UNIVERSIDADE FEDERAL DE SÃO CARLOS

CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE PRODUÇÃO

ROBSON FLÁVIO CASTRO

A SIMPLIFIED DRUM-BUFFER-ROPE SYSTEM EVOLUTION PROPOSAL FOR PRODUCTION AND DISTRIBUTION TO AVAILABILITY

SÃO CARLOS - SP

2022

ROBSON FLÁVIO CASTRO

A SIMPLIFIED DRUM-BUFFER-ROPE SYSTEM EVOLUTION PROPOSAL FOR PRODUCTION AND DISTRIBUTION TO AVAILABILITY

Thesis presented to the Post Graduation Program in Industrial Engineering at the Federal University of São Carlos, as part of the requirements for obtaining the PhD degree in Industrial Engineering.

Supervisor: Professor PhD Roberto Fernandes Tavares Neto

SÃO CARLOS - SP

2022

Dedico esta tese à minha família pelo apoio incondicional durante o caminhar na busca das minhas realizações, e o acolhimento nos momentos mais difíceis.

UNIVERSIDADE FEDERAL DE SÃO CARLOS

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

Folha de Aprovação

Defesa de Tese de Doutorado do candidato Robson Flávio Castro, realizada em 18/10/2022.

Comissão Julgadora:

Prof. Dr. Roberto Fernandes Tavares Neto (UFSCar)

Profa. Dra. Juliana Keiko Sagawa (UFSCar)

Prof. Dr. Reinaldo Morabito Neto (UFSCar)

Profa. Dra. Susana Isabel Carvalho Relvas (ULisboa)

Prof. Dr. Leandro Callegari Coelho (ULaval)

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

AGRADECIMENTOS

A produção de uma tese de doutorado é árdua, exige muito trabalho e envolve várias pessoas. Essa tese exigiu anos de trabalho e muitas pessoas fizeram parte dessa caminhada. A elas devo minha eterna gratidão, pois, me ajudaram a chegar a um lugar que era distante e pensei até que fosse impossível. Minha família foi quem mais me apoio em todos os momentos, me ouvindo e ajudando a tomar decisões importantes. Obrigado meu pai Paulo e minha mãe Ângela, obrigado minha esposa Ana Paula e minha filha Alice, obrigado meu irmão Vagner, minha cunhada Laura e meus sobrinhos Tomás e Ana Luiza. Quero agradecer ainda à família da minha esposa, minha sogra Ana Maria, meu sogro Sério, aos meus cunhados Sérgio e Célia, e sobrinhos Nicolas, Mateus e Murilo. Amo vocês.

Não poderia deixar de agradecer aos meus mestres e orientadores prof. Roberto Tavares e prof. Moacir Godinho, por terem me aceitado como orientando, por todos os momentos em que estivemos juntos, pelas palavras de apoio, por me respeitarem, paciência e amizade, sem contar as excelentes contribuições dada a pesquisa. O prof. Roberto Tavares, além de tudo, é um ser humano incrível e sempre me tranquilizou nos momentos em que pensei que não seria possível chegar ao fim. O prof. Moacir Godinho é um grande ser humano, empreendedor nato, visionário e acredita que sempre é possível. Foi a primeira pessoa que me incentivou a ingressar no doutorado.

Convivi com vários professores do departamento, que me ensinaram e me apoiaram. Devo gratidão aos professores Alceu, Gilberto Ganga, Fábio Molina, Pedro Munari e Reinaldo Morabito. Agradeço ainda ao Robson e Lucas, da secretaria do PPGEP, pelo suporte com a documentação.

Outras pessoas a quem devo muito são o Dr. Carlos Augusto Hueb, a Drª Denise Rocha Belfort Arantes Brero e a Sônia Lazari pelo suporte emocional e contribuição com o meu crescimento e amadurecimento.

Em minha banca tive o prazer de contar com quatro grandes nomes da ciência, as professoras Susana Relvas e Juliana Sagawa, e os professores Leandro Coelho e Reinaldo Morabito, que prontamente aceitaram meu convite. Muito obrigado pela seriedade e profissionalismo com que se colocaram e pelas excelentes contribuições. Digo com muito orgulho que fazem parte da minha história.

Por fim, agradeço a todos que trabalham para manter o programa de pós-graduação em Engenharia de Produção da Universidade Federal de São Carlos a pleno vapor.

ABSTRACT

The Simplified Drum-Buffer-Rope (S-DBR) is a system that encompasses the planning, scheduling, production control, and control inventory replenishment in the supply chain. S-DBR offers distinct approaches for make-to-order and make-to-stock manufacturing environments. For this research, we focus on the method applicable to make-to-stock environments, known as Make-To-Availability (MTA) in S-DBR. MTA commits to ensuring high product availability. To adapt to merchandise distribution environments, the MTA has been expanded by incorporating new rules. While this expansion isn't considered a new method by its original authors, we refer to it as Distribution-To-Availability (DTA) to differentiate it from MTA. Our previous studies on S-DBR revealed operational challenges with both MTA and DTA. Therefore, this research aims to propose enhancements for these two methods, which are supported by four specific objectives. The first objective (i) is to conduct a systematic literature review to identify suggestions and proposed improvements for S-DBR and its predecessor, DBR. The review uncovered that the previously identified problems have yet to be addressed, and studies are scarce on S-DBR compared to DBR. One of the difficulties encountered by MTA is effectively managing manufacturing environments with sequence-dependent setup time. The second (ii) and third (iii) objectives are dedicated to tackling this challenge. Objective (ii) involves identifying dispatch rules suitable for MTA, while objective (iii) focuses on proposing a solution for MTA in sequence-dependent setup time environments. In the case of DTA, the challenge lies in planning the replenishment of stock buffers within the supply chain, particularly in the distribution network. S-DBR emphasizes the need for frequent inventory allocation but does not offer a solution that effectively balances transport costs and inventory maintenance costs. Consequently, the specific objective (iv) aims to propose a solution that enables DTA to address the trade-off between transportation costs and inventory maintenance costs, ensuring both product availability and profitability. The research results demonstrate that utilizing different dispatch rules can enhance the performance of MTA. Furthermore, our proposed solution for sequence-dependent setup time environments has improved the performance of MTA. Regarding DTA, the proposed solutions effectively ensure inventory supply in the distribution network by striking a balance between transport and inventory maintenance costs, thereby maintaining product availability and profitability.

Keywords: Theory of Constraints, Simplified Drum-Buffer-Rope, Make-to-Availability, Distribution-to-Availability.

Funding: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

RESUMO

O Simplified Drum-Buffer-Rope (S-DBR) é um sistema para o planejamento, programação e controle da produção, que também pode controlar a reposição de inventários na cadeia de suprimentos. O S-DBR propõe métodos diferentes para ambientes de manufatura make-to-order e make-to-stock. Nesta pesquisa, estamos interessados no método para ambientes make-to-stock, que no S-DBR é chamado de Make-To-Availability (MTA) porque assume o compromisso de garantir alta disponibilidade de produtos e oferece mecanismos específicos para tal. O MTA foi ampliado para se adequar a ambientes de distribuição de mercadorias, com a adição de novas regras. Embora essa extensão não tenha sido considerada um novo método pelos seus autores, decidimos chamá-la de Distribution-To-Availability (DTA) para distingui-la do MTA. Em nossos estudos anteriores sobre o S-DBR identificamos alguns problemas no modo de operação do MTA e do DTA. O objetivo desta pesquisa é propor melhorias para os dois métodos, que desdobramos em quatro objetivos específicos. O primeiro objetivo específico é (i) identificar as adaptações e melhorias propostas ao S-DBR e em seu precursor, DBR, através de uma revisão sistemática da literatura. Foi constatado pela revisão da literatura, que os problemas identificados anteriormente ainda não haviam sido tratados, além de haver poucos estudos sobre o S-DBR em comparação ao DBR. Uma das dificuldades do MTA é lidar com ambientes de manufatura onde há tempo de setup dependente da sequência. O segundo e terceiro objetivos específicos foram dedicados a esse problema. O segundo objetivo específico (ii) é identificar regras de despachos adequadas ao MTA. Esse estudo embasou o estudo relacionado ao terceiro objetivo específico, (iii) que é propor uma solução para o MTA voltada a ambientes com de tempo de setup dependente da sequência. No DTA, o problema está no planejamento da reposição dos buffers de estoque espalhados pela cadeia de suprimentos, mais especificamente na rede de distribuição. O S-DBR prega que a reposição de inventário deve ser frequente, porém não aponta uma solução que lide com o trade-off entre os custos de transporte e o custo de manutenção de inventário. Relacionado a isso, o objetivo específico (iv) é propor uma solução que habilite o DTA a enfrentar o trade-off entre os custos de transporte e os custos de manutenção de estoques, além de planejar a reposição de inventário para garantir tanto alta disponibilidade de produtos quanto lucratividade. Os resultados da pesquisa mostraram que o uso de diferentes regras de despacho pode melhorar o desempenho do MTA. Além disso, o desempenho do MTA melhorou com a solução que propomos para ambientes com tempo de setup dependente da sequência. Em relação ao DTA, as soluções propostas mostraram-se

capazes de planejar a reposição de inventário na rede de distribuição equilibrando os custos de transporte e de manutenção de inventário, mantendo a disponibilidade dos produtos e o lucro.

Palavras-chave: Teoria das Restrições, Simplified Drum-Buffer-Rope, Make-to-Availability, Distribution-to-Availability.

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

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AT	Arrival time
BS	Buffer status
CCR	Constraint capacity resource
CONWIP	Constant Work-in-Process
CWH	Central Warehouse
DBM	Dynamic Buffer Management
DBR	Drum-Buffer-Rope
DDR	Dispatching rule
DTA	Distribution-to-Availability
DWIP	Downstream work-in-process
FGI	Finished Goods inventory
FIFO	First-in-first-out
FPL	Full planned load
FR	Fill rate
FT	Flow time
HEA	Hybrid evolutionary algorithm
MIP	Mixed integer programming
MTA	Make-to-availability
MTO	Make-to-order
MTS	Make-to-stock
PO	Production orders
POLCA	Paired-cell Overlapping Loops of Cards with Authorization
POS	Point of sale
PPC	Production planning and control
PSO	Particle Swarm Optimization
PSO-S	Particle Swarm Optimization for Sequence
PSP	Prioritization by buffer status
PSP1	Prioritization by buffer status without downstream work-in-process
PWH	Plant Warehouse
RPL	Regular planned load
RT	Replenishment time
SB	Stock buffers
S-DBR	Simplified Drum-Buffer-Rope
SKU	Stock keeping unit
SLR	Systematic literature review
SPT	Shortest processing time
SRPT	Shortest remaining processing time
ST	Setup time
TL	Target level
TOC	Theory of Constraints
UT	Utilization
VBP	Virtual Buffer Penetration
WIP	Work-in-process

TABLE OF CONTENTS

1 INTRODUCTION

1.1 Context and fundaments

Supply chains are currently experiencing significant transformations, including the pursuit of greater resilience, the rise of e-commerce, the adoption of omnichannel sales, the diversification of distribution channels, market pressures to reduce delivery times, and the introduction of new types of vehicles for transporting goods. Within this context, managing the production and distribution of goods while maintaining business profitability and a high service level has become increasingly challenging. To address these challenges, businesses are turning to management and decision-making tools, such as the Theory of Constraints-based system known as Simplified Drum-Buffer-Rope (S-DBR) (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). This research is dedicated to studying S-DBR and proposing solutions that contribute to its wider adoption. The fundamental principles of S-DBR will be present in the following sections.

1.1.1 Theory of Constraints

The S-DBR emerged from the Drum-Buffer-Rope (DBR), a system based on the Theory of Constraints (TOC) (GOLDRATT; COX, 1984; GOLDRATT, 1986; SCHRAGENHEIM, 2010). The notion of constraint is fundamental to understanding TOC and its applications. Cox et al. (2012, 28), editors of the Theory of Constraints International Certification Organization (TOCICO) Dictionary, define constraint as the factor that ultimately limits the performance of a system or organization. The factor that, if the organization were able to increase it, that is more fully exploit it or more effectively subordinate to it, would result in achieving more of the goal (IKEZIRI et al., 2018).

Conceived by Israeli physicist Dr. Eliyahu Moshe Goldratt, the TOC is based on the application in organizations of experimental science concepts (IKEZIRI et al., 2018). Its roots go back to the 70s, when Goldratt and his team developed a finite programming software for the optimization of production systems, called Optimized Production Technology (OPT) (COX; SCHLEIER, 2010). In 1984 the book The Goal was published, which presents a series of concepts focused on explaining certain phenomena that govern manufacturing and proposed a process of continuous improvement and decision-making support for organizations (GOLDRATT; COX, 1984). Since then, its application has broadened into various areas, such as production, supply chain, projects, accounting, distribution, and retail. Thus, the primary focus of TOC has moved on from factory bottlenecks, production planning, control, and scheduling techniques to becoming a global management philosophy focused on leveraging performance and offering decisive competitive advantages to organizations (de Souza and Pires 2010; Goldratt 1990a).

1.1.2 Theory of Constraints and manufacturing – Drum-Buffer-Rope

The implementation of the Theory of Constraints (TOC) in a manufacturing environment has significant implications for the evaluation of workstation performance (GOLDRATT; COX, 2003). According to Schragenheim (2010), the factory control systems used in the late 1970s and early 1980s rewarded individual workstation efficiency, leading to excessive inventory between stations. As a result, the Drum-Buffer-Rope (DBR) system emerged, focusing on workflow management rather than capacity. The DBR maximizes production line flow to generate company revenue while minimizing work-in-progress (WIP) in the production process (SCHRAGENHEIM, 2010). This approach prevents dispersion and ensures that the entire process operates at a pace that supports the slowest resource in the system.

Dettmer (2000) considers the Drum-Buffer-Rope (DBR) as one of Goldratt's most renowned constraint management tools. Goldratt and Cox (2003) explain the DBR mechanism using an analogy of scouts walking in a line, where the slowest scout has a rope tied to their waist, and the other scouts follow at the same pace. The drum sets the rhythm for the slowest scout, while the rope prevents the scouts from dispersing and establishes a buffer, which represents the maximum distance. In the manufacturing environment, the drum is considered a capacity constraint resource (CCR) (SAIF; YUE; AWADH, 2022). The flow of the manufacturing line is limited by the CCR.

1.1.3 Simplifying the Drum-Buffer-Rope

Despite the numerous successful implementations of the DBR around the world, the presence of an internal CCR is assumed even in situations in which this assumption does not hold (LEE et al., 2010a). For this type of situation, Schragenheim and Dettmer (2000) proposed a new approach based on the DBR but with relevant simplifications, naming the approach the Simplified Drum-Buffer-Rope.

In the comparison between simplified drum-buffer-rope (S-DBR) and DBR, it is important to understand why S-DBR is considered even simpler. Both approaches require a "drum" representing the rate of demand from the market, a "buffer" to protect the drum, and a "rope" to signal the release of materials for new orders. So, what sets S-DBR apart as "simple"? The simplicity of S-DBR arises from the elimination of two types of buffers, namely the capacity-constrained resource (CCR) buffer and the assembly buffer. Additionally, S-DBR employs a market-driven master production schedule instead of the CCR (or drum) schedule, resulting in clearer and more streamlined shop floor control.

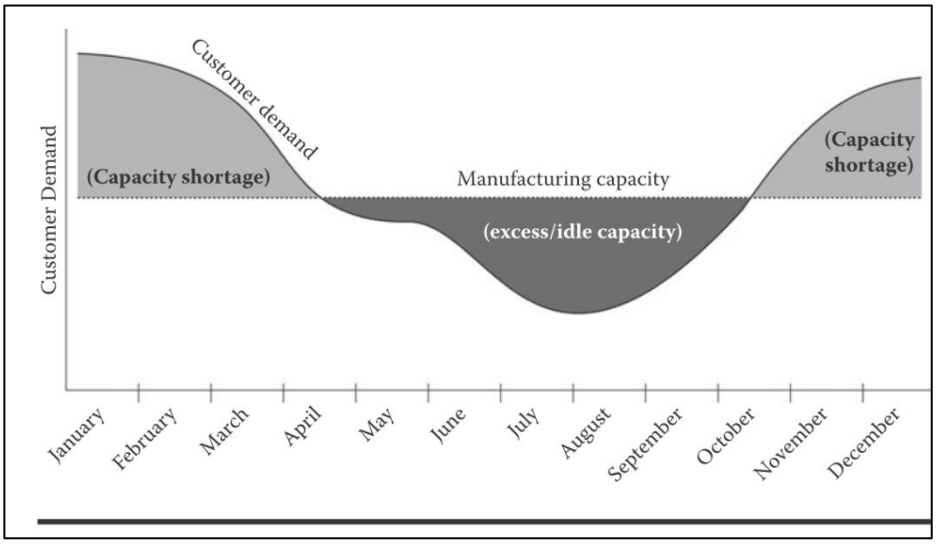
Schragenheim (2010) explains that S-DBR does not rely on a detailed schedule specifically for the CCR, even when the CCR is active within the system. Instead, the production line's overall load is controlled through the planned load mechanism. Planned load encompasses all orders that have been released but have not yet been processed by the CCR. By controlling the planned load, S-DBR effectively maintains capacity control without the need for a detailed CCR schedule. This approach safeguards production against variabilities and helps prevent issues such as lateness or stockouts. Overall, S-DBR offers a simpler yet effective means of managing the shop floor and ensuring efficient production processes.

The book "Manufacturing at Warp Speed: Optimizing Supply Chain Financial Performance" (SCHRAGENHEIM; DETTMER, 2000) introduces the concept of S-DBR and explores the integration and coordination of manufacturing, marketing, sales, purchasing, and finance to maximize sales revenue by effectively managing inventories and indirect costs. The book equips readers with the necessary tools to implement this strategy.

While S-DBR makes manufacturing management easier, it is not always the best choice. DBR is well-suited for situations where demand consistently exceeds the plant's capacity. On the other hand, S-DBR recognizes that demand may not always utilize the entire production capacity, particularly for certain companies (SOUZA; BAPTISTA, 2010). This concept is depicted in Figure 1.1, where the dashed line represents the plant's capacity.

Companies can experience seasonal peaks in demand, while others may have smoother demand patterns. In either case, during periods of high demand, production capacity may fall short of meeting the requirements. Managers are always faced with the challenge of aligning demand and capacity (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). One of the assumptions made by S-DBR is that market demand always acts as a constraint, which may be disguised by occasional demand peaks that introduce constraints into the system for a short duration (SCHRAGENHEIM; DETTMER, 2000).

Figure 1.1 - Demand Variations on Production Capacity



Source: Schragenheim, Dettmer and Patterson (2009)

In other words, companies often have idle capacity for most of the time, and therefore, the market must always be seen as a constraint within the system. Consequently, it is crucial to explore the market to increase sales and profitability. When idle capacity exists, the potential gain becomes limited by the level of demand.

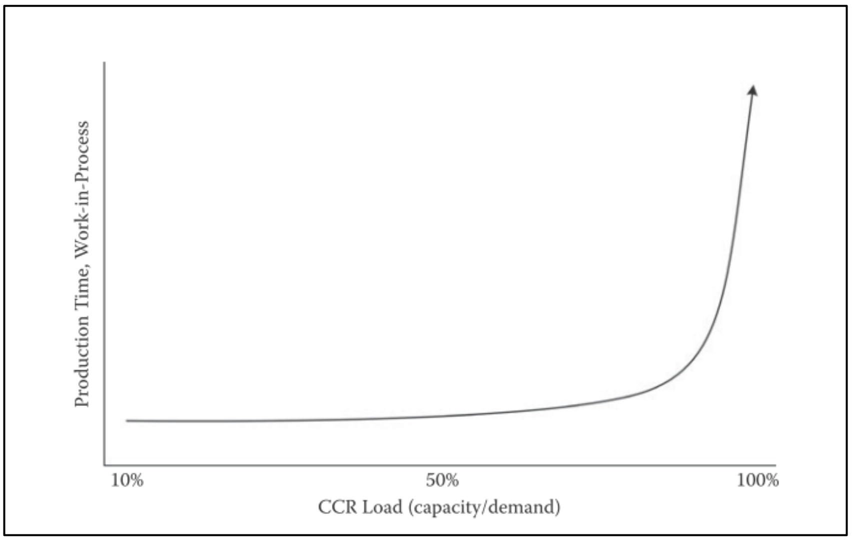
Looking at it in this light, it makes more sense to tie the rope to market demand rather than consider the uses of production resources. Furthermore, by accepting that the market is always a constraint on the system, one is asserting that it is part of the system whose purpose is, among other things, to make more money. Capacity constraints, therefore, must be understood as interactive constraints on the main market constraint, weakening the system.

Recognizing that market demand is the constraint of the system, it is important to maintain excess capacity in the CCR and, consequently, in all other resources. The system should not be fully loaded, even if demand requires greater capacity to meet it. The reason is that when a production process approaches 100% load, queues for orders, replenishment orders, and WIP start to increase exponentially, in addition to creating delays in deliveries to customers (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). Figure 1.2 illustrates this phenomenon.

Market demand fluctuates and there is always a risk of placing a large burden on the RRC. For this reason, its cargo must be carefully monitored, and market commitments must be limited (SCHRAGENHEIM; DETTMER, 2000). However, Souza and Baptista (2010), quoting Goldratt and Cox (2003), state that there must be only one restriction in the system. As demand must be understood as a permanent constraint, some excess capacity must be guaranteed in the CCR. This excess is called Protective Capacity (SCHRAGENHEIM; DETTMER, 2000).

Invariably, a CCR can also be considered a constraint on demand, as it also limits the supply of products to the market.

Figure 1.2 - Evolution of production time versus increased demand



Source: Adapted from Schragenheim, Dettmer, and Patterson (2009)

1.1.4 Simplified Drum-Buffer-Rope for make-to-stock

In 2009 was launched the book "Supply Chain Management at Warp Speed: Integrating the System from End to End" (SCHRAGENHEIM; DETTMER, 2009), amplified the S-DBR. The main objective of this book is to show how the Theory of Constraints (TOC) can be integrated with the distribution of finished products, the acquisition of raw materials, and the manufacturing process, and present the concept of make-to-availability (MTA). When the TOC-based planning system was rethought, it was realized that the make-to-stock (MTS) environment needed different principles (SCHRAGENHEIM, 2010), and thus born the concept of MTA.

According to Schragenheim (2010), the DBR does not see the difference between MTS and MTO strategies. At the time DBR was developed in the 1980s, there was no dispute over the assumption that there was no difference between servicing MTO and MTS environments. However, when Goldratt was dealing with the MTS theme and verifying the differences between this form of production and the MTO, one of his findings gave rise to the term MTA, where more marketing information was added to the operational meaning of MTS: “we assume a commitment with the market chosen by us to maintain a correct availability of a group of specific final products in a specific warehouse”. Thus, the objective of the MTA system is to offer a commercial opportunity to the company, allowing it to offer and guarantee immediate deliveries to customers. In some markets, this value offered to customers becomes a competitive advantage.

DBR and S-DBR are still best applied in MTO production environments, particularly in small and medium enterprises. They get better results with MTO. This does not happen with large companies, which often sell their products to retail chains, which in turn are served by distribution centers. So, production is always going into a warehouse somewhere. These industries are part of a long and complex supply chain. For this reason, the creation of an S-DBR approach for MTS environments was an important and natural evolution (SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

1.1.5 Simplified Drum-Buffer-Rope for distribution

The book titled "Supply Chain Management at Warp Speed: Integrating the System from End to End" (SCHRAGENHEIM; DETTMER; PATTERSON, 2009) introduces the extension of S-DBR to distribution environments. According to the authors, merely manufacturing products does not guarantee sales, even if the products are in high demand. The decision of whether to deliver products directly to end users or store them in warehouses is an obvious one. Without sales to consumers, finished goods inventories remain untouched, and future production of those products eventually ceases. This is where distribution networks play a crucial role in assisting producers, especially in the case of consumer products.

This approach takes advantage of the fact that relative variation in demand is significantly smaller for the manufacturer compared to typical retailers. While there may be less stock in the system, the availability of the product to the retailer is increased through frequent deliveries. Essentially, this process applies DBR principles throughout the supply chain (SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

In the TOCICO Dictionary, the distribution/replenishment solution in the Theory of Constraints (TOC) is defined as a pull-based method that involves determining stock buffer sizes and replenishing inventory within the supply chain based on actual consumption by end-users rather than forecasts. Each link in the supply chain maintains stock levels that accommodate the maximum expected demand within the average replenishment time, accounting for the level of unreliability in replenishment time. Generally, each link receives what was shipped or sold, with adjustments made when buffer management detects changes in demand patterns (SULLIVAN et al., 2007).

