que el PDS logra disminuir hasta un 10.95% los tiempos de finalización, en presencia de incrementos de hasta un 400% en los tiempos de producción.

En el artículo "**A product-driven system approach to generate fast solutions to the job shop scheduling problem"**, se estudia el tiempo de ejecución y la capacidad de un PDS para encontrar buenas soluciones en sistemas del tipo jobshop. Este trabajo se enfoca en resolver el Job Shop Scheduling Problem (JSSP) a través de un Sistema inteligente de manufactura. Dada la alta carga computacional (que se incrementa con el tamaño del problema), los métodos convencionales no son mayormente usados en la práctica industrial. Por esto se propone un modelo de inteligencia artificial que usa los productos inteligentes para tomar decisiones y resolver el JSSP en cortos periodos de tiempo. Se experimenta con instancias disponibles en la literatura y se compararon los resultados en 60 segundos de ejecución con metodologías optimas (programación entera), heurísticas de descomposición (SBH), y reglas de secuenciación evaluando el performance de acuerdo con el makespan obtenido y el presente en la literatura. Los resultados indican que el modelo propuesto entrega en algunas instancias mejores soluciones que metodologías óptimas en cortos periodos de ejecución.

En el artículo "**An adaptive product-driven system using evolutionary algorithms to increase the flexibility in scheduling problems at different scales"** se considera un modelo HMS-MAS para incrementar la flexibilidad de un sistema de producción a través de respuestas rápidas y con resultados cercanos al optimo en problemas de diferente tamaño. Para la toma de decisiones se considera una función de inteligencia basada en un algoritmo evolutivo, donde cada agente tiene la capacidad de tomar decisiones en base a las mejoras consecutivas de las soluciones encontradas. El modelo fue testeado en 102 instancias presentes en la literatura comparando los resultados con metodologías optimas, heurísticas y reglas de secuenciación de amplio uso en la industria. Las principales contribuciones de este articulo son: Proponer un nuevo modelo adaptativo que resuelve el JSSP en diferentes escalas incluyendo como función de inteligencia un algoritmo evolutivo para la búsqueda de mejores soluciones y la utilización del paradigma HMS-MAS con productos inteligentes en arquitecturas altamente distribuida en cortos tiempos de ejecución. Como resultado el articulo sugiere que el PDS produce soluciones cercanas al optimo en cortos periodos de tiempo sin importar la escala del problema estudiado.

En el artículo **"A product-driven system approach to reduce nervousness in master production Schedule"** se propone un sistema controlado por el producto que complementa los resultados obtenidos por modelos óptimos del plan maestro de producción. Este modelo se basa en productos inteligentes que toman las decisiones de producción con una función de inteligencia capaz de reducir la nerviosidad del sistema sin incrementar en una misma magnitud el costo de ejecución. Los resultados finales muestran una relación dispar entre el incremento del costo de producción y la disminución del nerviosismo del sistema. La sensibilidad en el incremento del costo versus la reducción de la nerviosidad logra disminuir el nerviosismo en 11.42% con un incremento del 2.39% de los costos.

1.2 Organización general de tesis

Este documento de tesis está organizado de la siguiente forma. El Capítulo 2 presenta el trabajo denominado "Implementation of a Holonic Product-Based Platform for Increased Flexibility in Production Planning", con sus principales contribuciones. El Capítulo 3 describe el trabajo "A product-driven system approach to generate fast solutions to the job shop scheduling problem". El Capítulo 4 describe el modelo propuesto en "An adaptive product-driven system using evolutionary algorithms to increase the flexibility in scheduling problems at different scales". El Capítulo 5 presenta el artículo A product-driven system approach to reduce nervousness in master production Schedule. El capítulo 6 las principales conclusiones de los trabajos presentados.

1.3 Artículos publicados

- Sáez Bustos, P., Herrera López, C. (2021). Implementation of a Holonic Product-Based Platform for Increased Flexibility in Production Planning. In: Trentesaux, D., Borangiu, T., Leitão, P., Jimenez, JF., Montoya-Torres, J.R. (eds) Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future. SOHOMA 2021. Studies in Computational Intelligence, vol 987. Springer, Cham. https://doi.org/10.1007/978-3-030-80906-5_12.

- P. Sáez, C. Herrera, J.E. Pezoa, A product-driven system approach to generate fast

solutions to the job shop scheduling problem., IFACPapersOnLine, Volume 55, Issue 10,2022, Pages 1930-1937, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2022.09.681.

- Sáez P, Herrera C, Booth C, Belmokhtar-Berraf S, Parada V (2023) A product-driven system with an evolutionary algorithm to increase flexibility in planning a job shop. PLoS ONE 18(2): e0281807. https://doi.org/10.1371/journal.pone.0281807.

Implementation of a Holonic Product-Based Platform for Increased Flexibility in Production Planning

Patricio Sáez Bustos(✉) and Carlos Herrera López

Department of Industrial Engineering, University of Concepción, Concepción, Chile
{patricsaez,cherreral}@udec.cl

Abstract. In the Industry 4.0 era, production planning problems are very relevant to production systems and are essential parts of the supply chain. Broadly speaking, production planning problems are tackled using models and methodologies, aiming for optimal solutions. This work introduces realism and stability to optimal production planning strategies using a holonic, product-driven manufacturing platform with increased flexibility. A model based on an anarchic holonic architecture and embedded intelligence logic provides decision-making capacity in a "production lot" in the face of disturbances. The proposed model is validated by comparing the results obtained with a lot-streaming mathematical programming model. Results show that significant changes in lot processing times (disturbances) generate significant changes in completion times. The proposed platform reduces up to 10.95% completion times in face of disturbances, generating significant benefits by increasing flexibility.

Keywords: Industry 4.0 · Holonic manufacturing system · Multi-agent system · Anarchic manufacturing · Lot-streaming · Smart product · Flexibility

1 Introduction

Production planning and control (PPC) is recognized as a complex problem in the industry, that requires achieving customer satisfaction and optimizing available resources. This complexity is explained by a large number of interrelated elements and variables. In specific cases, the problems are theoretical, without real application. Among the most studied production systems are manufacturing production systems organized as flow shops and job shops or task workshops [1–4]. In particular, both types of production systems are present in an industry (independently or jointly), becoming also a central part of the literature's works.

Conventional production planning systems work by developing hierarchies between different product aggregation levels [5], resulting in monotonous and static production scheduling [6]. New perspectives, such as multi-agent systems (MASs) and holonic manufacturing systems (HMSs), have attracted increasing interest in the industry by introducing agility, adaptability, autonomy, and, above all, flexibility in production systems [7, 8]. HMSs manage production in decentralized and distributed decision-making

D. Trentesaux et al. (Eds.): SOHOMA 2021, SCI 987, pp. 180–194, 2021.
https://doi.org/10.1007/978-3-030-80906-5_12

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-10 (2022) 1930–1937

A product-driven system approach to generate fast solutions to the job shop scheduling problem.

P. Sáez*, C. Herrera*, J. E. Pezoa**

*Departamento de Ingeniería Industrial, Universidad de Concepción, Chile (e-mail:patricsaez@udec.cl cherreral@udec.cl).

**Departamento de Ingeniería Eléctrica, Universidad de Concepción, Chile (e-mail: jpezoa@udec.cl)

Abstract: Many optimal algorithms, heuristics, metaheuristics, simulation approaches, agent-based models, and machine learning tools attempt to solve the job shop scheduling problem (JSSP). This article proposed a model of artificial intelligence with agents representing intelligent products from the perspective of product-driven systems (PDS) to solve this problem at different scales. The intelligent products make all decisions in a distributed way aiming to minimize the makespan and increase the computational efficiency for the JSSP. The agents embed the intelligence function using a based shifting bottleneck heuristic (SBH) approach. The novelty of the proposed approach lies in the automation of decisions in a highly distributed architecture to increase manufacturing flexibility. The results are compared with an optimal integer programming model (IP), SBH, and two conventional heuristics considering instances commonly used in the literature. Concerning the makespan, the proposed approach obtains a fast solution near optimal in instances with a low number of resources and better results than IP and conventional heuristic in instances with a more significant number of resources, increasing the response capacity with a similar computational time.

Keywords: Agent-based models, job-shop, sequencing, shifting bottleneck heuristic, intelligent products, product-driven system.

1. INTRODUCTION

In modern production, procedures are increasingly flexible in the face of internal or external disturbances. This process has caused the design of new techniques (such as intelligent agents) to adaptively control or hybridize methodologies to solve specific problems (Adam et al., 2010).

The assignment of tasks and processes to be performed by different machines and on different jobs is called a production scheduling problem. This assignment is one of the most challenging tasks that companies face today and has been the focus of extensive research (Fuchigami & Rangel, 2018). Different methods and approaches have been tested and applied to the problem of production scheduling to solve problems. However, the time difference between the beginning and the completion of a series of tasks or makespan represents one of the most important objectives because it is directly related to customer satisfaction performance indicators.

The job-shop scheduling problem (JSSP) is considered an NP-hard problem (Asadzadeh, 2015) and has been extensively studied (Fuchigami & Rangel, 2018).In the literature, there are exact methods, such as integer programming and branch and bound, to solve the JSSP. However, the high computational load exponentially increases with the size of the problem (Nowicki & Smutnicki, 2005). For real industrial problems, the computational time of a given algorithm or method should not be too long for practical use. It has been decided to use a wide variety of heuristic procedures in the industry, which provide good results in a reasonable amount of time (Bożek & Werner, 2018).

Decomposition-based heuristics and metaheuristics such as the bottleneck algorithm (Adams et al., 1988); Local search algorithms such as taboo search (Bożek & Werner, 2018) and Simulated Annealing (Monostori et al., 2006), are used for large-scale problem cases, trying to provide flexibility in the manufacturing processes. These heuristics develops solutions for complex problems by breaking a problem into a series of smaller subproblems, which are more manageable and easier to solve. One of the most used decomposition heuristics is the shifting bottleneck heuristic (SBH), which was proposed by Adams et al. (1988). This algorithm decomposes the JSSP into subproblems that iteratively program a single machine. According to Ovacik and Uzsoy (1992), a decomposition method has better results than dispatch rules in both its average and the worst case.

The intelligent product is the representation of an order or physical product linked to the information and the rules that govern its manufacture, storage, or transport, allowing it to influence operations (Wong et al., 2014). The use of intelligent products brings essential benefits to a product-driven production approach and the JSSP (Herrera et al. 2014; Herrera et al. 2016). In this sense, it has been used to improve the entire life cycle of products, i.e., design, production,

10.1016/j.ifacol.2022.09.681

PLOS ONE

RESEARCH ARTICLE

A product-driven system with an evolutionary algorithm to increase flexibility in planning a job shop

Patricio Sáez[1☯]*, Carlos Herrera[1☯], Camila Booth[1☯], Sana Belmokhtar-Berraf[2‡], Victor Parada[3,4‡]

1 Department of Industrial Engineering, Universidad de Concepción, Concepción, Chile, **2** Université Gustave Eiffel, ESIEE Paris, COSYS-GRETTIA, Marne-la-Vallée, France, **3** Department of Informatics Engineering, Universidad de Santiago de Chile, Santiago, Chile, **4** Instituto Sistemas Complejos de Ingeniería (ISCI), Santiago, Chile

☯ These authors contributed equally to this work.
‡ SBB and VP also contributed equally to this work.
* patricsaez@udec.cl

Citation: Sáez P, Herrera C, Booth C, Belmokhtar-Berraf S, Parada V (2023) A product-driven system with an evolutionary algorithm to increase flexibility in planning a job shop. PLoS ONE 18(2): e0281807. https://doi.org/10.1371/journal.pone.0281807

Editor: Praveen Kumar Donta, TU Wien: Technische Universitat Wien, AUSTRIA

Received: December 13, 2022

Accepted: February 1, 2023

Published: February 16, 2023

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: V. Parada gratefully acknowledge financial support from ANID PIA/PUENTE AFB220003 and DICYT-USACH 061919PD. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

Abstract

The scheduling of a job shop production system occurs using models to plan operations for a given period while minimizing the makespan. However, since the resulting mathematical models are computationally demanding, their implementation in the work environment is impractical, a difficulty that increases as the scale problem grows. An alternative approach is to address the problem in a decentralized manner, such that real-time product flow information feeds the control system to minimize the makespan dynamically. Under the decentralized approach, we use a holonic and multiagent systems to represent a product-driven job shop system that allows us to simulate real-world scenarios. However, the computational performance of such systems to control the process in real-time and for different problem scales is unclear. This paper presents a product-driven job shop system model that includes an evolutionary algorithm to minimize the makespan. A multiagent system simulates the model and produces comparative results for different problem scales with classical models. One hundred two job shop problem instances classified as small, medium, and large scale are evaluated. The results suggest that a product-driven system produces near-optimal solutions in short periods and improves its performance as the scale of the problem increases. Furthermore, the computational performance observed during the experimentation suggests that such a system can be embedded in a real-time control process.

Introduction

Industry 4.0 requires manufacturing systems to be flexible, dynamic, and able to react immediately to disruptions. These are critical aspects in the design stage, for which it is necessary to resort to modeling that integrates the advantages of traditional modeling [1]. Such models are the basis of the manufacturing process control system [2], and should consider the integration

CAPÍTULO 2.

Implementation of a Holonic Product-Based Platform for Increased Flexibility in Production Planning

Cita: Sáez Bustos, P., Herrera López, C. (2021). Implementation of a Holonic Product-Based Platform for Increased Flexibility in Production Planning. In: Trentesaux, D., Borangiu, T., Leitão, P., Jimenez, JF., Montoya-Torres, J.R. (eds) Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future. SOHOMA 2021. Studies in Computational Intelligence, vol 987. Springer, Cham. https://doi.org/10.1007/978-3-030-80906-5_12.

Abstract. In the Industry 4.0 era, production planning problems are very relevant to production systems and are essential parts of the supply chain. Broadly speaking, production planning problems are tackled using models and methodologies, aiming for optimal solutions. This work introduces realism and stability to optimal production planning strategies using a holonic, product-driven manufacturing platform with increased flexibility. A model based on an anarchic holonic architecture and embedded intelligence logic provides decision-making capacity in a "production lot" in the face of disturbances. The proposed model is validated by comparing the results obtained with a lot-streaming mathematical programming model. Results show that significant changes in lot processing times (disturbances) generate significant changes in completion times. The proposed platform reduces up to 10.95% completion times in face of disturbances, generating significant benefits by increasing flexibility.

Keywords: Industry 4.0, Holonic manufacturing system, Multi-agent system, Anarchic manufacturing, Lot-streaming, Smart product, Flexibility.

2.1 Introduction

Production planning and control (PPC) is recognized as a complex problem in the industry, that requires achieving customer satisfaction and optimizing available resources. This complexity is explained by a large number of interrelated elements and variables. In specific cases, the problems are theoretical, without real application. Among the most studied production systems are manufacturing production systems organized as flow shops and job shops or task workshops (Fan & Cheng, 2016; Raileanu, 2010; Shahzad & Mebarki, 2016; Wu et al., 2019). In particular, both types of production systems are present in an industry (independently or jointly), becoming also a central part of the literature's works.

Conventional production planning systems work by developing hierarchies between different product aggregation levels (Ma et al., 2019), resulting in monotonous and static production scheduling (Rolón & Martínez, 2012). New perspectives, such as multi-agent systems (MASs) and holonic manufacturing systems (HMSs), have attracted increasing interest in the industry by introducing agility, adaptability, autonomy, and, above all, flexibility in production systems (Kruger & Basson, 2019; Mcfarlane et al., 2002). HMSs manage production in decentralized and distributed decision-making architectures, while MASs provide a greater degree of flexibility and reconfiguration to production systems. In this case, the agents provide a physical representation of the system components, including machines, equipment, products, lots, etc., thus providing different perspectives and control scenarios (Leitão et al., 2015).

An HMS platform with an anarchic decision-making architecture was developed to study the need for more flexible production plans (Ma et al., 2019). This architecture promotes the need for the decentralization of decisions, delivering this action to the lowest links in the production chain, in this case to production lots. Therefore, production lots exhibit behavior similar to an intelligent product with features such as those set by (Herrera, 2011; Mcfarlane et al., 2002; Wong et al., 2014). The validation of our platform was performed by comparing the results of a practical example to minimize makespan. A lot-streaming mathematical programming model optimally solved the production planning problems considered in this work. Subsequently, the system was disturbed by changing machines' processing times (typical disturbances in production systems due to failures, machine lock, starving, etc.). In this way, the deterioration impact of planning and the contribution of a distributed decision-making system such as the one proposed can be assessed. Our goal is to show the need for systems that provide more flexibility in decision-making processes.

The article is organized as follows: Section 2.2 describes a bibliographic review associated with flexibility issues in PPC; measures and tools used for the analysis and incorporation of flexibility into production processes are analyzed along with the participation of MAS and HMS models for this purpose. Section 2.3 presents the materials and methods. The construction of the holonic model and the mathematical programming model developed to obtain optimal solutions are also presented. Section 2.4 describes the experimentation developed for each of the test instances. Section 2.5 shows our application's main results by considering a standard case and one with disruptions in process times. Finally, Section 2.6 summarizes our work.

2.2 Bibliographic Review

Flexibility is an attribute that gives manufacturing systems the possibility that at a certain level of variation in the quantities to be produced and/or at interruptions in the production line there will be no significant changes in planning (Yadav & Jayswal, 2018). In Slack (1983) examined the concept of flexibility in manufacturing, defining a framework of attributes that influence the different aspects of flexibility. However, the idea of flexible manufacturing systems was proposed in 1960 by David Williamson, who devised the so-called "24 system" of machines capable of producing 24 h a day without human intervention.

Scheduling problems have been extensively studied in the literature and are a fundamental part of production systems theory. Thus, many articles on flexible manufacturing systems can be found in the literature with techniques that provide the necessary flexibility (Demirel et al., 2018; Lee, 2008; Zhou et al., 2019). The techniques analyzed are in the fields of simulation, artificial intelligence (AI), and Petri networks, among others. However, there are also works based on mathematical programming, such as that developed by Stecke & Solberg (1981), who used mixed nonlinear programming for machine grouping to minimize part movement. While known to deliver optimal results, such methods can be affected during actual operation by the high execution times of the implemented algorithms (Topaloglu & Kilincli, 2009).

One of the most widely used performance indicators to be minimized in production planning is the completion time (makespan) (Ahmadi-Darani et al., 2018; Gu et al., 2010; Topaloglu & Kilincli, 2009). This concept must be accompanied by other indicators, such as costs or failure rates in more complex models to achieve a closer version of reality. In Choi and Wang (2012), flexibility was incorporated into a sequential production environment using a new decomposition method combined with sequencing rules (shorter process times are processed first) and a genetic algorithm, which minimize makespan by delivering flexibility to the process.

Distributed processes such as MASs and HMSs provide an excellent opportunity to respond to changes in production environment conditions effectively. Wang et al., (2018) provided real-time production planning supported by a new architecture based on a MAS model in conjunction with the Internet of Things (IoT). This model proposed an optimal machine allocation strategy depending on component status. However, this approach does not incorporate a complete decentralization of the decision-making process. On the other hand, in the work of S. Raileanu et al. (2010), production planning was implemented by a HMS generating control at different levels. Communication with these levels occurs in the upper layers, with recommendations sent to lower layers which communicate with each other to solve and optimize tasks.

2.3 Materials, Methods and Proposal

The use of scheduling and lot-streaming techniques to solve programming problems has been widely used to reduce completion times optimally. Lot-streaming is a technique that considers “n” jobs on “m” machines, where jobs are divided into sublots to minimize the delay and completion time of tasks (see Figure 2.1). In works such as those described in (Kumar et al., 2000; Potts & Baker, 1989; Trietsch & Baker, 2008; Tseng & Liao, 2008), lot-streaming is used in various programming problems, the most studied being those about sequential process (flow shop and job shop) organization.

In today’s industrial practices, many of the problems include objectives that conflict with each other. These problems are solved by various optimization models, which work with subproblems simultaneously or individually (Gharaei & Jolai, 2018). In this work, an HMS-based approach is proposed for smart lots that make production process decisions for the products contained therein.

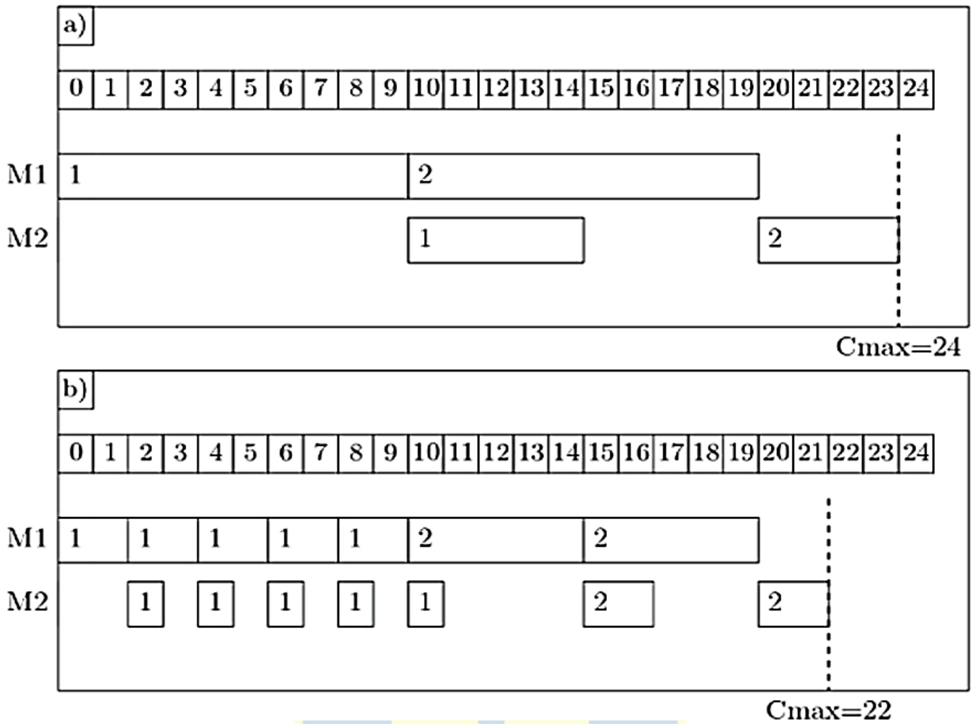

Figure 2.1. (a) Scheduling two jobs into two sequential machines (M1 and M2); (b) Scheduling jobs using the lot-streaming technique, dividing work into sublots (Tseng & Liao, 2008).

The proposed architecture has been simulated on the NETLOGO platform (Wilensky, 1999). The architecture is composed of two serial stations in which production processes are developed for batches. At the first station is the M1 machine, where the production process starts sequentially. At the second station, the machines M2, M3, and M4 finish the production process (each lot can be processed on only one second-stage machine), as shown in Figure 2.2.

Product lots planning calculates the target global function to minimize completion time or makespan. This objective refers to finishing production as quickly as possible and is calculated by lots at the end of their production processes. Decisions are reused to perform a learning process for future generations of solutions.