To maintain optimal inventory levels in each link, the TOC distribution solution relies on the continuous replenishment of consumed stocks from stock buffers. In this research, we referred to this solution as distribution-to-availability (DTA).

1.2 The S-DBR literature

The S-DBR is discussed in a few numbers of studies. The first one was realized by Chang and Wen-Tso (2011). They proposed a weighted layer production buffer and weighted production buffer to monitor the status of the buffer deviation in a re-entrant flow shop operated by S-DBR. Soon after, Chang and Huang (2011) proposed a solution to operate a re-entrant manufacturing environment with S-DBR. According to the authors, the S-DBR is not prepared to operate in these environments, due to its dispatching rule. Hence, they proposed the layer production buffer to monitor the buffer status and sequence orders according to it. Chang and Huang (2013) used the S-DBR to enhance a model for a re-entrant flow shop (RFS) environment. In the model, job processing times are generated from a discrete uniform distribution and machine breakdowns are subject to an exponential distribution. Improvements were made to the due-date assignment method, the order release, and the dispatching rules. An alternative method that enhances the S-DBR system performance was developed by J.-H. Lee et al. (2010). The new S-DBR approach presents rules for the operation of a fluctuating make-to-order environment with interactive or multiple CCRs, as well as rules for the inclusion of urgent orders.

Ikeziri et al. (2019) extensively reviewed the literature on the Theory of Constraints and showed that the S-DBR has received little attention from researchers. Recently, Govoni et al. (2021) compared the performance of a production system managed by make-to-availability under the action of different methods of continuous improvement. Finally, Ikeziri et al. (2021) evaluated the effectiveness of the Dynamic Buffer Management (DBM) method, which is responsible for adjusting the target level over time.

1.3 Research questions

In their book "Supply Chain Management at Warp Speed," Schragenheim, Dettmer, and Patterson (2009) indicate potential solutions to implement MTA and DTA methods. However, they do not support critical decision-making in real-world settings, nor do they offer empirical evidence to substantiate the effectiveness of these approaches or validate the anticipated benefits. To address the gaps left by Schragenheim, Dettmer, and Patterson (2009), we initiated this study by conducting a comprehensive literature review on S-DBR to identify solutions and evidence. To direct our inquiry, we formulated the following research question:

RQ1: What adaptations and improvements have already been proposed to the Drum-Buffer-Rope and Simplified Drum-Buffer-Rope systems?

To address this inquiry, we conducted a systematic literature review (SLR), the findings of which are outlined in Chapter 3. The outcomes revealed that the previously identified issues remained unresolved. Subsequently, the next phase of the study involved selecting the problems to be tackled. Among them, two were chosen and are elaborated upon below.

1.3.1 Make-To-Availability research questions

One notable issue within the MTA that has captured our attention is the detrimental impact of sequence-dependent setup time on availability. This challenging situation introduces significant variability in replenishment time, resulting in a sluggish and unreliable system. To mitigate this, production orders must be processed in a specific sequence to minimize setup time. Deviating from this sequence leads to wastage of workstation/machine capacity (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). Conversely, adhering strictly to the sequence can result in excessively long replenishment times, hindering prioritization based on buffer status and ultimately affecting availability commitment (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). If an urgent production order is placed at the end of the processing queue, the risk of stock depletion becomes significant. Thus, a tradeoff arises: either minimizing setup time and expediting the flow time of all orders or streamlining the order with higher urgency while delaying the remaining orders in the queue.

There are at least two ways to solve this problem: increasing inventory or increasing production capacity to reduce replenishment lead time. We must consider that the increase in stock generates the need to increase physical space and maintenance costs in addition to the cost of the stock itself. Inventories also decrease if production capacity is increased to reduce replenishment lead time. In any case, the dependent setup time negatively impacts the MTA as it requires some investment.

The systematic literature review showed that no proposed solution mitigates the impact of dependent setup time on MTA performance, even though this may be a common problem for companies that adopt it.

In the MTA, the sequencing of production orders occurs machine-to-machine, guided by the logic of Prioritization by Buffer Status (PBS) (SCHRAGENHEIM; DETTMER; PATTERSON, 2009) when selecting the next order for processing. PBS can be viewed as a dispatch rule (DR) utilized in production scheduling to prioritize tasks within workstation queues (HEGER et al., 2016; NGUYEN, 2017). Unlike optimization methods, DRs make decisions based on real-time information when machines are idle and tasks are in the queue, rather than pre-determined by an algorithm (NGUYEN, 2017). This mechanism enables DRs to leverage up-to-date system data and make decisive choices (NGUYEN, 2017). DRs are easy to implement and generally offer faster results, presenting a significant advantage (FRAMINAN; FERNANDEZ-VIAGAS; PEREZ-GONZALEZ, 2019).

Given the resemblance between PBS and dispatching rules, we opted to explore potential solutions within the realm of dispatching rules. This led us to two research questions:

RQ2: Which dispatch rules are suitable to be adopted by Make-To-Availability?

RQ3: What solution can be adopted by Make-To-Availability to lead with sequence-dependent setup time environments?

1.3.2 Distribution-To-Availability research question

Supply chains are undergoing significant changes, such as the search for more resilience, growth of e-commerce, omnichannel sales, diversification of distribution channels, market pressure to reduce delivery times, and new types of vehicles to transport goods. These changes have made the distribution of goods more complex, and increasingly challenge the management distribution of goods. The challenges include managing multi-sites distribution chain coordinate and synchronization, and the control inventory of multiple distribution centers, regional warehouses, and retailers.

To overcome these challenges, the companies can use systems such as Vendor-Managed Inventory (VMI), Collaborative Planning, Forecasting and Replenishment (CPFR), Just-in-time (JIT), and DTA. Vendor-Managed Inventory (VMI) is a system in which the supplier takes responsibility for managing inventory levels at the customer's location (GOVINDAN, 2013). Another system is Collaborative Planning, Forecasting, and Replenishment (CPFR), which involves collaboration between trading partners in the supply chain to develop a shared understanding of demand and inventory requirements (HOLLMANN; SCAVARDA; THOMÉ, 2015). CPFR includes joint planning, demand forecasting, and inventory replenishment. Just-in-time (JIT) is a policy of ordering stock in needed quantities only when needed (GOLHAR; STAMM, 1991). On the other hand, DTA seeks to improve the availability of items at all points of consumption (end users) based on the constant replenishment of consumed stocks from strategically positioned stock buffers in the supply chain (COX; SCHLEIER, 2010).

Another challenge in distributing goods is balancing cost, profit, and service level. In this sense, there is a vast literature that discusses the logistics tradeoffs. One of the most important and discussed is the tradeoff between transportation costs and inventory holding costs (CARDÓS; GARCÍA-SABATER, 2006; CHOUDHARY; SHANKAR, 2013; MOSCA; VIDYARTHI; SATIR, 2019; QIU et al., 2022; SARKAR et al., 2019; TURKENSTEEN; VAN DEN HEUVEL, 2023). If the replenishment frequency is increased, transportation costs increase too, and inventory holding costs are reduced. On the other hand, decreasing the frequency of replenishment reduces transportation costs, which require higher inventory levels at upstream nodes of the network, increasing inventory holding costs. The ideal is to keep costs low enough to ensure the desired profit and level of service.

DTA is not yet able to handle this trade-off due to the way it makes decisions about inventory replenishment. As DTA does not use demand forecasting techniques, the pressure on the stock replenishment function is high, as it is responsible for making all decisions about what, when, and how much to restock (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). Theoretically, whenever a single unit of inventory is consumed at a distribution network node, an action to replenish that item should be initiated immediately. Stock replenishment involves a series of resources from collecting products from the warehouse to delivering them to the final customer or point of sale.

It is vital to plan resupply to optimize resource usage and balance cost, profit, and service level as discussed earlier. Although it is essential to plan the replenishment, DTA does not have a planning solution, according to a systematic review of the literature (chapter 3). Schragenheim, Dettmer, and Patterson (2009) argue that frequent replenishments keep inventories low and buffers full, but do not point to a tool for making replenishment decisions considering the costs involved. Therefore, DTA is not able to resolve the tradeoff between transportation costs and inventory holding costs.

Deciding when to replenish stock buffers is a difficult task that requires much more than a policy; tools are needed. For companies that intend to implement DTA, it is vital to have a tool that balances availability and profitability. The lack of this tool motivated us and led us to the following research question:

RQ4: How to replenish stock buffers in a distribution network managed by DTA, protecting the availability of products and the business's profit?

To answer this question, we propose a mixed integer programming (MIP) and computational heuristic solution based on the studies of Fachini and Armentano (2020) and Koç et al. (2015). We intend to contribute to the literature on S-DBR, logistics, and supply chain literature. We also hope to contribute to companies that need alternative solutions to the distribution of goods.

1.3.3 Objectives

Based on the above considerations, the main objective of this research is ***to propose improvements for S-DBR system methods, make-to-availability, and distribution-to-availability.*** For this, the following specific objectives need to be achieved:

Specific objective 1 (RQ 1): Identify the proposed adaptations and improvements for the DBR and S-DBR systems and understand the reasons that led to them.

Specific objective 2 (RQ 2): Evaluate the S-DBR/MTA behavior incorporating different dispatching rules.

Specific objective 3 (RQ 3): Develop a dispatching method for S-DBR/MTA to sequence dependent-setup time environments.

Specific objective 4 (RQ 4): Propose a solution to the S-DBR/DTA capable of planning the replenishment of stocks to guarantee the availability of the products and the business's profit.

Each objective will be studied in a chapter, namely: specific objective one is studied in Chapter 3 through a systematic literature review; objective 2 is studied in Chapter 4, objective three in Chapter 5, and objective four in Chapter 6.

1.4 Contributions and ineditism

The S-DBR encompasses manufacturing and distribution environments, unlike other systems that focus only on manufacturing. It has emerged more recently, and few scientific works are dedicated to S-DBR, different of Kanban, CONWIP, and POLCA. Its main literature is composed by Lee et al. (2010), Chang and Wen-Tso (2011), Chang and Huang (2011), Chang and Huang (2013), Ikeziri et al. (2019), Govoni et al. (2021) and Ikeziri et al. (2021). Little is known about S-DBR, and little has been studied by science. However, this thesis contributes to the advancement and expands the scientific literature of the S-DBR. For companies and practitioners, the research points to solutions to practical problems that can help improve financial results, while maintaining or increasing service levels.

The originality of the thesis is based on the results of a systematic literature review, that confirms that there are no answers in the scientific literature to research questions RQ2, RQ3 and RQ4.

1.5 Research Method

This research has two stages: i) identification of the literature on DBR and S-DBR and ii) proposition of improvements in S-DBR. The first stage meets objective 1 and uses a systematic literature review. At this stage the studies are classified according to the mechanisms affected - Drum, Buffer, or Rope. The motivations that led to these adaptations were analyzed giving rise to a set of research avenues. The second stage meets objectives 2, 3 and 4, using discrete event simulation and computational experiments to evaluate solutions.

A systematic review seeks to identify, select, and critically evaluate relevant research. When included in the review, significant data are collected and analyzed in each study to generate a better understanding of the subject. The systematic review uses the method of Tranfield, Denyer and Smart (2003), which suggests the following steps:

a) Topic selection and subject definition.
b) Explore publications and databases of publications.
c) Create an organization and classification of each publication.
d) Classify documents and make a structure for the topic.
e) Analyze and criticize obtained structure.
f) Show results and future research.

Objective 2 was pursued by employing computational simulation to evaluate the performance of S-DBR/MTA operations under various dispatching rules. In the selection process of dispatching rules, it was considered that S-DBR/MTA does not assign deadlines or completion dates to tasks, making such information unnecessary for the rules. The evaluation of dispatching rules was conducted using a simulation model of a flow shop production line implemented in Python 3.5, utilizing the SimPy library version 3.0.10. The production line configurations were adopted from Nguyen et al. (2015) and Thürer and Stevenson (2018). For each rule, a simulation scenario was created and replicated 50 times to gather data. The analysis of the scenario results was based on the average values of service level (S), average system stock ($\bar{E}$), average flow time ($\bar{F}$), and the inventory rate as a percentage of the service level ($\bar{C}$). Descriptive statistics and box plot graphics were employed for the analysis.

To address objective 3, a solution based on the Particle Swarm Optimization (PSO) metaheuristic was developed. To evaluate the performance of the PSO in relation to the MTA dispatch rule, simulations were conducted with thirty replications and scenarios involving variations in the position of the bottleneck workstation. The results were analyzed using descriptive statistics, percent deviation, and the boxplot plot for the mean values of mean flow time, mean setup time, work in process, fill rate, target level, and machine utilization rate indicators.

Objective 4 involves addressing an inventory replenishment planning problem, which entails making decisions regarding what items to replenish, how much to replenish, where to replenish them, and how to route the delivery vehicles. Two solutions were proposed, drawing from the works of Fachini and Armentano (2020) and Koç et al. (2015). The first solution is a Mixed Integer Programming (MIP) model that integrates all decision-making aspects. To further enhance the planning process, a second solution was devised, which follows a two-step approach. In the first step, a MIP model is utilized to determine the replenishment details, including what items, how much, and where to replenish. In the second step, a metaheuristic algorithm is employed to optimize the vehicle routing. Both solutions were implemented using Python 3.8 and Cplex 20.1.0, executed on a computer with 8 gigabytes of RAM and an Intel Core i5 2.40 GHz CPU. These solutions were applied to 18 instances, and the computational performance was evaluated in terms of CPU time. The results obtained from the instances were analyzed using descriptive statistics and percentage deviation techniques.

1.6 Thesis structure

The thesis consists of six chapters, in addition to Chapter 1. Chapter 2 provides a comprehensive explanation of the MTA (Make-to-Availability) and DTA (Demand-Driven Adaptive Enterprise) concepts, which are crucial for understanding the research undertaken. In Chapter 3, the systematic literature review findings are presented, including an analysis of studies proposing adaptations and enhancements for the DBR (Drum-Buffer-Rope) and S-DBR (Simplified Drum-Buffer-Rope) methodologies. Additionally, research gaps are identified and discussed. Chapter 4 focuses on evaluating the impact of different dispatching rules on the behavior of S-DBR/MTA systems. The results of this research were presented at the 51st Brazilian Symposium on Operational Research in 2019, and the paper has been accepted for publication in the European Journal of Industrial Engineering, scheduled for 2023. Chapter 5 outlines the steps taken to develop a dispatching method specifically tailored for S-DBR/MTA systems operating in environments with sequence-dependent setup times. The findings of this work were announced in the Journal of Intelligent Manufacturing in 2020. Chapter 6 is dedicated to the development of solutions for replenishment planning for the DTA. The results of this research will be submitted to a journal in the future. Lastly, Chapter 7 offers concluding remarks and presents overall insights derived from the research conducted.

2 BACKGROUND

This chapter explains the principles, concepts, and way of working of the MTA and DTA methods that fundament this research.

2.1 Make-to-availability

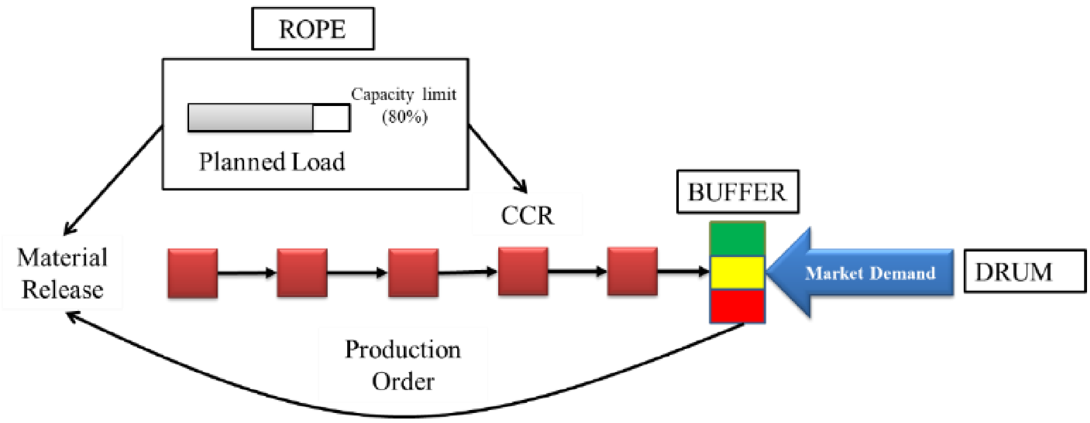
On the manufacturing management side, S-DBR works as a production planning and control (PPC) system (PEETERS; VAN OOIJEN, 2020). PPC activities aim to define what, how much, and when to produce, buy, and deliver so that the company can match manufacturing performance with customer demands (BUENO; GODINHO FILHO; FRANK, 2020). The PPC function is responsible for making decisions regarding planning, starting, controlling, monitoring, scheduling, and reprogramming of production planning, and ensuring the delivery of the products of a manufacturing company (BONNEY, 2000).

Two key guidelines sustain the tactical aspect of the MTA approach (SCHRAGENHEIM, 2010):

1) Production needs to focus on flow, making the orders flow as quickly as possible through the factory, until reaching the warehouse for finished products.
2) Unless there is a good reason to believe demand has changed or will change, a simple and direct way to react to any sale is to replenish whatever was sold.

Figure 2.1 illustrates the operation of the MTA, which meets demand through stock buffers supplied by the factory.

Figure 2.1 - MTA mechanisms



Source: Proposed by the author

MTA operates in cycles divided into four stages (SCHRAGENHEIM; DETTMER; PATTERSON, 2009):

3) The beginning of the cycle occurs with customer orders' arrival, which is served by stock buffers.
4) A stock replenishment request is sent to the factory as soon as products are removed from the buffer. The demand (orders arrival frequency) dictates the manufacturing pace, which is called a drum.
5) Replenishment requests become production orders at the factory, managed by the planned load, which represents the rope. The purpose of the rope is to avoid overloading production resources and align demand with production resources capacity. According to the planned load, production orders are released - workload on the slowest production resource or constraint capacity resource (CCR). The planned load is the sum of production orders that are in progress and have not yet been processed by CCR. The orders are released until the planned load reaches 80% of the average replenishment time, to avoid overloading the CCR. Replenishment time is the period between the sale of an item from the stock buffer and the replacement of that item.
6) The cycle ends with the fulfillment of production orders and the arrival of products to the stock buffer.

Defining the right size of inventory is very important to MTA. The “right” size of any inventory is determined primarily by two variables: demand and supply response time (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). Demand includes average consumption and its fluctuation range. Replenishment time refers to the time from inventory consumption until replacement. Shorter replenishment times reduce the need for clients to hold large inventories (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). However, fluctuations in demand must also be considered.

Five principles guide an effective planning and control process to support MTA and minimize the risk of damage from understock or overstock. They are (SCHRAGENHEIM; DETTMER; PATTERSON, 2009):

1. Inventory and replenishment time are closely correlated: shorter replenishment times lead to smaller inventories, ensuring availability and reducing lost sales. They also enable more accurate demand projections. On the other hand, longer replenishment times require larger volumes of finished stock to maintain availability. The stock level should be sufficient to cover potential demand during the time required for replenishment. As replenishment times increase, more inventory is needed to guarantee sales and mitigate demand fluctuations. Therefore, minimizing finished stock levels requires shortening production time as much as possible.

2. Work-in-process supplements protection of availability: protecting product availability involves not only maintaining finished inventory but also considering production orders in progress. Work-in-process inventory is necessary to ensure availability alongside finished stock. The rationale is that while work-in-process may not be immediately available, much of it is close to completion. To ensure availability, a fixed amount of stock, known as the target level (TL), is maintained in both finished goods and the production pipeline. Although actual finished stock may fluctuate, the overall system remains stable. As finished inventory is consumed, completed production orders replenish it to prevent stock-outs while avoiding excessive accumulation of finished inventory.

3. Tomorrow will be like today: the short-term forecast relies on established stock targets to ensure availability, considering both demand and supply. Unless there is a clear indication of a trend change, the current stock target is assumed to be correct for short-term availability. Each item produced should have a target level of units in either finished inventory or the production pipeline. Consumption of finished inventory triggers the generation of new production orders to maintain a relatively constant total stock in the system. The goal is to respond quickly to consumption, minimize replenishment time, and maintain stock stability.

4. Status of finished inventory dictates production floor priorities: the prioritization of work orders in production is determined by the deviation from target levels for each item. Three basic priorities are defined:

a. Green: If finished inventory exceeds two-thirds of the target level, it is considered higher than necessary. Replenishment is not urgent, and it may indicate excess finished goods inventory. Work-in-process for items with "green" finished inventory has low priority compared to other orders.

b. Yellow: If on-hand finished inventory is between one-third and two-thirds of the target level, it is considered normal. A yellow status implies no urgency or excessive inventory. Yellow status orders take priority over green orders but do not require management intervention.
c. Red: When finished inventory falls below one-third of the target level, the risk of stock-outs increases. Immediate action is needed to restore red inventory to yellow or green. Management intervention may be required for red orders.

These prioritization categories help determine the urgency of replenishment and guide decision-making in production to maintain optimal inventory levels and availability. Figure 2.2 demonstrate these concepts:

Figure 2.2 - The regions of a Stock Buffer

Source: Proposed by the author

5. Stagnation is undesirable: while items may occasionally fall into the red or green zones, prolonged residence in either condition indicates the need to adjust the inventory target level. If an item consistently stays in the red zone, it suggests a high risk of stock-outs, necessitating an increase in the inventory target level. Conversely, if an item consistently remains in the green zone, it indicates excess inventory, and the target level should be decreased. Regular monitoring and adjustments of the inventory target levels help maintain a balanced and efficient inventory management approach.

2.1.1 Make-to-availability operation

I) Determining the appropriate inventory

The concept of replenishing consumed items or products has implications for maintaining a fixed inventory in the shop. This fixed inventory encompasses both finished goods and work-in-process throughout the shop floor. It means that the necessary parts and assemblies required to fulfill committed availability may exist at different stages of completion. The total fabricated inventory represents the quantity needed for committed availability (SCHRAGENHEIM, 2010).

While it is ideal to have a fixed stock of finished goods, this is impractical as demand depletes the inventory, necessitating replenishment. Therefore, the concept of a "stock buffer" is introduced as a protective mechanism to ensure availability. The stock buffer consists of the total amount of finished goods and work-in-progress (WIP), serving as a straightforward approach to implement the appropriate protection mechanism (SCHRAGENHEIM, 2010).

The stock buffer, referred to as the target level, plays a crucial role in maintaining availability from the time an item is sold until its replenishment arrives at the finished-goods warehouse. The average time required to replenish an item is known as the replenishment time. The target level should not only consider the average demand within the replenishment time but also account for potential sales and situations where replenishment might take longer than the average replenishment time (SCHRAGENHEIM, 2010).

Schragenheim (2010) proposes two approaches for determining the target levels. One approach involves multiplying the above-average demand within the replenishment time by a "paranoia factor" to incorporate sales peaks and production disruptions. A minimum paranoia factor of 1.5 (50% above the average) is recommended when there are no sequence-dependent setups, prioritizing rapid workflow. In cases of high demand fluctuations and frequent flow disruptions, a factor of 2 should be used.

Another approach examines the maximum sales over the past 6 to 12 months within a defined reliable replenishment time. The reliable replenishment time ensures that items can be safely obtained within that timeframe when they are urgently needed.

It's important to note that the determination of the target level based on the criteria serves as an initial inventory setting. Future adjustments to the target levels, whether increasing or decreasing them, are made using a specialized algorithm that monitors the actual behavior of the finished-goods stock.

II) Buffer Management in MTA

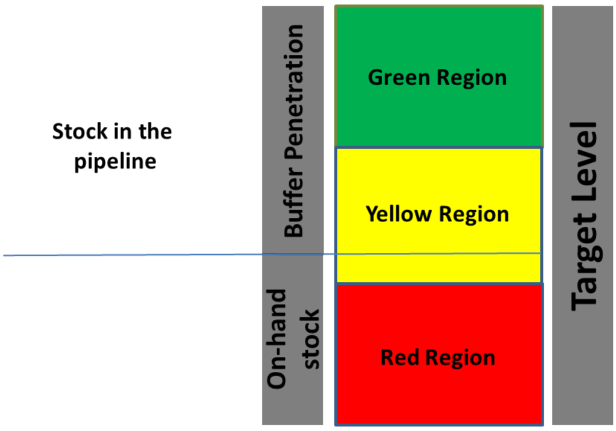
Once the target level is operational, replenishment orders are initiated based on the previous consumption. The MTA replenishment policy is like the base-stock policy, a continuous-review inventory system (KEGENBEKOV; JACKSON, 2021). The particularity of base-stock policies is that a replenishment order to restore the base-stock level S is made whenever the inventory position is below S, as described by Kouki et al. (2020). Therefore, the reorder point is S-1, and the policy is commonly referred to as a (S -1, S) policy in inventory literature. The base-stock concentrate just on inventory replenishment, but the MTA goes ahead and determines the priorities on the shop floor using buffer status. It is the percentage of the penetration into the stock buffer – stock withdrawn from the buffer –, calculated as shown equation 1:

$$BS = \frac{(TL - FGI - WIP)}{TL} \qquad (1)$$

where BS is the buffer status, TL is the Target Level, FGI is the finished goods inventory (on-hand stock) and WIP is the Work-In-Process (stock in the pipeline plus the production orders waiting to release). The production orders do not have a due date as in MRP.

Schragenheim, Dettmer and Patterson (2010) define the state of the stock buffer when containing two-thirds or more of the target level as green. In other words, one-third or less of the buffer is not in the FGI, but somewhere on the way. In a similar fashion, when the FGI contains between one-third and two-thirds of the target level, as shown in Figure 2.3, they call that state yellow. When the on-hand stock, the inventory at the finished-goods warehouse, is less than one-third, meaning more than two-thirds are not at the warehouse, then the state is red.

Figure 2.3 - Illustrate of the stock buffer



Source: Proposed by the author

At any given point in time, the stock buffer is divided into the part that exists as finished goods on-hand and available for immediate sale (FGI), and the stock that complements the previous part to the full target level (WIP). Assuming we keep the target level intact, then the latter part is in the form of all the product components required for the finished goods to be equal to the target level. The part of the buffer that is not in the finished goods is called “penetration into the buffer” because that stock has not yet completed manufacturing and therefore is not currently available for immediate shipment.

To calculate the priority of an order on the floor, we should use the equation 2 (SCHRAGENHEIM, 2010):

$$BS\ of\ a\ PO = \frac{(TL - DWIP - FGI)}{TL} \tag{2}$$

where $BS\ of\ a\ PO$ is the Buffer Status of a PO, TL is the Target Level, DWIP is the work-in-process concerning open PO located downstream from the PO for which want to know the BS, and FGI is the finished goods inventory. The priority of the orders depends on what lies downstream of the production order. The BS of a PO is the real indication of how urgent it is.

In Table 2.1, an example illustrates a target level inventory buffer for a product called P1. The scenario involves a production order of 200 units for P1 somewhere on the shop floor, with 100 units currently in the finished-goods stock. The desired target level, representing optimal availability, is set at 500 units. It is crucial to have the entire target level within the production system, whether in finished goods or at various stages of completion on the shop floor. Presently, only 20 percent of the target level resides in the finished-goods inventory, indicating an urgent need to replenish it. Notably, the size of Order 1 (200 units) does not determine the urgency. Instead, urgency is determined by the amount in the finished stock downstream from Order 1. To indicate order priority, a color code is used: green, yellow, or red. Order 1 is considered urgent and falls under the red buffer status. Order 2 is upstream from Order 1 and has 300 units (60 percent of the target inventory) downstream from it, resulting in a yellow buffer status. Order 3, which is for an additional 100 units, has 80 percent downstream from it, representing 80 percent of the target, and is classified under the green buffer status.

Table 2.1 - Availability targets and priority status of orders for a buffer target of five hundred

Inventory and Production Orders	**Quantity**	**Percentage of target in front of order (downstream)**	**Buffer status (priority)**
Finished goods	100		
Order 1	200	20	Red
Order 2	100	60	Yellow
Order 3	100	80	Green
Target level	500		

Source: Proposed by the author

III) Generating production orders and the state of capacity

The order generation process, triggered by the reduction of stock buffers due to demand, can lead to the overloading of production resources within the system. This becomes more pronounced when setups are required, as it accelerates the consumption of resource capacity. However, a consequence of longer replenishment times is the increased likelihood of more products being classified as "red" in terms of their order status. When the number of red orders exceeds 20 percent, the effectiveness of the entire priority maintenance scheme diminishes, resulting in a significant number of shortages (SCHRAGENHEIM, 2010).

In the context of manufacturing, various challenges such as seasonality and dependent setups exist. To ensure availability, stricter control over production resources is necessary. This additional load control must account for both system instability and sudden shifts in priorities (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). As a result, the manufacturing throughput accounting (MTA) approach requires a certain level of protective capacity. The concept of "protective capacity" refers to the threshold where the lack of immediate available capacity starts to cause tangible damage. The loss of protective capacity can occur due to excessive total demand or an excessive number of setups. Two approaches can be employed to address this issue (SCHRAGENHEIM; DETTMER; PATTERSON, 2009):

1. Setting a minimum production batch: The minimum batch size is separate from the target level and is added on top of it. When the combined inventory in the pipeline and on-hand falls below the target level, a production order is generated. However, the size of this order is set to be at least equal to the minimum batch quantity. It is possible that the total inventory exceeds the target level, but it should still be less than the target level plus the minimum batch.

2. Managing the capacity of the capacity-constrained resource (CCR): New production orders are released only when it is reasonable to expect that the CCR will be able to work on them soon. This responsibility falls under the purview of "planned load."

These elements contribute to the effective implementation of DBR and help maintain a balance between production and capacity utilization. The drum-buffer-rope (DBR) approach involves the management of planned load, which includes regular planned load (RPL) and full planned load (FPL). RPL is the total load on the capacity-constrained resource (CCR) from all released production orders that have not yet been processed by the CCR. FPL includes all required replenishments, even those that have not been released (SCHRAGENHEIM, 2010).

To maintain control over the release of new orders, the release is limited based on the RPL. Orders are released until the regular planned load approaches a predetermined limit. Each released order updates the regular planned load, and once it exceeds the limit, the release of orders is halted. After reducing the load, the remaining orders in the queue can be released (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). For example, if the replenishment time is five days with 16 hours of CCR time per day (80 hours in total), a natural limit for the regular planned load would be 80 percent of the replenishment time, which is 64 hours. This limit ensures that the replenishment time is maintained, considering only 80 percent of the available time (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). This approach considers the time required for operations downstream of the CCR, as those typically take less time compared to reaching the CCR and being processed. By following this release procedure, excessive work-in-process (WIP) on the shop floor can be avoided. Suppose the planned load reaches 50 hours on a given day, while the limit is 64 hours. In this scenario, up to 14 hours of work can be released. However, there is a need to release 19.3 hours, slightly exceeding the limit. In such cases, the orders are released based on priority, starting from the highest priority (e.g., P10, P3, P1, P7, P5, P8). The total time for these orders equals 13.2 hours. The decision to release the next order, such as P9, which would surpass the 14-hour limit, should be made by the person responsible for production, considering their judgment (SCHRAGENHEIM, 2010). It is important to note that orders P6, P2, and P4 would have to wait for at least one additional day in this scenario.

Table 2.2 - List of the orders in the queue awaiting release

Product	**Quantity to replenish**	**Target level**	**Priority (%)**	**Total time on CCR (hours)**
P1	13	120	10.83	1.5
P2	3	95	3.16	0.8
P3	120	1000	12	2
P4	45	3000	1.5	0.8
P5	24	400	6	3.2
P6	114	3500	3.26	2.5
P7	100	1000	10	1.5
P8	100	2000	5	2
P9	33	750	4.4	2
P10	50	400	12.5	3
			Total:	19.3 hours

Source: Proposed by the author

When implementing the drum-buffer-rope (DBR) approach, it is crucial to consider the minimum batch size as well as the load on the capacity-constrained resource (CCR) (SCHRAGENHEIM, 2010). The determination of priority considers the quantity needed to replenish to the target level and the batch size. For instance, if the target level is 100 units and there are 49 units on-hand and 50 units in the pipeline (across multiple unfinished production orders), only one unit is required to replenish to the target level. However, if the minimum batch size is 25 units, the priority for releasing the next replenishment order is calculated as 1 × 100/100 = 1%. Meanwhile, the time required on the CCR must consider the processing of a batch of 25 units, potentially affecting the release of other orders scheduled for the same day, as the CCR's capacity is planned based on the load of 25 units (SCHRAGENHEIM, 2010). Considering both the minimum batch size and the load on the CCR ensures effective planning and sequencing of orders in the DBR system, considering quantity requirements and resource capacity constraints.

IV)Monitoring the Target Level Size – Dynamic Buffer Management

The planning stage of target levels in the drum-buffer-rope (DBR) approach requires feedback to ensure adequacy and accommodate changes in demand or supply (SCHRAGENHEIM, 2010). The behavior of the on-hand stock provides valuable insights into the appropriateness of specific target levels. To recommend adjustments to target levels based on finished-goods stock behavior patterns, Dynamic Buffer Management (DBM) algorithms are employed (SCHRAGENHEIM, 2010).

Maintaining excessively high inventory levels can lead to unnecessary replenishments and capacity implications during peak times (SCHRAGENHEIM, 2010). Buffer targets consistently in the green zone for extended periods indicate an excessive buffer size. Schragenheim, Dettmer, and Patterson (2010) suggest reducing the target level when an item spends a continuous period in the green region, known as the "green check period." The default duration for the green check period is twice the replenishment time.

Conversely, a high frequency of red points in the buffer signal that the target level is too low (SCHRAGENHEIM, 2010). Spending significant time near the top of the red zone may indicate that the buffer is insufficient to prevent shortages. An increase in the buffer is recommended when there is a penetration into the red zone, with the depth of the penetration considered as a relevant signal. If the cumulative depth of penetrations within the replenishment time exceeds the size of the red level, increasing the buffer is advised.

When the target level is increased, the item enters the red zone, and a new replenishment production order is released. It takes time for the new buffer size to stabilize, so decisions to further increase the buffer should be postponed until the impact of the previous increase is observed. A cooling period, equivalent to one replenishment time, is recommended before re-evaluating penetrations into the red zone.

Schragenheim (2010) recommends increasing buffers by 20 percent and decreasing them by 15 percent. Manufacturing environments with relatively stable demand allow for smaller buffer adjustments to align with trends. It is crucial to analyze both demand and flow in the production shop floor, identifying critical changes in behavior and potential material shortages, to make informed decisions regarding finished-goods stock buffers.

While DBM assesses the combination of demand and supply, Schragenheim (2010) emphasizes the importance of conducting focused analyses of demand and flow within the production shop floor. These analyses quickly identify critical changes and potential material shortages, guiding decisions in manufacturing environments. However, it is worth noting that these analyses are currently not integrated into the known Theory of Constraints (TOC) solution for manufacturing to availability (MTA) (SCHRAGENHEIM, 2010).

2.1.2 Overcoming complications in using MTA

There are limitations to Synchronous Drum-Buffer-Rope (S-DBR) in handling highly complex production processes (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). It is essential to investigate various complicated factors to assess their potential to hinder a successful implementation of S-DBR. These are some complicating factors:

1. Multiple simultaneous Critical Chain Resources (CCRs).
2. "Wandering" bottlenecks.
3. Sequence-dependent setups.

I) Multiple Capacity-Constrained Resources (CCRs)

When multiple work centers with comparable capacities act as the "weakest links" on the shop floor, it presents a situation where one capacity-constrained resource (CCR) feeds another (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). In such cases, when options for overtime, extra shifts, or outsourcing are limited, Schragenheim, Dettmer, and Patterson suggest implementing control measures. This involves having adequate protective capacity and establishing a single focal point for planning and control, which is the CCR (SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

II) Wandering Bottlenecks

Bottlenecks can arise from changes in the product mix, as different product families may dominate at different times (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). When a particular product family becomes dominant, a specific machine or work center becomes the clear weakest link. However, if the product mix shifts and another product family starts heavily utilizing a different work center or machine, that work center or machine may become the new constraint.

During phases when bottlenecks act as interactive constraints, the overall ability to control reliability deteriorates (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). The capacity-constrained resource (CCR) alternates between the two, creating an interactive state without providing early warning. In such cases, it is crucial to carefully monitor one CCR to prevent overloading the other.

III) Dependent Setups in MTA

The Simplified Drum–Buffer–Rope (S-DBR) approach operates on the assumption that the sequence of work on the capacity-constrained resource (CCR) does not significantly impact its capacity (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). If this assumption holds, scheduling based on market demand priorities and monitoring the load on the CCR using planned load is sufficient to ensure high production floor reliability.

In the presence of sequence-dependent setups, the setup time at a resource depends on the previous setup nature (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). In such cases, production planning aims for a "preferred sequence" that minimizes the total number of setups. However, following the preferred sequence becomes challenging when it consumes excessive capacity and prevents order expediting, which could pose issues for the S-DBR methodology (SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

Deviation from the preferred sequence results in significant wasted capacity, potentially turning the constrained resource into a bottleneck. Moreover, maintaining the preferred sequence may lead to replenishment time equaling the entire cycle time, compromising availability. Changing the preferred sequence to expedite an urgent order is only possible in extreme cases.

Managing sequence-dependent setups requires higher target levels to account for the need to respond quickly in a system that can be slow at times. This encourages the creation of stock to ensure prompt availability when needed (SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

2.2 Distribution-to-availability

The Theory of Constraints (TOC) provides guidelines for distribution management. It emphasizes reporting buffer status daily and ensuring frequent transportation (COX; SCHLEIER, 2010). The solution comprises some basic rules.

Firstly, aggregating stock at higher levels, such as manufacturer and central warehouses, helps centralize control and streamline operations. Determining appropriate buffer sizes for each location in the supply chain requires careful analysis of demand patterns, supply availability, and replenishment lead time. Increasing replenishment frequency helps avoid stockouts and maintain optimal inventory levels, ensuring timely product availability. Buffers play a crucial role in managing inventory flow, absorbing fluctuations in demand and supply. Dynamic Buffer Management (DBM) optimizes buffer sizes based on demand and lead time variability. Setting replenishment priorities based on urgency and criticality aligns with strategic objectives and customer demands, improving stock management and customer satisfaction.

Implementing these strategies, along with tools like DBM, enables businesses to gain better control over their supply chains, reduce costs, enhance customer service levels, and achieve operational success. The integration of the Simplified Drum–Buffer–Rope (S-DBR) production operation with the distribution end of the supply chain ensures products reach customers efficiently (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). However, managing distribution becomes complex due to the large number of decisions involved in determining stock levels for each item and location. Uncertainty in demand, production, and replenishment time further complicates the decision-making process (SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

The TOC approach addresses these challenges by considering the impact of supply and demand to determine appropriate stock levels throughout the supply chain, while being mindful of cash and space limitations (SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

The objective is to ensure high availability of items at all consumption points while constantly renewing consumed stocks from strategically placed stock buffers (COX; SCHLEIER, 2010). This approach optimizes distribution and replenishment, considering the dynamic nature of market demand (SULLIVAN; REID; CARTIER, 2007).

> *"(a) pull distribution method that involves setting stock buffer sizes and then monitoring and replenishing inventory within a supply chain based on the actual consumption of the end user, rather than a forecast. Each link in the supply chain holds the maximum expected demand within the average replenishment time, factored by the level of unreliability in replenishment time. Each link generally receives what was shipped or sold, though this amount is adjusted up or down when buffer management detects changes in the demand pattern."*

The solution is comprised of six steps (COX; SCHLEIER, 2010):

1) Aggregate stock at the highest level in the supply chain: the Plant Warehouse/Central Warehouse (PWH/CWH).
2) Determine stock buffer sizes for all chain locations based on demand, supply, and replenishment lead time.
3) Increase the frequency of replenishment.
4) Manage the flow of inventories using buffers and buffer penetration.
5) Use Dynamic Buffer Management (DBM).
6) Set manufacturing priorities according to urgency in the PWH stock buffers.

2.2.1 Aggregate Stock at the Highest Level in the Supply Chain: The Plant/Central Warehouse (PWH/CWH)

The TOC solution suggests aggregating inventory at the supplying source, where stocks can be utilized to serve multiple destinations, and implementing a pull replenishment mechanism triggered by sales at the consumption point (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). This approach, aligned with statistical principles, ensures a more stable and responsive system compared to maintaining large inventories at individual consumption points or shops (SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

2.2.2 Determine Stock Buffer Sizes for All Chain Locations Based on Demand, Supply, and Replenishment Lead Time

The stock buffer size represents the maximum quantity of inventory held at a specific location within the supply chain to safeguard against demand fluctuations. Determining the appropriate stock buffer size is influenced by two distinct factors (SCHRAGENHEIM; DETTMER; PATTERSON, 2009):

7) Demand rate—*demand* is the need for an item while the *demand rate* represents the amount demanded per period (day, week, month, etc.).
8) Supply responsiveness—how quickly the consumed units can be replenished. The main factor here is the *TOC replenishment (lead) time (RLT),* which is defined in the *TOCICO Dictionary* (SULLIVAN; REID; CARTIER, 2007) as *"the time it takes from when a product is sold until a replacement is available at the point of sale/use."*

2.2.3 Increase the frequency of replenishment

The TOC distribution/replenishment solution emphasizes the importance of frequent replenishment to maintain low stock levels. According to Schragenheim, Dettmer, and Patterson (2009), the potential increase in shipping costs from more frequent shipments is outweighed by the boost in sales. They argue that there exists a tradeoff between investing in higher shipment frequency and the cost of lower availability. Increasing the frequency of deliveries improves availability while raising shipping costs. On the other hand, reducing the frequency leads to lower availability or necessitates higher inventory levels to accommodate demand variations. The authors assert that although transportation costs may rise, the reduction in inventory investment frees up cash that can be utilized to diversify product offerings from the same supplier.

To achieve the desired stock levels, a purchase order must be generated for each replenishment, ensuring that the available stock reaches the designated target level. The purchase quantity is determined based on the available inventory and issued purchase orders that are yet to be received. The combined total of available stock and pending purchase orders must align with the target level.

2.2.4 Manage the Flow of Inventories Using Buffers and Buffer Penetration

The Theory of Constraints (TOC) applies the concepts of buffer and buffer penetration to effectively manage inventories in the distribution process. The stock buffer size refers to the desired quantity of a specific SKU maintained at each stock location to safeguard against demand fluctuations. Schragenheim, Dettmer, and Patterson (2009) provide an example where a stock buffer size of 100 units is established for an SKU, and currently, 40 units are available on hand. This indicates that 60 units either need to be ordered or are in the process of being ordered from the supplying location. If the 60 units are not already on order or en-route, a replenishment order of 60 units should be immediately issued.

Buffer penetration is calculated by dividing the number of missing units from the buffer by the stock buffer size and expressing it as a percentage. The quantity of units missing from the buffer is determined by subtracting the available stock and the units already ordered from the stock buffer size. In the previous example, the buffer penetration for the stock at that site is 60 percent (100 - 40)/100). Like Material Time Analysis (MTA), the buffer size is divided into three equal regions, and the color of the buffer is assigned based on the buffer penetration:

- Green: Buffer penetration less than 33 percent
- Yellow: Buffer penetration between 33 and 67 percent
- Red: Buffer penetration between 67 and 100 percent
- Black: 100 percent buffer penetration (stockout situation)

The color of the buffer penetration indicates the urgency of replenishing the stock. Each color zone is associated with specific actions:

- Green: Inventory at the consumption point is sufficient for the time being. Action required: Order a replenishment amount (prioritize based on production capacity if replenishing from a plant).
- Yellow: Inventory at the consumption point is adequate. Additional units need to be ordered from the upstream supply chain. Action required: Order the replenishment amount (order even if there is a capacity shortage, addressing the capacity issue on the production floor if necessary).
- Red: Inventory at the consumption point is at risk of stockout. Units in transit or in manufacturing should be expedited, and an urgent replenishment order must be placed with the supplying source if no units are currently en route. Action required: Investigate, order, and possibly expedite.
- Black: The stock has completely depleted at the consumption point, resulting in potential lost sales opportunities. Swift resolution is crucial, particularly for downstream links in the supply chain where the ability to respond to replenishment and buffer changes diminishes. Action required: Expedite and order immediately.

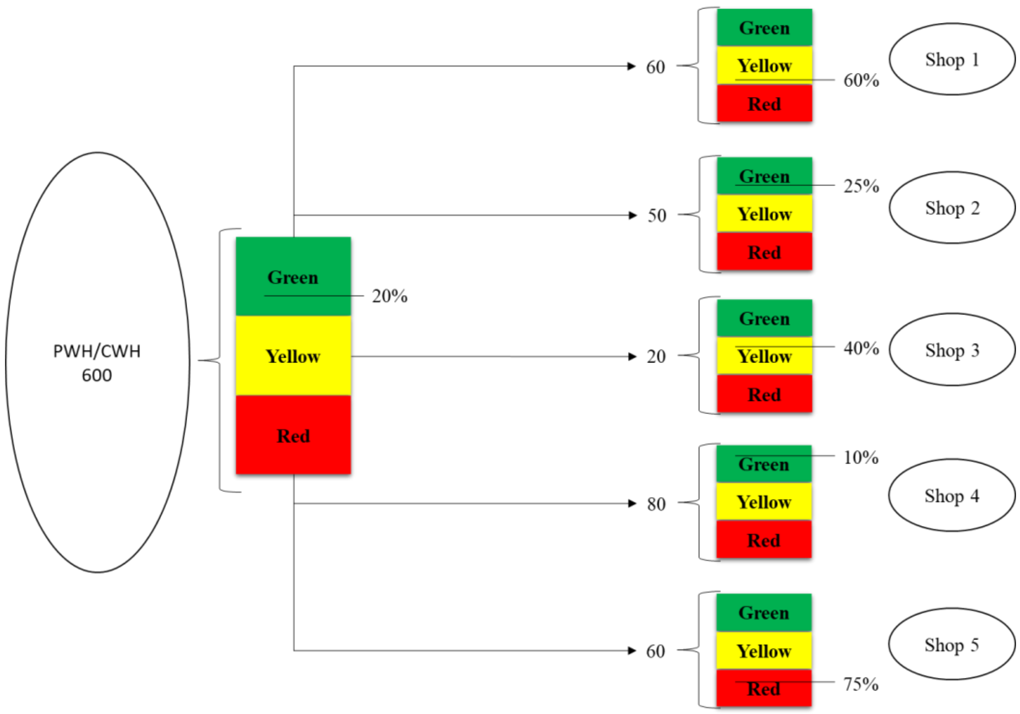
Schragenheim, Dettmer, and Patterson (2009) illustrate the placement of buffers and the utilization of color-coded regions for prioritization in Figure 2.4. The figure demonstrates that each location has distinct buffers for the same item, which are managed separately. For instance, the buffer at the Plant Warehouse/Central Warehouse (PWH/CWH) has a size of 600 units, and currently, it has a buffer penetration of 20 percent (480 units out of 600). Therefore, the buffer is assigned a green priority color. Similarly, at Shop 1, the buffer size for the item is 60 units, with only 24 units currently available, resulting in a buffer penetration of 60 percent and a yellow priority color. This arrangement determines the placement of buffers and their prioritized replenishment at upstream links. However, this prioritization alone may not suffice as stock can be present at the location and in transit simultaneously.

Multiple perspectives of the same buffer are valuable and can be obtained. The Theory of Constraints (TOC) has introduced the concept of Virtual Buffer Penetration (VBP) to determine the priority of stock at any given point in the supply chain based on the status of downstream links (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). However, this priority is only relevant until the next stocking point. For instance, the VBP for an SKU at the PWH/CWH considers only the physical stock available there, while the VBP for a shipment considers stock from previous shipments and the target location.