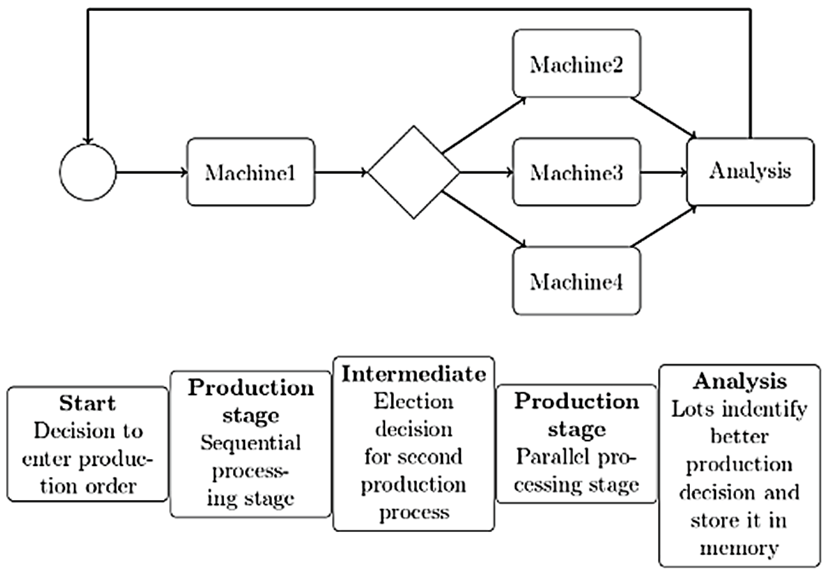

Figure 2.2. Stages of the production process.

2.3.1 Interactions Between Agents

A MAS simulation is used to solve this problem for a part lot in an anarchic structure perspective (Ma et al., 2019). The type of HMS system is given because the agents of the MAS model represent physical system components. In the work of Yu et al. (2018), three categories of agents were developed: work agents, machine agents and control agents; however, our system, having an anarchic architecture, delegates authority and autonomy in decision- making to the lowest level of system entities without centralized control or supervision, so there will be only two categories: lot agents and machine agents. Machine agents are static and only receive nonparticipating lot agents; lot agents are dynamic entities that control the realization of the products they represent.

The parameters and steps used in our algorithm are presented in the following pseudocode:

```
Home:
Create lot variable and lot family;
Create variables for lots: process times, layout position,
andmemory;
Setting function ();
Create differentiated machines between type I and II (series
and parallel, respectively);
Create undifferentiated lots and positioned in the initial
machine;
Differentiation Function ()
Identify existing lot types and categorizes lots;Normal
production times are allocated per lot; Initial Position
Function ();
Assign the position variable in each lot a random i value be-
tween 1 and n with n s number of lots;
Execute function ();
If the lots have not finished processing then;
  If the lots are in the initial machine then;
       Compare their positions, the one with priority
       initiates processing (Priority: lower position in the
       layout);
   If lots finish processing then;
       Ask machines in parallel if they are processing
       If machines are not processing then;
          Lot moves to unemployed machine
       Else
       Lot asks lots in process who finishes beforeStarts
       processing
   Else
   Continue processing
  Else
  Continue processing
Else
Makespan Function (); Calculate completion time
Lots keep in memory the position in which the best result was
obtained
Lots move to starting position
```

According to the pseudocode above, each lot corresponds to an agent defined by the minimum accepted size lot. The generated agents follow the concept of intelligent products proposed by C. Wong et al. (2018) and C. Herrera (2011), where each entity (product lot) consists of the following characteristics:

1. Agent has a unique identity.
2. Agent can communicate effectively with its environment.
3. Agent can retain or store data about itself.
4. Agent implements a language to display its features, production requirements, and so on.
5. Agent can participate or make decisions relevant to its results.

In particular, an agent has the following characteristic profile:

1. Unique identity: Each agent has a specific and unrepeatable ID (although there are shared features when talking about lots of the same product).
2. Effective communication: Communication between agents (lots) is active and indifferent to location. This is based on the decision-making process and choosing the best sequence for the common goal.
3. Retention and storage of information about itself: An agent has a memory of its processing time on each machine. Besides, it saves the position that improves each of the objective functions.
4. Language: Based on group consultations with all agents or with agents in sectors of the production process.
5. Relevant decision making: Decision to process or not, in addition to the choice of the machine on which to process.

Communication between agents is classified according to the different possibilities of interactions present in the model. These interactions are defined as visual, auditory, and verbal.

Visual interactions refer to whether the product agent displays its entity (level 1), its nearby environment (level 2), or the entire medium (level 3). Auditory interactions correspond to the ability to capture media information from level 1 that identifies the agent as independent of the environment. It develops at a level 2 where it captures local and selective information, to reach a level 3 in which it captures information from the entire environment. Finally, verbal interactions indicate the level of delivery of agent-product information, from disconnection with the medium (level 1), delivery of timely information (level 2), or delivery of information to the entire medium (level 3) (See Figure 2.3).

The interaction between agents has a fixed query-based structure. The structure of the interaction between system components is shown in Figure 2.4. This representation follows the UML sequence structure and shows the conceptual architecture of an intelligent system.

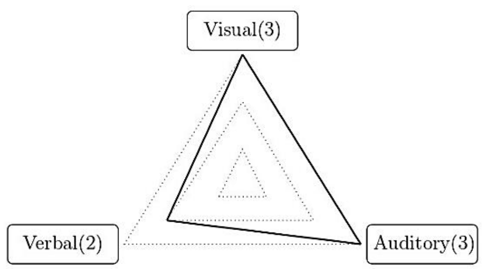

Figure 2.3. Shows the characterization of the agent-product for testing.

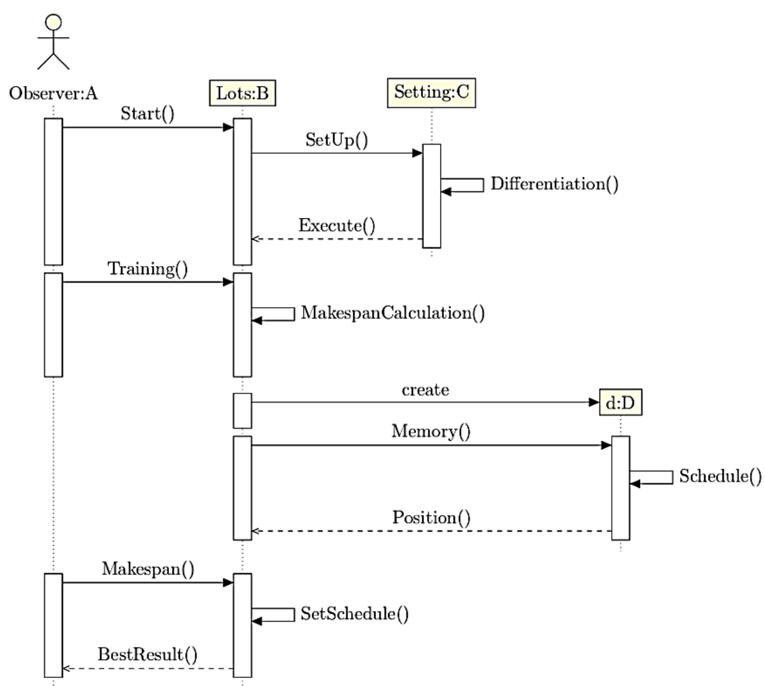

Figure 2.4. Sequence UML diagram

The first object in the diagram presented in Figure 2.4 is the “observer”. It describes user interaction with the model and the receipt of the best results by agents. The second multi-object component (comprises all lots present in the simulation) displays the sequential structure of lot operations, taking the indications entered by the observer. Additionally, it explains the sequence of actions executed by agents (lots), from the configurations of their attributes to indicating the best result to the observer, taking the best position stored in their memory. The third and fourth objects are created from the lots by configuring each agent and saving the best position for each of the objective functions: makespan and utilization.

The structure of our work is exemplified in the UML diagram of classes presented in Figure 2.5. Model interactions focus on the observer and lots through a configuration layer that creates machines and lots, allowing lot differentiation depending on the products they contain. The diagram continues with an accurate class, attending to decision-making at the lowest levels, which reads and writes attributes autonomously. Finally, the scheduling object corresponds to the result of the production configurations that the agents run.

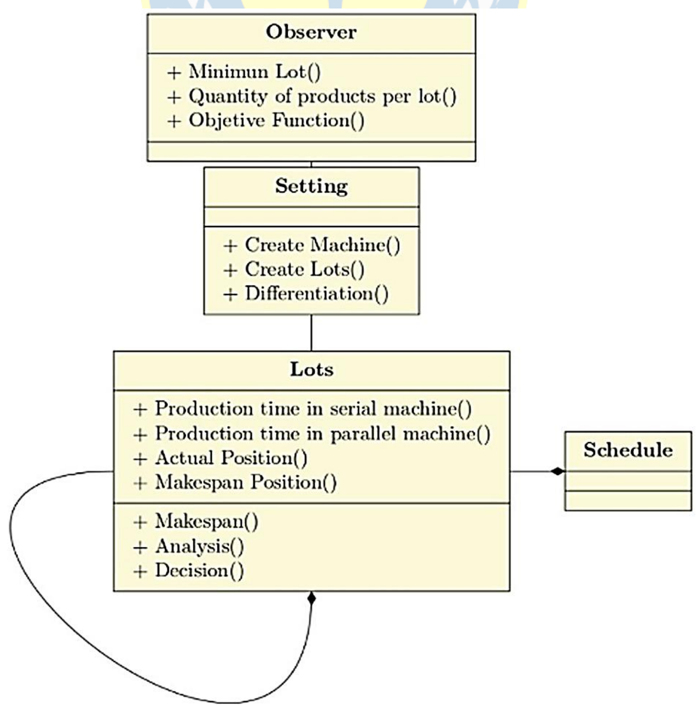

Figure 2.5. UML diagram of classes for the proposed model

The lot class corresponds to all the agents present in the model and aims to complete its production process through the sequence of machines established for the system (Figure 2.5). This model consists of a sequential machine where the production process begins and continues with the decision-making process of agents, and a second production process involving three identical and unrelated parallel machines. No process has a setup time; however, each machine could have a different rhythm (due to breakdowns or some other cause external to the process). Each machine can serve one lot at a time (with a number of products defined by the minimum lot accepted) and ends in an analysis stage. Machines learn from the generated sequence to position themselves back in the production queue and restart the process.

2.3.2 Mathematical Programming Model

For comparison, a mathematical lot-streaming model is formulated as well. This mathematical programming model will allow us to develop a generic configuration with optimal results to validate the actual results. The configuration that was considered for the mathematical programming model is shown in Figure 2 (2-stage hybrid system), where the first stage corresponds to mass production and the second to parallel production. This model was developed and validated in (Herrera, 2011).

The problem with the lot-streaming that we address in our work is to divide the quantities to be produced of each product into sub lots to reduce the total sequencing duration (makespan). Sub lots are constrained by minimum amounts defined by the decision-making process, and the model parameters are included in Table 2.1.
The complete model is developed below:

Table 2.1. Parameters and variables for the mathematical model

Parameters

P_i	Number of products per lot
Q_{min}	Minimum sub lot size
TPA_p	Unit production time in machine A for product p
TPB_p	Unit production time in machine B for product p

J	Number of lots
K	Number of machines in stage two (parallel)
M	Big number

Variables

x_{ijk}	Number of products i in lot j assigned to machine k
w_{ij}	1 if product i is assigned to lot j in stage 1 and 0 otherwise
w_{ijk}	1 if product i is assigned to position j in machine k in stage 2 and 0 otherwise
STA_j	Start time of stage 1 for lot j
STB_{jk}	Start time of stage 2 for lot j in machine k

Objective Function:

min C_{max}

subject to:

(1) $$\sum_{j=1}^{J}\sum_{k=1}^{K} x_{ijk} = N_i \qquad i = 1,2$$

(2) $$x_{ijk} \geq Q_{min} * w_{ij} \qquad i = 1,2, \forall j, \forall k$$

(3) $$x_{ijk} \leq M * w_{ijk} \qquad i = 1,2, \forall j, \forall k$$

(4) $$w_{1j} + w_{2j} = 1 \qquad \forall j$$

(5) $$\sum_{k=1}^{K} w_{ijk} = 1 \qquad i = 1, 2, \forall j$$

(6) $$STA_1 = 0$$

(7) $$STA_j = STA_{j-1} + TPA_1 * \sum_{k=1}^{K} x_{1jk} + TPA_2 * \sum_{k=1}^{K} x_{2jk} \qquad \forall j$$

(8) $$STB_{(j-1)k} \geq STA_j \qquad \forall j, \forall k$$

(9) $$STB_{jk} \geq STB_{(j-1)k} + TPB_1 * x_{1jk} + TPB_2 * x_{1jk} \qquad \forall j, \forall k$$

(10) $$Cmax \geq STB_{Jk} + TPB_1 * x_{1Jk} + TPB_2 * x_{1Jk} \qquad \forall k$$

Restrictions (1), (2) and (3) refer to the number of products assigned to each lot and their lower (Q_{min}) and upper limits M. Constraints (4) and (5) are assignments for each lot j and machine k, respectively. Restrictions (6), (7), (8) and (9) indicate the start times of each lot in both stage one and stage two. Finally, constraint (10) sets the maximum end time value.

2.4 Experiments

Next, we analyze the consequences of lack of flexibility in production. A practical example was generated with 2 products distributed in minimum lots of 100 units, with a maximum production of 1200 units (total units considering the two products). This example follows the structure presented in Figure 2.2, with two production stages to manufacture a product. Machines can work these products with different production times. The first stage starts working semi-finished products and the second stage finishes products. The programming results generated by the lot-streaming model, as well as those obtained by the MAS platform, will be affected by disturbances in the processing time of the machines when identifying the planning consequences and analyzing their behavior.

2.4.1 Description of Experimentation

Our goal is to verify the need for flexible production plans capable of adapting to disruptions during production processes. Lot production will be simulated in a planning horizon that allows existing lots to be completed. The production will be generated respecting its maximum volume (established at 1200 units) and the demand originated by combinations of minimum lots of both products (e.g., DP1   DP2   1200, with DP1 as the demand for product 1 in lots of 100 units and DP2 as the demand for product 2 in lots of 100 units).

The simulation parameters are as follows (Table 2.2):

Table 2.2. Simulation parameters

Parameters	Values
Types of products	2

Lots	Demand dependent (100 products per lot)
Parameters	Values
Production	Maximum of 1200 units in total (product 1 + product 2)
Normal production time	Production time: product 1 in stage 1 × 1 Production time: product 1 in stage 2 × 3 Production time: product 2 in stage 1 × 2 Production time: product 2 in stage 2 × 8
Internal disturbances	
Increased production time for products 1 and 2 in stage 1	100, 200 and 400 (%)
Increased production time for products 1 and 2 in stage 2	100, 200 and 400 (%)

2.5 Results

Table 2.3 lists the production data associated with the analyzed practical example. Nine volumes of production were assessed. Additionally, the number of lots per product and completion times are set for the mathematical programming (PM) algorithm and multiagent platform (PMAS).

Table 2.3. Production overview and volume of lot (T.F.: End time in hours)

Demand	Prod.1	Prod. 2	Lots (Prod. 1)	Lots (Prod. 2)	T.F. PM	T.F. PMAS
Demand1	1000	200	10	2	19	19
Demand2	900	300	9	3	20	20
Demand3	800	400	8	4	22	22
Demand4	700	500	7	5	24	24
Demand5	600	600	6	6	26	26
Demand6	500	700	5	7	28	28
Demand7	400	800	4	8	29	29
Demand8	300	900	3	9	32	32
Demand9	200	1000	2	10	34	34

It is clear from Table 2.3 that both the linear programming and the solutions found in the adaptive platform are identical, meaning that at any production rate under normal conditions, both solutions are optimal. The increase in production time for lots 1 and 2 in stage 1 is shown in the data expressed in Table 2.4.

The multiagent simulation results for different production times are identical in both cases (PM and PMAS). However, the production planning obtained under normal conditions is affected by increasing its value by approximately 67% for the 400% increase in processing time. Since this stage is sequential, all processes are delayed; therefore, the agents' adaptation does not influence the completion time.

Table 2.4. Increased production time in stage 1 (times in hours)

Demand	TP stage increase 1 100%		TP stage increase 1 200%		TP stage increase 1 400%	
	Pm	PMAS	Pm	PMAS	Pm	PMAS
Demand1	31	31	45	45	59	59
Demand2	33	33	48	48	63	63
Demand3	35	35	51	51	67	67
Demand4	37	37	54	54	71	71
Demand5	39	39	57	57	75	75
Demand6	41	41	60	60	79	79
Demand7	43	43	63	63	83	83
Demand8	45	45	66	66	87	87
Demand9	48	48	69	69	91	91

The results for increasing production time for lots 1 and 2 in stage 2 are shown in Table2. 5.

Table 2.5. Increased production time in stage 2 (times in hours)

Demand	TP stage increase 2 100%		TP stage increase 2 200%		TP stage increase 2 400%	
	Pm	PMAS	Pm	PMAS	Pm	PMAS
Demand1	35	34	69	66	137	130
Demand2	39	37	77	71	153	139
Demand3	41	40	81	78	161	154
Demand4	47	45	93	89	185	177
Demand5	51	47	101	91	201	179
Demand6	53	51	103	100	203	196
Demand7	56	55	110	109	218	217
Demand8	59	57	113	111	221	219
Demand9	66	66	130	130	258	258

The results of this simulation demonstrate the importance of flexibility in production scheduling. This is because the completion times obtained from the programming given by the mathematical model are greater than those obtained by the simulation platform. Production planning given by exact models such as PM only guarantees optimality in typical or theoretical situations, while the model generated in the simulation platform can adapt to environmental conditions and modify the initial planning. When this modification is made online as designed in an IoT architecture, that is, as disturbances occur, agents can react and modify their sequencing.

While completion times increase considerably compared to the data found under normal conditions, this model improves completion times by adapting to production conditions. Table 2.6 shows the percentage of damping results for end time in each instance. There is an average damping of 3.92%, with a max. of 10.95% and a min. of 0.

Table 2.6. Increased production time in stage 2 (times in hours)

Demand	TP increase in stage 2 100%	TP stage increase 2 200%	TP stage increase 2 400%
	Reduction (%)	Reduction (%)	Reduction (%)
Demand1	−2,86	−4,35	−5,11
Demand2	−5,13	−7,79	−9,15
Demand3	−2,44	−3,70	−4,35
Demand4	−4,26	−4,30	−4,32
Demand5	−7,84	−9,90	−10,95
Demand6	−3,77	−2,91	−3,45
Demand7	−1,79	−0,91	−0,46
Demand8	−3,39	−1,77	−0,90
Demand9	0	0	0

2.6 Conclusion

This article's objective was to analyze the need of flexibility in production systems and deliver alternatives to existing static models. The proposed platform incorporates two distributed anarchic structures, in an intelligent product approach. This provides a tremendous advantage for production planning systems because human intervention is not needed to change the planning. More interestingly, the production lots reorganize and communicate themselves to achieve the common goal.

Compared to the lot-streaming technique, the platform achieved the same results for both initial conditions and disturbances within the sequential machines. This shows us that the developed platform can identify the optimal solution for all instances of the batch defined. We note that, for sustained increases in production times of 100%, 200%, and 400%, long completion times up to 88% are obtained, where alternatives such as the presented platform would deliver a reduction of up to 10.95% in completion times.

In future research, we will consider larger problems on a real scale. Additionally, the incorporation of new disturbances will be considered when analyzing their impacts on production and the effect that the architecture, as the one presented here, would have.

References

Ahmadi-Darani, M. H., Moslehi, G., & Reisi-Nafchi, M. (2018). A two-agent scheduling problem in a two-machine flowshop. International Journal of Industrial Engineering Computations, 9(3), 289–306. https://doi.org/10.5267/j.ijiec.2017.8.005

Choi, S. H., & Wang, K. (2012). Flexible flow shop scheduling with stochastic processing times: A decomposition-based approach. Computers and Industrial Engineering, 63(2), 362–373. https://doi.org/10.1016/j.cie.2012.04.001

Demirel, E., Özelkan, E. C., & Lim, C. (2018). Aggregate planning with Flexibility Requirements Profile. International Journal of Production Economics, 202, 45–58. https://doi.org/10.1016/j.ijpe.2018.05.001

Fan, B., & Cheng, T. (2016). Two-agent scheduling in a flowshop. European Journal of Operational Research, 252(2), 376–384. https://doi.org/http://dx.doi.org/10.1016/j.ejor.2016.01.009 0377-2217

Gharaei, A., & Jolai, F. (2018). A multi-agent approach to the integrated production scheduling and distribution problem in multi-factory supply chain. Applied Soft Computing Journal, 65, 577–589. https://doi.org/10.1016/j.asoc.2018.02.002

Gu, J., Gu, M., Cao, C., & Gu, X. (2010). A novel competitive co-evolutionary quantum genetic algorithm for stochastic job shop scheduling problem. Computers and Operations Research, 37(5), 927–937. https://doi.org/10.1016/j.cor.2009.07.002

Herrera, C. (2011). Cadre générique de planification logistique dans un contexte de décisions centralisées et distribuées. Université Henry Poincaré - Nancy . Francais.

Kruger, K., & Basson, A. H. (2019). Evaluation criteria for holonic control implementations in manufacturing systems. International Journal of Computer Integrated Manufacturing, 32(2), 148–158. https://doi.org/10.1080/0951192X.2018.1550674

Kumar, S., Bagchi, T. P., & Sriskandarajah, C. (2000). Lot streaming and scheduling heuristics for m-machine no-wait flowshops. Computers and Industrial Engineering, 38(1), 149–172. https://doi.org/10.1016/S0360-8352(00)00035-8

Lee, K. K. (2008). Fuzzy rule generation for adaptive scheduling in a dynamic manufacturing environment. Applied Soft Computing Journal, 8(4), 1295–1304. https://doi.org/10.1016/j.asoc.2007.11.005

Leitão, P., Rodrigues, N., Barbosa, J., Turrin, C., & Pagani, A. (2015). Intelligent products: The grace experience. Control Engineering Practice, 42, 95–105. https://doi.org/10.1016/j.conengprac.2015.05.001

Ma, A., Nassehi, A., & Snider, C. (2019). Anarchic manufacturing. International Journal of Production Research, 57(8), 2514–2530. https://doi.org/10.1080/00207543.2018.1521534

Mcfarlane, d., sarma, s., chirn, j. L., wong, c. Y., & ashton, k. (2002). The intelligent product in manufacturing control and management. Ifac Proceedings Volumes, 35(1), 49–54. https://doi.org/10.3182/20020721-6-ES-1901.00011

Potts, C. N., & Baker, K. R. (1989). Flow shop scheduling with lot streaming. Operations Research Letters, 8(6), 297–303. https://doi.org/10.1016/0167-6377(89)90013-8

Raileanu, Silviu. (2010). Production scheduling in a holonic manufacturing system using the open-control concept. U.p.b. Sci Bull Series C, 72(Chircu), 1–4. https://doi.org/ISSN 1454-234

Rolón, M., & Martínez, E. (2012). Agent-based modeling and simulation of an autonomic manufacturing execution system. Computers in Industry, 63(1), 53–78. https://doi.org/10.1016/j.compind.2011.10.005

Shahzad, A., & Mebarki, N. (2016). Learning Dispatching Rules for Scheduling: A Synergistic View Comprising Decision Trees, Tabu Search and Simulation. Computers, 5(3), 16. https://doi.org/10.3390/computers5010003

Slack, N. (1983). Flexibility as a Manufacturing Objective. International Journal of Operations Management, 3(3), 4–13. https://doi.org/https://doi.org/10.1108/eb054696

Stecke, K. E., & Solberg, J. J. (1981). Loading and control Problem for a Flexible

Manufacturing System. International Journal of Production Research, 19, 481–490.

Topaloglu, S., & Kilincli, G. (2009). A modified shifting bottleneck heuristic for the reentrant job shop scheduling problem with makespan minimization. International Journal of Advanced Manufacturing Technology, 44(7–8), 781–794. https://doi.org/10.1007/s00170-008-1881

Trietsch, D., & Baker, K. R. (2008). Basic Techniques for Lot Streaming. Operations Research, 41(6), 1065–1076. https://doi.org/10.1287/opre.41.6.1065

Tseng, C. T., & Liao, C. J. (2008). A discrete particle swarm optimization for lot-streaming flowshop scheduling problem. European Journal of Operational Research, 191(2), 360–373. https://doi.org/10.1016/j.ejor.2007.08.030

Wang, J., Zhang, Y., Liu, Y., & Wu, N. (2018). Multi-agent and Bargaining-game-based Real-time Scheduling for Internet of Things-enabled Flexible Job Shop. IEEE Internet of Things Journal, PP(c), 1. https://doi.org/10.1109/JIOT.2018.2871346

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling Northwestern University Evanston.

Wong, C. Y., Mcfarlane, D., Zaharudin, A. A., & Agarwal, V. (2014). The intelligent product driven supply chain. November 2002. https://doi.org/10.1109/ICSMC.2002.1173319

Wu, C.-C., Chen, J.-Y., Lin, W.-C., Lai, K., Bai, D., & Lai, S.-Y. (2019). A two-stage three-machine assembly scheduling flowshop problem with both two-agent and learning phenomenon. Computers and Industrial Engineering, 130, 485–499. https://doi.org/10.1016/j.cie.2019.02.047

Yadav, A., & Jayswal, S. C. (2018). Modelling of flexible manufacturing system: a review. International Journal of Production Research, 56(7), 2464–2487. https://doi.org/10.1080/00207543.2017.1387302

Yu, F., Wen, P., & Yi, S. (2018). A multi-agent scheduling problem for two identical parallel machines to minimize total tardiness time and makespan. Advances in Mechanical Engineering, 10(2), 1–14. https://doi.org/10.1177/1687814018756103

Zhou, Y., Yang, J. J., & Zheng, L. Y. (2019). Multi-Agent Based Hyper-Heuristics for Multi-Objective Flexible Job Shop Scheduling: A Case Study in an Aero-Engine Blade Manufacturing Plant. IEEE Access, 7, 21147–21176. https://doi.org/10.1109/ACCESS.2019.2897603

CAPÍTULO 3.

A product-driven system approach to generate fast solutions to the job shop scheduling problem.

Cita: P. Sáez, C. Herrera, J.E. Pezoa, A product-driven system approach to generate fast solutions to the job shop scheduling problem., IFACPapersOnLine, Volume 55, Issue 10,2022, Pages 1930-1937, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2022.09.681.

Abstract: Many optimal algorithms, heuristics, metaheuristics, simulation approaches, agent-based models, and machine learning tools attempt to solve the job shop scheduling problem (JSSP). This article proposed a model of artificial intelligence with agents representing intelligent products from the perspective of product-driven systems (PDS) to solve this problem at different scales. The intelligent products make all decisions in a distributed way aiming to minimize the makespan and increase the computational efficiency for the JSSP. The agents embed the intelligence function using a based shifting bottleneck heuristic (SBH) approach. The novelty of the proposed approach lies in the automation of decisions in a highly distributed architecture to increase manufacturing flexibility. The results are compared with an optimal integer programming model (IP), SBH, and two conventional heuristics considering instances commonly used in the literature. Concerning the makespan, the proposed approach obtains a fast solution near optimal in instances with a low number of resources and better results than IP and conventional heuristic in instances with a more significant number of resources, increasing the response capacity with a similar computational time.

Keywords: Agent-based models, job-shop, sequencing, shifting bottleneck heuristic, intelligent products, product-driven system.

3.1 Introduction

In modern production, procedures are increasingly flexible in the face of internal or external disturbances. This process has caused the design of new techniques (such as intelligent agents) to adaptively control or hybridize methodologies to solve specific problems (Adam et al., 2010).

The assignment of tasks and processes to be performed by different machines and on different jobs is called a production scheduling problem. This assignment is one of the most challenging tasks that companies face today and has been the focus of extensive research (Fuchigami & Rangel, 2018). Different methods and approaches have been tested and applied to the problem of production scheduling to solve problems. However, the time difference between the beginning and the completion of a series of tasks or makespan represents one of the most important objectives because it is directly related to customer satisfaction performance indicators.

The job-shop scheduling problem (JSSP) is considered an NP- hard problem (Asadzadeh, 2015) and has been extensively studied (Fuchigami & Rangel, 2018). In the literature, there are exact methods, such as integer programming and branch and bound, to solve the JSSP. However, the high computational load exponentially increases with the size of the problem (Nowicki & Smutnicki, 2005). For real industrial problems, the computational time of a given algorithm or method should not be too long for practical use. It has been decided to use a wide variety of heuristic procedures in the industry, which provide good results in a reasonable amount of time (Bożek & Werner, 2018).

Decomposition-based heuristics and metaheuristics such as the bottleneck algorithm (Adams et al., 1988); Local search algorithms such as taboo search (Bożek & Werner, 2018) and Simulated Annealing (Monostori et al., 2006), are used for large-scale problem cases, trying to provide flexibility in the manufacturing processes. These heuristics develops solutions for complex problems by breaking a problem into a series of smaller subproblems, which are more manageable and easier to solve. One of the most used decomposition heuristics is the shifting bottleneck heuristic (SBH), which was proposed by Adams et al. (1988). This algorithm decomposes the JSSP into subproblems that iteratively program a single machine. According to Ovacik and Uzsoy (1992), a decomposition method has better results than dispatch rules in both its average and the worst case.

The intelligent product is the representation of an order or physical product linked to the information and the rules that govern its manufacture, storage, or transport, allowing it to influence operations (Wong et al., 2014). The use of intelligent products brings essential benefits to a product-driven production approach and the JSSP (Herrera et al. 2014; Herrera et al. 2016). In this sense, it has been used to improve the entire life cycle of products, i.e., design, production, distribution, operation, and disposal phases. Moreover, it improves the quality and performance of the product by applying self-learning, self-diagnosis, self-adaptation, and self-optimization methods (Leitão et al., 2015; Barata & da Cunha, 2019).

This article presents the JSSP and how to solve it through an artificial intelligence-based approach that uses intelligent products that make scheduling decisions in instances with a different number of resources (machines and products). The proposed approach uses an agent-based model (ABM) to implement a product-driven system (PDS) with SBH as the embedded intelligence function for decision-making. For the experimentation, we compare the results obtained in 60 seconds of execution (fast solution), with the results of an optimal integer programming model, SBH, and two heuristic methods, considering the makespan and the computational effort as performance measures. Thus, the main contribution of this work is to propose a methodology based on intelligent products represented by agents that communicate with each other to obtain near-optimal JSSP makespan in an efficient and decentralized way.

The article develops as follows: Section 2 shows a literature review of JSSP, SBH, ABM and Intelligent products. Section 3 explains the proposed approach and shows the design of the computational experiments. Section 4 shows the main results of the proposed approach by considering two instances from the literature. Section 5 discusses the results. Finally, section 6 ex-presses the conclusion.

3.2. Literature review

3.2.1 Intelligent manufacturing for JSSP

The problem of production scheduling corresponds to allocating tasks on available machinery over time. Commonly, the term scheduling in production systems refers to the sequencing of operations that must ensure compliance with a series of constraints established in the process and the optimization of some performance measure of interest.

The physical distribution of available resources in the manufacturing system must be known to solve this kind of problems. It is also necessary to know the required manufacturing process for each product and define the flow that each process must follow through the system. Finally, the capacity is also essential, i.e., the number of products that each resource must be able to process, which is limited and specified (Fan & Cheng, 2016).

3.2.2 Job-shop scheduling problem

The JSSP is described as a set of n P_i jobs, with $i = 1, \dots, n$, which must be processed in a set of m machines Mk with $k = 1, \dots, m$. Each P_i job has a sequence of j operations executed in a specific order through a process that occurs during an uninterrupted period (Yu et al., 2015). Each operation has a processing time T_{ik}, the processing time of product P_i on machine M_k. Figure 3.1 shows an example diagram of the JSSP with m = 4 and n = 3.

This kind of system must comply with the following constraints:

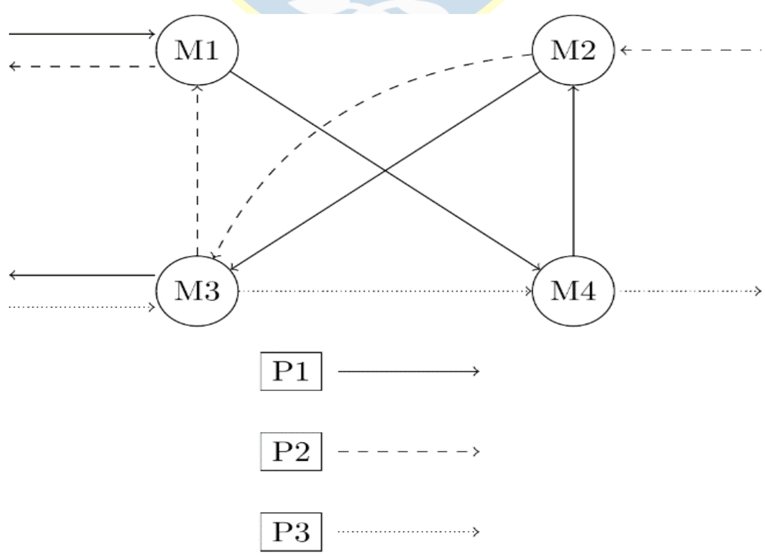

Figure 3.1. Block diagram of a job-shop type of manufacturing system with 4 machines and 3 products

- A job cannot visit the same machine twice.
- Each machine can process only one job at a time.
- Each job can be processed by one machine at a time.
- Jobs Pi must comply with the rules of precedence of its sequence.

A schedule determines the sequence of execution of all operations for all jobs on the machines. The result of this scheduling is represented on a Gantt chart. Thus, there is a global concern to improve the performance of the job-shop systems, improving the efficiency of various operations in the production system.

3.2.3 Integer programming application

To solve the job-shop scheduling problem in an optimal way, integer programming was used. An integer programming model from a disjunctive network is implemented, as explained by Mason (2002). This formulation defines times ST_{ij} as the start time for operations O_{ij} (product i on machine j). Also, there are a set OPS of all operations and set C of all edges of precedence; $O_{ij} < O_{ij} + 1$; m_{ij} refers to the machine on which operations O_{ij} are executed. Generalizing yields, the following mathematical model.

$$\min C_{max} \tag{1}$$

s.t.

$$\mathrm{ST_{ih}} - \mathrm{ST_{ij}} \geq \mathrm{p_{ij}}, \forall\, (ij)(ih) \;\; with\; h > j \tag{2}$$

$$\mathrm{C_{max}} - \mathrm{ST_{ij}} \geq \mathrm{p_{ij}}, \forall\, \mathrm{(ij)} \;\in \mathrm{OPS} \tag{3}$$

$$\mathrm{ST_{ij}} - \mathrm{ST_{rs}} \geq \mathrm{p_{rs}} \text{ or } \mathrm{ST_{rs}} - \mathrm{ST_{ij}} \geq \mathrm{p_{ij}}\,, \forall\, \mathrm{(ij)(rs)} \;\in \mathrm{OPS \text{ with } m_{ij}} = \mathrm{m_{rs}} \tag{4}$$

$$\mathrm{ST_{ij}} \geq 0 \tag{5}$$

In this model, (1) is the function objective; (2) and (3) are the precedence constraints for jobs and competition time, respectively; (4) is the disjunction constraint for the jobs and (5) is a non-negativity constraint.

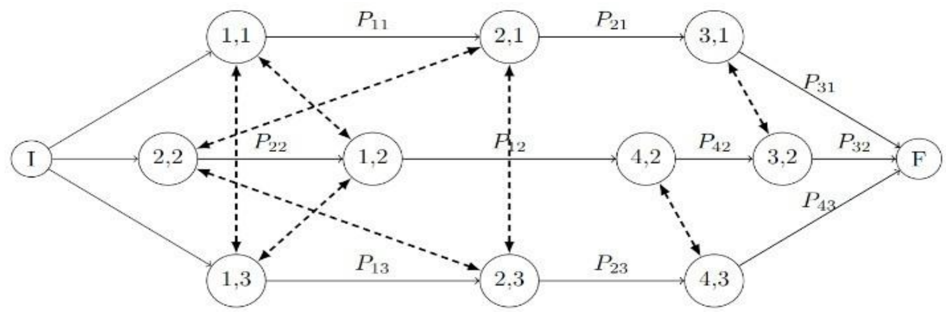

Figure 3.2. Disjunctive graph for the JSSP. Source: (Mason et al.,2002)

The main difficulty in solving the model is in the third set of constraints, where the conditional "or" (disjunction) does not enable the evaluation using typical solution techniques.

3.2.4 SBH

SBH was originally proposed by Adams et. al (1988) as a powerful decomposition method for the JSSP problem aimed at minimizing delay. This method uses the advantages of the disjunctive network introduced by B. Roy and B. Sussmann (1964) to model the interactions between subproblems generated in the decomposition. A network G considers the JSSP of type nxm (n jobs sequenced in m machines) and generates N-type nodes (operations) and two different sets of arcs A and B, defining the graph G = (N, A, B), see Figure 3.2. This Figure appears in In S. Topaloglu and G. Kilincli (2009), to show an example of a disjunctive network with three jobs and four machines. In this example, job 2 has sequence 2,1,4,3,4 (conjunctive), while the groups of nodes {(4,2a), (4,3), (4,2b), (4,3)} and {(2,2), (2,1), (2,3)} are connected by disjunctive arcs.

Each N node denoted as $N\ (i, k)$, represents the operation of job i on machine k in the network (except virtual nodes with zero process times at the start and end of the network). Conjunctive arcs A represent the jobs routes and correspond to the precedence constraints. If there is an arc from node (i, k) to node (j, k), operation (i, k) is precedent to operation (j, k). Disjunctive arcs B correspond to the resource constraints. Thus, they are undirected arcs that connect the operations executed on the same machine. Each JSSP operation can start when it has finished the execution of its predecessor operations (if any).

To complete the SBH, the following steps are followed:

- Identify the available machines and establish an initial set M0.
- Identify and solve the subproblems of each machine i of set M-M0.
- Identify critical machine k of set M-M0.
- Sequence the critical machine using the solution in step 2 by arranging the disjunctive arcs associated with the critical machine in the appropriate direction. We set M0:M0 U $\{k\}$.
- (Optional) Reoptimize the sequence for each machine m of the M0-k set by exploring information.
- If M0=M, stop.

Applegate and Cook (1991) show that the quality of the obtained solutions is directly affected by the disaggregation of the main problem and how the generated subproblems are solved. Recently, the SBH has been studied with other performance measures such as the total delay (Sahin et al., 2013), maximum delay (Lin & Uzsoy, 2016) and number of late jobs (Yadav & Jayswal, 2018). In addition, the SBH has been worked in conjunction with other heuristics such as genetic algorithms (Mönch et al., 2007) and simulations (Mönch & Zimmermann, 2011).

3.2.5 Agent-Based Modelling

ABM is a modeling approach that describes a complex system as a set of autonomous decision-making entities termed as agents. The ABM essentially reproduces a community of autonomous agents. Through their interactions, ABM simulates the appearance of collective behavior phenomena from the behavior of individual agents. The agents are entities with different levels of intelligence, whose construction is not exempt from problems. In the architecture perspective, we talk about a symbolic physical system that can generate an intelligent action from a system of physical symbols to a processing automaton. This capacity to generate intelligent actions has the complication of translating a real-world description into a symbolic description by representing the information captured in entities to make the agents reason (Shukla et al., 2019).

A particular class of ABMs uses different classes of agents, each with specific roles, called Multi-Agent Systems (MAS). In Wang (2018), a program of a JSSP system (flexible type) is generated in real-time based on a MAS. In turn, in Guizzi (2019), a system of the interactions between two types of agents is proposed: job and resource agents.

3.2.6 Intelligent products

In recent years, the industry's trend has been to incorporate technologies and AI, such as the use of robots, automated production programming, autonomous operation systems, etc., (Shukla et al., 2019). To incorporate these new technologies, detailed processes knowledge is necessary to generate a system, whose components are intelligent products (Meyer et al., 2011; Mcfarlane, 2012).

The concept of an intelligent product for many is simply the tangible physical entity, which can be part of a core product, an actual product (tangible physical product), or an augmented product (nonphysical part containing product information) (Wu et al., 2019). Another critical point is to specify the level of intelligence possessed by the product in question. To measure this characteristic, G. Meyer (2009) provides a three-dimensional orthogonal frame to classify the intelligence levels, as shown in Figure 3.3.

In the work of I. Kovalenko et al. (2019), multi-agent control strategies are used (product agents, which follow the definitions and concepts established by intelligent products), where the flexibility of complex and dynamic manufacturing systems is improved, corroborating the benefit of an Intelligent Product immersed in the industry.
The characteristics of the proposed agent-product are:

- It has a unique identity.
- It can communicate both with other product agents and with machine agents.
- It has memory capacity according to the objective function delivered
- Set your own production requirements by making important decisions for your destination.

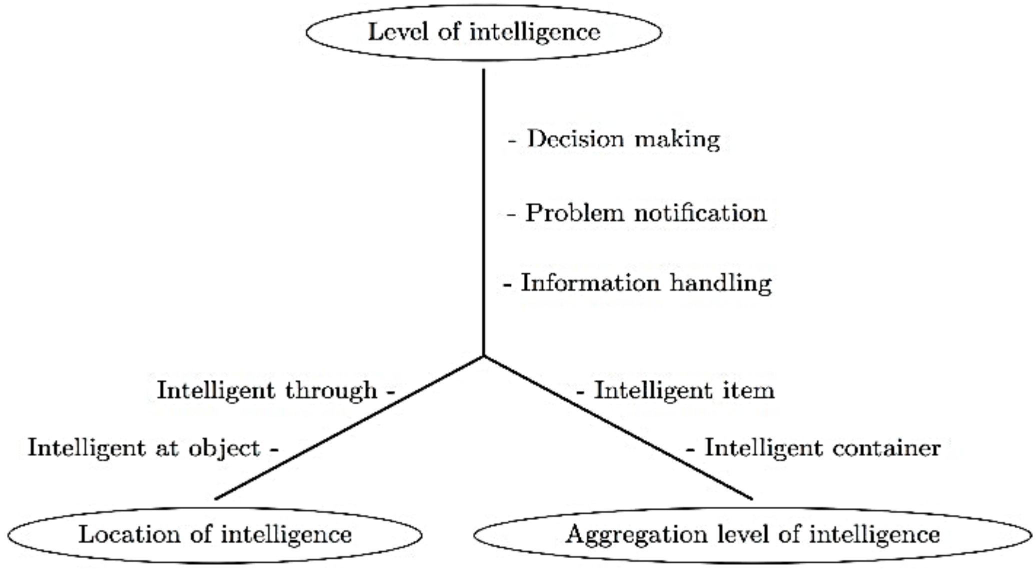

Figure 3.3. Classification model of an intelligent product. Source: (Meyer et al., 2009).

In addition, in the explored context, he has knowledge of the real world with information processing, actively communicating with the different actors of the system.

3.2.7 The product-driven approach

The goal of product-driven approach is to synchronize material flows, and information flows in the manufacturing environment. On the one hand, the physical product interacts with other physical entities; on the other hand, its digital part interacts with the environment to control and manage the production. Shahzad and Mebarki (2016) ed on JSSP and developed an algorithm, based on intelligent agents, using an ant colony approach to minimize the makespan. However, it does not address the complete case where each product is intelligent and controls its own process. In Gharaei & Jolai (2018) work, a MAS is used in production programming. The authors incorporate different roles for the agents in the model, e.g., machine agents, monitoring agents, and evaluation agents. The main reason for its use is to add the necessary flexibility to address changes in production conditions (Sáez & Herrera, 2021). However, the decisions are not entirely decentralized (due to the presence of evaluation agents) nor focused on the product as in our proposal.

Although studies address this problem using multi-agent models (Jarvis et al., 2018; Kovalenko et al., 2019; Leitão et al., 2015), they present hierarchical or heterarchical architectures with process controllers. In our case, the proposed approach generates a machine allocation strategy based on a complete decentralized decision-making process, reusing this information to carry out a learning process for future generations of solutions. Furthermore, this perspective allows each agent to make decisions relevant to their destination, following a type of architecture as indicated in the work of A. Ma (Ma et al., 2019), guaranteeing the complete distribution of the decision. Therefore, the novelty of our proposal is the use of the intelligent product paradigm in a product-driven system under a distributed architecture to obtain better performance in reduced computing times.

In recent works W. Bouazza et al., (2021), implemented a generic system model controlled by Intelligent Products to generate a decisional strategy. That allows an efficient change of a programming rule to another using a novel approach based on Hyper-Heuristics (HH).

The HH unite decision strategies with an optimization- simulation mechanism. This methodology can be interesting under certain conditions, where combining different rules at the appropriate time improves the manufacturing system performance overall and reactivity. The work of J. Campos et al., (2020) provides a specific solution for the dynamic programming of product-driven production with a design based on MAS. They implemented this model in a flexible hybrid flow with multiple constraints inspired by the pharmaceutical industry. The designed model has simple agents that behave under condition-action rules. However, these agents are limited in their actions and knowledge. These works differ from ours in the type of application problem. Our proposal focuses on a JSSP with agents that incorporate an embedded intelligence function (SBH heuristics). Each agent decides in a distributed manner, analyzing the makespan based on the response speed.

3.3. Experimentation

3.3.1 Application of the SBH methodology

Now, we consider the steps proposed in S. Topaloglu and G. Kilincli (2009) to implement the SBH methodology. The makespan is defined as the length of the longest path from the start node to the end node. This path is formed by a set of operations that starts at time 0 and ends at a time equal to the makespan. The direction of each undirected edge (corresponding to the resource relations) is executed in a plan, so that an operation starts immediately when its predecessor operations have finished. A start time ST_{ij} is defined for an O_{ij} operation, with an O_{kl} predecessor operation and with $TF_{ij} = ST_{ij} + p_{ij}$.