In Figure 2.4, the retailer's stock buffer size for a specific SKU is set at 100 units, with 25 units currently available and a shipment of 25 units en-route from the PWH/CWH to the shop. The Virtual Buffer figures are displayed above each stock along the way to the retailer. The VBP considers the combined stock from in-transit and downstream stocking points. The SKU's priority is determined by the Virtual Buffer Penetration of the next downstream stock location (as shown in Figure 2.5). This concept of VBP provides a highly effective tool, enabling complete visibility across the supply chain and offering a clear and straightforward priority mechanism for decision-makers at various stock points within the supply chain. Figure 2.4 serves as a demonstration of this supply chain management concept (SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

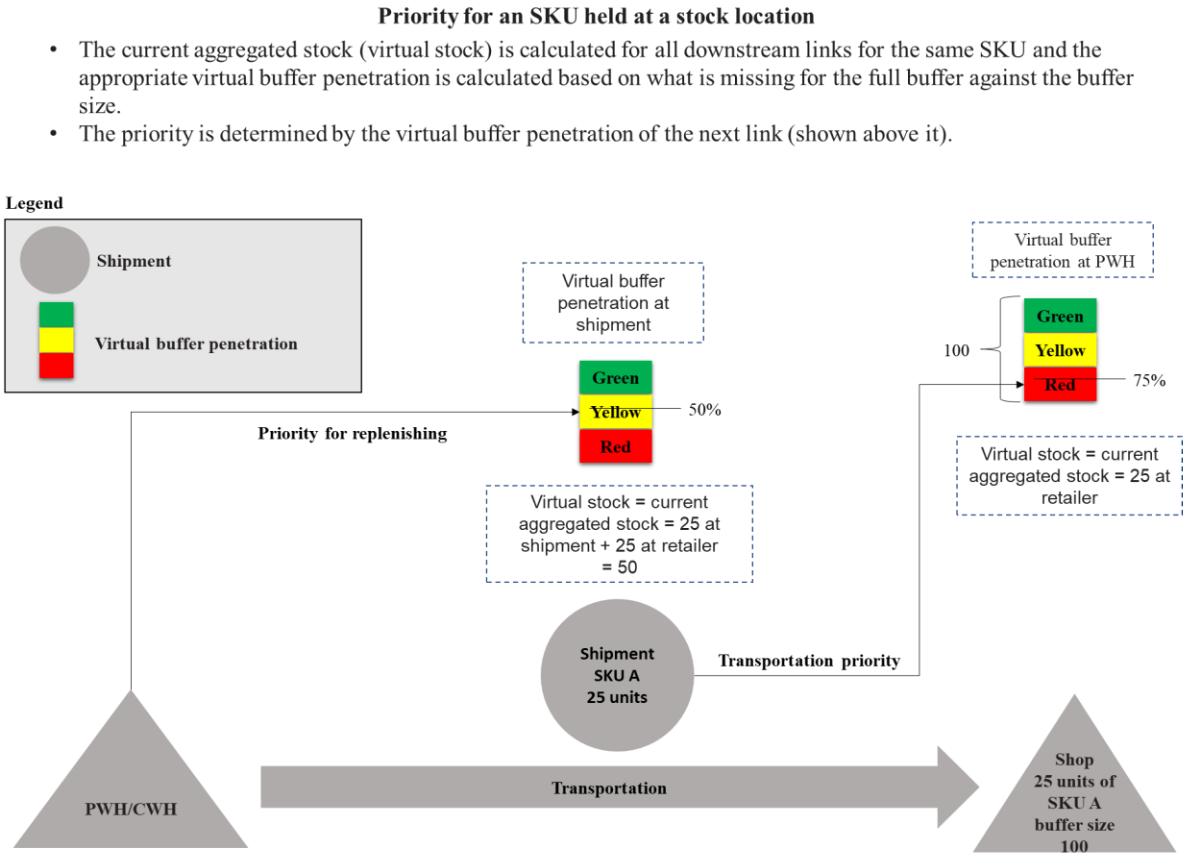
In Figure 2.5, the warehouse manager (or shop manager) can easily discern that the priority of this SKU is marked as red with a 75 percent buffer penetration. With a buffer size of 100 units and 25 units already at the shop, there is a shortage of 75 units. Therefore, the shop must promptly determine how to acquire more stock for this SKU.

Figure 2.4 - Item stock buffer sizes and buffer penetrations across the pull supply chain



Source: Adapted from Schragenheim, Dettmer and Patterson (2009)

Figure 2.5 - Virtual buffer concept applied to a shop item and in-transit shipments to this shop



Source: Adapted from Schragenheim, Dettmer and Patterson (2009)

The translation of the current information for various supply chain links for this example is (SCHRAGENHEIM; DETTMER; PATTERSON, 2009):

- The transportation manager is responsible for determining the priority of shipments, particularly those that require expedited handling. In this specific scenario, there is a need to expedite the shipment of 25 units for this SKU. The decision is based on a buffer penetration of 75 percent, which aligns with the Virtual Buffer Profile (VBP) observed by the plant warehouse manager. The virtual buffer for the PWH/CWH manager is calculated by combining the shop buffer with the transportation shipments. If the virtual buffer status indicates a red alert, the transportation manager must investigate the expected arrival time of the order at the shop. If there are any delays, immediate action should be taken to expedite the shipment.
- The warehouse manager is responsible for establishing the replenishment priority for this SKU. The virtual buffer considers the current stock levels both in transit and at the shop for this SKU. In this case, there is a need to replenish 50 percent of the buffer size for this item in the PWH/CWH, which amounts to 50 units. The replenishment shipment is assigned a priority status of yellow based on a buffer penetration of 50 percent. The buffer size for this SKU is set at 100 units, with 25 units currently available on-site and an additional 25 units en-route, resulting in a shortage of 50 units.

2.2.5 Use Dynamic Buffer Management

Dynamic Buffer Management (DBM) is a proactive approach that adjusts stock buffer sizes based on real-time usage data (SCHRAGENHEIM, 2010). By monitoring the buffer penetration of each SKU, DBM assesses whether the designated buffer size is appropriate. The core idea is to evaluate the combined impact of incoming supply and outgoing demand at the stocking point. Through this monitoring and adjustment process, DBM enables managers to determine the optimal stock buffer level necessary to meet demand while considering delivery capabilities.

The DBM mechanism provides two distinct warnings to managers: one for an excessively large buffer size and the other for a buffer size that is too small (SCHRAGENHEIM, 2010). To identify an oversized buffer, the actual stock of a specific SKU is compared to the target over an extended period. In the Theory of Constraints (TOC), a buffer is deemed too large if the buffer penetration consistently remains in the green zone for three consecutive replenishment periods. However, relying solely on replenishment time for buffer size analysis can be risky, especially when it exhibits significant variability. Factors such as multiple vendors with varying replenishment times for an SKU or different replenishment times for each SKU can contribute to this variability.

In cases where a SKU's buffer size has remained in the green zone for an extended period, the default recommendation is to decrease the buffer size. The general guideline suggests reducing the buffer size by 33 percent, although this recommendation is subject to several factors (SCHRAGENHEIM, 2010):

- The speed desired to lower inventories.
- The risk/importance placed on this SKU.
- The risk/importance of this stock location.

A similar mechanism is employed to determine whether the buffer size is set too low (SCHRAGENHEIM, 2010). This involves assessing whether the inventory for a particular SKU remains in the red zone after replenishment. In other words, it examines whether, based on the stock buffer size, the actual stock level remains in the red zone following sequential replenishments. The algorithm determines whether an SKU remains in the red zone for multiple days, typically using the replenishment time as a parameter. To address this situation, the guideline suggests increasing the buffer level by 33 percent (SCHRAGENHEIM, 2010). The definition of "too long" in a zone and the specific adjustments for increasing or decreasing the stock buffer level may vary among SKUs. These parameters serve as useful rules of thumb for establishing the system.

Following the buffer adjustment, the SKU undergoes a "cooling period" during which no further buffer changes are recommended to allow the system to adapt to the revised buffer size. The cooling period should be long enough for the adjustment to take effect (ensuring that the newly ordered quantities arrive at the stock location), but short enough to prevent any sudden market demand changes from going unnoticed (SCHRAGENHEIM, 2010). Typically, for an increased buffer, the cooling period is set to the full replenishment time, while for a decreased buffer, the cooling period involves waiting for the inventory at the location to transition from above the buffer size level to the green zone (as reducing the buffer size likely resulted in the current inventory exceeding the buffer size level) (SCHRAGENHEIM, 2010).

3 A SYSTEMATIC LITERATURE REVIEW ON THE EVOLUTION OF DRUM-BUFFER-ROPE AND SIMPLIFIED DRUM-BUFFER-ROPE SYSTEMS

3.1 Introduction

The Drum-Buffer-Rope (DBR) system, which originated from the Theory of Constraints (TOC), was developed to control production flow based on TOC principles. The original version of this system was introduced in the books "The Goal" (GOLDRATT; COX, 1984) and "The Race" (GOLDRATT, 1986). A simplified version called Simplified Drum-Buffer-Rope (S-DBR) emerged in 2000, as outlined in the book "Manufacturing at Warp Speed: Optimizing Supply Chain Financial Performance" authored by Schragenheim and Dettmer (2000).

Both DBR and S-DBR methods require the release of raw materials to the shop floor following resource constraints. The drum mechanism sets the production pace, the buffer safeguards the constraints against variability, and the rope regulates the release of work orders to production based on the drum's guidance. Buffer management determines the prioritization of production orders.

In the case of S-DBR, the primary constraint is market demand. The term "simplified" refers to the presence of only one buffer, which is the shipping buffer for make-to-order (MTO) environments (SCHRAGENHEIM; DETTMER, 2000), or the stock buffer for make-to-availability (MTA) environments (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). MTA is a type of make-to-stock environment that aims to ensure product availability without relying on demand forecasts. Only in S-DBR is there a mechanism called Planned Load (PL), which controls the load on the capacity-constrained resource (CCR) and supports the rope's functioning. PL identifies situations where new production orders should not be released to the factory floor due to CCR overload.

Numerous aspects have been studied in the context of DBR/S-DBR, including their applications (DARLINGTON et al., 2015; UMBLE; UMBLE; MURAKAMI, 2006;

WALKER, 2002), performance comparisons with other systems (THURER et al., 2017; GILLAND, 2010; WATSON; PATTI, 2008), and attempts to understand and explain their mechanisms (BLACKSTONE JR; COX III, 2002; GUPTA; KO, MIN, 2002; SCHRAGENHEIM; COX; RONEN, 1994).

Several studies have proposed modifications to the original DBR and S-DBR systems to adapt them to various organizational environments. These studies are valuable as they provide advancements, new application possibilities, and practical contributions. However, they are still limited in scope. To the best of our knowledge, no comprehensive comparative literature study has been conducted. Addressing this research gap is the primary objective of the present study. To achieve this, the article conducts a systematic literature review based on Tranfield, Denyer, and Smart's (2003) methodology, focusing on changes made to DBR/S-DBR systems. The main contribution of this study is to present the current state of adaptations to these systems and propose new avenues for research in this field.

The remainder of this article is structured as follows: Section 2 describes the research methodology, Section 3 presents the results, and Section 4 draws conclusions.

3.2 Research method

The systematic literature review (SLR) conducted in this study adhered to the methodology proposed by Tranfield, Denyer, and Smart (2003). It encompassed three distinct stages outlined in Table 3.1, to guarantee a comprehensive and thorough analysis of the existing literature.

Table 3.1 - Systematic literature review stages

I. Planning the review	II. Carrying out the review	III. Reporting and disseminating it
Research questions definition. Protocol review definition.	Identification of studies. Selection of studies by reading full text. Data extraction of studies;	Results and discussions Future research suggestion.

Source: Proposed by the author

I) Planning the review

The SLR was driven by three research questions:

RQ1: How DBR and S-DBR are being changed to better deal with practical requirements?
RQ2: Why these changes are being required?
RQ3: What are news research avenues concerning these changes?

The SLR was conducted by the protocol in Table 3.2.

Table 3.2 - Parameters of SLR

Parameters	**Values**
Databases	Web of Science, Compendex and Scopus
Keyword	"Drum-buffer-rope"
Fields	All
Document type	Articles published in journals
Language	English
Inclusion criteria	The study should propose some adaptation in the original DBR or S-DBR systems.
Data extracted	Article title, authors, date of publication date, adaptation proposed, affected DBR mechanism, and proposal evaluation.

Source: Proposed by the author

The SLR was limited to journal articles to ensure the high quality of the research.

II) Carrying out the review

The papers were identified in the databases using string searches based on the parameters defined in Table 3.1, including databases, keywords, document type, and language. Table 3.2 provides an overview of the search strings used. The results of the search, along with the consultation records for each database, are presented in Table 3.3. Initially, 637 articles were retrieved, out of which 175 were duplicates. The remaining 462 papers were carefully analyzed by reading their full texts, resulting in the selection of 32 papers that met the inclusion criteria. The database search was conducted on June 17, 2022.

Table 3.3 - Databases consulting register

Database	**Consult date**	**Search string**	**Number of articles found**
Scopus	17-06-2022	ALL ("drum-buffer-rope") AND (LIMIT-TO (DOCTYPE , "ar ")) AND (LIMIT-TO (LANGUAGE , "English ")) AND (LIMIT-TO (SRCTYPE , "j"))	451
Web of Science	17-06-2022	TOPIC:("drum-buffer-rope") Refined by: DOCUMENT TYPES: (ARTICLE) AND LANGUAGES: (ENGLISH) Timespan: All years. Indexes: SCI-EXPANDED, SSCI, A&HCI, CPCI-S, CPCI-SSH, ESCI.	97
Compendex	17-06-2022	(((("drum-buffer-rope") WN ALL)) AND (({ja} WN DT) AND ({english} WN LA)))	89
		Total	637
		Papers after removing duplicates	462

Source: Proposed by the author

Some topics studied in the excluded articles were DBR applications – e.g., Darlington et al. 2015; Umble, Umble, and Murakami 2006; Walker 2002 –, comparisons of DBR performance with that of other systems – e.g., Thürer et al. 2017; Gilland 2010; Watson and Patti 2008 – or about understanding and explaining DBR mechanisms – e.g., Blackstone Jr and Cox III 2002; Gupta, Ko, and Min 2002; Schragenheim, Cox, and Ronen 1994.

From 32 studies selected, we extracted the title, authors, publication date, adaptation proposed, affected DBR mechanism and proposal evaluation. Section 3 presents the results of the SLR with information about publication period, publication sources as well as a classification scheme for such studies.

III) Reporting and disseminating it

The results, discussions and future researchers are presented in subsequent sections.

3.3 Results analysis

The analysis of the results begins with a quantitative exploration, aiming to provide an overall perspective of the collected data. Out of the 462 articles obtained from the database search, approximately 7% focused on proposing modifications to either DBR or S-DBR. It is important to note that most studies concentrated on a single mechanism within DBR/S-DBR, and thus they were classified accordingly. Table 3.4 presents this classification, with the rows representing the affected mechanisms, and the columns differentiating between DBR and S-DBR studies. The frequency of studies for each mechanism is presented in terms of absolute and relative frequency. The studies that encompass all mechanisms are indicated in the DBR and S-DBR columns.

Table 3.4 - Papers classification

	DBR	S-DBR	Absolute Frequency	Relative Frequency
Buffer	9	2	11	34,38%
Rope	8	0	8	25,00%
Drum	5	0	5	15,63%
DBR	4	0	4	12,50%
S-DBR	0	2	2	6,25%
Drum-Buffer	1	0	1	3,13%
Drum-Rope	1	0	1	3,13%
Buffer-Rope	0	0	0	0,00%
Total	28	4	32	100,00%
% Total	87,50%	12,50%	100,00%	

S-DBR received significantly less attention compared to DBR, with only 12.50% of the studies focusing on it. The data indicates that the Buffer was the most extensively investigated aspect. Furthermore, studies proposing improvements for both DBR and S-DBR accounted for only 18.75% of the total. A substantial number of investigations concentrated solely on specific components of the system. In terms of application domains, manufacturing environments garnered the attention of 93.7% of the studies, while one study was dedicated to business processes and another focused-on container loading. The subsequent section presents an overview of the modifications proposed by each study.

3.3.1 Changes on DBR and S-DBR

The papers are divided into three parts. Part I detailed the papers that modified one of the follow mechanisms: drum, buffer, or rope. Part II detailed the papers that modified two of these three mechanisms, and Part III is dedicated to papers that modified the whole system.

I) Only one mechanism change

a) Drum

This section presents the 5 (15,63%) studies that propose changes to the drum mechanism and its operations – identification, scheduling, constraint exploitation, and work orders sequencing.

Table 3.5 presents the studies grouped by operation and exposes the proposition of each one, indicating the testing procedure used. No studies relating to S-DBR focus on changes to the drum.

Urban (2019) contributes to production management theory by proposing a new method for TOC implementation and a method for constraint identification that considers the irregularities in processes' capacity utilization. Khalil, Stockton, and Fresco (2008) also proposed a constraint identification method. According to the authors, the effectiveness of DBR in systems with high throughput levels is threatened by high levels of variability, which can cause constrained resources to shift to different stages of the manufacturing processes. The authors approach this issue by developing a methodology for measuring the variability of manufacturing processes by combining the variability originating from various sources.

Concerning constraint exploitation, Wu and Liu (2008) created a solution that can allow marketing personnel to establish realistic order promise dates by observing the system capacity over a planning horizon. The authors proposed a capacity available-to-promise

(CATP) model for DBR systems. The model was developed based on the drum to help the DBR users improve the due date promise and to exploit the constraint. By using this model, the CATP model provides a detailed and time-phased diagram of unused production capacity and allows marketing personnel to establish realistic order promise dates and concentrate on selling idle capacity in the future.

Table 3.5 - Studies that affected the drum

DRUM operation affected	Reference	Changes	Evaluation method
Constraint Identification	Urban (2019)	A new method of bottleneck identification is elaborated in this study.	Not tested
	Khalil, Stockton, and Fresco (2008)	Method to identify the constraint by measuring variability levels of each manufacturing process.	Simulation
Constraint Exploitation	Wu and Liu (2008)	A capacity available-to-promise model for constraint exploitation.	Numerical example
Constraint Identification and Exploitation	Lizarralde-Aiastui, De Eulate e Media Villa-Guisasola (2020)	An adaptation of the DBR is suggested for bottleneck selection and exploitation by a systematic decision-making process based on resource-based view (RBV) and practice-based view (PBV).	Real application
Work order Sequence	Gonzalez-R, Framinan, and Ruiz-Usano (2010)	A method to obtain a robust dispatching rule.	Simulation

Source: Proposed by the author

Lizarralde-Aiastui, De Eulate e Media Villa-Guisasola (2020) proposed a systematic process for deploying the selection and exploitation of the production system constraint by integrating a strategic perspective. The systemic process was based on the seminal work from Goldratt’s TOC (GOLDRATT; COX, 2003) and included critical aspects from theories on strategy, such as the resource-based view (RBV) and practice-based view (PBV). The RBV and PBV argue the importance of resources and practices in achieving competitive advantage and/or improved firm performance. A case study was used to present this adaptation, in which the researchers actively participated. The results show a reduced lead time of 10%, reduced WIP by 20%, reduced the volume of semi-finished material in progress by 40%, reduced quality-related incidents by 20%, and increased service levels from 50% to 70%.

Gonzalez-R, Framinan, and Ruiz-Usano (2010) highlighted the impact of dispatching rules on the performance of DBR in multiproduct manufacturing environments where order sequencing is constrained. According to the authors, no single dispatching rule universally outperforms others in all environments. Consequently, they proposed a methodology to derive a robust dispatching rule from a preselected set of rules based on specific performance measures sought by managers. The chosen dispatching rules included Shortest Process Time (SPT), Shortest Remaining Processing Time (SRPT), Shortest Imminent Operation (SI), Longest Processing Time (LPT), Earliest Due Date (EDD), Least Slack (LS), Critical Ratio (CR), First-Come-First-Served (FCFS), and Slack per Remaining Operation (SRO). To assess the robustness of the dispatching rules, an initial experimental design was conducted, incorporating various sources of variability observed in real manufacturing environments, such as stochastic processing times, unbalanced production lines, machine breakdowns, and setup times. Simulation techniques were employed to calculate performance measures, including average tardiness, maximum tardiness, and average work in process. The global performance of each dispatching rule was then evaluated across different scenarios using Taguchi's robustness concepts. The findings revealed that, for environments without setups, the most robust dispatching rules for constrained systems were EDD, SI, and FCFS. In scenarios involving setups, FCFS and SI emerged as the most robust rules.

b) Buffer

This section presents 11 studies (34.38% of the total) that proposed changes related to the buffer mechanism, specifically focusing on buffer sizing and buffer management. Table 3.6 provides an overview of these studies, categorizing them by operation, presenting their propositions, and indicating the testing procedures employed. Among these eleven studies, two specifically address S-DBR.

Wang et al. (2015) introduced a buffer sizing method for hospital inventory management, implementing DBR. The proposed approach, known as "dynamic drum-buffer-rope" (DDBR), adopts a demand-pull replenishment strategy to overcome the limitations of existing re-order point approaches commonly used in this domain. The DDBR approach determines the most appropriate buffer size and replenishment amount.

Radovilsky and Frankel (2013) aimed to identify the optimal buffer size using a model based on a finite multi-server queue. Their novel modeling approach calculates the ideal number of units waiting in line (i.e., the optimal buffer size) in front of a multi-server CCR (Constrained Capacity Resource). The goal is to maximize the net profit derived from the CCR's throughput while preventing the constrained resource from becoming idle.

Furthermore, Fallah et al. (2011) developed a buffer sizing method based on the principles of factory physics. Their proposed method enhances overall protection by determining buffer sizes based on the variability of constraint predecessor stations.

Table 3.6 - Studies that affected the buffer

BUFFER operation modified	**References**	**Adaptations**	**Evaluation method**
Buffer sizing	Wang, Cheng, Tsen and Liu . (2015)	An approach to determine the most appropriate buffer size and replenishment amount.	Real application
	Radovilsky and Frankel (2013)	A modeling approach of identifying the optimal buffer size in front of a CCR with parallel processes.	Numerical example
	Fallah et al. (2011)	An approach for buffer sizing based on the principles of factory physics.	Numerical examples
	Louw and Page (2004)	An open queuing network modelling approach to estimate the buffer size in production systems.	Simulation
Buffer management	Georgiadis and Politou (2013)	A dynamic time-buffer control mechanism.	Simulation
	Lenort, Klepek, aWhicher and Besta (2013)	A method for determination and control of buffers to protect the floating constraints.	Case Study
	Yung-Chia Chang (2011)	A weighted layer production buffer and weighted production buffer to monitor the status of the buffer deviation in re-entrant flow shop with S-DBR.	Simulation
	Chang and Huang (2011)	A layer production buffer to monitor the buffer status in S-DBR.	Simulation
	Hadas, Cyplik, and Fertsch (2009)	A dedicated planning system and shop floor control for manufacturing high-power marine engines.	Simulation
	Woo, Park, and Fujimura (2009)	A real-time buffer management method.	Simulation
	Riezebos, Korte, and Land (2003)	An integration of workload control principles to improve a DBR buffering approach.	Real application

Source: Proposed by the author

A queuing network model for estimating buffer size in production systems was created by Louw and Page (2004). The length of the time buffer is determined by utilizing the average flow time and standard deviation flow time of a production network modeled as GI/G/m queues.

In the realm of buffer management, Georgiadis and Politou (2013) devised a dynamic time-buffer control mechanism. They acknowledged that the challenge with the DBR approach lies in maintaining constant time buffers throughout the planning horizon. However, real-world manufacturing systems are subjected to variations in key factors such as demand and production times, making constant throughput impractical. To address this, the authors proposed a dynamic time-buffer control mechanism for short- to medium-term Production Planning and Control. Their approach incorporates an adaptive response to demand changes and ensures robustness in both internal and external shop environments. Additionally, they developed a system dynamic model to support decision-making on production time-buffer policies, integrating the proposed mechanism into a flow shop system.

Lenort, Klepek, and Whicher (2013) designed a method for determining and controlling buffers that protect floating bottlenecks from capacity losses resulting from the transfer of the constraint to another manufacturing stage.

To monitor the buffer deviation in a re-entrant flow shop operated by S-DBR, Chang, and Wen-Tso (2011) introduced the concept of a weighted layer production buffer and weighted production buffer.

Chang and Huang (2011) proposed an alternative approach for re-entrant manufacturing environments. They argued that S-DBR's dispatching rule is limited in such environments. To address this limitation, they introduced the layer production buffer to monitor buffer status and sequence orders accordingly.

In the context of high-power marine engine production, Hadas, Cyplik, and Fertsch (2009) emphasized the significance of reducing manufacturing lead time to decrease work-in-progress inventory and capital employed. They proposed a dedicated planning system and shop floor control method for buffering critical resources in complex make-to-order manufacturing. Their approach included buffer management procedures, system disruption compensation, and feedback protection based on the influence of "wandering bottlenecks."

Woo, Park, and Fujimura (2009) proposed a real-time buffer management method to address a limitation of DBR. They noted that DBR only manages a buffer based on the arrival status of raw materials or products at the constraint, without considering their position. To overcome this issue, they introduced a real-time buffer management method that extends the concept of a buffer using DBR scheduling techniques in response to real-time information. This method incorporates updated information, such as changes in the schedule made by the operator and real-time progress updates of the process.