The latest start time ST_{ij} of an operation is defined as the maximum start time of an operation, without causing an increase in makespan. The latest end time TF'_{ij} (with $TF'_{ij} = ST'_{ij} + p_{ij}$) is defined as the maximum end time of an operation without causing an increase in makespan.

Two types of operations are distinguished: those that comply with $ST_{ij} = ST'_{ij}$ and those that satisfy $ST_{ij} < ST'_{ij}$. The operations that fulfill the first condition are critical because a delay in time would increase the makespan. Those that fulfill the second condition are noncritical since a delay in their time of no more than $ST'_{ij} - ST_{ij}$, will not change the makespan.

3.3.2 Application of product-driven system with ABM methodology

The proposed approach is implemented from the perspective of an intelligent product. This approach has two types of agents: products and machines. For each family of agents, the possible states for each agent are:

- Agent-product: free, in process, and finished
- Agent-machine: available and in process

An agent-product can move from the free state to being in process and subsequently to being finished, while a machine can pass from an available state to be in process and afterwards become available again. The agent-machines are considered static entities that only provide information about the processing time of their tasks. The agent-products comply with the characteristics of an intelligent product, i.e., they have a unique identification, they can communicate with the machines around them and among other agent-products, and they take the operation decisions or wait at an agent machine.

Version 6.1.1 of the NetLogo simulation platform is used for modeling (Wilensky, 1999). Such platform provides a suitable environment to test and monitor the performance of the model. The proposed approach uses the parallelism in ABM models, where all simultaneously execute each instruction given to the agents. Figure 3.4 shows the flow chart for the decision-making process for all agent-products. These decisions follow the process described for the SBH methodology, generating initial conditions, and solving subproblems.

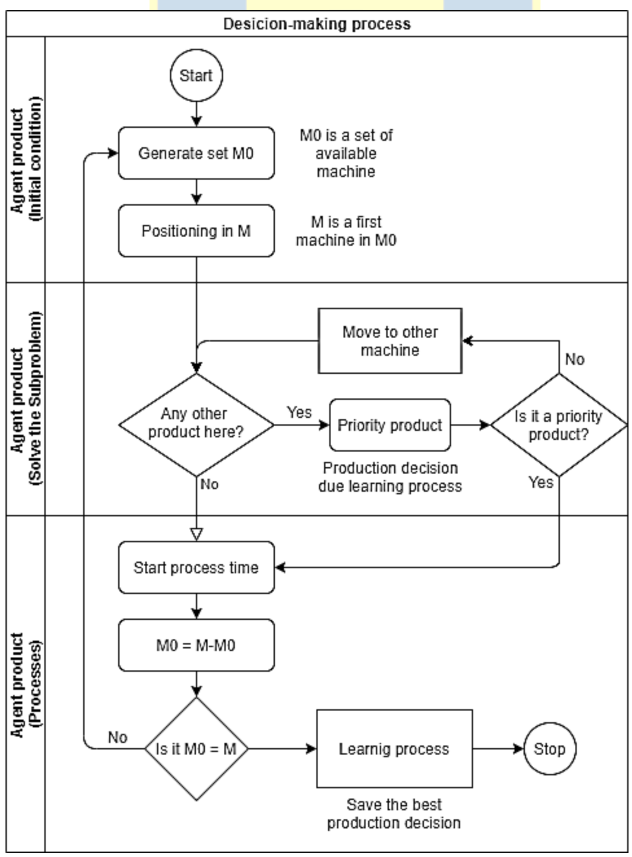

Figure 3.4. Flow chart for the intelligent product decision making process.

There are three stages in the decision-making process. In the first stage, the production sequence is stored in each agent- product, positioning it in the first machine of its sequence. Then, we continue with the resolution of the subproblems (when two agents-products need the same machine at the same time instant). In the final stage, all the agents-products have completed their work and generated the learning process. In the learning process, each agent-product saves the position of the machines in its production sequence and the production decisions of the sub-problems for use in future generations.

3.3.3 Experimental design

The instances used were presented in Adams et al. (1988) and consist of 15 different problems named abz5, abz6, abz7, abz8, and abz9 (instances with low number of resources), and swv11, swv12, swv13, swv14, swv15, swv16, swv17, swv18, swv19, and swv20 (instances with a larger number of resources). The instances are generated with random processing times in different intervals of uniform distributions. The characterization is ten jobs on ten machines (for the instances abz5 and abz6); 20 jobs on 15 machines (for the instances abz7, abz8, and abz9), and 50 jobs on ten machines (for the instances swv11, …, swv20) with different process times.

The integer programming model was implemented and solved with CPLEX under NEOS Server services. The server used is a Dell PowerEdge R410 with the following configuration: CPU - 2x Intel Xeon X5660 @ 2.8GHz (12 cores total), HT Enabled; Memory - 64GB RAM; Disk - 2x 500GB/2TB SATA drives setup in RAID1; Network - 1Gb/s Ethernet.

The proposed approach with a product-driven system (PDS) was implemented in version 6.2 of the Netlogo platform. The heuristics and proposal approach were implemented in a processor AMD Ryzen 5 3550H @ 2.1 GHz (8 cores in total); memory – 12GB RAM.

3.4. Results

The results of the performance of the proposed approach are compared with the different

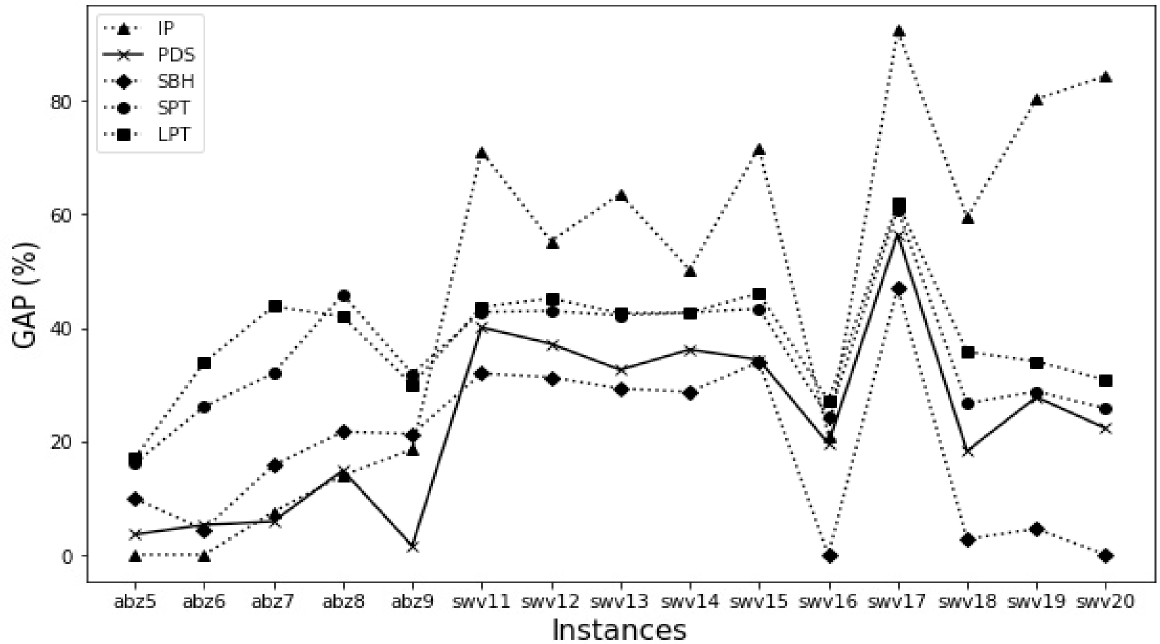

Figure 3.5. GAP obtained for the models for each instance.

methods mentioned above. In addition, we use the best results obtained in 60 seconds to generate a proper comparison and guarantee agility and flexibility in decision making. Optimal results are used as a baseline measure and are obtained from the literature.

3.5. Discussion and analysis of results

The results obtained by integer programming model (IP), PDS, and SBH are the closest to the optimum in instances with a low number of resources. The results show that computational times in the proposed approach are considerably less than IP in instances with a more significant number of resources. These computational times are closer to those obtained by heuristics such as SBH. However, the performance margin obtained by PDS over SBH is higher in 4 of 5 instances (abz5, abz7, abz8 and abz9). In the other cases, SBH obtains better results leaving PDS in second place. This new perspective (with a fast solution) integrates the possibility of making the process more flexible, responding with good results regardless of the number of resources.

The SPT and LPT heuristics generate good results according to the computational time required to complete the process. However, completion times are longer than all used models.

Figure 3.5 shows the GAP obtained by each model in each instance studied. In the case of IP, instances abz5 and abz6 had an optimal resolution. However, for instances abz7, abz8, abz9, and all instances svw, the proposed model obtains better results than IP in 60 seconds. Remarkably, the results for the proposed approach in instances with a low number of resources (abz7, abz8, and abz9) are better in short periods than all the methodologies used. However, for the instance, abz5 and abz6, the IP and SBH (exclusively abz6) heuristic were superior to the proposed approach. For the instances with a larger number of resources, the proposed approach obtained better results than IP, SPT, and LPT in 60 seconds of execution. However, SBH obtained better results in this instance.

Computational evidence suggests that our model based on intelligent products delivers comparable results to other more popular methods with high computational complexity (Yu et al., 2015). We note that results agree with those found by other authors using these techniques (Abar et al., 2017; Senouci et al., 2019). Thus, ABM can solve sequencing processes in workshop environments with almost optimal values, making this method a flexible methodology to solve JSSPs with different amounts of resources.

3.6. Conclusion

The proposed approach used the product-driven system to solve the JSSP, and evaluate the makespan as a performance measure. The adopted methodology enables one to face the complexity in this type of problems by describing the dynamics of manufacturing processes in terms of an intelligent product. Our results are consistent with those of other researchers and argue that heuristics perform better than sequencing rules. Thus, procedures such as integer programming obtain the best results. However, the IP result depends on the formulation of the mathematical model.

Based on the implemented methodology, it was possible to prove that an agent model with decisions made by intelligent products can address production scheduling problems and provide greater flexibility. Better results are obtained in short periods than tools highly used in the literature, allowing adaptation to unforeseen changes. The design of this methodology allows interactions between agents that provide a detailed description of a holonic dynamic.

The results concerning makespan are near to the optimal in instances with a low number of resources and obtained better results than IP and conventional heuristic in instances with more resources. The experimental evidence shows that the modeled intelligence directly affects the ability to obtain good results with fewer replications. However, the preliminary results encourage further exploration of this line of research due to the versatility of obtaining production plans without human intervention and reduced computational time.

Future research will be necessary to generate agents with improved communication capacity and implement more realistic events in planning, such as work cells, simultaneous products, and failures. In addition, it will include the implementation of a more sophisticated simulation with agents generated under the logic of intelligent products to minimize the nervousness of the system.

References

Adam, E., Berger, T., Sallez, Y., Trentesaux, D., & Mandiau, R. (2010). Robustness in adaptative holonic multiagent systems: The open-control concept. IFAC Proceedings Volumes (IFAC-PapersOnline), 10(PART 1), 114–119. https://doi.org/10.3182/20100701-2-PT-4011.00021

Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. Mana, 34(3), 391–400. https://www.jstor.org/stable/2632051

Applegate, D., & Cook, W. (1991). A computational study of job shop problem. ORSA Journal on Computing, 3, 149– 159. https://doi.org/10.1287/ijoc.3.2.149

Asadzadeh, L. (2015). A local search genetic algorithm for the job shop scheduling problem with intelligent agents. Computers and Industrial Engineering, 85, 376–383. https://doi.org/10.1016/j.cie.2015.04.006

Barata J., da Cunha P.R. (2019) In Search for a Viable Smart Product Model. In: Abramowicz W., Corchuelo R. (eds) Business Information Systems Workshops. BIS 2019. Lecture Notes in Business Information Processing, vol

373. Springer, Cham. https://doi.org/10.1007/978-3- 030-36691-9_9

Bouazza, W., Sallez, Y., & Trentesaux, D. (2021). Dynamic scheduling of manufacturing systems: a product-driven approach using hyper-heuristics. International Journal of Computer Integrated Manufacturing, 34(6), 641–665. https://doi.org/10.1080/0951192X.2021.1925969

Bożek, A., & Werner, F. (2018). Flexible job shop scheduling with lot streaming and sublot size optimisation. International Journal of Production Research, 56(19), 6391–6411. https://doi.org/10.1080/00207543.2017.1346322

Campos, J. T. de G. A. e A., Blumelova, J., Lepikson, H. A., & Freires, F. G. M. (2020). Agent-based dynamic scheduling model for product-driven production. Brazilian Journal of Operations & Production Management, 17(4), 1–10. https://doi.org/10.14488/bjopm.2020.044

Fan, B., & Cheng, T. (2016). Two-agent scheduling in a flowshop. European Journal of Operational Research, 252(2), 376–384. https://doi.org/http://dx.doi.org/10.1016/j.ejor.2016.01. 009 0377-2217

Fuchigami, H. Y., & Rangel, S. (2018). A survey of case studies in production scheduling: Analysis and perspectives. Journal of Computational Science, 25, 425–436. https://doi.org/10.1016/j.jocs.2017.06.004

Gharaei, A., & Jolai, F. (2018). A multi-agent approach to the integrated production scheduling and distribution problem in multi-factory supply chain. Applied Soft Computing Journal, 65, 577–589. https://doi.org/10.1016/j.asoc.2018.02.002

Guizzi, G., Revetria, R., Vanacore, G., & Vespoli, S. (2019). On the open job-shop scheduling problem: a decentralized multi-agent approach for the manufacturing system performance optimization. Procedia CIRP, 79, 192–197. https://doi.org/10.1016/j.procir.2019.02.045

Herrera, C., Thomas, A., & Parada V., (2014) A product- driven system approach for multilevel decisions in manufacturing planning and control, Production & Manufacturing Research, 2:1, 756-766, DOI: 10.1080/21693277.2014.949895

Herrera, C., Berraf, S.B., Thomas, A., & Parada, V. (2016). A reactive decision-making approach to reduce instability in a master production schedule. International Journal of Production Re-search, 7543, 0.

https://doi.org/10.1080/00207543.2015.1078516

Jarvis, D., Jarvis, J., Kalachev, A., Zhabelova, G., & Vyatkin, V. (2018). PROSA/G: An architecture for agent-based manufacturing execution. IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2018-Septe, 155–160. https://doi.org/10.1109/ETFA.2018.8502598

Kovalenko, I., Tilbury, D., & Barton, K. (2019). The model- based product agent: A control oriented architecture for intelligent products in multi-agent manufacturing systems. Control Engineering Practice, 86(March), 105– 117. https://doi.org/10.1016/j.conengprac.2019.03.009

Leitão, P., Rodrigues, N., Barbosa, J., Turrin, C., & Pagani, A. (2015). Intelligent products: The grace experience. Control Engineering Practice, 42, 95–105. https://doi.org/10.1016/j.conengprac.2015.05.001

Lin, P. C., & Uzsoy, R. (2016). Chance-constrained formulations in rolling horizon production planning: an experimental study. International Journal of Production Research, 54(13), 3927–3942. https://doi.org/10.1080/00207543.2016.1165356

Ma, A., Nassehi, A., & Snider, C. (2019). Anarchic manufacturing. International Journal of Production Research, 57(8), 2514–2530. https://doi.org/10.1080/00207543.2018.1521534

Mason, S. J., Fowler, J. W., & Matthew Carlyle, W. (2002). A modified shifting bottleneck heuristic for minimizing total weighted tardiness in complex job shops. Journal of Scheduling, 5(3), 247–262. https://doi.org/10.1002/jos.102

Mcfarlane, D. (2012). Product Intelligence : Theory and Practice. In IFAC Proceedings Volumes (Vol. 45, Issue 6). IFAC. https://doi.org/10.3182/20120523-3-RO-2023.00441

Meyer, G. G., Hans Wortmann, J. C., & Szirbik, N. B. (2011). Pro-duction monitoring and control with intelligent products. Inter-national Journal of Production Research, 49(5), 1303–1317. https://doi.org/10.1080/00207543.2010.518742

Meyer, G. G., Holmstro, J., & Fra, K. (2009). Computers in Industry Intelligent Products : A survey. 60, 137–148. https://doi.org/10.1016/j.compind.2008.12.005

Mönch, L., Schabacker, R., Pabst, D., & Fowler, J. W. (2007). Genetic algorithm-based subproblem solution procedures for a modified shifting bottleneck heuristic for complex job shops. European Journal of Operational Research, 177(3), 2100–2118. https://doi.org/10.1016/j.ejor.2005.12.020

Mönch, L., & Zimmermann, J. (2011). A computational study of a shifting bottleneck heuristic for multi-product complex job shops. Production Planning and Control, 22(1), 25–40. https://doi.org/10.1080/09537287.2010.490015

Monostori, L., Váncza, J., & Kumara, S. R. T. (2006). Agent- based systems for manufacturing. CIRP Annals - Manufacturing Technology, 55(2), 697–720. https://doi.org/10.1016/j.cirp.2006.10.004

Nowicki, E., & Smutnicki, C. (2005). An advanced tabu search algorithm for the job shop problem. Journal of Scheduling, 8(2), 145–159. https://doi.org/10.1007/s10951-005-6364-5

Ovacik, I., & Uzsoy, R. (1992). A shifting bottleneck algorithm for scheduling semiconductor testing operationes. Journal of Electronic Manufacturing, 2, 119–134. https://doi.org/10.1142/S0960313192000157

Roy, B., & Sussman, B. (1964). Les problémes d'ordonnancements avec contraintes disjonctives. Proceedings of SEMA, Montrouge.

Sáez, P., & Herrera, C. (2021). Implementation of a Holonic Product-Based Platform for Increased Flexibility in Production Planning. Studies in Computational Intelligence, 987, 180–194. https://doi.org/10.1007/978- 3-030-80906-5_12

Sahin, F., Narayanan, A., & Robinson, E. P. (2013). Rolling horizon planning in supply chains: Review, implications and directions for future research. International Journal of Production Re-search, 51(18), 5413–5436. https://doi.org/10.1080/00207543.2013.775523

Senouci, A., Warith, K. A., & Eldin, N. (2019). Resource- Constrained Construction Scheduling Using Agent Based Modeling Technique. Journal of Civil Engineering and Construction, 8(1), 25–33. https://doi.org/10.32732/jcec.2019.8.1.25

Shahzad, A., & Mebarki, N. (2016). Learning Dispatching Rules for Scheduling: A Synergistic View Comprising Decision Trees, Tabu Search and Simulation. Computers, 5(3), 16. https://doi.org/10.3390/computers5010003

Shukla, O. J., Soni, G., Kumar, R., Sujil, A., & Prakash, S. (2019). Harmony Search and Nature Inspired Optimization Algorithms (Vol. 741). Springer Singapore. https://doi.org/10.1007/978-981-13-0761-4

Topaloglu, S., & Kilincli, G. (2009). A modified shifting bottleneck heuristic for the reentrant job shop scheduling problem with makespan minimization. International Journal of Advanced Manufacturing Technology, 44(7– 8), 781–794. https://doi.org/10.1007/s00170-008-1881

Wang, J., Zhang, Y., Liu, Y., & Wu, N. (2018). Multi-agent and Bargaining-game-based Real-time Scheduling for Internet of Things-enabled Flexible Job Shop. IEEE Internet of Things Journal, PP(c), 1. https://doi.org/10.1109/JIOT.2018.2871346

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling Northwestern Universi-ty Evanston.

Wong, C. Y., Mcfarlane, D., Zaharudin, A. A., & Agarwal, V. (2014). The intelligent product driven supply chain. November 2002.
https://doi.org/10.1109/ICSMC.2002.1173319

Wu, C.-C., Chen, J.-Y., Lin, W.-C., Lai, K., Bai, D., & Lai, S.-Y. (2019). A two-stage three-machine assembly scheduling flowshop problem with both two-agent and learning phenomenon. Computers and Industrial Engineering, 130, 485–499. https://doi.org/10.1016/j.cie.2019.02.047

Yadav, A., & Jayswal, S. C. (2018). Evaluation of batching and layout on the performance of flexible manufacturing system. International Journal of Advanced Manufacturing Technology. https://doi.org/10.1007/s00170-018-2999-1

Yu, J.-M., Doh, H.-H., Kwon, Y.-J., Shin, J.-H., Kim, H.-W., Nam, S.-H., & Lee, D.-H. (2015). Decision Tree based Scheduling for Static and Dynamic Flexible Job Shops with Multiple Process Plans. Journal of the Korean Society for Precision Engineering, 32(1), 25–37. https://doi.org/10.7736/kspe.2015.32.1.25

CAPÍTULO 4.

A product-driven system with an evolutionary algorithm to increase flexibility in planning a job shop

Sáez P, Herrera C, Booth C, Belmokhtar-Berraf S, Parada V (2023) A product-driven system with an evolutionary algorithm to increase flexibility in planning a job shop. PLoS ONE 18(2): e0281807. https://doi.org/10.1371/journal.pone.0281807

Abstract: The scheduling of a job shop production system occurs using models to plan operations for a given period while minimizing the makespan. However, since the resulting mathematical models are computationally demanding, their implementation in the work environment is impractical, a difficulty that increases as the scale problem grows. An alternative approach is to address the problem in a decentralized manner, such that real-time product flow information feeds the control system to minimize the makespan dynamically. This paper presents a product-driven system model that includes an evolutionary algorithm to minimize the makespan of the job shop scheduling problem. A multiagent system simulates the model and produces comparative results for different problem scales with classical models. One hundred two job shop problem instances classified as small, medium, and large scale are evaluated. The results suggest that a product-controlled system produces near-optimal solutions in short periods and improves its performance as the scale of the problem increases. The results corroborate the advantage of using real-time information to optimize a production plan.

4.1 Introduction

The planning of production systems has benefited from information and communication technologies, allowing the emergence of models with the capacity for self-learning, self-diagnosis, self-adaptation, and self-optimization (Barbosa et al., 2015). Although there is a wide variety of production systems, job shop systems constitute a significant part, and their scheduling to optimize resources is a computational challenge (Pinedo, 2012). The job shop scheduling problem (JSSP) emerges from that situation as a problem belonging to the NP-hard class (Asadzadeh, 2015). Consequently, the computational requirement precludes its practical use in large-size situations. JSSP has been widely studied, visualizing a centralized and static approach. However, the integration of current technologies broadens the conception of production systems, giving rise to more sophisticated organizational ideas (Oborski & Wysocki, 2022). Thus, the JSSP can be approached in a decentralized and dynamic way, achieving that the flow of products during the operation contributes with real-time information that supports the decision-making of the production control system.