Riezebos, Korte, and Land (2003) conducted research aimed at improving the lead-time performance of a small packaging manufacturer that had implemented the Theory of Constraints (TOC). As part of their approach, they integrated the principles of Workload Control into DBR to manage the buffer system.

c) Rope

This section presents the 8 (25%) studies that propose changes to the rope mechanism, relative to the constraint scheduling, present in Table 3.7.

Table 3.7 - Studies that affected the rope

ROPE operation affected	Reference	Changes	Evaluation method
Constraint Scheduling	Yue et al. (2022)	A heuristic based on the DBR method is proposed for order scheduling and multi-item scheduling problem considering capacity constrained resources (CCR).	Simulation
	Qiao and Wu (2013)	A layered constraint scheduling algorithm.	Case study
	Kasemset and Kachitvichyanukul (2010)	A method to schedule constraints in a job-shop environment with a bi-level multi-objective mathematical model.	Numerical example
	Wu,Chen, Tsai and Yangl. (2010)	A DBR customized algorithm for constraint scheduling on TFT-LCD cell plants.	Numerical example
	Guan et al. (2008)	An algorithm for constraint scheduling which minimizes the maximum tardiness time.	Not tested
	Wu and Yeh (2006)	Algorithm to sequencing constraint operations of re-entrant flows.	Numerical example and real application
	Chen and Lee (2001)	A two-stage exhaustive constraint-based group scheduling procedure (BGSP) to improve cells with load imbalance.	Simulation
	Onwubolu (2001)	A tabu search heuristic to optimize the product mix on multiple constraint scheduling.	Simulation

Source: Proposed by the author

Yue et al. (2022) conducted research on order scheduling and the multi-item scheduling problem considering capacity-constrained resources (CCR). They proposed a heuristic based on the DBR method for a multi-item production environment. The heuristic was utilized to schedule customer order deliveries in each planning horizon and determine the release sequence of production orders for upstream resources in the production system. The proposed heuristic was tested on various real-life problems, and the results demonstrated its superiority over other heuristics discussed in the literature, yielding more significant outcomes.

To schedule the constraint in a job-shop environment, Kasemset and Kachitvichyanukul (2010) formulated a bi-level mathematical model based on the DBR system. The first level aimed to generate an initial schedule that minimizes idle time at the bottleneck.

The second-level model further optimized the schedule to meet multiple objectives while adhering to the constraints that dictate the job sequence froze at the bottleneck obtained from the first level. Simultaneous management of other resources was carried out at this level to ensure maximum utilization of the constraint.

In the context of constraint scheduling in thin-film transistor liquid crystal display (TFT-LCD) cell imaging, Wu et al. (2010) developed a customized DBR solution that specifically addressed schedule constraints. The proposed DBR algorithm calculated ruins - ideal production schedules for all batches at the constraint station - for each batch and allocated these batches to machines.

Guan et al. (2008) proposed minimizing the maximum tardiness in constraint scheduling as their objective. They developed an algorithm for group scheduling on the constraint, aiming to minimize the maximum tardiness time. The algorithm grouped products based on similar manufacturing processes and performed sequencing within each group.

Chen and Lee (2001) presented a solution for constraint-based scheduling in cellular manufacturing. Their approach involved a two-stage exhaustive bottleneck-based group scheduling procedure (BGSP) integrated with the DBR system to improve cells with load imbalance.

Onwubolu (2001) pointed out that the traditional TOC algorithm is not effective in maximizing profit returns in systems with multiple constraints. As a solution, he developed a tabu search-based TOC product mix heuristic to identify optimal or near-optimal product mix when multiple constrained resources are present.

Addressing the challenge of implementing DBR in the presence of re-entrant process constraints, Wu and Yeh (2006) developed an algorithm to ensure a more suitable sequencing of constraint operations and provide sufficient time between adjacent constraint operations within a batch. Similarly, Qiao and Wu (2013) investigated re-entrant manufacturing systems, which are recognized as highly complex. They proposed a DBR-based programming approach for re-entrant manufacturing of layered parts, incorporating a layered scheduling algorithm (LSA).

II) Two mechanisms changed

Two studies focused on two mechanisms, as shown in Table 3.8. Daniel and Guide (1997) targets the Drum and Buffer and Pass and Ronen (2003) aims on the Drum and Rope mechanisms. There is no study on the S-DBR that has changed the two mechanisms.

Regarding dispatching rules and buffer sizing, Daniel and Guide (1997) propose a change to the drum and the buffer. They highlight that there have been no formal examinations to best support DBR on what priority dispatching rules should be used at non-constraint work centers. Furthermore, a few researchers studied the effectiveness of DBR in the remanufacturing environment. Therefore, the authors carried out a study to exam some priority dispatching rules in combination with DBR under a variety of utilization levels, via a simulation model of a remanufacturing facility.

Table 3.8 - Studies that affected two mechanisms

Changed mechanisms	Reference	Adaptations	Evaluation method
Drum and Buffer	Daniel and Guide (1997)	Application of dispatching rules and a change in the buffer size determination.	Simulation
Drum and Rope	Pass and Ronen (2003)	Rules for scheduling and control of the jobs.	Not tested

Source: Proposed by the author

To schedule the manufacturing according to the market constraint in the Hi-Tech industry, Pass and Ronen (2003) created a systematic approach. It is argued that in a market-constrained environment, the marketing and sales department together with the research and development department are Permanent Constraints and need to be managed as such. The authors modify the Theory of Constraint’s five focusing steps to accommodate the market constraint, which affected the constraint scheduling (drum) and release of jobs (rope).

III) Three mechanisms changed

The studies about three mechanisms affected were divided into two groups – one that studied DBR and another group that studied the S-DBR, as can be seen in Table 3.9.

Saif et al. (2019) developed a production planning and scheduling method, based on DBR, that can be utilized in Industry 4.0. They proposed a DBR-based heuristic algorithm (DBR-HA) for multi-level planning considering shifting bottleneck resource to make an efficient schedule in a rolling horizon in a mixed model production environment and utilize capacity constraint resource (CCR) at maximum. According to the authors, the DBR-HA identifies the drum, i.e., CCR, makes an efficient schedule on it in each lower-level scheduling period and utilizes a feedback method to update customer orders in each medium level planning horizon. The proposed method is useful to implement Industry 4.0 in mixed model industries and update their plan and schedule in real time. The performance of the proposed DBR-HA algorithm is measured and compared with the performance of the basic scheduling rules used in the Case Company based on a Case Company problem data. Results indicate that the proposed method is significant to reduce the gap between medium level planning and lower-level schedules and gives an efficient medium level plan and lower-level schedule in each planning horizon as compared to the other methods.

Table 3.9 - Studies that affected the drum, buffer and rope

Changed system	Reference	Adaptations	Evaluation method
DBR	Saif et al. (2019)	This research proposed a DBR-based heuristic algorithm (DBR-HA) for multi-level planning considering shifting bottleneck resource to make efficient schedule in rolling horizon in mixed model production environment and utilize capacity constraint resource (CCR) at maximum.	Simulation
	Rhee, Cho, and Bae (2010)	A method to enhance the efficiency of business processes.	Simulation
	Liu, Tian and Sawaragi (2007)	A heuristic algorithm for solving container loading problems using the five-focusing steps, based on the DBR of the TOC and the multi-agent cooperative negotiation model.	Simulation
	Sirikrai and Yenradee (2006)	A Modified drum-buffer-rope (MOD-DBR) for a non-identical parallel machine flow-shop environment.	Simulation
S-DBR	Chang and Huang (2013)	An enhanced S-DBR model to be applied in a re-entrant flow shop (RFS)	Simulation
	J.-H. Lee et al. (2010)	A new S-DBR approach with rules for the operation of make-to-order environments with multiple or interactive CCR, as well as rules for the inclusion of urgent orders.	Numerical examples

Source: Proposed by the author

A method to enhance business processes was proposed by Rhee, Cho, and Bae (2010). The method, based on the DBR, was created to enhance the efficiency of business processes execution by controlling task allocation.

The DBR was applied to solve the container loading problem (CLP) and two-row pattern by Liu, Tian and Sawaragi (2007). CLP is a real-world problem arising from the centers of physical distributions, manufactories, and warehouses. In the two-row pattern all cargoes are considered rectangular parallelepipeds of known sizes and weights, where each cargo is positioned parallel to the side walls of a container - hence the use of the term two lines. To solve this problem, they proposed a heuristic algorithm based on the DBR, five-step focusing process and the cooperative negotiation strategy. The goal is to improve the positioning packing in the container, exploiting the constraints: First-In-Last-Out (FILO) packing ordering, the total load capacity of the container and the loading of all packing in containers.

A modified scheduling mechanism for non-identical parallel machine flow shop environment was created by Sirikrai and Yenradee (2006), called Modified Drum-Buffer-Rope (MOD-DBR). The MOD-DBR aims to combine the DBR scheduling mechanism with the finite-capacity loading mechanism and replace the buffer and rope mechanisms by the backward loading and scheduling procedure.

Next, S-DBR studies are commented on. Chang and Huang (2013) applied the S-DBR to enhance a model for a re-entrant flow shop (RFS) environment. In the model, job processing times are generated from a discrete uniform distribution and machine breakdowns are subject to an exponential distribution. Improvements were made to the due-date assignment method, and the order release and the dispatching rules.

An alternative method that enhances the S-DBR system performance was developed by J.-H. Lee et al. (2010). The new S-DBR approach presents rules for the operation of a fluctuating make-to-order environments with interactive or multiple CCRs, as well as rules for the inclusion of urgent orders.

3.3.2 Motivations for changes on DBR and S-DBR

Out of the fifteen studies, which account for 46% of the total, the motivation was the evolution of DBR or S-DBR rather than a specific industry requirement. These studies proposed advancements in the identification of floating and multiple constraints(KASEMSET; KACHITVICHYANUKUL, 2010; KHALIL; STOCKTON; FRESCO, 2008; LEE et al., 2010b; NOWOTARSKI; PASLAWSKI, 2017; URBAN, 2019; WANG; LI, 2011; YANG; TSAI; LIU, 2008), scheduling of constraints (ONWUBOLU, 2001; WU; LIU, 2008), buffer time sizing (GEORGIADIS; POLITOU, 2013; LOUW; PAGE, 2004; WOO; PARK; FUJIMURA, 2009), dispatch rules (GONZALEZ-R; FRAMINAN; RUIZ-USANO, 2010a)(GONZALEZ-R; FRAMINAN; RUIZ-USANO, 2010), and Industry 4.0 implementation (SAIF et al., 2019).

Urban (2019) proposed a method to identify constraints based on a DBR implementation in a manufacturing system with multiple floating constraints. Z. W. Wang and Li (2011a) developed a method to identify floating constraints arising from variations in product mix and processing times at individual workplaces. Khalil, Stockton, and Fresco (2008) devised another method to identify constraints due to the compromised effectiveness of DBR caused by high levels of variability, which can lead to the shifting of constraint resources and ineffective scheduling, buffer positions, and sizes. Lee et al. (2010) modified S-DBR to handle environments where constraints are not located in the middle of the routing, as assumed in S-DBR, and to identify multiple constraints.

Onwubolu (2001) proposed a method for scheduling multiple constrained resources, considering the product mix. Wu and Liu (2008) implemented a new mechanism in DBR to enable available-to-promise since the DBR scheduling algorithm only considers firm orders.

Georgiadis and Politou (2013) developed a method for dynamic buffer time sizing to accommodate changes in demand and stochastic production times. Louw and Page (2004) provided a solution for time buffer sizing aimed at reducing production lead time and gaining a competitive advantage by meeting market demand. Woo, Park, and Fujimura (2009) addressed the supervision of product progression through the production process, which, according to the authors, can confuse production.

Due to DBR having only one dispatch rule, Gonzalez-R, Framinan, and Ruiz-Usano (2010) tested several alternative dispatching rules to improve DBR performance. Saif et al. (2019) developed a DBR-based production planning and control system for Industry 4.0, driven by the lack of intelligent systems in this new context.

Eight studies (27%) customized DBR or S-DBR for specific environments. Wu et al. (2010) adapted DBR for thin film transistor liquid crystal display (TFT-LCD) manufacturing, acknowledging the need for customization based on the unique characteristics and requirements of different manufacturing environments. Chen and Lee (2001) proposed changes to DBR for cellular manufacturing, recognizing that DBR does not adequately address the complexity of cellular manufacturing. Guan et al. (2008) modified DBR to balance and stabilize production lines in high-mix, low-volume environments with diverse customer requirements and uncertainty. Wang, Cheng, Tsen, and Liu (2015) modified DBR to address inventory management challenges in hospitals, aiming to guarantee the availability of medical and surgical supplies at the lowest cost. Hadas, Cyplik and Fertsch (2009) focused on the production of high-power marine engines, where the production process is expensive and lengthy, requiring changes in DBR to reduce work in progress and capital employed. Riezebos, Korte, and Land (2003) adapted DBR to meet the peculiarities of a small packaging manufacturer in the Netherlands. Daniel and Guide (1997) developed a new version of DBR for remanufacturing, as few established scheduling systems for this environment were reported in the literature. Pass and Ronen (2003) customized DBR for the Hi-Tech industry.

Another eight studies (27%) customized DBR for specific manufacturing processes. DBR cannot establish the Rope for a non-identical parallel machine flow-shop environment, leading Sirikrai and Yenradee (2006) to propose changes to DBR. Wu and Yeh (2006), Qiao and Wu (2013), Yung-Chia Chang (2011), and Chang and Huang (2013) addressed the challenges of implementing DBR in re-entrant flows present in manufacturing systems. Rhee, Cho, and Bae (2010), Liu, Tian, and Sawaragi (2007), Radovilsky and Frankel (2013), and Fallah et al. (2011) did not explicitly state their motivation for changing DBR.

3.3.3 Research avenues

Based on the studies presented here, we recommend some research avenues. Regarding DBR, new studies can be developed with case studies. Also, new research can be developed on the programming of the interactive CCRs - where one CCR feeds others - and the floating CCRs. Studies on shipping and assembly buffers are also required.

Studies can be performed about the S-DBR / MTO and S-DBR / MTA, since they have been studied little so far. We recommend an integration between S-DBR / MTO and S-DBR / MTA, since they have different rules, but can be applied together in hybrid environments. Capacity management in these hybrid environments is quite challenging. It may also be interesting to study the adaptations made to the S-DBR during its implantation in the companies, through case studies. Further studies may improve the method of production control and sequencing based on buffer status for both S-DBR / MTO and S-DBR / MTA. It is also interesting to carry out investigations and propositions about the algorithm conventionally used for dynamic dimensioning of the buffers in S-DBR.

New research could be carried out about shifting DBR to S-DBR, where DBR is already implemented. Other studies may also apply the proposed adaptations to the DBR on S-DBR. Studies addressing S-DBR could focus on planned load with floating, multiple, or non-existent active RRCs. Stock buffer management, in the case of the MTA, integrated with the distribution and production is another opportunity not covered by current literature. The S-DBR can also be adapted to consider the limitation of space for work in process and stock of finished products. Studies that present S-DBR applications in real environments and the adaptations made can be a great contribution. These are just a few suggestions from the many possible studies on S-DBR.

The S-DBR simplifies the DBR that assumes new rules to manage make-to-stock environments and inventories of the distribution networks. Therefore, the S-DBR is a vast field of research little explored. Complications exposed by Schragenheim, Dettmer, and Patterson (2009) imposed by manufacturing environments can inspire future research. Know what they are:

- Multiple Operations of the CCR: there are two distinct situations in which a resource must perform multiple operations (in some sequence) in the same order.
- Multiple Capacity-Constrained Resources (CCRs): there are two or more work centers with approximately equal capacity, and those work centers are the “weakest links” of the shop floor.
- Wandering Bottlenecks: a situation where bottleneck moves between workstations that mix changes can cause.
- Dependent Setups in S-DBR: Simplified Drum–Buffer–Rope operates on the underlying assumption that the actual sequence of work on the CCR (or other resources, for that matter) does not adversely affect capacity in any significant way. Sequence-Dependent setups break this assumption.

There is at least one unanswered question concerning the S-DBR mode for managing inventories of the distribution networks. It is supposed to frequently replenish inventories. Schragenheim, Dettmer, and Patterson (2009) argue that frequent replenishments maintain low inventories and the buffers full, but they did not point to a tool that supports the decision-making regarding when to replenish. This is not an easy decision; future research can deal with this question.

3.4 Conclusions

This study utilizes a systematic literature review approach to identify research that proposes adaptations to DBR and S-DBR, as well as research gaps relevant to the thesis. Many studies have focused on individual system activities, such as buffer sizing, buffer management, constraint scheduling, identification, and exploitation. These studies were primarily motivated by the need for system evolution or adaptation to address the specific characteristics of different manufacturing environments.

It is worth noting that there is a relatively low number of research studies specifically dedicated to S-DBR, indicating that it remains less known and studied compared to DBR. While DBR gained attention among executives and entrepreneurs in the '80s through the book "The Goal" (GOLDRATT; COX, 2003), S-DBR only emerged in 2000.

In the section discussing research avenues, we highlight gaps identified in the book "Supply Chain Management at Warp Speed," which provides insights into how S-DBR functions in make-to-stock production environments and manages inventory within distribution networks (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). The authors of the book point out that S-DBR faces challenges when managing production lines with multiple operations of the CCR, multiple capacity-constrained resources, wandering bottlenecks, and dependent setups. Additionally, they note that inventory management for distribution networks in S-DBR is incomplete.

The review of existing literature did not yield a specific solution for the identified problems, such as dependent setups and inventory replenishment planning in goods distribution networks. Therefore, we have identified these as research gaps for the thesis. Given that these problems are common in factories and logistics companies, they deserve further attention and investigation.

4 EVALUATING DISPATCHING RULES FOR MAKE-TO-AVAILABILITY UNDER SIMPLIFIED DRUM-BUFFER-ROPE: A COMPUTATIONAL EXPERIMENT

4.1 Introduction

Make-to-availability (MTA) is a make-to-stock production method within the simplified drum-buffer-rope (S-DBR) production planning and control system (SCHRAGENHEIM; DETTMER; PATTERSON, 2009), which, in turn, is an evolution of the drum-buffer-rope (DBR) system (GOLDRATT, 1986) whose purpose is to guarantee the availability of finished products to meet demand through stock buffers (SCHRAGENHEIM, 2010; SCHRAGENHEIM; DETTMER; PATTERSON, 2009). DBR emerged from the Theory of Constraints (TOC) (GOLDRATT; COX, 1984; SCHRAGENHEIM; DETTMER; PATTERSON, 2009), conceived in the 1970s by Eliyahu M. Goldratt as a programming algorithm that became a broader concept of production planning and control (SIMONS; SIMPSON, 2009). The DBR is a programming mechanism consisting of a drum, buffer, and rope mechanisms, that controls the release of tasks to the system according to the production bottleneck (or constraint) (HUNG; HUANG; YANG, 2022; THÜRER; STEVENSON, 2018b). The drum sets the production pace, the buffer protects the drum against system variability, and the rope controls the release of materials to the factory floor. In the MTA method, the drum is the market demand, the buffer is the stock of each product – the sum of the *finished goods inventory* (FGI) plus the stock or *work in process* (WIP) –, and the rope controls the workload on the production line and releases the production orders (OP) to the shop floor (SCHRAGENHEIM, 2010). Production must quickly replenish the units sold to keep the buffer full; therefore, the production lead time must be reduced (BARCO; FILHO, 2021). Lead time reduction depends on the dispatching rule (DR) adopted. Dispatch rules are used in production scheduling to prioritize tasks in workstation queues (HEGER et al., 2016; NGUYEN; ZHANG, 2017a).

Some studies found in the literature and commented hereafter show that changes in the DR can improve production performance. Heger *et al.* (2016) proposed a hybrid approach that combined global information based on offline simulations with local adaptive decision rules, reducing the average delay by approximately 9%. El-Bouri and Nairy (2011) proposed a cooperative dispatch (CD) rule, showing that it reduced the average flow time compared with other rules. Huang and Chen (2018) created a mixed dispatch rule that assigned priorities to tasks in two stages. The created rule proved to be more effective in terms of on-time deliveries than the *shortest processing time* (SPT), *least work remaining* (LWKR), *and total weighted processing time* (TWPT) strategies. Nasiri *et al.* (2017) developed a DR composed of other DRs to minimize the average waiting time for tasks in an open-shop environment. The results indicated that the proposed DR was superior to the *first-in-first-out* (FIFO), *last-in-first-out* (LIFO), *longest processing time* (LPT), and SPT rules. The S-DBR/MTA method has a rule called prioritizing by buffer status (PSP), which prioritizes tasks considering the inventory level of finished products and processes (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). In this study, we created a rule derived from the PSP one, which we call PSP1, by removing the variable WIP. As in other studies, we wanted to evaluate a set of dispatching rules to improve the production performance in an environment managed by an S-DBR/MTA system.

Therefore, this study aims to evaluate the performance of S-DBR/MTA by incorporating other DRs. Four literature rules (FIFO, SPT, arrival time (AT), and *Shortest remaining processing time* (SRPT), two S-DBR/MTA rules (PSP and PSP1), and six combinations of the above (PSP-AT, PSP-SPT, PSP-SRPT, PSP1-AT, PSP1-SPT, and PSP1-SRPT) were applied. Because S-DBR/MTA does not assign deadlines or completion dates to tasks, we chose the SPT, SRPT, AT, and FIFO rules, which do not use such information. We collected data through simulation, and the results were evaluated in terms of service level (S), average stock in the system ($\bar{E}$), average flow time ($\bar{F}$), and rate of inventory by the percentage of service level ($\bar{C}$).

The rest of the article has the following structure: Section 4.2 presents the notation used; Section 4.3 presents details on the operation of the MTA method, as well as the dispatching rules evaluated; Section 4.4 presents the configuration and results of the computer simulation; and finally, Section 4.5 presents the conclusions of the study.

4.2 Notation

The notation used in this work is as follows:

P_{ir} priority assigned to task i for product r using the dispatch rule.

EA_r target level of product buffer r.

NP_r buffer stock level r.

WIP_r work-in-process of product r.

FGI_r finished goods inventory of product r.

t_{im} processing time of task i on machine m.

T_i time of arrival of task i to the system.

F_{im} time of arrival of task i to the queue of machine m.

τ moment when the dispatch takes place.

O_i remaining processing time for task i in the system.

SP_{ir} buffer status of task i for product r.

4.3 Theoretical background

4.3.1 Make-to-availability

The purpose of the S-DBR/MTA system is to replenish the units sold without requiring an accurate sales forecast process (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). The system consists of three components: a drum, buffer, and rope. The drum is the market demand, which dictates the pace of production by consuming stock items and triggers the tasks for replenishing the stock (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). Each product has its buffer, which is the sum of the FGI plus the WIP and aims to protect deliveries against variations in demand and supply (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). The target level (TL) limits the buffer calculated for each product (SCHRAGENHEIM, 2010). Production must keep stocks close to the TL. To control the buffers, the S-DBR/MTA system uses dynamic buffer management (DBM), responsible for adjusting the TL as spare time and demand change (SCHRAGENHEIM, 2010; SCHRAGENHEIM; DETTMER; PATTERSON, 2009). The rope controls the release of tasks to the shop floor (SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

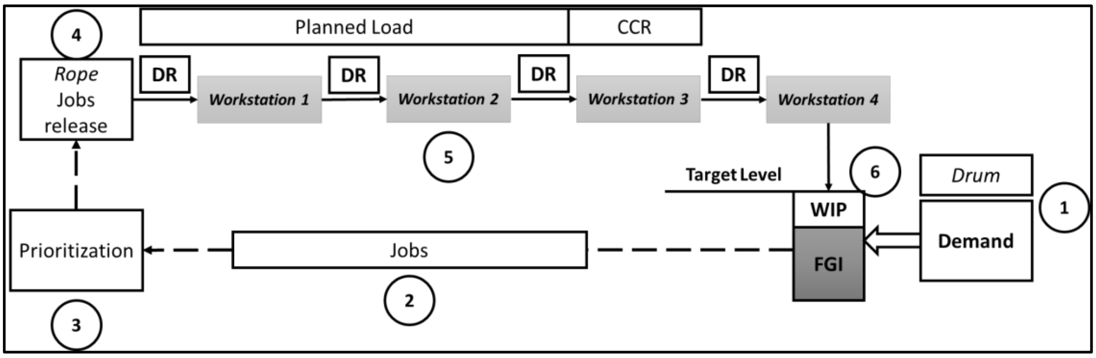
When the sum of the FGI and WIP for a given product is below the established TL, a production order (SCHRAGENHEIM; DETTMER; PATTERSON, 2009) that goes to the release queue on the shop floor is generated. A planned load (PL) releases tasks while limiting the load on the resource with the least capacity on the production line called the capacity constraint resource (CCR) (SCHRAGENHEIM, 2010; SCHRAGENHEIM; DETTMER; PATTERSON, 2009). Therefore, tasks are released only when the load on the CCR is below a pre-established limit (SCHRAGENHEIM, 2010; SCHRAGENHEIM; DETTMER;

PATTERSON, 2009). The first task to be released is that with the highest percentage of the buffer that needs to be filled – penetration into the buffer – as indicated by:

$$SP_{ir} = \frac{(EA_r - (WIP_r + FGI_r))}{EA_r} \qquad (1)$$

Figure 4.1 helps understand the S-DBR/MTA process, with enumerated circles showing the sequence of activities from the consumption of buffer items until their replenishment. A dispatching rule operates in the queues at the workstations, as shown in the figure.

Figure 4.1 - Make-to-availability operation



Source: Proposed by the author

The consumption of the FGI triggers the manufacturing process under the S-DBR/MTA (1). The withdrawal of products from the FGI generates production orders (2) to a queue in the factory, where they are prioritized (3) according to equation 1. Prioritized orders wait to be released to the shop floor (4) until the planned load signals that there is capacity at the workstations (5) for processing. Finished production orders replenish the FGI (6). This cycle repeats continuously to keep the FGI close to the TL.

4.3.2 Dispatching rules

This section begins by briefly considering the dispatch rules used in the study. Dispatch rules are used in production scheduling to prioritize tasks in workstation queues (HEGER et al., 2016; NGUYEN, 2017). Unlike optimization methods, DRs decide when there are tasks in the queue and idle machines, instead of being determined in advance by an algorithm (NGUYEN, 2017). This mechanism allows DRs to use updated system information and make bold decisions (NGUYEN, 2017). DRs are generally fast and can be quickly implemented, which is a significant advantage (FRAMINAN; FERNANDEZ-VIAGAS;

PEREZ-GONZALEZ, 2019). However, there is no DR that is good for all production environments (GONZALEZ-R; FRAMINAN; RUIZ-USANO, 2010).

4.3.3 The make-to-availability dispatch rule

The S-DBR/MTA dispatch rule is called prioritization by buffer status (PSP) (SCHRAGENHEIM et al., 2009) and is expressed as:

$$P_{ir} = \frac{(TL_r - (DWIP_r + FGI_r))}{TL_r} \qquad (2)$$

The PSP equation (Equation 2) is like the one used to release the tasks (Equation 1). The only difference is the replacement of parameter WIP_r by $DWIP_r$. We also derived a PSP expression, which we call PSP1, by removing the $DWIP_r$ variable:

$$P_{ir} = \frac{TL_r - FGI_r}{TL_r} \qquad (3)$$

The idea is to use PSP1 to determine whether the use of WIP in prioritization improves or worsens the performance of S-DBR/MTA.

4.4 Computational experiments

After implementing a model based on Nguyen et al. (2015) and Thürer and Stevenson (2018), simulation experiments were conducted under 12 scenarios that contemplate the following DRs: PSP, PSP1, FIFO, AT, SPT, SRPT, PSP-AT, PSP-SPT, PSP-SRPT, PSP1-AT, PSP1-SPT, and PSP1-SRPT. The simulation model was implemented in the programming language Python 3.5, using the SimPy library, version 3.0.10. A PC with an Intel Core i5-6200U 2.3 GHz processor and 8 GB RAM was used to run the experiments. The warm-up time and the time to collect statistical data corresponded to the completion of 1000 and 5000 tasks, respectively, according to Nguyen *et al.* (2015). Each scenario was replicated 50 times (NGUYEN et al., 2015).

4.4.1 Simulation setup

The simulated system consisted of a non-permutational flow-shop production line with seven stations composed of a single machine (LADJ; TAYEB; VARNIER, 2021; THÜRER; STEVENSON, 2018a) and ten different products. As presented in Section 4.3.1, the S-DBR/MTA was implemented without the target stock adjustment mechanisms and the planned load, so they did not influence the results. Table 4.1 lists the target levels used in the simulation, together with the average times between the arrival of the demands simulated by the exponential distribution. Each demand is equivalent to the buffer unit of a product as well as for each task. The processing times of the products in the machines follow a uniform distribution according to the lower and upper bounds listed in Table 4.2. With these configurations, the production line utilization was approximately 90% under the PSP rule.

Table 4.1 - Target stock units and the time between demands (in time units)

	Products									
	1	2	3	4	5	6	7	8	9	10
Target level	29	36	46	41	54	30	33	36	33	30
Time between demands	16.00	12.65	9.84	11.24	8.43	15.46	14.05	12.50	14.05	15.46

Source: Proposed by the author

Table 4.2 - Processing times (in units of time)

	Machines						
Products	1	2	3	4	5	6	7
1	1.0; 1.5	0.3; 0.8	2.2; 2.7	1.1; 1.3	0.6; 1.1	0.7; 1.2	1.0; 1.5
2	0.5; 1.5	2.2; 2.7	0.5; 1.0	0.4; 0.9	0.5; 1.0	0.8; 1.3	3.9; 3.4
3	0.1; 0.5	0.5; 1.0	2.6; 3.1	0.7; 1.2	0.8; 1.3	2.0; 2.5	0.3; 0.8
4	2.0; 2.5	0.1; 0.3	0.3; 0.5	0.2; 0.4	3.3; 3.5	0.5; 1.0	0.5; 1.0
5	1.0; 1.5	0.2; 0.7	0.8; 1.3	2.9; 2.4	0.1; 0.6	2.5; 3.0	0.1; 0.5
6	1.2; 1.7	0.3; 0.6	0.5; 1.0	2.0; 2.5	0.1; 1.7	0.4; 0.6	1.3; 1.8
7	0.1; 0.5	0.9; 1.4	0.5; 1.5	2.1; 2.6	1.0; 1.5	0.7; 1.3	0.1; 0.6
8	0.5; 1.0	2.0; 2.5	0.1; 0.6	1.0; 1.5	1.3; 1.8	0.5; 1.5	1.5; 2.0
9	0.1; 0.6	0.9; 1.1	2.0; 2.5	0.3; 0.8	1.0; 1.5	0.3; 0.8	1.0; 2.0
10	0.5; 1.0	1.2; 1.7	1.3; 1.8	0.6; 1.1	0.5; 1.0	0.6; 1.1	0.1; 0.6

Source: Proposed by the author

4.4.2 Selected and combined rules

The instruction derived from a DR depends on the production objective that must be met, such as minimizing the average flow time or average delay. In the S-DBR/MTA system, the most appropriate rules are those that minimize the average flow time because they increase the frequency of product replenishment, although rules that are not based on deadlines or dates can also be applied. Considering these assumptions, we selected the following DRs:

1) *First-in-first-out* (FIFO): the highest priority is given to the task that first arrived at the queue (RAJENDRAN; HOLTHAUS, 1999). Priority equation: P_i=F_{im}.

2) *Arrival time* (AT): it prioritizes the task that arrived earlier in the system (RAJENDRAN; HOLTHAUS, 1999). Priority equation: P_i=τ-T_i.
3) *Shortest processing time* (SPT): it is efficient in minimizing both the average flow time and average delay on a highly loaded shop floor (GONZALEZ-R; FRAMINAN; RUIZ-USANO, 2010). It prioritizes the task with the least processing time. Priority equation: P_i=T_{im}.
4) *Shortest remaining processing time* (SRPT): seeks to minimize the average flow time (GONZALEZ-R; FRAMINAN; RUIZ-USANO, 2010). It prioritizes the task with the least remaining processing time. Priority equation: P_i=O_i.

S-DBR/MTA rules use stock, whereas rules selected from the literature consider time data. We consider that rules combining both data can better balance the need to replenish the buffer's items with a higher percentage of penetration and speed of replenishment, increasing the service level.

These are the combined rules:

1) PSP-AT: chooses the task with the highest priority P_i = (τ-T_i)*PSP.
2) PSP-SPT: chooses the task with the lowest priority P_i = T_{im}/PSP.
3) PSP-SRPT: chooses the task with the lowest priority P_i = O_i/PSP.
4) PSP1-AT: chooses the task with the highest priority P_i = (τ-T_i)*$PSP1$.
5) PSP1-SPT: chooses the task with the lowest priority P_i = $T_{im}/PSP1$.
6) PSP1-SRPT: chooses the task with the lowest priority P_i = $O_i/PSP1$.

4.4.3 Performance indicators

To evaluate the performance of the S-DBR/MTA when using different DRs, the following indicators were computed:

1) Service Level (S): fraction of the demand immediately from stock in the long run. It indicates the effectiveness of the system to ensure availability. Its registration occurs at the end of each replication and is calculated by:

$$S = \frac{IAE}{TID} \qquad (4)$$

where IAE is the total number of items served by *the FGI,* and TID is the total number of items demanded.

2) Average inventory in the system ($\bar{E}$): sum of the averages of the *WIP* and *FGI*. The value record occurs every time an item enters or leaves the *FGI*, or a task enters or leaves the production line, which changes the *WIP*. At the end of each replication, the average was calculated according to:

$$\bar{E} = \frac{\sum_{i=1}^{M} WIP_i}{M} + \frac{\sum_{i=1}^{N} FGI_i}{N} \qquad (5)$$

where WIP is the sum of all stocks in the process at time i, M is the number of entries and exits of the tasks on the production line during the replication, FGI is the sum of the stocks of the finished products of all products, and N is the number of records FGI made throughout replication.

3) Average flow time ($\bar{F}$): average time a task remains in production. At the end of each replication, the average was calculated according to:

$$\bar{F} = \frac{\sum_{i=1}^{N} F_i}{N} \qquad (6)$$

where F_i is the flow time of task *i*, and *N* is the total number of tasks passed through the production line.

4) Inventory rate per service level percentage ($\bar{C}$): indicates how many inventory units are required for each service level percentage. It identifies the rule that best balances the stock level with the service level. It was calculated as follows:

$$\bar{C} = \frac{\bar{E}}{S} \qquad (7)$$

where $\bar{E}$ is the average stock measure in the system and *S* is the service-level measure.

4.5 Results

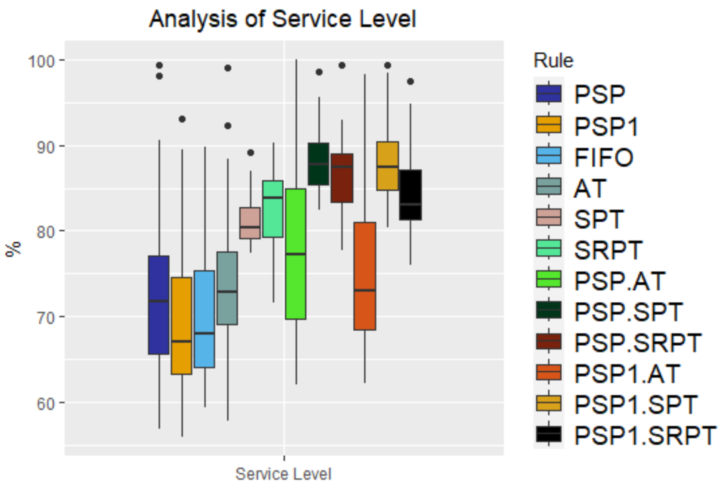
In this section, we analyze the survey results summarized in Table 4.3. Starting with the service level indicator (S), the best performances are observed for the PSP-SPT and PSP1-SPT rules, as can be seen in Figure 4.2. The combination of PSP and PSP1 with SPT increased the service level by 21%, which improved product availability. The PSP-SRPT rule achieved a similar result, with a 19% increase in service level compared to the PSP.

Table 4.3 - Rules performance

	S		$\bar{E}$		$\bar{F}$	
Rule	Average	Std. Dev.	Average	Std. Dev.	Average	Std. Dev.
PSP	0.73	0.10	301.81	23.87	379.38	36.54
PSP1	0.69	0.08	301.55	25.19	393.84	40.27
FIFO	0.70	0.08	304.85	24.44	387.19	38.20
AT	0.74	0.09	292.21	26.51	367.44	40.54
SPT	0.81	0.03	135.80*	5.33	140.12*	8.56
SRPT	0.83	0.05	139.29*	15.98	117.91*	16.92
PSP-AT	0.79	0.11	282.55	32.77	352.74	49.20
PSP-SPT	0.88*	0.04	228.93	20.92	272.94	31.56
PSP-SRPT	0.87*	0.04	269.41	22.35	335.27	36.01
PSP1-AT	0.75	0.08	295.19	24.83	372.65	38.47
PSP1-SPT	0.88*	0.05	220.70*	22.90	260.78*	34.04
PSP1-SRPT	0.84	0.04	271.64	21.23	333.13	31.24

Source: Proposed by the author
*Better results

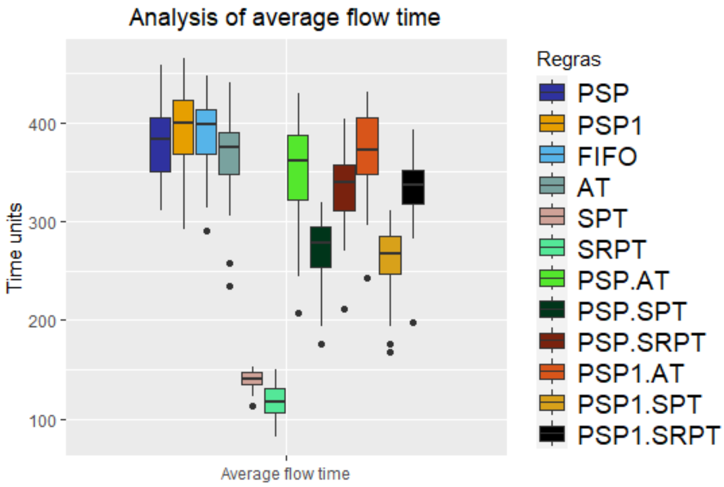
Figure 4.2 - Service level boxplot S



Source: Proposed by the author

Regarding the mean flow time ($\bar{F}$), the SPT and SRPT rules obtained the lowest averages, 69%, and 63% respectively, compared to the PSP, as shown in Figure 4.3. The PSP-SPT and PSP1-SPT rules, which demonstrated the highest service levels, reduced the average flow time by 31% and 28% respectively. Both rules are recognized for their effectiveness in reducing the average production flow time. The Shortest Processing Time (SPT) rule prioritizes tasks based on their shortest processing time, ensuring that the shortest jobs are completed first. This approach decreases wait time in the system as these jobs spend less time in the queue waiting for processing to begin. By minimizing the wait time, job completion is accelerated, resulting in a decrease in throughput time (KOULAMAS; KYPARISIS, 2007). Something similar happens with the Shortest Remaining Processing Time (SRPT) rule, which prioritizes the task with the shortest remaining processing time, ensuring that the task closest to completion is processed first. This dynamic prioritization strategy effectively minimizes flow time by promptly completing the most imminent jobs (BECCHETTI; LEONARDI, 2001; HUNG; CHANG, 2002).

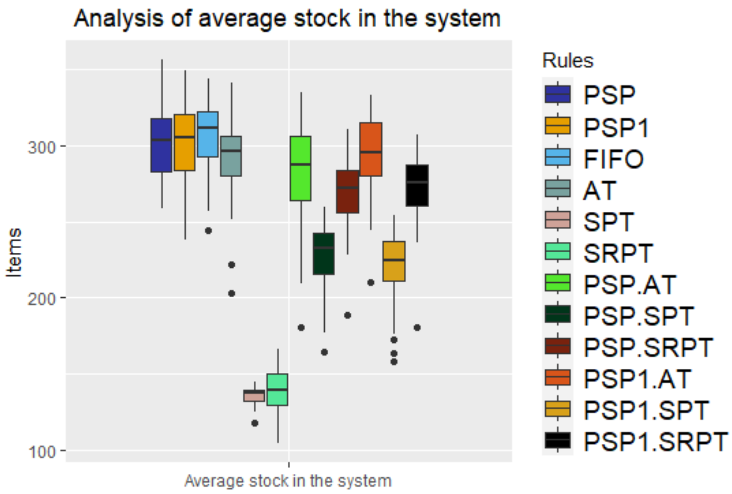
Figure 4.3 - Boxplot of mean flow time $\bar{F}$



Source: Proposed by the author

When it comes to the system's average stock ($\bar{E}$), Figure 4.4 clearly shows that both the SPT and SRPT rules achieved significant reductions compared to the other rules. The SPT rule reduced the stock by 55%, while the SRPT rule reduced it by 54% when compared to the PSP rule. This information is illustrated in Figure 4.4.

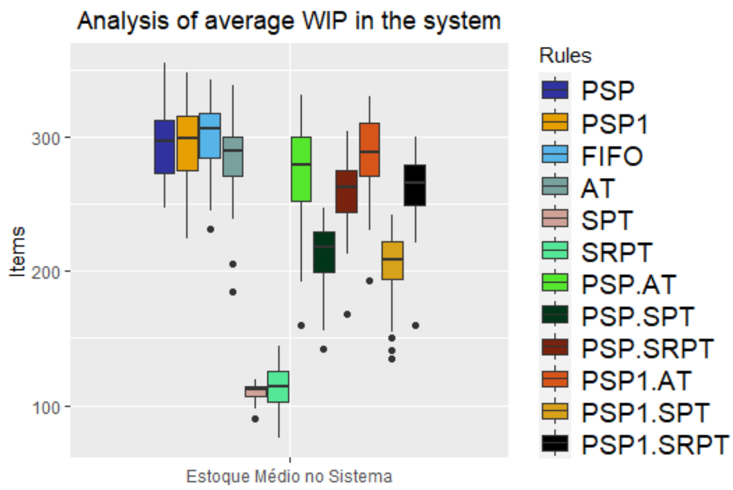
Figure 4.4 - Boxplot of the average stock in the system $\bar{E}$



Source: Proposed by the author

Figure 4.5 makes clear that a decrease in inventory is closely related to a reduction in work-in-process (WIP). The WIP decrease, in turn, is attributed to a decrease in the average throughput time ($\bar{F}$). This interesting phenomenon can be explained by Little's law, which states that, in a stationary system, the average long-term throughput is equal to the ratio of the average total work in process (WIP) to the average flow time (FT) (LITTLE, 1961).

Figure 4.5 - Boxplot of average WIP in the system



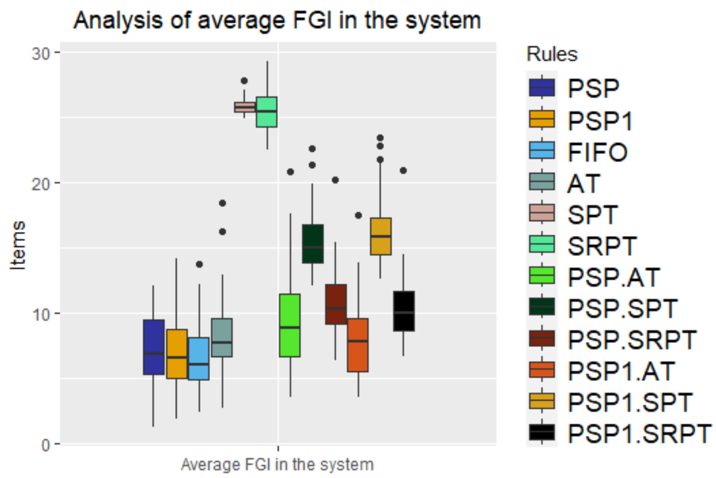
Source: Proposed by the author

The increase in production speed resulting from the SPT and SRPT rules led to a higher level of finished goods inventory (FGI), as shown in Figure 4.6. However, it's important to note that having more FGI didn't give these rules an advantage in terms of service level, as depicted in Figure 4.2. This can be attributed to not considering inventory levels when prioritizing tasks. To illustrate, let's consider a scenario where a job with low inventory and a long processing time must wait for a significant duration until other jobs are processed. During this waiting period, the FGI for that job diminishes. Consequently, the service level performance is negatively impacted. Due to this phenomenon, it is noteworthy that the PSP rule outperformed the SPT and SRPT rules in terms of service level, as illustrated in Figure 4.2, despite having a lower level of finished goods inventory (FGI).

The service level is the principal indicator of the performance of the S-DBR/MTA system; however, there must be a balance between it and the system's average stock. Without this balance, the application of S-DBR/MTA may become unfeasible, owing to the increased inventory costs. Therefore, S-DBR/MTA must work to ensure availability while keeping inventory levels low. This measure was used to determine which rule had the best balance.

Table 4.4 shows the results thus obtained. The classification in Table 4.4 shows that the SPT and SRPT rules provided the best results, with SRPT showing an advantage because of its best service level. The PSP rule was only in the 10th position.

Figure 4.6 - Boxplot of average FGI in the system



Source: Proposed by the author

Table 4.4 - Inventory rate by percentage of service level

Classification	Rule	$\bar{E}$	*S(%)*	E/S(%)
1	SRPT	139.29	83%	1.68
2	SPT	135.8	81%	1.68
3	PSP1-SPT	220.7	88%	2.51
4	PSP-SPT	228.93	88%	2.60
5	PSP-SRPT	269.41	87%	3.10
6	PSP1-SRPT	271.64	84%	3.23
7	PSP-AT	282.55	79%	3.58
8	PSP1-AT	295.19	75%	3.94
9	AT	292.21	74%	3.95
10	PSP	301.81	73%	4.13
11	FIFO	304.85	70%	4.36
12	PSP1	301.55	69%	4.37

Source: Proposed by the author

4.6 Conclusions

This study aims to compare the performance of the PSP dispatching rule to other already known rules (FIFO, SPT, AT, and SRPT) and create a PSP-based rule called PSP 1, which disregards the DWIP parameter. In the study, were created combinations of such rules. The rules were evaluated in a simulation of a flow shop line operated by the MTA.

The results show that the PSP and PSP 1 did not outperform the other rules in any of the four performance indicators: service level, the average stock in the system, average flow time, and inventory rate by the service percentage level. The PSP 1 was no better than the PSP, which reduced the service level by 4% and increased the average streaming time by the same proportion.

In ranking inventory rate by the service level percentage, which shows the most efficient rules and what needs less inventory to reach the service level, PSP 1 had the worst result. Therefore, it is not advisable to disregard the DWIP parameter. This changes when PSP and PSP 1 are combined with the SPT rule, called PSP-SPT and PSP 1-SPT. Both presented the same service level in this case, but the PSP1-SPT obtained 5% less average flow time. In the ranking of the rate of inventory by the percentage of service level, the PSP1-SPT obtained third place and the PSP-SPT fourth place. It is advisable to withdraw the PSP DWIP when combined with the SPT rule, which can compensate for the increase in average flow time generated by the withdrawal of the DWIP.

In ranking the inventory rate by service level percentage, the SRPT and SPT rules were tied for the first position, and the PSP1-SPT was in the third position. The SRPT had a 2% service level advantage compared to the SPT. Among the three, the PSP1-SPT achieved the best level of service. PSP and PSP 1 only obtained the penultimate ante and last positions.

The experiments showed that the PSP rule may not be the best for the MTA, at least in environments like the one studied, and that other rules should be considered to improve the system's performance. This study contributes to the S-DBR literature showing that the proposed production sequencing, the PSP, should be revised. It is important to note that advancing a task whose finished goods inventory is closer to the end can harm the stock of other products. This contribution can be extended to practice as a warning to companies that use S-DBR and those that intend to implement it.

Further research is required to confirm the evidence presented. To achieve this, it is recommended to conduct experiments that test the activated mechanisms of dynamic buffer management and prioritization in releasing tasks based on buffer status and seasonal demand, while considering the prevailing trends and high degree of uncertainty. Additionally, a new study could be conducted to examine the stability of the rules results when applied to different systems with varying characteristics, such as an increased number of products and a larger quantity of machines. This analysis would offer valuable insights into the consistency of the results and whether the rules apply consistently across different production systems. It is important to emphasize that the findings of this study cannot be generalized to other types of production systems. Therefore, future studies should encompass diverse production line arrangements, including job shop, assembly, and cellular manufacturing, to obtain a comprehensive understanding of the topic.

In the context of the thesis, this study is complementary to the study of specific objective 3 - develop a dispatching method for S-DBR/MTA to dependent setup.