The development of intelligent systems makes it possible to dynamically meet the computational challenge offered by production systems. Such systems occur in situations of different sizes, making it difficult to use a single method for production scheduling (Bożek & Werner, 2018; Nowicki & Smutnicki, 2005). Exact methods for JSSP cannot always be used in practice due to their computational cost; thus, it is necessary to resort to heuristic methods that produce a solution sacrificing optimality (Kim et al., 2020). Product-driven systems (PDS) face such difficulties for JSSP. They are systems that consider the information coming from the product cycle to support decision-making in the control system. Then, the products are equivalent to agents actively participating in the control system (Herrera et al., 2016; Sallez, 2014).

The advantage of using a PDS lies in the increased agility and reactivity of the production system. A PDS provides the ability to react to disturbances related to itself or other parts of the manufacturing system (Pannequin & Thomas, 2012). Furthermore, the PDS considers product intelligent artificial entities to implement and coordinate the control process. Such products allow the reconfiguration of resources to provide agility in the face of production changes (Oltean et al., 2018). The PDS implementation occurs by applying concepts of a holonic manufacturing system (HMS) with a multiagent system (MAS). An HMS has a fundamental unit called a holon, which describes an entity in its physical and virtual forms. HMSs are not simple automated physical structures but entities capable of autonomous self-organization, mixing the physical and virtual worlds to avoid waste and inefficiencies (Mcfarlane et al., 2002). In turn, a MAS constitutes a form of development based on the distribution, autonomy, and cooperation of virtual entities called agents (Leitão et al., 2015). Consequently, a PDS dynamically addresses the optimization of the JSSP (Peng et al., 2019).

Several studies demonstrate the benefits of using a PDS even when decision-making is decentralized. Mihoubi et al. (2020), found good performance in minimizing production system execution times. In turn, Bouazza et al. (2021), considered a PDS as a decision strategy for efficient scheduling rule changes using a hyperheuristic and found good performance in minimizing execution times. Shen and Norrie (1999), used agents as negotiating entities, emphasizing flexibility by combining an MAS with a genetic algorithm. Likewise, Wang and Choi (2014) presented a proposed holonic decomposition to minimize the makespan of a flexible JSSP. The proposed method uses autonomous and cooperative holons to construct solutions. It follows that, in the decentralized approach to the problem, difficulties arise in dealing with combinatorics resulting from the fact that products may change from one rule to another. Consequently, the search space increase produces an increase in computational time. This issue, which has been less explored in the literature, can be addressed with an evolutionary algorithm that could potentially improve the solution.

This manuscript proposes a PDS model that considers parameters associated with the HMS and MAS to increase flexibility in planning a Job Shop production system. The model produces a fast response with a near-optimal solution for problems of different sizes. In addition, the model considers intelligent products to support decision-making by considering a function based on evolutionary algorithms. The use of evolutionary algorithms brings more efficiency to the search process by modifying the representation of the modeled system. The model's performance was tested with JSSP instances studied in the literature and compared with an integer programming model, a heuristic method, and dispatching rules. A comparison with this method allows us to analyze the robustness of the proposed model. Dispatch rules offer a fast alternative solution, although with lower accuracy. In turn, heuristics that are more elaborate methods obtain better solutions. Exact models guarantee the determination of the optimal solution but with a high computational cost.

The main contributions of this manuscript are as follows:

- It proposes a new model for the production planning of a job shop system. The model includes an evolutionary algorithm to search for better solutions applied through intelligent products.

- The proposed model uses the holonic-multiagent paradigm with intelligent products for decision-making in a highly distributed architecture. The product agents can apply evolutionary algorithms to optimize perturbed outcomes.
- Experimentation is performed on JSSP instances used in the literature.
- The model is adaptive in solving JSSP of different sizes and obtains near-optimal results in short execution times.

4.2 Proposed model

This section describes the model proposed to solve the JSSP and the algorithm used to determine the optimal solution. This model is a product-driven system with an evolutionary algorithm (PDS-EA) that represents a productive job shop system in which machines and products interact through production jobs or operations. The process corresponds to an MAS that considers machines and products as agents of the system. In the holonic model, the machine agent and the product agent correspond to a virtual representation of their physical entities. The JSSP solution search occurs with an evolutionary algorithm operated by the product type agents. This process is generic and useful for representing several real industry situations.

Operating agents that possess the characteristics of an intelligent product, defined in the work of C. Wong et al. (2014), represent the system's mobile entities. That means they have a unique identification, can communicate with machines around them and other operation agents, and make operational decisions on a machine. In addition, machine agents are considered static entities that only provide information about the processing time of their tasks. Figure 1 depicts a scheme of the general model that follows an example selected from the literature (Shahzad & Mebarki, 2016). The example considers the scheduling of three products on three machines, as shown in Table 4.1. Figure 4.1 show, in the first stage, the model configured through the product and operation agents, configuring the processing sequence of each of them. In the second stage, the operation agents generate feasible production sequences and calculate the associated makespan. In this stage, the operation agents learn the sequences that generate the best results. The best results from the second stage improve through an evolutionary process in the third stage.

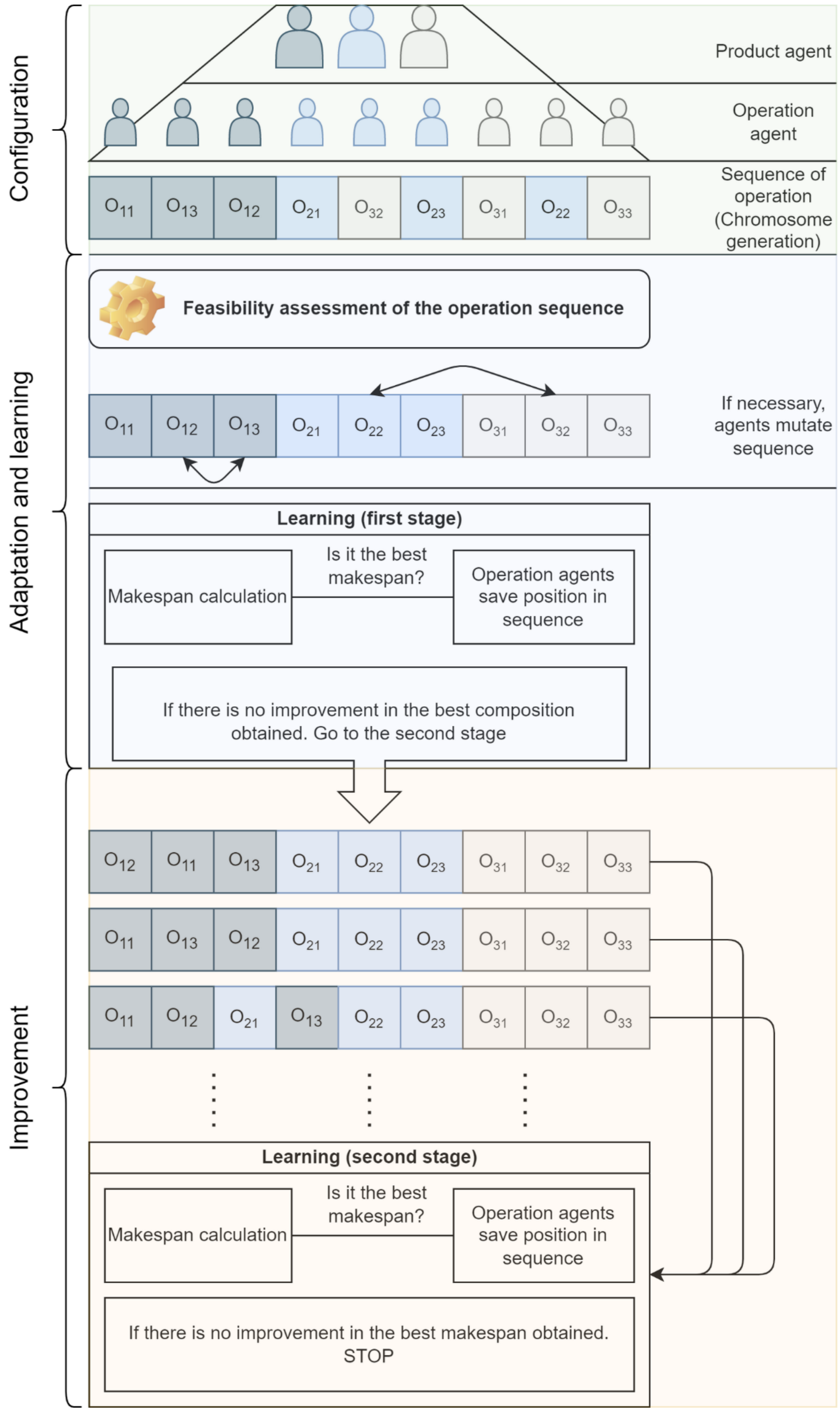

Figure 4.1. Execution procedure of the proposed approach.

Table 4.1. A problem instance with three products and three machines

	M1	M2	M3
J1	O_{11}	O_{12}	O_{13}
J2	O_{21}	O_{22}	O_{23}
J3	O_{31}	O_{32}	O_{33}

The PDS-EA model can be represented schematically by separating the physical and virtual stages. Both stages are represented horizontally in the diagram in Figure 4.2; the columns represent the elements of the model that, in the physical part, correspond to acquisition, entities, and visualization. In turn, the virtual part represents the information inputs, architecture, intelligence function, representation of the entities, agent's response to the problem, interaction, and results.

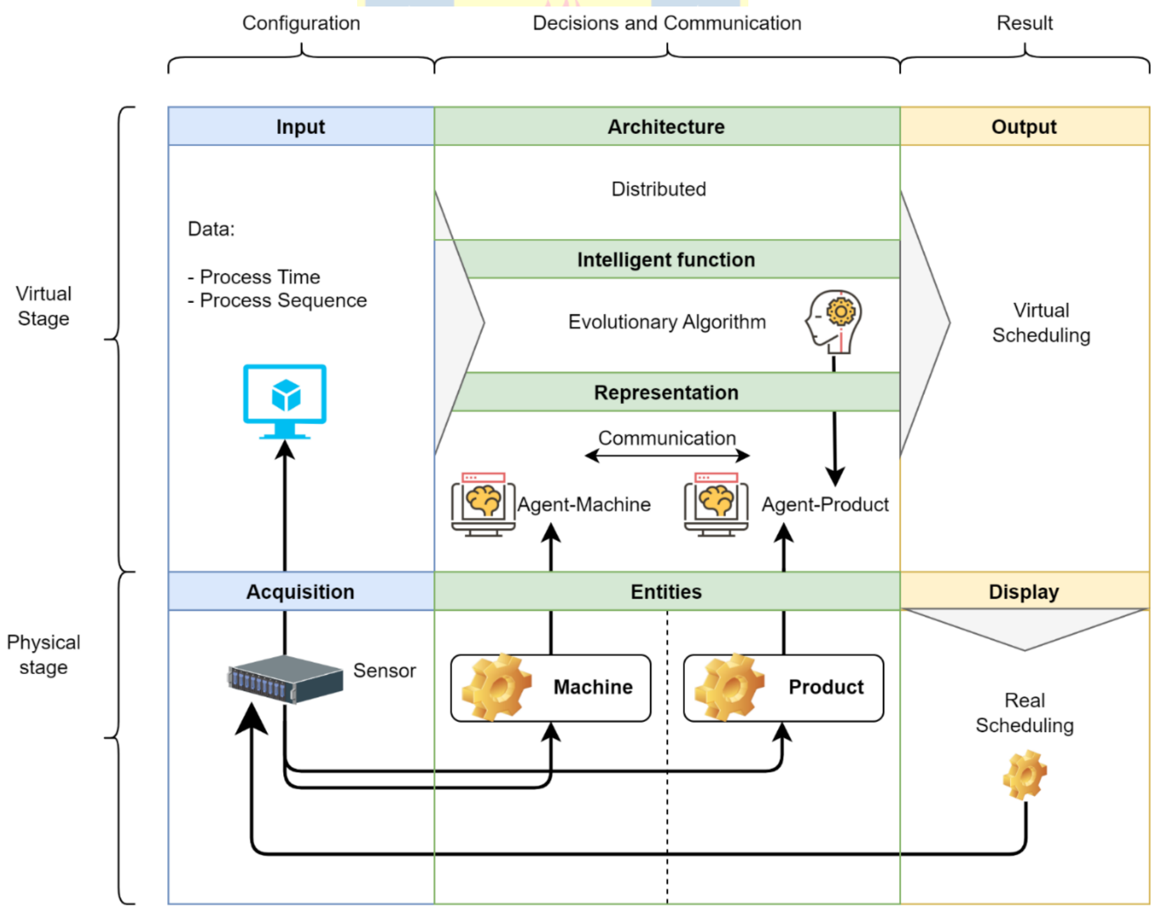

Figure 4.2. Schematic presentation of PDS/EA.

The information flow of the PDS-EA model starts at the acquisition modules that represent physical sensors. Such modules capture information and transform it into data for each virtual entity. The entities represented virtually by agents follow a distribution given by the highly distributed architecture. In this architecture, all the entities associated with the products guide the decisions of their process. Such decisions are generated based on an intelligence function embedded in each agent to evaluate individual and collective performance. This evaluation occurs through three stages: data collection and reading, solution learning and improvement, and scheduling generation. In the data collection and reading stage, product and machine agents collect sequence data and operation times required to complete jobs. In the learning and improvement stage, the operation agents calculate the makespan of the problem through the intelligence function.

The PDS-EA model uses an intelligence function that is an evolutionary algorithm with an elitist selection strategy for makespan minimization. The advantage of using an elitist strategy instead of a probabilistic reproduction is that the best solution improves monotonically concerning the previous generation. The potential disadvantage is the convergence of the population to a local minimum. The balance between the two aspects occurs by regulating the mutation rate. Thus, a mutation evolved on a single chromosome is proposed instead of a gene-by-gene mutation. This process avoids the violation of the production sequence of each product in the JSSP.

The development of the actions of the PDS-EA model is represented through an UML-type sequence diagram. Figure 4.3 depicts the order of actions performed by the operation agents. The first decision occurs with the sequencing of operations generated by all product agents. Then, the operation agents evaluate the sequence's feasibility, verifying the assignment of jobs to the machines according to the JSSP. If the generated sequence is not feasible, mutations are performed to make the sequence feasible. The makespan is then evaluated. If this improves with subsequent iterations, the product agents store the position in the generated sequence in memory and use it to minimize the makespan. Suppose there is no change in the best makespan after generations. In that case, a second stage begins in which the operation agents evaluate new sequences through an evolutionary swapping mechanism between the operation agents' positions. If the operation agent generates an unfeasible sequence, they mutate their positions until feasible. The procedure ends when no improvement occurs after a certain number of iterations. This whole procedure is represented by a flowchart, as shown in Figure 4.4.

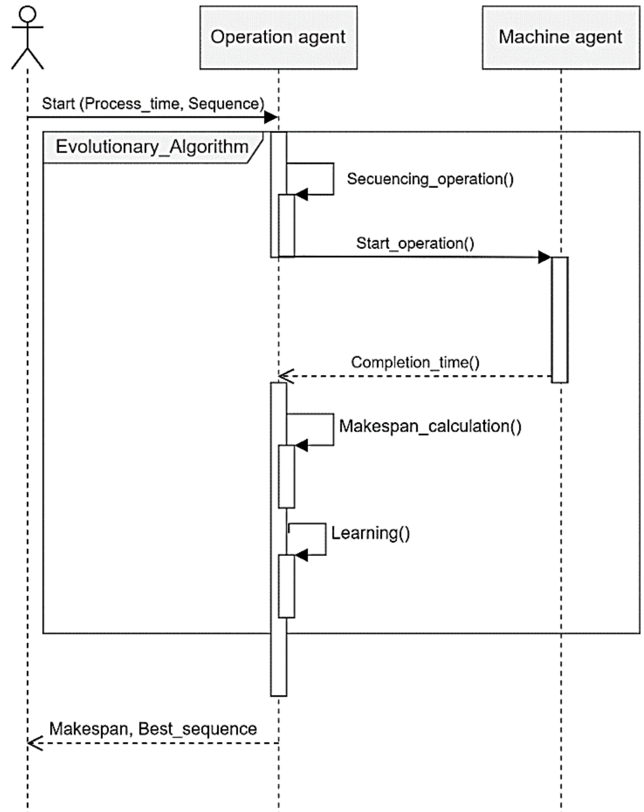

Figure 4.3. Diagram of the sequence for the action of the product agent.

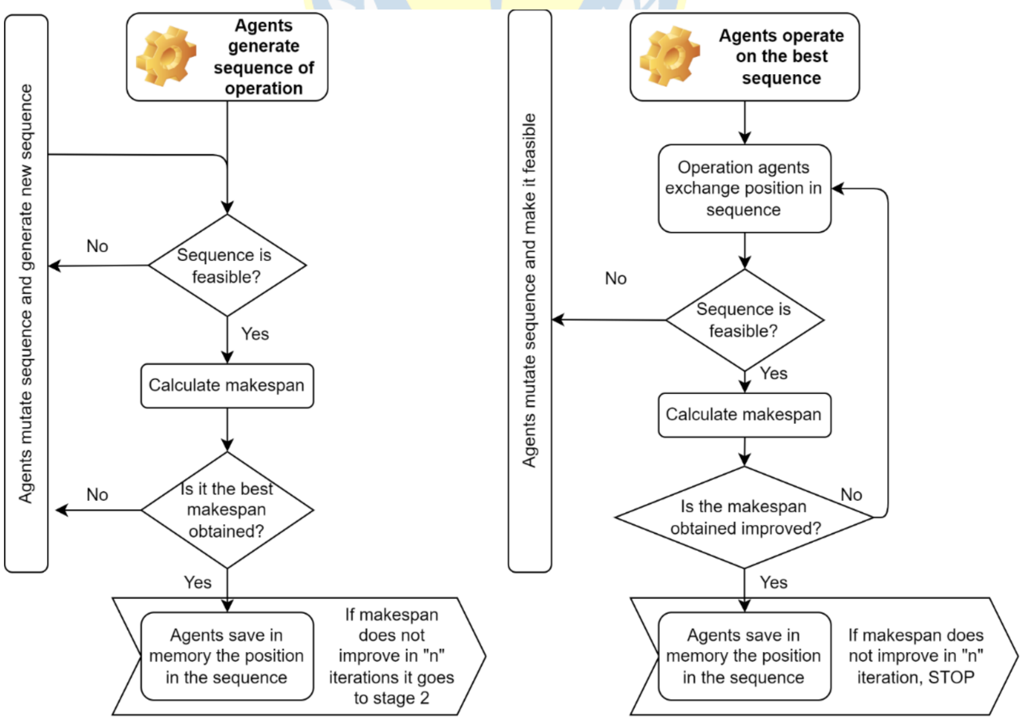

Figure 4.4. Flow diagram for the intelligent process in product decision-making.

The model simulation was performed with NetLogo Version 6.2 (Wilensky, 1999). The proposed approach uses parallelism in MAS, where each instruction is executed by agents simultaneously. In addition, the platform provides a suitable environment for testing and monitoring the model's performance.

We compare the results obtained by the PDS-EA model with three heuristics and an exact model. The heuristics used are the bottleneck heuristic (SBH) proposed by Adams et al. (1988), sequencing by shortest processing time (SPT), and sequencing by longest processing time (LPT). The exact model is an integer programming problem (IP) (Zhang et al., 2019). The problem instances were classified according to their scale as small (SS), medium (MS), and large (LS). This classification results from the number of operations, products, and machines. One hundred-two instances from the literature are used (Table 2). The first column of Table 4.2 shows the authors proposing the test instances. The following columns contain the name of the problem instances and the number of jobs and machines. The last column presents the ranking by the scale of each group of instances. SS contains instances with fewer than 100 operations, MS with more than 100 and fewer than 400 operations, and LS with more than 400 operations. The results of the PDS-EA model with SBH, SPT, LPT, and IP obtained for all test instances are compared, measuring the makespan at different execution times.

The proposed algorithm is compared with three types of standard approaches to the problem. The first approach decomposes the problem into subproblems of less complexity, known as the shifting bottleneck heuristic (SBH) (Mönch & Zimmermann, 2011). The second approach considers dispatching rules that are computationally easy to implement because they assign jobs according to the processing time. Specifically, the short processing time (SPT) assigns jobs from shortest to longest, and conversely, the long processing time (LPT) assigns jobs from longest to shortest. In turn, the integer programming (IP)-based method determines the optimal solution of the problem when it is possible according to the available computational resources.

The makespan evaluates the PDS-EA performance by a deviation ratio. Let m_p be the makespan obtained by the PDS-EA and m_i be the makespan obtained by the IP algorithm. The deviation ratio R for a runtime of 10 minutes is defined according to Equation 1. In addition, the gap between m_p and the lower bound value known for each instance problem is evaluated.

$$R = m_p / m_i \ (1)$$

4.3 Results

The PDS-EA, on average, matches the results of exact methodologies at medium and low scales and performs better with large-scale instances. Such performance is presented in Table 2, whose first column presents the scales under study, followed by the average, minimum and maximum results for the different execution times. For SS instances, PDS-EA obtains values close to those obtained by the exact method (R≈1). The average for MS instances is 23.8% higher, obtaining the best result with one hour of execution (6% higher than the value of the exact methodology). For LS instances, PDS-ES outperforms IP on average with R=0.523. In addition, with LS instances, all run times obtain R<1, and the best result occurs at 2 min of execution.

Table 4.2. Average (Av.) makespan of the first 10 minutes and best result within 1 hour of execution time for SS, MS, and LS instances.

Inst.	1min	2min	3min	4min	5min	6min	7min	8min	9min	10min	60min	Av.
SS	1.21	1.10	1.10	1.06	1.05	1.04	1.04	1.04	1.04	1.04	1.03	1.07
MS	1.34	1.30	1.29	1.27	1.25	1.23	1.22	1.21	1.22	1.20	1.06	1.23
LS	0.85	0.31	0.34	0.40	0.46	0.47	0.50	0.53	0.53	0.53	0.79	0.52

The performance comparison of PDS-EA with IP, SBH, SPT, and LPT is performed with the gap calculated for each instance's optimal or lower bound value. With small-size instances, IP outperforms the other algorithms. In turn, PDS-EA outperforms all heuristics after 2 minutes of execution. Figure 4.5a shows the gap for the five methods during one hour of computational time for SS instances. During such a period, PDS-EA evolves gradually, decreasing the makespan difference related to the best-known makespan. After an hour, the exact method is still in process to determine the static optimal solution. Straight lines in the figure represent the heuristic-determined value found immediately at the beginning of the period. The PDS-EA obtains the best result when one hour of computational time is reached. This result differs from the best result found with IP by 3% but improves by 9.08% the best heuristic value obtained with SBH.

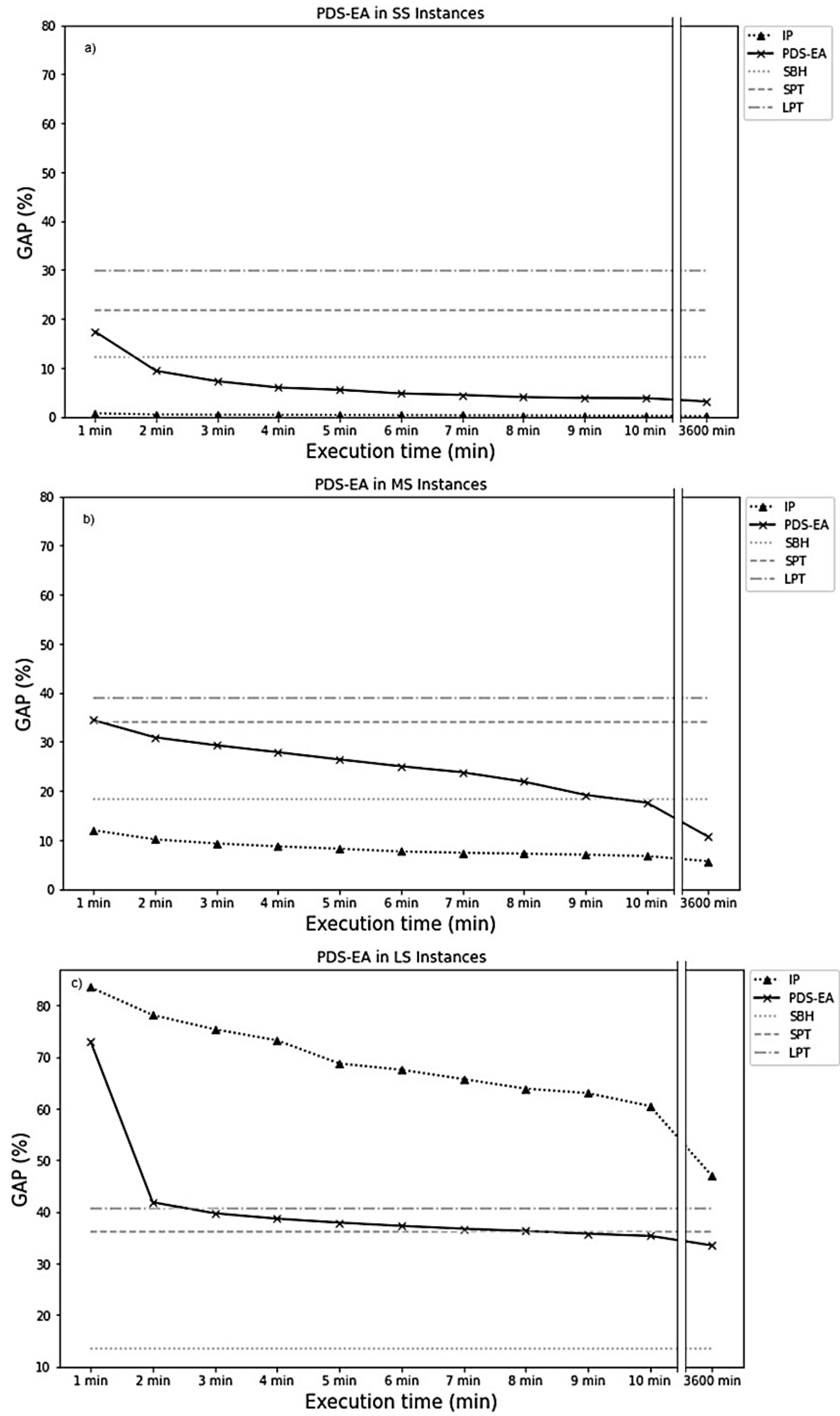

Figure 4.5. PDS-EA gaps for SS, MS, and LS instances.

a. GAP values each minute of simulation Average GAP for SS instances; b. GAP values for each minute of simulation Average GAP for MS instances; c. GAP values each minute of simulation Average GAP for LS instances.

With MS instances, the PDS-EA model performs better than the heuristics from minute 10 onward. The best results occur with IP, which outperforms PDS-EA by 4.97% on average. Figure 4.5b shows the gap for the five methods during one hour of computational time for MS instances. The PDS-EA outperforms the heuristics by 7.58% on average. With the LS instances, the SBH maintains its performance and achieves a gap of less than 20%. This behaviour is observed in Figure 4.5c. Although PDS-EA performs better than the exact method after one hour of computational time, the gaps are larger than for smaller instances. At the end of the period, PDS-EA and IP continue with a decreasing trend, suggesting that convergence is slower for larger instances.

PDS-EA obtains good solutions for the three problem sizes studied. Its main advantage is that the method dynamically optimizes the makespan as time progresses. Figs 4.5a, 4.5b, and 4.5c show that the gradual algorithm produces a better makespan than the heuristic algorithm and the dispatch rules.

Although the inclusion of the evolutionary algorithm in the PDS-EA involves an increase in computational time, its performance is maintained for large instances. Figure 4.5c shows a gradual decrease in the gap to values close to 30% after 1 hour of computational time. Compared to the other sizes, the algorithm requires more time to enter a phase of lower gaps. Even so, it is observed that the process continues to move toward lower gap values, suggesting that a longer simulation time could further reduce the gap. Figure 4.6 shows the gap obtained in all instances for PDS-EA and IP. A similar behavior is observed when the number of operations is approximately 100. However, for higher numbers, both algorithms have a clear difference. For problems of approximately 300 operations, the difference seems to stabilize.

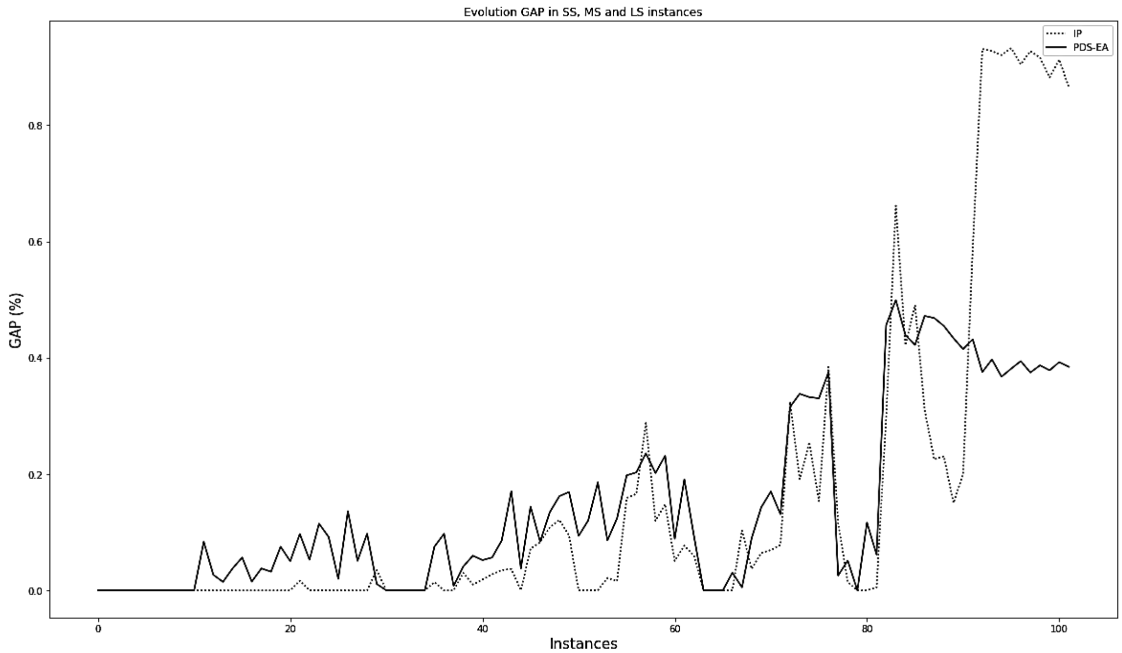

Figure 4.6. Gap for PDS-EA and IP for all instances ordered by the number of operations.

Computational experiments suggest that the proposed PDS-EA offers competitive results compared to exact and heuristic methods. The PDS-EA architecture generates a stable system capable of reacting to changes without human intervention in short periods. Therefore, PDS-EA can solve sequencing processes in job shop environments at different scales, adapting to the complications of each instance. Thus, PDS-EA positions distributed decision-making as a competitive alternative for job scheduling problems in production systems.

4.4 Conclusion

This paper presents a decentralized decision-making model to minimize the makespan of a job shop problem. From the flow of products in the different machines, real-time information feeds the model to correct the course of the operations, keeping in mind the minimization of the makespan. The model contemplates a genetic algorithm selecting the best decision at each instant. To test the model, an agent-based system simulates the operations. Data from the literature allow comparison with four standard approaches: an integer programming algorithm and three approximate methods. Simulation of one hour of computational time, the gap concerning the best-known solution for each approach's small, medium and large-size instances is recorded.

The proposed model's comparative result varies with the number of instances it faces. With small instances, the proposed model underperforms during the whole simulation hour against the exact method, which can find the best solution quickly. With medium-sized instances, a balance is observed between the proposed method and the exact method approaching the hour of simulation time. With large instance sizes, the proposed method outperforms the exact method during that period. In turn, despite being very fast in finding the solution, the approximate methods do not present good performance, with small and medium-sized instances being surpassed by the proposed and exact methods. However, with large instances, the SBH rule produces a better solution than all the solutions generated by the proposed method during the hour of simulation time.

References

Adams, J., Balas, E., & Zawack, D. (1988). The shifting bottleneck procedure for job shop scheduling. The Intitute of Management Sciences, 34(3), 391–400. https://doi.org/0025-1909/88/3403/0391

Asadzadeh, L. (2015). A local search genetic algorithm for the job shop scheduling problem with intelligent agents. Computers and Industrial Engineering, 85, 376–383. https://doi.org/10.1016/j.cie.2015.04.006

Barbosa, J., Leitão, P., Adam, E., & Trentesaux, D. (2015). Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. Computers in Industry, 66, 99–111. https://doi.org/10.1016/j.compind.2014.10.011

Bouazza, W., Sallez, Y., & Trentesaux, D. (2021). Dynamic scheduling of manufacturing systems: a product-driven approach using hyper-heuristics. International Journal of Computer Integrated Manufacturing, 34(6), 641–665. https://doi.org/10.1080/0951192X.2021.1925969

Bożek, A., & Werner, F. (2018). Flexible job shop scheduling with lot streaming and sublot size optimisation. International Journal of Production Research, 56(19), 6391–6411. https://doi.org/10.1080/00207543.2017.1346322

Herrera, C., Belmokhtar-berraf, S., Thomas, A., & Parada, V. (2016). A reactive decision-making approach to reduce instability in a master production schedule. International Journal of Production Research, 7543, 0.

https://doi.org/10.1080/00207543.2015.1078516

Kim, J. G., Jun, H. B., Bang, J. Y., Shin, J. H., & Choi, S. H. (2020). Minimizing tardiness penalty costs in job shop scheduling under maximum allowable tardiness. Processes, 8(11), 1–15. https://doi.org/10.3390/pr8111398

Leitão, P., Rodrigues, N., Barbosa, J., Turrin, C., & Pagani, A. (2015). Intelligent products: The grace experience. Control Engineering Practice, 42, 95–105. https://doi.org/10.1016/j.conengprac.2015.05.001

Mcfarlane, D., Sarma, S., Chirn, J. L., Wong, C. Y., & Ashton, K. (2002). The intelligent product in manufacturing control and management. IFAC Proceedings Volumes, 35(1), 49–54. https://doi.org/10.3182/20020721-6-ES-1901.00011

Mihoubi, B., Bouzouia, B., Tebani, K., & Gaham, M. (2020). Hardware in the loop simulation for product driven control of a cyber-physical manufacturing system. Production Engineering, 14(3), 329–343. https://doi.org/10.1007/s11740-020-00957-w

Mönch, L., & Zimmermann, J. (2011). A computational study of a shifting bottleneck heuristic for multi-product complex job shops. Production Planning and Control, 22(1), 25–40. https://doi.org/10.1080/09537287.2010.490015

Nowicki, E., & Smutnicki, C. (2005). An advanced tabu search algorithm for the job shop problem. Journal of Scheduling, 8(2), 145–159. https://doi.org/10.1007/s10951-005-6364-5

Oborski, P., & Wysocki, P. (2022). Intelligent Visual Quality Control System Based on Convolutional Neural Networks for Holonic Shop Floor Control of Industry 4.0 Manufacturing Systems. Advances in Science and Technology Research Journal, 16(2), 89–98. https://doi.org/10.12913/22998624/145503

Oltean, E., Borangiu, T., & Răileanu, S. (2018). Holonic Hybrid Supervised Control of a Radiopharmaceutical Production Plant. IFAC-PapersOnLine, 51(11), 1249–1254. https://doi.org/10.1016/j.ifacol.2018.08.419

Pannequin, R., & Thomas, A. (2012). Another interpretation of stigmergy for product-driven systems architecture. Journal of Intelligent Manufacturing, 23(6), 2587–2599. https://doi.org/10.1007/s10845-011-0588-3

Peng, C., Wu, G., Warren Liao, T., & Wang, H. (2019). Research on multi-agent genetic algorithm based on tabu search for the job shop scheduling problem. PLoS ONE, 14(9), 1–19. https://doi.org/10.1371/journal.pone.0223182

Pinedo, M. (2012). Scheduling, Theory, algorithms, and systems (Springer (ed.); Fourth edi).

Sallez, Y. (2014). Proposition of an analysis framework to describe the “Activeness” of a product during its life cycle. Serv. Orientat. Holonic Multi-Agent Manuf. Robot, 257–270.

Shahzad, A., & Mebarki, N. (2016). Learning Dispatching Rules for Scheduling: A Synergistic View Comprising Decision Trees, Tabu Search and Simulation. Computers, 5(3), 16. https://doi.org/10.3390/computers5010003

Shen, W., & Norrie, D. H. (1999). Agent-Based Systems for Intelligent Manufacturing : A State-of-the-Art Survey. 1(2), 129–156. https://doi.org/10.1007/BF03325096

Wang, K., & Choi, S. H. (2014). A holonic approach to flexible flow shop scheduling under stochastic processing times. Computers and Operations Research, 43(1), 157–168. https://doi.org/10.1016/j.cor.2013.09.013

Wilensky, U. (1999). NetLogo. http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling Northwestern University Evanston.

Wong, C. Y., Mcfarlane, D., Zaharudin, A. A., & Agarwal, V. (2014). The intelligent product driven supply chain. November 2002. https://doi.org/10.1109/ICSMC.2002.1173319

Zhang, J., Wang, L., & Xing, L. (2019). Large-scale medical examination scheduling technology based on intelligent optimization. Journal of Combinatorial Optimization, 37(1), 385–404. https://doi.org/10.1007/s10878-017-0246-6

CAPÍTULO 5.

A product-driven system approach to reduce nervousness in master production schedule.

Cita: Patricio Sáez, Carlos Herrera, Victor Parada. A product-driven system approach to reduce nervousness in master production schedule.

Abstract: A critical problem for raw material processing companies occurs in the construction of the master production schedule. Scheduling is highly susceptible to fluctuations in demand, which is the primary source of instability and nervousness in the system. Flexible systems that avoid nervousness without increasing the overall cost are necessary to minimize the impact of demand fluctuations. Product-oriented systems are an alternative to face this problem; however, it is unclear how they perform computationally under specific conditions. This paper proposes a product-driven system to complement the master production plan generated by a mathematical model. This system considers intelligent agents that make production decisions with an intelligence function capable of reducing nervousness without significantly increasing the production cost. In the analyzed case, a decrease in nervousness of 11.42% is obtained, involving a cost increase of 2.39%.

Keywords: Product-driven, Nervousness, Schedule, Intelligent product, Agent based model, Holonic manufacturing system.

5.1 Introduction

Conventional manufacturing management is constantly evolving due to the incorporation of new technologies that make it possible to reduce the problems caused by fluctuations in market demand and operational disturbances. Thus, conventional production planning and control models have been transformed into new flexible models to react dynamically during the production period. They are models capable of reacting to disturbances arising from changes in scheduling. Consequently, they respond to disturbances in routing due to operating machinery, production, expansion, processes, products, and production volumes (Cardin et al., 2018; Gräßler & Pöhler, 2017; Kovalenko et al., 2019; Yadav & Jayswal, 2018).

Flexibility in production systems is commonly included in the master production plan. This plan provides the production quantity of each product according to the requirements and market demands and is therefore used for strategy definition and decision-making (Mortezaei & Zulkifli, 2013). The master production plan is devised using optimization models that in general do not contemplate the details of the operations; therefore, they do not guarantee viable production. To correct unfeasibility, the operations are adjusted, generating instability in the system and giving rise to the phenomenon known as *production plan nervousness* (Mortezaei & Zulkifli, 2013).

Nervousness has frequently been cited as an obstacle to implementing stable production systems. The phenomenon produces distrust in planning and a need for permanent supervision (Damand et al., 2013). Since the leading cause of nervousness is the fluctuation of demand, incorporating this phenomenon into a model is a complex task (Atadeniz & Sridharan, 2020). Incorporating new technologies in manufacturing systems and using artificial intelligence tools have made it possible to mitigate the effects produced by nervousness (Campos et al., 2020).

Despite the strong impact of nervousness on production stability, the topic has received limited attention in the literature. Instability is considered the cause of nervousness because, when a high level of nervousness is recorded, an increase in production plan instability arises (Kabak & Ornek, 2009; Sivadasan et al., 2013). Often, instability works indifferently with the concept of nervousness; however, each concept can be considered a consequence of the other (Pujawan & Smart, 2012; V. Sridharan & LaForge, 1990). Some of the most widely accepted proposals to mitigate nervousness have been based on automatic reprogramming for the system to react to exception conditions (Li & Disney, 2017). However, jobs cannot be reprogrammed in practice due to the inflexibility of conventional routines.

Recent developments considering nervousness in production systems have been based on experimental studies and quantitative modeling (Azouz et al., 2018). Although the literature lacks clarity regarding the most appropriate way to mitigate nervousness, some studies have suggested that frequent rescheduling provides better responsiveness to demand fluctuations. In contrast, others have suggested that frequent schedule changes should be avoided (Pujawan & Smart, 2012). In turn, including the cost of production in the analysis results in the conclusion that an improvement in stability does not mean a substantial increase in the total cost of production (Herrera et al., 2016). However, to have more clarity on the performance of a given model, the proposal must be computationally simulated.

Product-driven production systems (PDSs) are models that naturally allow for the inclusion of the nervousness phenomenon. A PDS is an interoperable system in which the product is the controller of resources and adapts to disturbances because it considers products as artificial and intelligent entities to implement and coordinate the control process (Herrera, 2011; Mcfarlane et al., 2002; Meyer et al., 2009). Then, the products allow for the dynamic reconfiguration of resources to provide agility in the face of production changes generated by nervousness. The implementation of a PDS occurs with the holonic system concept (HMS) through a multiagent system (MAS). In an HMS, the process entities, which can be machines, robots, or workers, are modeled as holons consisting of physical and virtual components. Similarly, holons are entities capable of autonomous self-organization, mixing the physical and virtual worlds (Mcfarlane et al., 2002). In turn, an MAS constitutes a form of development based on the distribution, autonomy, and cooperation of virtual entities called agents (Leitão et al., 2015). Bearing in mind the characteristics of PDSs, it seems natural to include the nervousness of the system as an additional component. However, there is no clarity on the effect of this phenomenon's inclusion on the computational performance of PDSs.

This paper presents a PDS that considers the nervousness management of a production planning system. The PDS considers intelligent products as functional units and makes autonomous production decisions to manage nervousness in an environment under realistic conditions. A decrease in system nervousness occurs due to the decentralized decision-making generated by the information coming from intelligent products. The evaluation of the computational performance of the proposed PDS occurs with a production planning scenario with 12 products over a planning horizon of one year. The novelty of this proposal is the generation of flexible production planning with the ability to decrease the nervousness of the system, generating more stable plans and dampening the increase in production costs.

The article is organized as follows. Section 2 presents a literature review and describes key concepts, such as master production plan, nervousness, PDSs, and intelligent products. Section 3 describes the proposed PDS. Section 4 shows the experimental design, Section 5 presents the results obtained, and Section 6 concludes the paper.

5.2 Related work

Production planning determines the quantity of a product to be produced, the time at which it should be produced, and in some cases, the section of the facility where each production stage should be performed. The planning takes place by a mathematical model that optimizes decision-making while minimizing costs or maximizing profits. The model determines the quantity produced in each period over a finite horizon without exceeding the system's capacity and satisfying the demand of future periods. Lot-sizing modeling is one of the relevant techniques that allows for planning to be performed (Ramya et al., 2019).

Production planning that considers time has been addressed in the literature through moving horizon planning for various production processes (Demirel et al., 2018; Ju et al., 2018; Kabak, 2009; Lalami et al., 2017; Lin & Uzsoy, 2016). However, although moving horizon planning is a common practice in the industry, there is still little clarity in the literature on its impact on the stability of the production process when combined with artificial intelligence tools (Ziarnetzky et al., 2018). A study with real-world data was developed in the automotive industry, considering multiple impact assessment tests to meet the requirements of a plant (Lalami et al., 2017).

In modern industry, it is necessary for production planning to respond to dynamic market conditions by mitigating the effects of nervousness. Planning should reduce lead times, provide greater agility to processes, improve product quality, and reduce manufacturing costs (S. C.L. Koh et al., 2002). However, fulfilling these objectives usually requires a sequence of reconfigurations of operations, causing a permanent modification of the established schedule, generating instability, and increasing production nervousness (Salido et al., 2017). Several studies have presented methodologies and tools for measuring, detecting, and eliminating production instability (Schuh et al., 2019). Instability and nervousness have often been studied indifferently (Pujawan & Smart, 2012; V. Sridharan & LaForge, 1990). Other works have considered instability as a consequence of system nervousness (Kabak & Ornek, 2009; Sivadasan et al., 2013). With a higher level of specificity, Tunc et al. (2013) mentioned two types of nervousness that occur as a consequence of the quantities involved or of the configurations made. Some of the most widely accepted proposals to mitigate nervousness resort to automatic reprogramming, so the system reacts to exceptional conditions (Li & Disney, 2017). However, reprogramming is sometimes not straightforward due to the inflexibility of conventional programs.

Several studies have considered the mitigation of nervousness based on the quantity of production, inventory, or safety stock (Hasachoo & Masuchun, 2016a; Koh & Saad, 2006; V. Sridharan & LaForge, 1990). Other proposals for nervousness mitigation have focused on the planning horizon and the amount of production or storage. In the former, planning horizon freezing has been used (Atadeniz & Sridharan, 2020; Kadipasaoglu & Sridharan, 1995; Lalami et al., 2017; S. v. Sridharan et al., 1988). Additionally, the rolling horizon method (Inderfurth, 1994; Kazan et al., 2000; Mönch & Zimmermann, 2011; van der Sluis, 1993) and increases in the forecast horizon (Carlson et al., 1982; Hasachoo & Masuchun, 2016b) have been studied. Other authors have considered the dynamic lot-sizing model (Carlson et al., 1979; Xie et al., 2003) and control rules (de Kok & Inderfurth, 1997).