5 DISPATCHING METHOD BASED ON PARTICLE SWARM OPTIMIZATION FOR MAKE-TO-AVAILABILITY

5.1 Introduction

Some structural keystones are required to design a manufacturing system; one of the most important is identifying how the enterprise will deliver products to clients. This is known as the fulfill demand strategy. Two main strategy branches for fulfilling the demand of production systems are widely adopted in manufacturing systems: the first is referred to as make-to-order, which produces goods according to a set of client orders. The existence of client orders is a common premise in production scheduling literature, as seen in the study by (BAKER; TRIETSCH, 2018). The second is based on maintaining or establishing the stock levels of produced goods and is used a premisse for lot-sizing planning approaches (DREXL; KIMMS, 1997; JANS; DEGRAEVE, 2008). This strategy is known as make-to-stock.

The make-to-availability (MTA) strategy is derived from make-to-stock and aims to ensure the availability of products by providing rapid stock replenishement, without a formal method of demand forecasting. The MTA was proposed whilst developing the simplified drum buffer rope (S-DBR), a production control system based on the foundations of the theory of constraints (TOC). The TOC has been used extensively to tackle manufacturing issues (GHORBANI et al., 2014; PANIZZOLO, 2016; URBAN; ROGOWSKA, 2020). Urgan and Rogowska (2020) designed a method for identifying a bottleneck in manufacturing system managed by TOC. Panizzolo (2016) conducted an empirical study of the relationship between TOC production and operational performance in manufacturing plants. Ghorbani et al. (2014) applied the TOC thinking process to determine the critical factors in a cellular manufacturing system.

Contrary to production planning approaches (that can be found, for example, in the scheduling literature of Baker and Trietsch (2009), and Zhang and Roy (2019) and the S-DBR operates at the control level of the manufacturing process. Thus, the S-DBR aims to react to inventory levels changes, instead of generating a set detailed resource allocation plan. This continuously updates a production plan, according to a set of well-established rules (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). Similar behavior can be observed in production control techniques, such as Kanban and CONWIP (AL-TAHAT; DALALAH; BARGHASH, 2012; GAURY; PIERREVAL; KLEIJNEN, 2000; KHOJASTEH-GHAMARI, 2012).

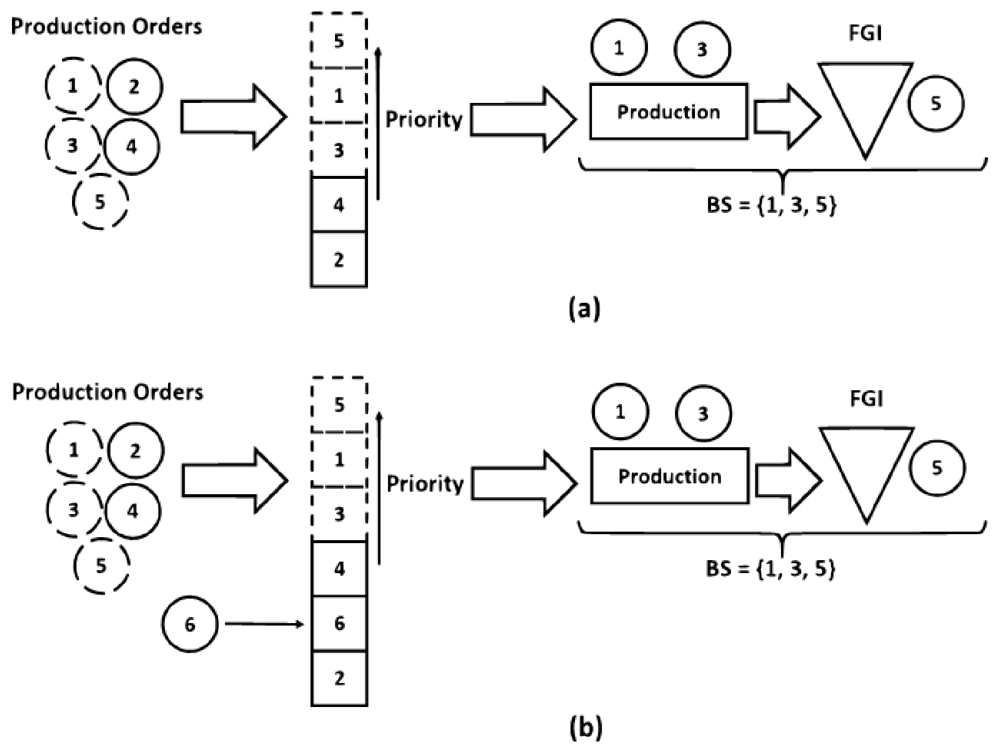
Thus, in S-DBR, production orders (POs) are prioritized, both at the time of production release and in workstations. The S-DBR is based on the following elements. The buffer status (BS) defines the degree of importance of each PO (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). The buffer is the sum of stock in the form of finished goods inventory (FGI) and the work-in-process (WIP) (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). The dispatching PO logic of the MTA, referred to as prioritization by buffer status (PBS), does not consider dependent setup time, thereby increasing the mean flow time, and delaying the replenishment of the stock that was consumed by demand.

The simplified behavior of the S-DBR is illustrated in Figure 5.1. In this figure, five POs initially arrive at the manufacturing system. The production priority of these orders is set as [5, 1, 3, 4, 2] owing to the adopted dispatching method. After some time, PO 5 is produced and stored in the FGI, and orders 1 and 3 are still in the system. This stage is illustrated in Figure 5.1a. At this point, PO 6 arrives in the system. The dispatching method re-prioritizes the remaining POs to [4, 6, 2]. This is presented in Figure 5.1b.

As presented in Figure 5.1, the choice of a proper dispatching method is a vital element for implementing the S-DBR. There are many studies that have present interesting results in terms of solving the dispatching problem with metaheuristics, such as particle swarm optimization (PSO) (MARICHELVAM; GEETHA; TOSUN, 2020; NGUYEN; ZHANG, 2017b; WANG et al., 2016), genetic algorithms (HABIB ZAHMANI, 2017; ROLF et al., 2020; TEPPAN; DA COL, 2020), tabu search (ALI; TELMOUDI; GATTOUFI, 2019; LEE; YU; LEE, 2013; SHAHZAD; MEBARKI, 2016), simulated annealing (BEKTUR; SARAÇ, 2019; PRATA; DE ABREU; LIMA, 2020; VITAL-SOTO; AZAB; BAKI, 2020), and ant colony (KORYTKOWSKI; WIŚNIEWSKI; RYMASZEWSKI, 2013).

However, during our initial review of the literature, we found no studies that presented a solution for S-DBR to effectively incorporate sequence-dependent setup time into its production order prioritization logic.

Figure 5.1 - Example of the operation of an S-DBR system



Source: Proposed by the author

Thus, this study aims to develop a dispatching method for S-DBR/MTA to dependent-setup time environments. To achieve this, we use the continuous optimization found by the particle swarm optimization (PSO) to adjust the production orders prioritization, thereby minimizing the mean flow time and total setup time in environments with a dependent setup time, named particle swarm optimization for Sequence (PSO-S). Although there are further optimization methods that can be applied, we believe that PSO has interesting, documented results, which makes it a good candidate for this problem and has been successfully applied to solve various optimization problems (LIANG; CUEVAS JUAREZ, 2016; MARICHELVAM; GEETHA; TOSUN, 2020). The advantages of the PSO algorithm are its simple structure, ease of implementation, speed in obtaining solutions, and robustness, as demonstrated in the literature. (TASGETIREN et al., 2007).

To evaluate the performance of the new dispatching method, we used a computational simulation to compare this method and the MTA dispatching logic. The results demonstrated that PSO-S achieved better performance, reducing the mean flow time, setup time, and stock level.

The remainder of this paper is structured as follows: Section 2 presents the literature related to MTA and PSO; Section 3 presents the problem; Section 3.2 describes the method proposed to solve it; Section 4 describes the simulation implementation details and presents the results; and Section 5 draws the concluding remarks of this paper, including suggestions for future research.

5.2 Background

5.2.1 Make-To-Availability

MTA is a subtype of make-to-stock, which aims to guarantee the availability of final pro ducts to the selected market (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). This definition differs from traditional make-to-stock, in which no firm commitment to availability is given (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). Two key guidelines maintain the tactical aspect of the MTA approach (GOLDRATT, 2009; SCHRAGENHEIM; DETTMER; PATTERSON, 2009):

1) Production needs to focus on maximizing the flow of orders through the warehouse, until they reach the finished product warehouse.
2) Unless there is a good reason to believe that demand has changed, or will change, a simple and direct way to react to any sale is to replenish all the sold stock.

The procedure to operate a production system in the MTA mode comprises of four steps (SCHRAGENHEIM; DETTMER; PATTERSON, 2009):

Step 1: Definition of the initial stock target levels:

In MTA, the quantity of stock in the factory or system must remain fixed for each product. This quantity is called the target level (TL) and consists of the average demand during the replenishment time (RT). The RT it is the time needed to replenish items that have been consumed, weighted by a factor of demand variability during the RT period. To ensure availability with low stock levels, the RT must be as short as possible. This TL is an initial and conservative value and is adjusted dynamically according to the behavior profile over time of the finished good inventory (FGI) available for delivery (SCHRAGENHEIM, 2002; SCHRAGENHEIM; DETTMER; PATTERSON, 2009).

Step 2: Production order generation:

As stated previously, POs are created according to rules based on stock levels. In this case, a production order is generated when the total stock level of certain items (considering both the WIP and the FGI) is below the TL. To avoid the inconvenience of small production orders across the production system, Schragengheim, Dettmer and Patterson (2009) suggested a derived policy that sets each order release priority according to the BS. In this policy, a PO is enabled when the level of (WIP + FGI) is lower than the TL. The set of enabled orders are sorted by the value of BS, $BS = \frac{TL-WIP-FGI}{TL}$.

Step 3: Buffer Management

The role of buffer management (BM) is to allocate appropriate priorities and notify operators when an extraordinary effort is required to move specific orders forward.

The principal protection for immediate delivery is the FGI. The TL buffer can be divided into three equal zones. The status of the buffer (FGI+WIP) is considered to be green, yellow, or red when it is equal to or greater than 66.7%, between 33.3% and 66.7%, or less than 33.3% of the TL, respectively. The green and yellow status form the buffer penetration zone. Figure 5.2 illustrates this concept.

Figure 5.2 - Three zones of the TL

Source: Adapted from Schragenheim (2010)

Another key concept of the MTA is the BS of a PO, defined by Schragenheim (2010) in equation (1). In this equation, the BS is given as a measurement of the deviation between the stock TL and the existing stock, considering both the downstream work In process (DWIP) and the FGI. Each order receives a priority according to the value of BS: for the range 100%-66%, the PO has a high priority; for $BS_{PO} = [66\% - 33\%]$, a medium priority; and if an order has a BS bellow 0.33%, it receives a low priority.

$$BS_{PO} = \frac{TL-DWIP-FGI}{TL} \tag{1}$$

Step 4: Maintaining the correct TL

The frequency and intensity of penetrations in the green and red regions in a period demonstrate the effects of the TL (SCHRAGENHEIM, 2010). The FGI levels trigger changes in the TL; if FGI stays on the green zone for a specified amount of time, the TL decreases; similarly, the TL increases if the FGI stays in the red zone for too long.

5.3 Problem description and dispatching method

As described in Section 4.1, one of the main issues of the MTA method is its difficulty in responding to the dynamics of the system, such as the changes caused by sequence-dependent setup times. The dependent setup time can influence the system flow time and, consequently, increase the stock levels. Thus, this study aims to develop a more suitable algorithm for controlling a manufacturing shop-floor. We assume that the arrival of the orders is not known in advance and the processing times are stochastic. As a testing environment, we adopt three simulation models inspired by the work of Thurer et al. (2017). These models are non-permutational flowshops, with 7 stations. The arrivals are ordered according to an Earlang distribution. Each order is composed of their processing times at each stage. The simulation runs until 30 replications and 10,000 time units have been completed for each order. The goal is to minimize the mean flowtime plus setup time, according to equation (2). We also analyze the stock levels of the final solution. More details regarding the specifics of the simulation model are given in Section 4.

$$FST(S_\pi) = \left(\frac{\sum_{j=1}^{n} C_j}{N}\right) + \sum_{j=1}^{n} ST_{j,j-1} \qquad (2)$$

Where:

- FST: Value of the fitness;
- S_π: A solution (particle) represents a sequence of production orders; each one contains a value of C_j and $ST_{(j,j-1)}$;
- N: Total number of production orders in the queue;
- C_j: Estimation of completion time of production order j;
- $ST_{(j,j-1)}$: Setup time of production order j, dependent on production order $j-1$;

5.4 The Particle Swarm Optimization

PSO is a metaheuristic proposed by Kennedy and Eberhart in 1995 (CHOPARD; TOMASSINI, 2018). PSO emulates the foraging behaviors of birds and fish schooling by moving agents named "particles" through a search space (CHEN et al., 2018; XIA et al., 2019). The position $X_i = [x_{i1}, x_{i2}, \ldots, x_{id}, \ldots, x_{iD}]$, $x_{id} \in [x_{min}, x_{max}]$ of each particle i represents a viable solution to the problem with D dimensions. A particle also contains a dynamically adjusted velocity vector $V_i = [v_{i1}, v_{i2}, \ldots, v_{id}, \ldots, v_{iD}]$, $v_{id} \in [v_{min}, v_{max}]$ to allow exploration of the search space. Each particle also contains a dynamically adjusted velocity vector V to allow exploration of the search space and two acceleration coefficients (c_1 and c_2). Algorithm 1 presents the pseudo-code of the PSO. Mathematically, the particle velocity and position are updated according to the following equations (TIAN; SHI, 2018):

$$v_{id}(t+1) = \omega \cdot v_{id}(t) + c_1 \cdot r_1[p||id(t) - x_{id}(t)] + c_2 \cdot r_2[p_{gd}(t) - x_{id}(t)] \quad (3)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1)$$
(4)

where:

- c_1 and c_2 are acceleration coefficients reflecting the weight of the stochastic acceleration terms, which pull each particle toward the local best ($pbest$) and global best ($gbest$) positions, respectively.
- r_1 and r_2 denote two random numbers, uniformly distributed in the range $(0,1)$.
- ω is the inertia weight used for balancing the global and local search. In general, a large inertia weight facilitates global exploration, whereas a small inertia weight tends to facilitate local exploration.
- The best previous position (the position that yields the best fitness value) of the i_{th} particle is recorded as $pbest$ and denoted by $P_i = [p_{i1}, p_{i2}, \ldots, p_{id}, \ldots, p_{iD}]$, whereas the global best position of the entire swarm achieved so far is recorded as $gbest$ and indicated as $P_g = [p_{g1}, p_{g2}, \ldots, p_{gd}, \ldots, p_{gD}]$.

The pseudo-code of the PSO metaheuristic is presented in Algorithm 1.

Algorithm 1: Pseudo-code for a standard PSO algorithm

```
begin
    Randomly initialize particle swarm;
    while number of iterations or the stopping criterion is not met do
        Evaluate fitness of the particle swarm;
        for n = 1 to number of particles do
            Find pbest;
            Find gbest;
            for d = 1 to number of dimensions of particles do
                Update the velocity of d particle dimension by Eq. 2 ;
                Update the position of d particle by Eq. 3 ;
```

The PSO algorithm has been used for a wide range of applications, such as logistics (SHIMIZU; IKEDA, 2010; SHIMIZU; SAKAGUCHI; MIURA, 2014), scheduling (NOUIRI et al., 2018), among others (LIU et al., 2019; XU et al., 2020).

5.4.1 Proposed dispatching method based on PSO

The objective of PSO-S is to sequence the production orders to be processed by a single machine with dependent setup time, minimizing mean flow time plus setup time, according to Eq. 5. The PSO-S is described in detail in Algorithm 2. The following notation is adopted to algorithm:

- $d_p^{(t+1)}$: percentage deviation from actual position X_i^t to the best particle position P_i, used in the next iteration $t+1$;
- $d_g^{(t+1)}$: percentage deviation from actual position X_i^t to the best global position P_g, used in the next iteration $t+1$;
- $d_{pg}^{(t+1)}$: mean of percentage deviation $d_p^{(t+1)}$ and $d_g^{(t+1)}$;
- $v_i^{(t+1)}$: new particle velocity, used to update the particle position, for the next iteration $t+1$. This value is the number of swaps that occur in the update of the position between the POs contained in the particle;
- D = dimensions of the particle;
- $improved$: Indicates whether $gbest$ has improved in the iteration.

$$FST(S_\pi) = \left(\frac{\sum_{j=1}^{n} C_j}{N}\right) + \sum_{j=1}^{n} ST_{j,j-1} \qquad (5)$$

where

- FST: Value of the fitness;
- S_{π}: A solution (particle) represents a sequence of production orders; each one contains a value of C_j and $ST_{(j,j-1)}$;
- N: Total number of production orders in the queue;
- C_j: Estimation of completion time of production order j;
- $ST_{(j,j-1)}$: Setup time of production order j, dependent on production order $j-1$;

The PSO-S algorithm initializes the swarm (X_i), randomizing initial particles (X_0), which have a sequence of production orders (see lines 6 and 9). From line 9, the algorithm finds the sequence that minimize the mean flowtime plus total setup time, executing various interactions on swarm (X_i). Each interaction passes by all the particles (line 12), applying the following logic:

1) Updating $pbest$ (lines 14 and 15): the best particle position ($pbest$) is updated if its fit ($FST(X_i)$) is less than the fit of the actual best particle position ($FST(X_i)$).
2) Updating $gbest$ (lines 16 and 17): the best global position ($gbest$) is updated if its fit ($FST(X_i)$) is less than the fit of the actual best global position ($FST(gbest)$).
3) Percentage deviation calculation of $d_p^{(t+1)}$ and $d_p^{(t+1)}$ (lines 19, 20, and 21): if the actual best global position ($gbest$) is not updated, the algorithm calculates the percentage deviation from the particle fit ($FST(X_i)$) to the best particle fit ($FST(pbest)$), and from the particle fit ($d_p^{(t+1)}$) to the best global fit ($FST(pbest)$) - $d_p^{(t+1)}$ and $d_p^{(t+1)}$, respectively. Subsequently, the mean of them ($d_p g^{(t+1)}$) is calculated to determine the particle velocity of movimentation.
4) Particle velocity updating (v_i) (lines from 23 to 26): the velocity (v_i) is the number of swaps between the production orders (x_{id}) into particle i, calculated by multiplying the partricle length (D) by $d_p g^{(t+1)}$.
5) Particle position updating (lines from 28 to 33): at the position update, the algorithm passes through v_i positions and swaps production order x_{id} (posterior) by x_{id+1} (previously). The swap occurs if it minimizes the sum of the process time (MP) and dependent setup time (S).

Algorithm 2: Pseudo-code for PSO-S to MTA

begin

  $N_p \leftarrow numberofparticles$;
  $X_0 \leftarrow setofPOs \in thequeueoftheworkstation$;
  $D \leftarrow lengthofX_0$;
  $S \leftarrow dependentsetuptime$;
  $MP \leftarrow processtime$;
  $t \leftarrow N_p$;
  foreach $\underline{i \in N_p}$ **do**
    $X_i \leftarrow randomofX_0$;
  while $\underline{t > 0}$ **do**
    $i \leftarrow N_p$;
    $improved \leftarrow false$;
    while $\underline{i > 0}$ **do**
      $i \rightarrow i - 1$;
      if $\underline{FST(X_i^t) < FST(pbest)ORFST(pbest) = 0}$ **then**
        $pbest = X_i$;
      if $\underline{FST(pbest) < FST(gbest)ORFST(gbest) = 0}$ **then**
        $gbest = pbest$;
        $improved = true$;
      else
        $d_p^{t+1} = \frac{(FST(X_i^t) - FST(pbest))}{FST(pbest)}$;
        $d_g^{t+1} = \frac{(FST(X_i^t) - FST(gbest))}{FST(gbest)}$;
        $d_{pg}^{t+1} = \frac{d_p^{t+1} + d_g^{t+1}}{2}$;
      /* Update the velocity */
      if $\underline{d_{pg}^{t+1} > 0}$ **then**
        $v_i^{t+1} = DXd_{pg}^{t+1}$;
      else
        $v_i^{t+1} = 1$;
      /* Update the position */
      while $\underline{v_i^{t+1} > 0 \text{ and foreach } x_{id} \in X_i}$ **do**
        $x_{id} = MP(x_{id}) + S(x_{id}, x_{id+1})$;
        $x_{id+1} = MP(x_{id+1}) + S(x_{id+1}, x_{id})$;
        if $\underline{x_{id} > x_{id+1}}$ **then**
          Swap x_{id} and x_{id+1};
          $v_i^(t + 1) = v_i^{t+1} - 1$;
    if $\underline{improved = false}$ **then**
      $t \rightarrow t - 1$;

6) Iteration validation (lines 34 ad 35): the iteration is valid denpending on the global best ($gbest$) is improved or not. Thus, the algorithm is executed until it can no longer improve the result.

5.5 Computational Experiments

5.5.1 Simulation

To analyze the performance of the PSO-S, we implemented the MTA described in the previous section in Python 3.5. Based on Thurer et al. (2017), the simulation model was implemented using the SimPy v. 3.0.10 library. Three different methods were used to sequence the POs in production: first-in-first-out (FIFO), PBS, and PSO-S. This model was executed on a single core of a PC i7. The simulated manufacturing environment is based on the work of based on Nguyen et al. (2015) and Thürer and Stevenson (2018), composed of a non-permutational flow shop, which contains 7 stations, where each station is a single machine.

This model was validated by replicating the configuration described by Thurer et al. (2017), and performing the following procedure: three different simulation models were implemented, altering only the position of the single bottleneck station to stations 1, 4, and 7. For each instance, there is one product, and the operation processing times follow a truncated 2-Erlang distribution, with a mean of one time unit after truncation and a maximum of four time units. The maximum processing time was reduced by 20% in the non-bottleneck stations, from 4 time units to 3.2 time units. The inter-arrival time of the POs follows an exponential distribution, with a mean of 1.111 time units, which results in a utilization level of 90% at the bottleneck. The orders were released onto the shop floor immediately upon arrival at the system. The mean flow time was the baseline used to validate the simulation model. For each model 30 replications were executed. Table 5.1 shows that there is only a 5% difference between the results from Thurer et al (2017) and the algorithm proposed in this study.

Table 5.1 - Comparison of the results of mean flow time with the results of Thürer (2017)

The position of the bottleneck station	Results from Thürer et al. (2017)	Results found by the implemented model	Deviation in absolute numbers (1 - 2)	Deviation in percent (1 - 2)
First	12.83	12.37	0.46	3.60%
Center	12.85	12.38	0.47	3.70%
Last	12.96	12.44	0.52	4.00%

Source: Proposed by the author

The original simulation model was adapted by modifying the following elements. It included ten products, and all machines had dependent setup time for all products. Three different models were implemented on the MTA: FIFO, PBS, and PSO-S. The FIFO was adopted as a comparison parameter to demonstrate the difference between sequencing POs by PSO-S and without changing the sequence.

There is a stock buffer for each product in the modified simulation model. The inter-arrival time of the demands that consume the buffers follows an exponential distribution, and the mean for each product is shown in Table 5.2. Each demand consumes one item from the FGI. The processing time of the machines follow the Erlang distribution, and Table 5.3 contains the values of the scale parameter. The shape parameter of the Erlang is equal to 2 for all products and machines. Table 5.4 presents the dependent setup time matrix, which is the same for all machines. The general simulation, PSO-S parameters, and MTA parameters are given in Table 5.5, Table 5.6, and Table 5.7, respectively.

The parameters from Table 5.5 were defined as replications, warmup time, and simulation time, following the findings of Thurer et al (2017). The parameters defined using previous simulations were: number of particles (N_p) and number of iterations (N_t). It can be observed from the results that PSO approach can provide better results than the other analyzed methods.