A fundamental component in the design of a PDS is the intelligent product, which has different definitions (Wong et al., 2014; Valckenaers et al., 2009; Kiritsis, 2011;Kärkkäinen et al., 2003; Ventä, 2007; Mcfarlane, 2012). The definition used in our proposal is that of Wong et al. (2014). Their definition states that an intelligent product must have five characteristics: it must possess a unique identity, be able to communicate effectively with its environment, retain or store data about itself, be able to participate or make decisions relevant to its destiny, and have a communication language to display its

characteristics. Thus, intelligent products are the entities responsible for making decisions and planning the system's future. Consequently, an intelligent product gives the PDS a particular orientation toward the synchronization of material flows and information flows. The characteristics of intelligent products provide the basis for the product-controlled production approach. They are entities that take the initiative during the execution of the production plan by reacting appropriately to disturbances that might occur (Herrera et al., 2016). This approach facilitates the design, distribution, and operation phases of production. The consequence is improved product quality and performance resulting from self-learning, self-diagnosis, self-adaptation, and self-optimization (Barbosa et al., 2015).
A PDS is a distributed control system to support operational decision-making, the design of which is facilitated by including the holonic paradigm, which specifies that each product is represented by physical and virtual components (Mihoubi et al., 2020). In turn, the virtual component is interpreted as an agent, which is why a PDS corresponds to a multiagent system. Agent-based models have entities with an active and autonomous role, originating actions without direct human intervention. C. Herrera et al. (2014) studied a production system with such characteristics by simulating the coordination of the different decision levels. The authors analyzed production planning and control processes and found that coordination between active batches is effective at distributed levels compared to conventional approaches. In addition, the study by J. Campos et al. (Campos et al., 2020) provided a solution to a dynamic scheduling problem by dividing the process into three stages. Each stage involves different agents with specific roles, although the authors did not directly consider a master scheduling model.

Integrating a PDS with a holonic system and its implementation through a multiagent system could generate computational times that do not allow for real-time production control. In turn, the decentralized decision-making of such systems could provide feasible solutions that minimize nervousness for a given period but with higher production costs. In addition, practical solutions adopted in the industry have considered static production modeling that could be incorporated into a PDS as an initial situation to be dynamically corrected with individual intelligent product decisions. Integrally, these topics have received less attention in the production planning literature, and there is little clarity regarding the computational performance that a PDS with such features could have.

5.3 PDS proposed

The proposed PDS implements a master plan for a production system that operates with production cycles and periods, considering the existence of nervousness. The master production plan is obtained from the optimization problem solution, which provides the optimal quantity to produce in each cycle and period of each product. Each product is represented by a virtual agent that transforms the information into valuable data for decision-making. Thus, the agents constitute a highly distributed architecture. In turn, each agent contains an intelligence function that evaluates its individual and collective performance. Nervousness is evaluated as the difference between the quantity to be produced of each product in a given cycle and a given period. Section 5.3.1 presents the optimization model, and Section 5.3.2 contains the nervousness evaluation. Section 5.3.3 presents the PDS architecture.

5.3.1 The optimization problem

The mathematical model that produces the master production plan considers the minimization of the production cost subject to the constraints that specify the quantity to be produced at a given time. This formulation extends the formulation presented in the literature for lot-sizing problems by including production costs, inventory, setup, and backorder costs (Quadt, 2004; Ramya et al., 2019). Let the following decision variables be defined as follows:

$x_{it} = \ Quantity\ of\ product\ i\ in\ period\ t\ in\ cycle\ k.$

$s_{it} = \ Quantity\ of\ inventory\ product\ i\ in\ period\ t\ in\ cycle\ k.$

$r_{it} = \ Backlog\ of\ product\ i\ in\ period\ t\ in\ cycle\ k.$

$y_{it} = \ Setup\ of\ product\ i\ in\ period\ t\ \ (y_{it} = 1 \leftrightarrow x_{it} > 0, \forall i, \forall t, \forall k).$

The model requires the following input:

$d_{it} = \ Demand\ of\ product\ i\ in\ period\ t.$

$p_{it} = \ Production\ cost\ of\ product\ i\ in\ period\ t.$

$h_{it} = \text{Inventary cost of product } i \text{ in period } t.$

$b_{it} = \text{Backorder cost of product } i \text{ in period } t.$

$q_{it} = \text{Setup costo of product } i \text{ in period } t.$

$C_t = \text{Capacity in period } t.$

The mathematical formulation is in Equations (1) to (5).

$$min\ f^k = \sum_{i=1}^{n} \sum_{t=k}^{t\prime} (p_{it}x_{it} + h_{it}s_{it} + b_{it}r_{it} + q_{it}y_{it}) \tag{1}$$

Subject to:

$$x_{it} \le My_{it}, i \in [1, \dots, n], t \in [k, \dots, t'] \tag{2}$$

$$\sum_{i}^{n} x_{it} \le C_t, \mathrm{t} \in [k, \dots, t'] \tag{3}$$

$$s_{i0}^0 = s_{ini}^0, \quad r_{i0}^0 = r_{ini}^0, \quad i \in [1, \dots, n] \tag{4}$$

$$s_{i(t-1)} - r_{i(t-1)} + x_{it} = d_{it} + s_{it} - r_{it} \quad , i \in [1, \dots, n], t \in [k, \dots, t'] \tag{5}$$

The objective function of model f^k in Equation (1) corresponds to the minimization of the production cost in the intervals of time horizon sliding $[k, \dots, t']$. In this way, k and t' = $k + n - 1$ are the first and last periods, respectively, of the mobile planning horizon of length n in each cycle k. Constraints (2) relate production and the corresponding setup, where setup = 1 when there is production and 0 otherwise. Constraints (3) restrict the production according to the available capacity during the period. Constraints (4) and (5) set the initial conditions of inventory, backorders, and the balance between the two.

5.3.2 Measurement of system nervousness

Nervousness measures the difference in the quantity to be produced of product i in period t during production cycle k compared to the previous cycle and period. The calculation is based on two parameters -- the magnitude of change and the frequency of changes -- so that significant changes or a high frequency of changes in production implies high values of nervousness. Two metrics express the nervousness per cycle and period. Let N be the planning horizon, P be the number of products in the planning, C_{ki} be the number of schedule changes of product i in cycle k and C_{ti} be the number of schedule changes of product i in period t. Furthermore, let Q_{it}^{k} be the production quantity for product i in period t in cycle k. Then, in Equation (6), N_{cki} is the nervousness in cycle k for product i, and in Equation (7), $N_{pti}$is the nervousness in period t for product i. Equation (8) presents the measure of nervousness N.

$$N_{cki} = C_{ki} * \left\{\sum_{t=0}^{n} |Q_{i\,(t+1)}^{k} - Q_{it}^{k}\,|\right\}, \forall\ k \quad (6)$$

$$N_{pti} = C_{pi} * \left\{\sum_{t=0}^{n} |Q_{it}^{(k+1)} - Q_{it}^{k}\,|\right\}, \forall\ t \quad (7)$$

$$N = N_{cki} + N_{pti} \ (8)$$

The parameter Φ_k in Equation (9) quantifies the ratio between cost and nervousness for each cycle k. This parameter identifies the magnitude of the change in each cycle by calculating the area under the curve of cost and nervousness.

$$\Phi_k = \frac{\int_0^k c(k)dk}{\int_0^k n(k)dk}, \forall\ k \in \{k = 1, \ldots, k = 60\}\ (9)$$

5.3.3 PDS Architecture

The system architecture contains physical and virtual layers, each of which has three levels -- configuration, interactions, and results -- as shown in Figure 5.1. Each product is represented in the virtual layer by an agent, transforming the information into valuable data for decision-making. The configuration of the virtual layer represents the results generated by the optimization model as data for communication and decision-making by each agent. Thus, the physical layer of the system interacts with other physical entities, and its virtual layer interacts with the environment for production control and management. Decision-making and communication among agents are distributed on the same hierarchical scale. The intelligence function of the agents considers decision rules for obtaining a global objective considering all of the system's entities. Such decision rules are known and applied by all of the agents of the system through internal and interagent communication. This information is processed and stored in the physical part of the components. At the results level, the model outputs correspond to the production planning, virtually and physically representing the planning.

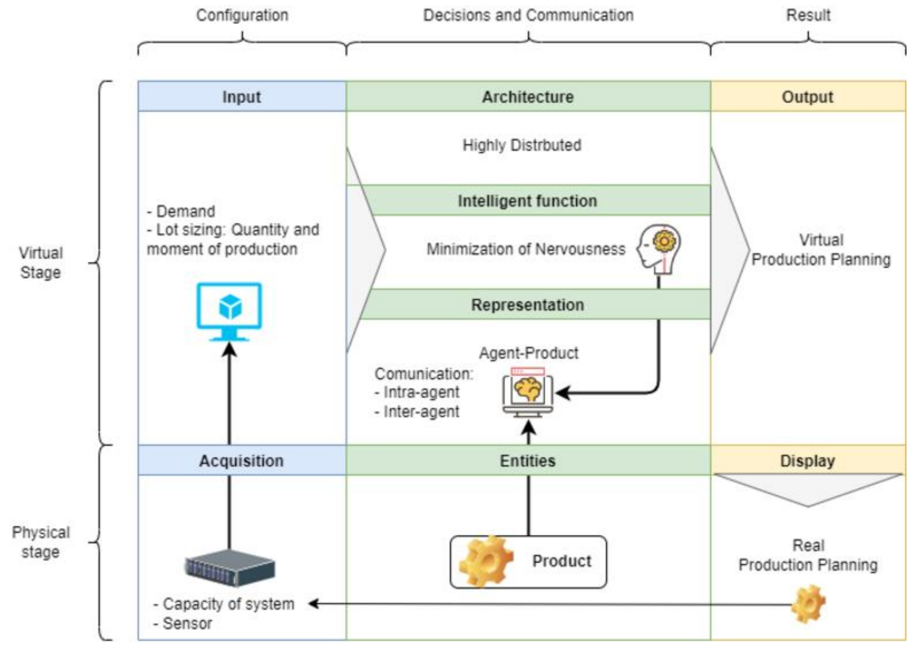

Figure 5.1. PDS architecture

The virtual layer contemplates an intelligence function that evaluates individual and collective performance, looking for the system's stability with a sustained cost increase. To this end, the intelligence function measures the nervousness of each agent using Equation (8). Each agent complies with the characteristics of an intelligent product defined by Wong et al. (2014); i.e., they have a unique identifier and can communicate with the surrounding agents of the same product type, as shown in Figure 5.2.

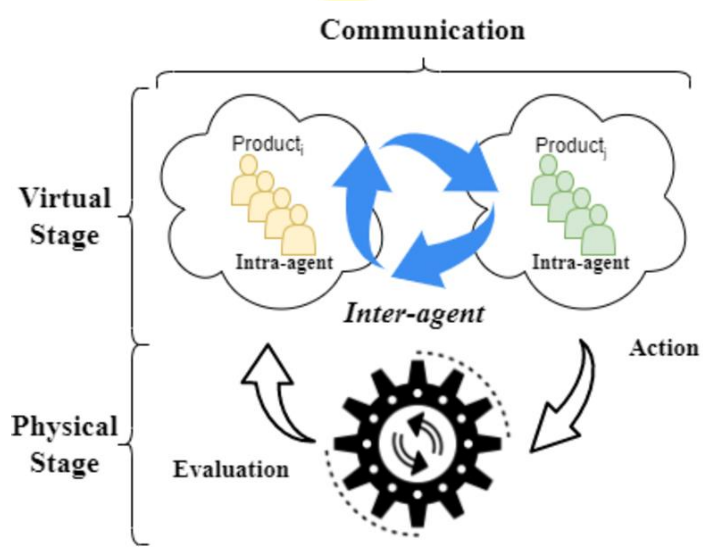

Figure 5.2. Diagram of agent communication.

The architecture operation comprises a sequence of actions represented in Figure 5.3. Initially, the algorithm that solves the mathematical model generates the optimal production. The agents evaluate the nervousness and cost of the scheduling performed, determining the production required for a minimum increase in production cost. Then, the agents communicate with agents of the same product type in the respective cycle and period to evaluate the production quantity. Simultaneously, agents communicate with agents of another product family to avoid exceeding the system's production capacity and to satisfy each product's demand (see Figure 5.4). Then, the possibilities of decreasing nervousness are evaluated by modifying the production quantities and calculating the costs associated with such modifications. When a production quantity modification occurs that improves the value of nervousness, the agents store the production values. This communication architecture and agent interactions respond to a perturbation of the system because of the permanent evaluation of quantities.

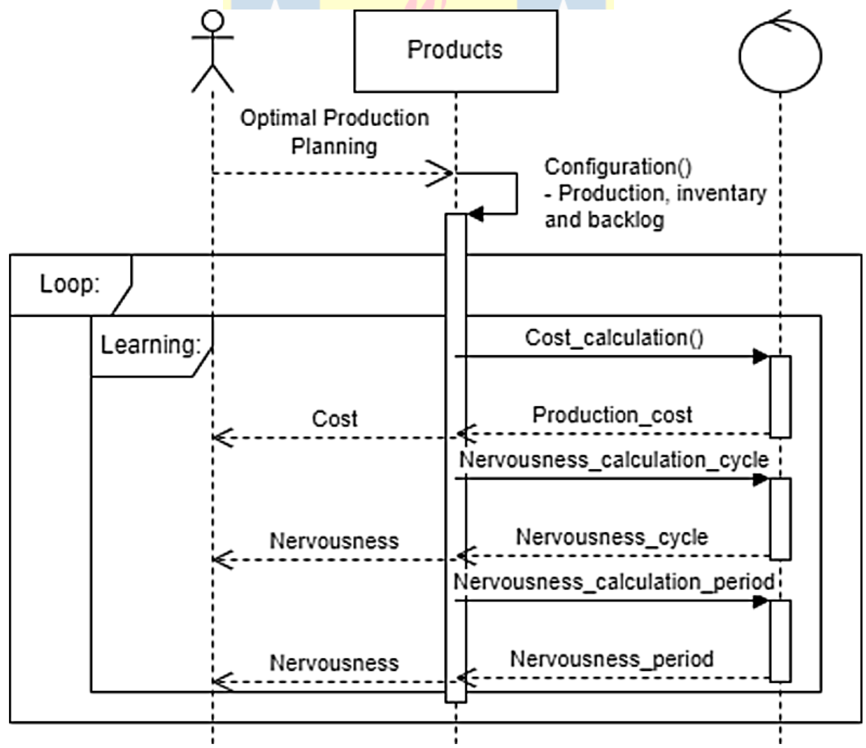

Figure 5.3. Sequence diagram.

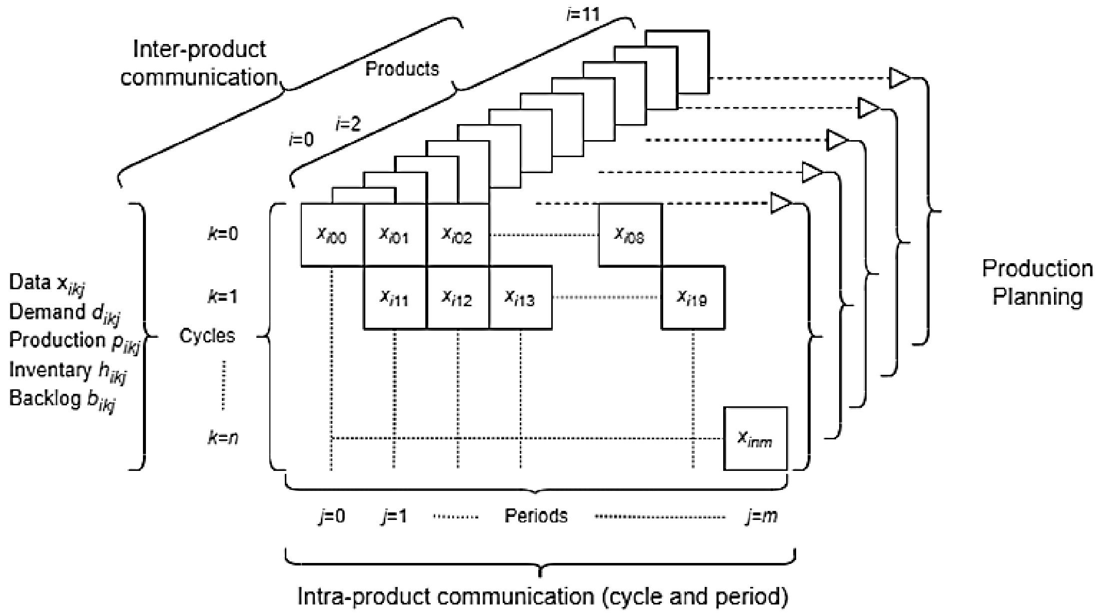

Figure 5.4. Simulation dynamics and agent communication.

5.4 Experimental design

The production plan considers 12 products and a production horizon of 52 periods. The planning horizon is $n = 8$ with an interval between periods of $\Delta t = 1$. The demand for each product obeys a normal distribution $d_{it}^{k} \sim \eta(\mu, \rho) = \eta(120, \rho), \forall i, \forall t, \forall k$ to simulate different variations of ρ. In the first stage, a master production plan is generated for each product in the active period and a demand projection for the subsequent periods. The complete simulation has parameters close to an industrial case.

Version 6.2 of the NetLogo simulation platform is used for the simulation, which provides a suitable environment for testing and monitoring model performance (Wilensky, 1999).

5.5 Results

The PDS presents an initial phase of significant variation in cost and nervousness until it reaches a steady state. This phenomenon emerges from a simulation with three control variables: per period, per cycle, and per period cycle. In period-based control, intelligent products analyze the production quantities during each period and modify the production plan to reduce nervousness. In cycle control, intelligent products analyze the production

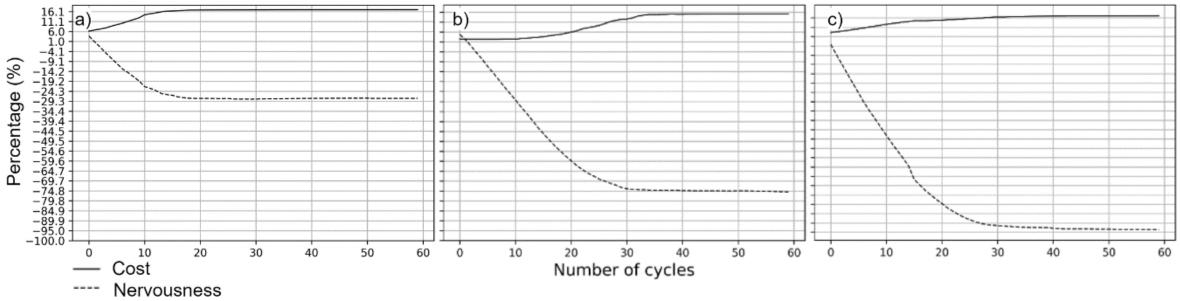

Figure 5.5. a) Results of the model based on control by period. b) Results of the model based on control by cycle. c) Results of the model based on control by simultaneous period cycle.

quantities in the planning horizon. In period-cycle control, intelligent products look for stability per period and per cycle by considering consecutive periods of the planning horizon. In each type of control, Equations (6), (7), and (8) update the nervousness. Figure 5 shows the results of the variations of cost and nervousness for each type of control. The decrease in nervousness occurs with the consequent increase in cost concerning the initial values. For example, considering control by period (Figure 5.5a), there is an increase of 11.21% in cost and a reduction of 14.72% in nervousness in the eighth cycle. In turn, in control by cycle (Figure 5.5b), an increase of 2.39% in cost and a reduction of 18.27% in nervousness are observed. Figure 5.5c shows the behavior of the PDS according to the period-cycle control. A more significant decrease in nervousness is observed than with the two previous types of controls. In the same programming cycle, an increase of 11.27% in cost and a reduction of 34.44% in nervousness are observed.

The PDS results indicate an uneven relationship between decreased nervousness and increased costs. Thus, the more significant that the decrease in nervousness is, the smaller that the increase in the cost of the production plan is. This dynamic generated by the system is consistent with the modification of the plan that minimizes the cost; i.e., any change in the production plan calculated through the mathematical model generates an increase in cost. However, the benefit of such a modification implies more stable plans. As the production cycles proceed, both cost and nervousness reach an equilibrium because it is no longer possible to modify production quantities.

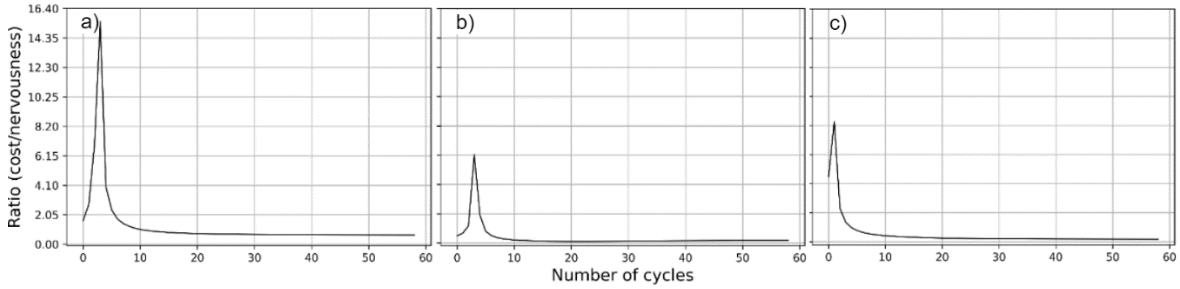

Figure 5.6. a) Values of Φ for the model based on control per period. b) Values of Φ for the model based on control per cycle. c) Values of Φ for the model based on control per period cycle.

In Figure 5.6, we observe the results for different values of Φ_k, which compares the initial cost increase with the benefits of nervousness reduction. The behavior is similar in the three types of control applied, obtaining a more noticeable change when using the cycle-period control, which optimizes in a balanced way between cycle and product.

In all types of control, cost increases with decreasing nervousness are observed in the first cycles of the simulation. However, after this initialization stage, a period of stability is reached during which there are no substantial differences in the magnitude of the changes associated with costs and nervousness. For example, by applying cycle control, a reduction of 11.42% in nervousness is achieved with an increase of 2.39% in the cost of the production plan. The computational results suggest that using a PDS is promising in reducing nervousness without substantial increases in production costs. Thus, a PDS can improve the master production plan by minimizing nervousness and adapting to changing environments.

5.6 Conclusions

This work proposes a production planning system that contemplates nervousness management. The system reduces nervousness and adjusts production using the concept of intelligent products and starting from a production plan for the planning period. The initial production plan is produced with a mathematical cost minimization model. The proposed system is evaluated numerically considering a 12-product production system and a planning period of one year. The evolution of the system performance during the period is reflected in the production costs and system nervousness.

Experimental evidence shows that PDSs reduce nervousness without increasing the cost in the same proportion. In this sense, a 2.39% increase in cost results in a nervousness reduction of up to 11.42% when using cycle control. The PDS generates flexible solutions without the need to perform multiple executions of the algorithm that solves the mathematical model, which, although it generates the optimal solution, requires a long computational time.

The system built includes a mathematical model, a metric to measure nervousness and a definition of an intelligent product. In the literature, several variants for each of these 3 options have been observed, generating a combination of possible situations. Future research could allow us to explore this combination of possibilities generating a production planning control system according to the needs of each industry.

References

Atadeniz, S. N., & Sridharan, S. v. (2020). Effectiveness of nervousness reduction policies when capacity is constrained. *International Journal of Production Research*, *58*(13), 4121–4137. https://doi.org/10.1080/00207543.2019.1643513

Azouz, N., Silva Belisário, L., Ammar, A., & Pierreval, H. (2018). Addressing over-correction in adaptive card-based pull control systems. *International Journal of Computer Integrated Manufacturing*, *31*(12), 1189–1204. https://doi.org/10.1080/0951192X.2018.1509132

Barbosa, J., Leitão, P., Adam, E., & Trentesaux, D. (2015). Dynamic self-organization in holonic multi-agent manufacturing systems: The ADACOR evolution. *Computers in Industry*, *66*, 99–111. https://doi.org/10.1016/j.compind.2014.10.011

Campos, J. T. de G. A. e A., Blumelova, J., Lepikson, H. A., & Freires, F. G. M. (2020). Agent-based dynamic scheduling model for product-driven production. *Brazilian Journal of Operations & Production Management*, *17*(4), 1–10. https://doi.org/10.14488/bjopm.2020.044

Cardin, O., Derigent, W., & Trentesaux, D. (2018). Evolution of holonic control architectures towards Industry 4.0: A short overview. *IFAC-PapersOnLine*, *51*(11), 1243–1248. https://doi.org/10.1016/j.ifacol.2018.08.420

Carlson, R. C., Beckman, S. L., & Kropp, D. H. (1982). the Effectiveness of Extending the Horizon in Rolling Production Scheduling. In *Decision Sciences* (Vol. 13, Issue 1, pp. 129–146). https://doi.org/10.1111/j.1540-5915.1982.tb00136.x

Carlson, R. C., Jucker, J. Y., & Kropp, D. H. (1979). Less nervous MRP systems: a dynamic economic lot sizing approach. *Management Science*, *25*, 754–761. https://doi.org/http://dx.doi.org/10.1287/mnsc.25.8.754

Damand, D., Derrouiche, R., & Barth, M. (2013). Parameterisation of the MRP method: Automatic identification and extraction of properties. *International Journal of Production Research*, *51*(18), 5658–5669. https://doi.org/10.1080/00207543.2013.810819

de Kok, T., & Inderfurth, K. (1997). Nervousness in inventory management: Comparison of basic control rules. *European Journal of Operational Research*, *103*(1), 55–82. https://doi.org/10.1016/S0377-2217(96)00255-X

Demirel, E., Özelkan, E. C., & Lim, C. (2018). Aggregate planning with Flexibility Requirements Profile. *International Journal of Production Economics*, *202*, 45–58. https://doi.org/10.1016/j.ijpe.2018.05.001

Gräßler, I., & Pöhler, A. (2017). Implementation of an Adapted Holonic Production Architecture. *Procedia CIRP*, *63*, 138–143. https://doi.org/10.1016/j.procir.2017.03.176

Hasachoo, N., & Masuchun, R. (2016a). Factors affecting schedule nervousness in the production operations of airline catering industry. *IEEE International Conference on Industrial Engineering and Engineering Management*, *2016-Janua*, 499–503. https://doi.org/10.1109/IEEM.2015.7385697

Hasachoo, N., & Masuchun, R. (2016b). Reducing schedule nervousness in production and operations under non-stationary stochastic demand: The case of an airline catering company. *IEEE International Conference on Industrial Engineering and Engineering Management*, *2016-Decem*, 941–945. https://doi.org/10.1109/IEEM.2016.7798016

Herrera, C. (2011). *Cadre générique de planification logistique dans un contexte de décisions centralisées et distribuées.* Université Henry Poincaré - Nancy . Francais, <NNT : 2011NAN10046>. <tel-01746192v2>.

Herrera, C., Belmokhtar-berraf, S., Thomas, A., & Parada, V. (2016). A reactive decision-making approach to reduce instability in a master production schedule. *International Journal of Production Research*, *7543*, 0. https://doi.org/10.1080/00207543.2015.1078516

Herrera, C., Thomas, A., & Parada, V. (2014). A product-driven system approach for multilevel decisions in manufacturing planning and control. *Production & Manufacturing Research*, *2*(1), 756–766. https://doi.org/10.1080/21693277.2014.949895

Inderfurth, K. (1994). Nervousness in inventory control: Analytical results. *OR Spektrum*, *16*(2), 113–123. https://doi.org/10.1007/BF01719468

Ju, L., Zhang, Q., Tan, Z., Wang, W., Xin, H., & Zhang, Z. (2018). Multi-agent-system-based coupling control optimization model for micro-grid group intelligent scheduling considering autonomy-cooperative operation strategy. *Energy*, *157*, 1035–1052. https://doi.org/10.1016/j.energy.2018.06.097

Kabak, K. E. (2009). An Improved Metric for Measuring Schedule Instability in Material An improved metric for measuring multi-item multi-level schedule instability under rolling schedules. *Computers & Industrial Engineering*, *56*(2), 691–707. https://doi.org/10.1016/j.cie.2006.11.001

Kabak, K. E., & Ornek, A. M. (2009). An improved metric for measuring multi-item multi-level schedule instability under rolling schedules. *Computers & Industrial Engineering*, *56*(2), 691–707. https://doi.org/10.1016/j.cie.2006.11.001

Kadipasaoglu, S. N., & Sridharan, V. (1995). Alternative approaches for reducing schedule instability in multistage manufacturing under demand uncertainty. *Journal of Operations Management*, *13*(3), 193–211. https://doi.org/10.1016/0272-6963(95)00023-L

Kärkkäinen, M., Ala-Risku, T., & Främling, K. (2003). The product centric approach: A solution to supply network information management problems? *Computers in Industry*, *52*, 147–159. https://doi.org/https://doi.org/10.1016/S0166-3615(03)00086-1

Kazan, O., Nagi, R., & Rump, C. M. (2000). New lot-sizing formulations for less nervous production schedules. *Computers and Operations Research*, *27*(13), 1325–1345. https://doi.org/10.1016/S0305-0548(99)00076-3

Kiritsis, D. (2011). Computer-Aided Design Closed-loop PLM for intelligent products in the era of the Internet of things. *Computer-Aided Design*, *43*(5), 479–501. https://doi.org/10.1016/j.cad.2010.03.002

Koh, S. C. L., & Saad, S. M. (2006). Managing uncertainty in ERP-controlled manufacturing environments in SMEs. *International Journal of Production Economics*, *101*(1 SPEC. ISS.), 109–127. https://doi.org/10.1016/j.ijpe.2005.05.011

Koh, S. C. L., Saad, S. M., & Jones, M. H. (2002). Uncertainty under MRP-planned manufacture: Review and categorization. *International Journal of Production Research*, *40*(10), 2399–2421. https://doi.org/10.1080/00207540210136487

Kovalenko, I., Tilbury, D., & Barton, K. (2019). The model-based product agent: A control oriented architecture for intelligent products in multi-agent manufacturing systems. *Control Engineering Practice*, *86*(March), 105–117. https://doi.org/10.1016/j.conengprac.2019.03.009

Lalami, I., Frein, Y., & Gayon, J. P. (2017). Production planning in automotive powertrain plants: a case study. *International Journal of Production Research*, *55*(18), 5378–5393. https://doi.org/10.1080/00207543.2017.1315192

Leitão, P., Rodrigues, N., Barbosa, J., Turrin, C., & Pagani, A. (2015). Intelligent products: The grace experience. *Control Engineering Practice*, *42*, 95–105. https://doi.org/10.1016/j.conengprac.2015.05.001

Li, Q., & Disney, S. M. (2017). Revisiting rescheduling: MRP nervousness and the bullwhip effect. *International Journal of Production Research*, *55*(7), 1992–2012. https://doi.org/10.1080/00207543.2016.1261196

Lin, P. C., & Uzsoy, R. (2016). Chance-constrained formulations in rolling horizon production planning: an experimental study. *International Journal of Production Research*, *54*(13), 3927–3942. https://doi.org/10.1080/00207543.2016.1165356

Mcfarlane, D. (2012). Product Intelligence : Theory and Practice. In *IFAC Proceedings Volumes* (Vol. 45, Issue 6). IFAC. https://doi.org/10.3182/20120523-3-RO-2023.00441

Mcfarlane, D., Sarma, S., Chirn, J. L., Wong, C. Y., & Ashton, K. (2002). THE INTELLIGENT PRODUCT IN MANUFACTURING CONTROL AND MANAGEMENT. *IFAC Proceedings Volumes*, *35*(1), 49–54. https://doi.org/10.3182/20020721-6-ES-1901.00011

Meyer, G. G., Holmstro, J., & Fra, K. (2009). *Computers in Industry Intelligent Products : A survey*. *60*, 137–148. https://doi.org/10.1016/j.compind.2008.12.005

Mihoubi, B., Bouzouia, B., Tebani, K., & Gaham, M. (2020). Hardware in the loop simulation for product driven control of a cyber-physical manufacturing system. *Production Engineering*, *14*(3), 329–343. https://doi.org/10.1007/s11740-020-00957-w

Mönch, L., & Zimmermann, J. (2011). A computational study of a shifting bottleneck heuristic for multi-product complex job shops. *Production Planning and Control*, *22*(1), 25–40. https://doi.org/10.1080/09537287.2010.490015

Mortezaei, N., & Zulkifli, N. (2013). Integration of lot sizing and flow shop scheduling with lot streaming. *Journal of Applied Mathematics*, *2013*. https://doi.org/10.1155/2013/216595

Pujawan, I. N., & Smart, A. U. (2012). Factors affecting schedule instability in manufacturing companies. *International Journal of Production Research*, *50*(8), 2252–2266. https://doi.org/10.1080/00207543.2011.575095

Quadt, D. (2004). Lot-sizing and scheduling for flexible flow lines. In *Lot-Sizing and Scheduling for Flexible Flow Lines*. https://doi.org/ISBN: 978-3-642-17101-7

Ramya, R., Rajendran, C., Ziegler, H., Mohapatra, S., & Ganesh, K. (2019). Capacitated Lot Sizing Problems in Process Industries. In *Capacitated Lot Sizing Problems in Process Industries*. https://doi.org/10.1007/978-3-030-01222-9

Salido, M. A., Escamilla, J., Barber, F., & Giret, A. (2017). Rescheduling in job-shop problems for sustainable manufacturing systems. *Journal of Cleaner Production*, *162*, S121–S132. https://doi.org/10.1016/j.jclepro.2016.11.002

Schuh, G., Prote, J. P., Luckert, M., Hünnekes, P., & Schmidhuber, M. (2019). Effects of the update frequency of production plans on the logistical performance of production planning and control. *Procedia CIRP*, *79*, 421–426. https://doi.org/10.1016/j.procir.2019.02.115

Sivadasan, S., Smart, J., Huaccho Huatuco, L., & Calinescu, A. (2013). Reducing schedule instability by identifying and omitting complexity-adding information flows at the supplier-customer interface. *International Journal of Production Economics*, *145*(1), 253–262. https://doi.org/10.1016/j.ijpe.2013.04.043

Sridharan, V., & LaForge, L. (1990). An analysis of alternative policies to achieve schedule stability. *Journal of Manufacturing and Operations Management*, *3*(1), 53–73 DOI:10.1111/j.1540-5915.1988.tb00259.x.

Sridharan, S. v., Berry, W. L., & Udayabhanu, V. (1988). Measuring Master Production Schedule Stability Under Rolling Planning Horizons. *Decision Sciences*, *19*(1), 147–166. https://doi.org/10.1111/j.1540-5915.1988.tb00259.x

Tunc, H., Kilic, O. A., Tarim, S. A., & Eksioglu, B. (2013). A simple approach for assessing the cost of system nervousness. *International Journal of Production Economics*, *141*(2), 619–625. https://doi.org/10.1016/j.ijpe.2012.09.022

Valckenaers, P., Germain, B. Saint, Verstraete, P., Belle, J. Van, & Brussel, H. Van. (2009). *Computers in Industry Intelligent products : Agere versus Essere § , §§*. *60*, 217–228. https://doi.org/10.1016/j.compind.2008.12.008

van der Sluis, E. (1993). Reducing system nervousness in multi-product inventory systems. *International Journal of Production Economics*, *30–31*(C), 551–562. https://doi.org/10.1016/0925-5273(93)90120-A

Ventä, O. (2007). Intelligent products and systems: Technology theme - Final report. In *VTT Publications* (Issue 635) ISBN: 951387009X.

Wilensky, U. (1999). *"NetLogo." http://ccl.northwestern.edu/netlogo/. Center for Connected Learning and Computer-Based Modeling*. Northwestern University Evanston.

Wong, C. Y., Mcfarlane, D., Zaharudin, A. A., & Agarwal, V. (2014). *The intelligent product driven supply chain*. *November 2002*. https://doi.org/10.1109/ICSMC.2002.1173319

Xie, J., Zhao, X., & Lee, T. S. (2003). Freezing the master production schedule under single resource constraint and demand uncertainty. *International Journal of Production Economics*, *83*(1), 65–84. https://doi.org/10.1016/S0925-5273(02)00262-1

Yadav, A., & Jayswal, S. C. (2018). Evaluation of batching and layout on the performance of flexible manufacturing system. *International Journal of Advanced Manufacturing Technology*. https://doi.org/10.1007/s00170-018-2999-1

Ziarnetzky, T., Mönch, L., & Uzsoy, R. (2018). Rolling horizon, multi-product production planning with chance constraints and forecast evolution for wafer fabs. *International Journal of Production Research*, *56*(18), 6112–6134. https://doi.org/10.1080/00207543.2018.1478461

CAPÍTULO 6. CONCLUSIONES

La presente tesis doctoral propone una arquitectura holónica basada en productos inteligentes para el aumento de la flexibilidad en planificación de la producción y logística. Este sistema híbrido, permite simular sistemas industriales en los que los productos puedan estar activos, tomando decisiones y garantizando una cierta estabilidad de los planes de producción, así como de agilidad a nivel de taller. Esta estabilidad se debe a la capacidad de adaptación a perturbaciones internas o externas del sistema, a través de la implementación de nuevos paradigmas como los HMS, MAS, PDS y productos inteligentes, permitiendo la cooperación de los recursos para la obtención de objetivos en común.

El desarrollo de estudios que combinen los paradigmas de auto-organización, auto-adaptación y auto-optimización, como el PDS, podría ofrecer nuevas perspectivas no exploradas hasta ahora, ampliando así el espectro de herramientas para futuros trabajos. Si bien las medidas cualitativas y cuantitativas clásicas pueden ofrecer información valiosa, estas medidas sólo añaden valor verdadero si su evaluación es factible de implementar. En lo que se refiere a la implantación de nuevos sistemas de control, los trabajos futuros tendrán como objeto ingresar nuevos criterios de evaluación con el fin de representar fehacientemente los procesos de la industria.

Los artículos presentados en esta tesis se basan en una arquitectura de modelos de optimización, completamente descentralizados y distribuidos. Estos modelos son implementados a través de agentes inteligentes (representación virtual de un recurso físico) capaces de cooperar, organizar y tomar decisiones relevantes tanto para su ciclo productivo como para el de los demás agentes presentes en el modelo. Esto genera un espacio para una nueva línea de investigación la que busca hibridar metodologías para lograr planes de producción mucho más estables y capaces de afrontar las dinámicas de los mercados. Los artículos presentados obtuvieron las siguientes conclusiones.

En el artículo denominado "Implementation of a Holonic Product-Based Platform for Increased Flexibility in Production Planning" se analizó la necesidad de incrementar la flexibilidad en los sistemas de producción en arquitecturas altamente distribuidas. El modelo propuesto entregó una gran ventaja en los sistemas de producción sin la intervención humana. Más específicamente fue capaz de reorganizar lotes de producción en torno a un objetivo común. En la comparación de los resultados se utilizó un sistema de producción estándar resuelto a través de la técnica de lot streaming, al cual se le aplicó una perturbación en los tiempos de producción. Los resultados indicaron que la plataforma desarrollada puede responder satisfactoriamente a cambios en los tiempos de producción reduciendo un 10.95% el makespan post perturbación.

En el artículo denominado “A product-driven system approach to generate fast solutions to the job shop scheduling problem” se propuso un PDS para resolver el JSSP evaluando el makespan como medida de desempeño. El modelo propuesto incorporó los conceptos del producto inteligente con una función de inteligencia basada en el algoritmo SBH para la toma de decisiones. Los resultados obtenidos fueron comparados con técnicas optimas (programación entera), heurísticas (SBH) y reglas de despacho, comprobando que un modelo PDS con decisiones tomadas por productos inteligentes puede abordar problemas de programación de producción y brindar mayor flexibilidad. Se obtuvo mejores resultados que herramientas ampliamente utilizadas en la literatura en cortos tiempos de ejecución, permitiendo la adaptación a cambios no previstos. Particularmente, referentes al makespan se obtuvo resultados cercanos del óptimo en instancias con pocos recursos y mejores resultados que la programación entera y las heurísticas convencionales en instancias con más recursos.

En el artículo denominado “An adaptive product-driven system using evolutionary algorithms to increase the flexibility in scheduling problems at different scales”, se utilizó un modelo denominado PDS-EA para la toma de decisiones descentralizadas para la minimización del makespan en un JSSP. La función de inteligencia incorporada es un algoritmo evolutivo para la selección del mejor resultado. Se utilizaron 102 instancias de prueba ampliamente utilizadas en la literatura divididas en instancias de baja, media y gran escala. Los resultados se compararon con técnicas estándar, de modelos óptimos (programación entera), métodos aproximados (heurísticas) y reglas de despacho. La experimentación se basa en una simulación que analiza el gap en los primeros 10 min y el mejor resultado a 60 min de ejecución.

Los resultados de este artículo dependen de la escala en la cual se aplicó. En instancias de baja y media escala el modelo propuesto tiene un rendimiento cercano al método exacto, acercándose a la hora de tiempo de simulación. Sin embargo, con instancias de gran tamaño, el método propuesto supera al método exacto durante ese período. La arquitectura PDS-EA genera un sistema estable capaz de reaccionar a cambios sin intervención humana en periodos cortos de tiempo. Por tanto, PDS-EA puede resolver procesos de secuenciación en entornos de taller a diferentes escalas, adaptándose a las complicaciones de cada instancia.

En el artículo denominado “A product-driven system approach to reduce nervousness in master production Schedule”, se presentó un modelo PDS bajo el concepto del producto inteligente. Para la experimentación se generó un plan maestro de producción para simular condiciones reales de la industria. En una primera etapa se calcularon los resultados óptimos del lot-sizing para inicializar el modelo (menor costo posible). Sin embargo, cualquier cambio en el plan de producción obtenido impacta directamente en los costó, debido a que no posee la capacidad de responder a perturbaciones. Por esto, la metodología propuesta demostró que un modelo basado en productos inteligentes puede tomar las decisiones necesarias para generar un plan de producción flexible en conjunción con modelos óptimos como punto de partida. La evidencia experimental mostró que el modelo PDS logra reducir el nerviosismo sin incrementar el costo en una misma magnitud. Por ejemplo, con solo un incremento del 2.39% del costo lo logra obtener una reducción del nerviosismo de hasta un 11.42% en un control por ciclo.

Se considera para futuras investigaciones explorar instancias de problemas a real escala, incorporando nuevas perturbaciones y mejorando la toma de decisiones a través de una función de inteligencia más robusta. Esto debido a la necesidad de aportar conocimiento en necesidades reales de la empresa moderna. Si bien la evidencia experimental muestra que la arquitectura y las inteligencias utilizadas poseen la capacidad de obtener buenos resultados independiente de la escala del problema. Estos resultados también indican que serán necesarias investigaciones futuras para generar agentes con mayor capacidad de comunicación e implementar eventos más realistas en la planificación, como celdas de trabajo, productos simultáneos y fallas.

Por otra parte, la necesidad de más exploración en esta línea de investigación se debe a la versatilidad que se entrega en el plan maestro sin intervención humana generando flexibilidad y adaptación a cambios sin la necesidad de generar múltiples ejecuciones de modelos óptimos como el lot-sizing (modelos de harta carga computacional). Por esto es necesario generar agentes con funciones de inteligencia más complejas asociadas a herramientas de inteligencia artificial que nos entreguen mayor conocimiento del proceso optimizando no solo la nerviosidad del sistema, sino que otras variables de decisión importantes, como la producción a tiempo u optimización de recursos.

CAPÍTULO 8. REFERENCIAS

Barbosa, J., & Leitao, P. (2011). Simulation of multi-agent manufacturing systems using agent-based modelling platforms. IEEE International Conference on Industrial Informatics (INDIN), 477–482. https://doi.org/10.1109/INDIN.2011.6034926.

Cardin, O., Derigent, W., & Trentesaux, D. (2018). Evolution of holonic control architectures towards Industry 4.0: A short overview. IFAC-PapersOnLine, 51(11), 1243–1248. https://doi.org/10.1016/j.ifacol.2018.08.420

Gräßler, I., & Pöhler, A. (2017). Implementation of an Adapted Holonic Production Architecture. Procedia CIRP, 63, 138–143. https://doi.org/10.1016/j.procir.2017.03.176

Herrera, C. (2011). Cadre générique de planification logistique dans un contexte de décisions centralisées et distribuées. Université Henry Poincaré - Nancy . Francais, <NNT : 2011NAN10046>. <tel-01746192v2>.

Herrera, C., Thomas, A., (2010). Simulation of less master production schedule nervousness model To cite this version : HAL Id : hal-00511733 Simulation of Less Master Production Schedule Nervousness Model. https://doi.org/10.3182/20090603-3-RU-2001.0554.

Lee, K. K. (2008). Fuzzy rule generation for adaptive scheduling in a dynamic manufacturing environment. Applied Soft Computing Journal, 8(4), 1295–1304. https://doi.org/10.1016/j.asoc.2007.11.005

Mcfarlane, D., Sarma, S., Chirn, J. L., Wong, C. Y., & Ashton, K. (2002). THE INTELLIGENT PRODUCT IN MANUFACTURING CONTROL AND MANAGEMENT. IFAC Proceedings Volumes, 35(1), 49–54. https://doi.org/10.3182/20020721-6-ES-1901.00011

Muhuri, P. K., Shukla, A. K., & Abraham, A. (2019). Industry 4.0: A bibliometric analysis and detailed overview. Engineering Applications of Artificial Intelligence, 78(November 2018), 218–235. https://doi.org/10.1016/j.engappai.2018.11.007

Sáez, P., & Herrera, C. (2021). Implementation of a Holonic Product-Based Platform for Increased Flexibility in Production Planning. Studies in Computational Intelligence, 987, 180–194. https://doi.org/10.1007/978-3-030-80906-5_12