Table 5.2 - Mean values (expressed in time unit) of inter-arrival time of products demands

Products	1	2	3	4	5	6	7	8	9	10
Mean value	0.5	1	1.5	2	2	2.5	3.5	2.5	3.5	3

Source: Proposed by the author

Table 5.3 - Processing time (expressed in time unit) matrix of each product expressed

Machines	**Products**									
1	1	2	3	4	5	6	7	8	9	10
2	0.2	0.2	0.3	0.5	0.7	0.1	0.7	0.1	0.7	0.7
3	0.3	0.3	0.1	0.7	0.8	0.6	0.9	0.7	0.1	0.9
4	0.4	0.4	0.5	0.6	0.9	0.1	0.2	0.2	0.6	0.8
5	0.5	0.5	0.1	0.8	0.3	0.2	0.3	0.1	0.2	0.3
6	0.6	0.6	0.7	0.2	0.4	0.2	0.4	0.3	0.9	0.6
7	0.7	0.7	0.1	0.3	0.5	0.4	0.5	0.1	0.4	0.1
8	0.8	0.1	0.2	0.4	0.6	0.4	0.6	0.4	0.3	0.5

Source: Proposed by the author

Table 5.4 - The matrix of dependent setup time (expressed in time unit) of each product

Products	1	2	3	4	5	6	7	8	9	10
1	0.0	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.1
2	0.1	0.0	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
3	0.9	0.1	0.0	0.2	0.3	0.4	0.5	0.6	0.7	0.8
4	0.8	0.9	0.1	0.0	0.2	0.3	0.4	0.5	0.6	0.7
5	0.7	0.8	0.9	0.1	0.0	0.2	0.3	0.4	0.5	0.6
6	0.6	0.7	0.8	0.9	0.1	0.0	0.2	0.3	0.4	0.5
7	0.5	0.6	0.7	0.8	0.9	0.1	0.0	0.2	0.3	0.4
8	0.4	0.5	0.6	0.7	0.8	0.9	0.1	0.0	0.2	0.3
9	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.1	0.0	0.2
10	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.1	0.0

Source: Proposed by the author

Table 5.5 - General simulation parameters

Parameter	Value
Replications	30 times
Warmup time	3,000 time unit
Simulation time	10,000 time unit
Order production size	10 units of the products

Source: Proposed by the author

Table 5.6 - PSO-S parameters

Parameter	Value
Number of particles (N_p)	20
Number of iterations (N_t)	20

Source: Proposed by the author

Table 5.7 - MTA parameters

Parameter	Value
One day	10.0 time unit
Initial Replenishment Time (RT)	5.0 days
Initial CCR station	7
Initial Mean of Setup time on CCR	1
Initial Mean of Process time on CCR	1
Initial Target Level	Multiplication of diary demand by RT parameter.

Source: Proposed by the author

The MTA parameters in Table 5.7 were updated every 100 time units. Upon updating, the "Initial CCR Station" received the number of the most-utilized machine in that period. The "Initial Mean of Process Time on CCR" was updated according to the mean process time for each job on the CCR, and the "Initial Mean of Setup Time on CCR" was updated in the same way. The "Initial Replenishment Time (RT)" was updated according to the mean RT for each PO.

The measures used to compare the performance of the dispatching solutions were mean FT, ST, mean WIP, FR, mean TL, and mean UT. The FT was registered on completion of the PO, by subtracting the PO completion time entered in the production line. The ST registration occurred in any machine, for any product. The WIP level was measured when a PO entered or left the production line. The TL was registered when there was a new adjustment to its level by the BM. The FR was registered at the end of each replication. The demands delivered with the FGI stock (DS), were summed during the replication. Ultimately, the total of the DS was divided by the demand total number of items. The UT was the mean of the utilization time of all machines at the end of the replication.

At the end of each replication, the mean of each measure was calculated. The values registered were summed and divided by the number of registrations.

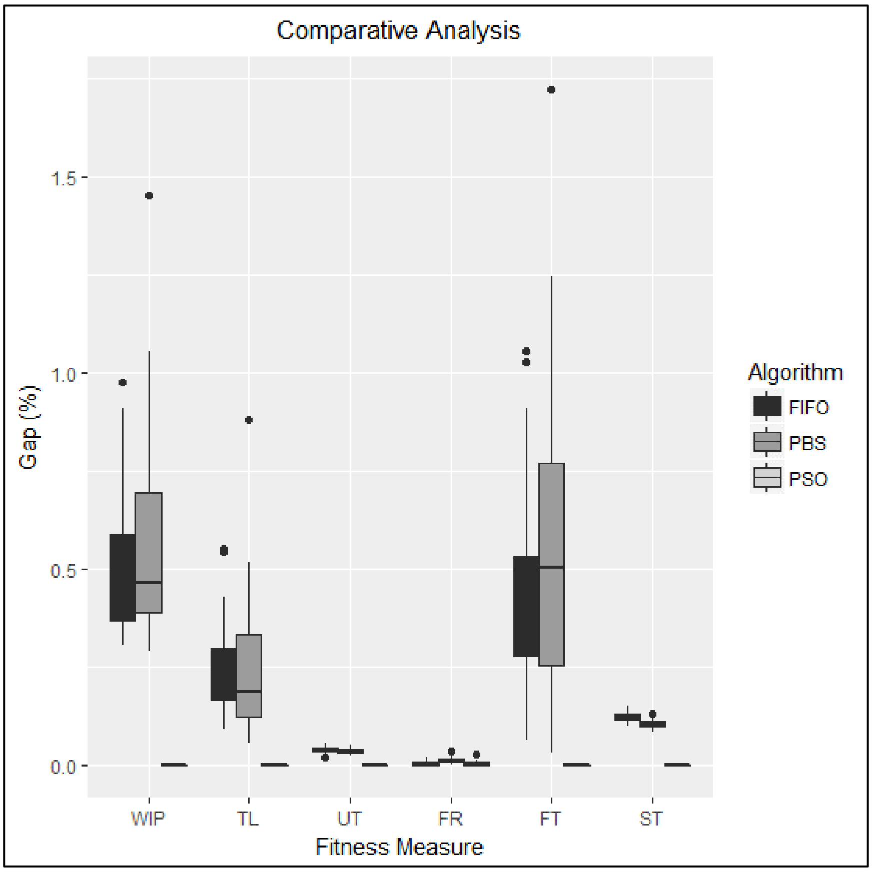
5.5.2 Results

The results are the mean of the 30 replications. To present the general performance of the dispatching methods, the results were normalized according to the percentage deviation ($PD = \frac{(xdm-min)}{min}$), where xdm is the result of the dispatching method (FIFO, or PBS, or PSO-S) and min is the minimum value found. These results are shown in Figure 5.3.

Figure 5.3 makes clear that the PSO-S achieved the best results for most fitness measurements; the PBS only achieved the best results in the FR but the difference was not significant. It is noteworthy that the PSO-S reduced the FT and ST considerably. The PSO-S minimized the mean flow time and setup time proving its effectiveness. Due to the reduction of the FT and ST, the WIP drastically reduced. The FIFO and PBS methods needed a higher TL to ensure product availability, which also increased the WIP. In the UT measure, the PSO-S had a smaller gain.

The results of the simulations of all analyzed methods are found in Table 5.8. The first three columns presents the absolute values, and the last present the deviation of the FIFO and PBS methods the results of the PSO algorithm. According to the data presented, there was a reduction in FT of 31% and 35%, compared to FIFO and PBS, respectively. Furthermore, the PSO presented lower standard deviation values. Despite the ST, there was a reduction of 11% and 9% compared to FIFO and PBS, respectively. The standard deviation was the same for all measures.

Figure 5.3 - Percentage deviation of all dispatching methods



Source: Proposed by the author

Table 5.8 - Results found by all the methods

		FIFO (1)	PBS (2)	PSO-S (3)	(3) vs. (1)	(3) vs. (2)
Mean flow time (FT)	**Avg**	64.70	69.31	44.89	-31%	-35%
	Std dev	13.79	17.04	3.76	-73%	-78%
Setup time (ST)	**Avg**	0.45	0.44	0.40	-11%	-9%
	Std dev	0.003	0.003	0.003	0%	0%
Work-in-Process (WIP)	**Avg**	227.74	236.30	151.82	-33%	-36%
	Std dev	28.42	40.56	4.03	-86%	-90%
Target Level (TL)	**Avg**	44.70	44.40	35.96	-20%	-19%
	Std dev	4.37	6.51	0.90	-79%	-86%
Utilization (UT)	**Avg**	83.47	83.17	80.34	-4%	-3%
	Std dev	0.55	0.47	0.37	-33%	-21%
Fill rate (FR)	**Avg**	0.99	0.99	0.99	0%	0%
	Std dev	0.008	0.003	0.003	-62%	0%

Source: Proposed by the author

Thus, the adoption of these methods provides the lowest range of PO flow time. With this result, it can be acknowledged that for the instances analyzed, the proposed PSO-S allows the practitioner to better estimate the completion time of the POs. The PSO-S reduced the WIP by 33% and 36% with FIFO and PBS, respectively. The low standard deviation of PSO-S indicates that it can maintain WIP stability, with a 20% and 19% reduction, compared to FIFO and PBS, respectively. The decrease in work-in-progress (WIP) can be attributed to the reduction in setup time, resulting in a decrease in the average flow time. This intriguing phenomenon can be elucidated by Little's law, which establishes that, in a steady-state system, the average long-term throughput is equivalent to the ratio of the average total work in process (WIP) to the average flow time (FT) (LITTLE, 1961). Furthermore, the PSO-S caused minor variation, in the TL, which can be verified by the low standard deviation.

For UT and FR there was a small difference between the dispatching methods, thus demonstrating that the PSO-S does not need more capacity to produce with a lower flow time and achieve a similar FR. Although the PSO-S has been designed to minimize FT and ST, other important indicators for the MTA, such as WIP and TL, have also been improved.

5.6 Conclusions

The objective of this study was to develop a dispatch method, denominated PSO-S, to enhance the performance of the S-DBR/MTA in environments with sequence-dependent setup time. Inspired by the Particle Swarm Optimization metaheuristic, PSO-S was evaluated using a simulation of a flow shop line managed by the MTA, considering sequence-dependent setup times on all machines for all products. The performance of PSO-S was compared to the PBS rule and FIFO rule, assessing indicators such as mean flow time, mean setup time, mean work-in-process, mean target level, mean machines utilization, and mean fill rate (service level).

The results demonstrate significant improvements achieved by PSO-S compared to the other dispatch rules. PSO-S reduced the average setup time by a minimum of 9%, resulting in a decrease of at least 33% in the average work-in-process and 31% in the average flow time. Additionally, it led to a reduction of at least 19% in the target level and 3% in machine utilization. No significant difference was observed in fill rate performance. The effectiveness of PSO-S lies in its capability to minimize setup time, thereby reducing work-in-process inventory and target levels, which can ultimately contribute to cost reduction.

Further experimentation with PSO-S is encouraged to explore its potential across different shop-floor structures. Although this study validated the proposed method based on existing literature, investigating diverse control systems like Workload Control, Kanban, and POLCA could provide valuable insights and expand upon the proposed method.

The promising results obtained through the PSO algorithm warrant further exploration and application of metaheuristics in production control scenarios. Additionally, future research can delve into alternative control systems beyond the S-DBR/MTA framework to further enhance production control strategies.

6 INVENTORY REPLENISHMENT IN THE SIMPLIFIED DRUM-BUFFER-ROPE SYSTEM: A SOLUTION PROPOSAL BASED ON HEURISTICS AND MIXED INTEGER PROGRAMMING

6.1 Introduction

Supply chains are undergoing significant changes, such as the search for more resilience, growth of e-commerce, omnichannel sales, diversification of distribution channels, market pressure to reduce delivery times, and new types of vehicles to transport goods. These changes have made the distribution of goods more complex, and increasingly challenge the management distribution of goods. The challenges include managing multi-sites, distribution chain coordinate and synchronization, the control inventory of multiple distribution centers, regional warehouses, and retailers.

To overcome these challenges, the companies use systems such as Vendor-Managed Inventory (VMI), Collaborative Planning, Forecasting and Replenishment (CPFR), Just-in-time (JIT), and DTA. The Vendor-Managed Inventory (VMI) is a system in which the supplier takes responsibility for managing inventory levels at the customer's location (GOVINDAN, 2013). Another system is Collaborative Planning, Forecasting, and Replenishment (CPFR), which involves collaboration between trading partners in the supply chain to develop a shared understanding of demand and inventory requirements (HOLLMANN; SCAVARDA; THOMÉ, 2015). CPFR includes joint planning, demand forecasting, and inventory replenishment. Just-in-time (JIT) is a policy of ordering stock in needed quantities only when needed (GOLHAR; STAMM, 1991). On the other hand, DTA seeks to improve the availability of items at all points of consumption (end users) based on the constant replenishment of consumed stocks from strategically positioned stock buffers in the supply chain (COX; SCHLEIER, 2010).

The DTA approach addresses these challenges by considering the impact of supply and demand to determine appropriate stock levels throughout the supply chain while being mindful of cash and space limitations (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). The objective is to ensure the high availability of items at all consumption points while constantly renewing consumed stocks from strategically placed stock buffers (COX; SCHLEIER, 2010). This approach optimizes distribution and replenishment, considering the dynamic nature of market demand (SULLIVAN; REID; CARTIER, 2007).

> *"(a) pull distribution method that involves setting stock buffer sizes and then monitoring and replenishing inventory within a supply chain based on the actual consumption of the end user, rather than a forecast. Each link in the supply chain holds the maximum expected demand within the average replenishment time, factored by the level of unreliability in replenishment time. Each link generally receives what was shipped or sold, though this amount is adjusted up or down when buffer management detects changes in the demand pattern."*

Like other systems, the Distribution and Transportation Analysis (DTA) also encounter challenges in effectively balancing cost, profit, and service level during the planning and distribution of goods. Extensive literature delves into the discussion of logistics tradeoffs, with one of the most significant and widely debated being the tradeoff between transportation costs and inventory holding costs (CARDÓS; GARCÍA-SABATER, 2006; CHOUDHARY; SHANKAR, 2013; MOSCA; VIDYARTHI; SATIR, 2019; QIU et al., 2022; SARKAR et al., 2019; TURKENSTEEN; VAN DEN HEUVEL, 2023). Increasing the frequency of replenishment leads to a rise in transportation costs but a reduction in inventory holding costs. Conversely, decreasing the replenishment frequency lowers transportation costs but necessitates higher inventory levels at upstream nodes of the network, subsequently increasing inventory holding costs. The goal is to strike a balance where costs remain sufficiently low to ensure the desired profit and service level.

DTA is not yet able to handle this trade-off due to the way it makes decisions about the inventory replenishment. As DTA does not use demand forecasting techniques, the pressure on the stock replenishment function is high, as it is responsible for making all decisions about what, when, and how much to restock (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). The basic rule of the DTA is whenever a single unit of inventory is consumed at a distribution network node, an action to replenish that item should be initiated immediately.

Currently, DTA cannot effectively address this trade-off due to its decision-making approach concerning inventory replenishment. Since DTA does not utilize demand forecasting techniques, there is increased pressure on the stock replenishment function, which assumes responsibility for all decisions regarding what, when, and how much to restock (SCHRAGENHEIM; DETTMER; PATTERSON, 2009). The fundamental principle of DTA is to trigger an immediate replenishment action whenever a single unit of inventory is consumed at a distribution network node.

Inventory replenishment encompasses a range of activities, from retrieving products from the warehouse to delivering them to the final customer or point of sale. Efficient planning of resupply is crucial to optimize resource utilization and strike a balance between cost, profit, and service level, as previously discussed. However, a systematic literature review (Chapter 3) indicates that DTA currently lacks a planning solution. Schragenheim, Dettmer, and Patterson (2009) argue that frequent replenishments help maintain low inventories and full buffers but do not provide guidance on making replenishment decisions that consider associated costs. Consequently, DTA is unable to effectively address the tradeoff between transportation costs and inventory holding costs.

Deciding when to replenish stock buffers is a difficult task that requires much more than a policy; tools are needed. For companies that intend to implement DTA, it is vital to have a tool that balances availability and profitability. The lack of this tool motivated us and led us to the following research question:

RQ4: How to replenish stock buffers in a distribution network managed by DTA, protecting the availability of products and the business's profit?

The DTA problem can be seen as a specific instance of the Inventory Routing Problem (IRP), which combines inventory and transportation management challenges. To model the DTA problem, we have adapted the basic IRP model presented by Coelho et al. (2014). As a solution approach, we propose a mixed integer programming (MIP) model and a computational heuristic based on the studies conducted by Fachini and Armentano (2020) and Koç et al. (2015). We aim to make contributions to the literature on S-DBR, logistics, and supply chain management. Additionally, we hope that our work can provide alternative distribution solutions for companies in need.

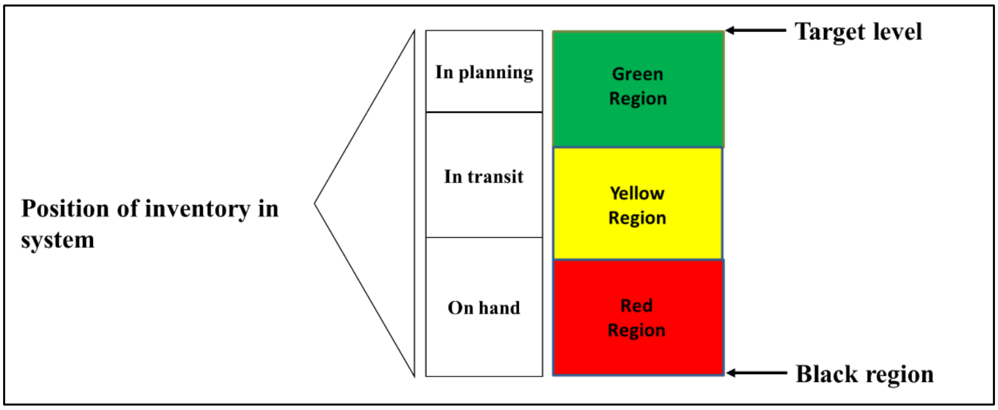
The paper is structured as follows: Section 6.2 presents the inventory replenishment process in Distribution-To-Availability; Section 6.3 describes the problem and presents the MIP solution; the hybrid evolutionary algorithm solution is presented in Section 6.4; Section 6.5 describes the setup of the computational experiment; Section 6.6 presents the outcomes and discussions; and Section 6.7 concludes the research. Because of the size limit, part of the paper is presented in Appendix A.

6.2 Inventory replenishment in Distribution-To-Availability

DTA is a pull distribution method that operates by establishing stock buffer sizes and monitoring and replenishing inventory within a supply chain based on actual consumption by end-users, rather than relying on forecasts. Stock buffers serve to safeguard product availability against variations in replenishment and demand timelines. Figure 6.1 provides an illustration of a stock buffer, where the stock buffer size denotes the maximum inventory quantity held at a specific location within the supply chain. The target level represents the highest stock keeping unit (SKU) level permitted at each site, resulting in potential variations in SKU stock buffer sizes across different locations. The buffer status (BS), indicated by color, represents the urgency of replenishing the stock and is calculated as the number of missing units from the buffer divided by the target level (COX; SCHLEIER, 2010):

- Green (buffer status is less than 33%): The inventory at the consumption point is high, providing more than enough protection for now.
- Yellow (buffer status is between 33 and 67%): The inventory at the consumption point is adequate. There is a need to order more units from the upstream supply chains.
- Red (buffer status between 67 and 100): The inventory at the consumption point is at risk of stocking out.
- Black (buffer status is 100%): The stock has run out at the consumption point; every hour that passes at this stage means (potential) lost sales opportunities.

Figure 6.1 - Stock buffer



Source: Adapted from Schragenheim, Dettmer and Patterson (2009)

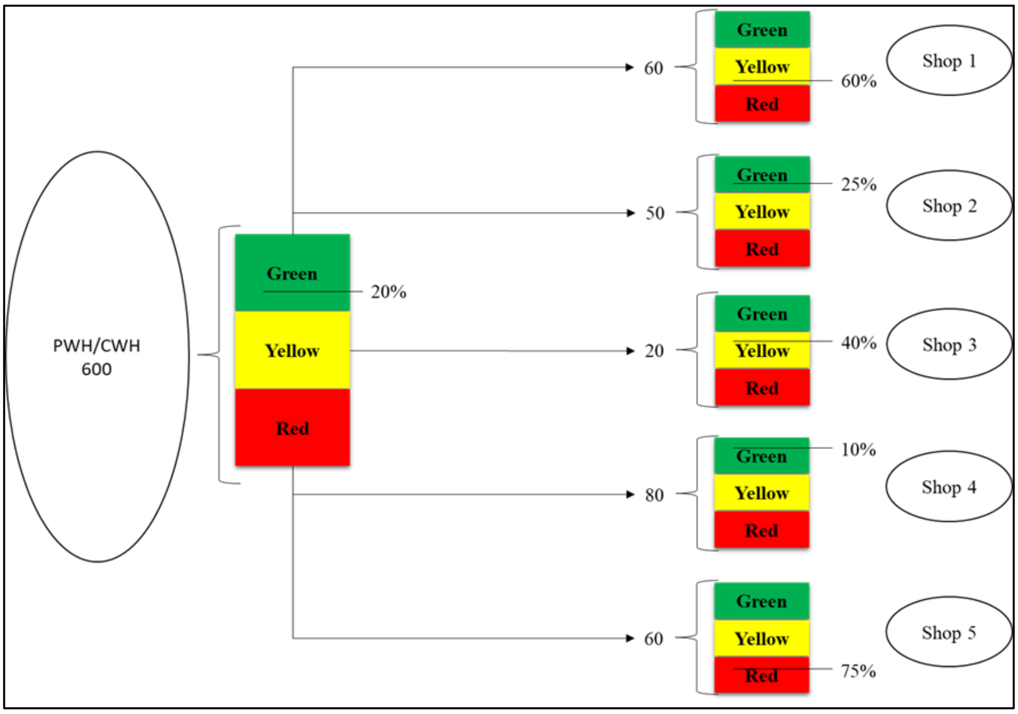
A higher buffer status indicates a greater urgency for replenishment, with the black buffer status being assigned the highest priority and the green buffer status the lowest. When the buffer status is black, the ready rate measure determines the fraction of time following replenishment that the buffer experiences stockouts.

DTA employs Dynamic Buffer Management (DBM) as a mechanism for managing buffers in a dynamic environment. In this approach, DTA does not rely on any forecasting models but instead dynamically measures the actual usage of stocks and readjusts the target level accordingly. For further insights into DBM, please refer to Ikeziri et al. (2021).

The buffer can exist in three different system positions: planning at a supplier, in transit, or at the point of sale (POS) as on-hand inventory. The sum of all the items across these positions should be equal to the target level. For instance, if a stock buffer has a target level of 100 units for a specific product, and there are currently 40 units on hand and 20 units in transit, then there will be 40 units in planning. The number of units that are missing in the buffer can be calculated as the target level minus the inventory in transit, which in this case would be 40 units (100 - 60). Consequently, the buffer status of the stock would be 40 percent ((100 - 60)/100). The buffer status is determined based on the items in hand and those in transit.

As stated previously, inventories are independently controlled. Figure 6.2 illustrates an example of a buffer of a single product type in one distribution center and five points of sale. Note that each buffer has a different status.

Figure 6.2 – Buffers in distribution network



Source: Adapted from Schragenheim, Dettmer and Patterson (2009)

In Figure 6.2, the management of buffers is depicted as separate entities. For example, if the buffer at the plant warehouse/central warehouse (PWH/CWH) has a capacity of 600 units and currently has 20% buffer penetration (with 480 units out of the total 600), it is represented by green to indicate its priority level. Similarly, in Shop 1, the buffer for the same item has a buffer status of 60%, but only 24 units are present, resulting in a yellow priority color.

The supplier must continuously monitor the downstream buffers and determine when to transfer items. This decision is complex and critical due to various factors such as costs, profitability, and transportation considerations. A distribution network typically comprises numerous geographically dispersed buffers that can be transferred in multiple ways.

Literature suggests that the DTA stock replenishment policy is insufficient when it comes to handling complex and large-scale problems that demand more advanced tools. Moreover, the focus solely revolves around quickly replenishing stock and ensuring high availability, with little consideration for the profitability of the overall operation.

6.3 Formal problem description

The DTA problem is a specific instance of the Inventory Routing Problem (IRP), which involves the integration of inventory management and transportation (CAMPBELL et al., 1998). The objective of the Inventory Routing Problem (IRP) is to minimize the total cost of inventory distribution while meeting the demand of each customer (Coelho et al., 2014). The inventory replenishment plan is subject to a set of constraints outlined by Coelho et al. (2014), including the following:

1. The inventory level for each customer must not exceed its maximum capacity.
2. Inventory levels must not go below zero.
3. The supplier's vehicles are limited to one route per period, starting and ending at the supplier.
4. Vehicle capacities must not be exceeded.

For the DTA problem, constraint 1 is analogous to the Target Level, while constraint 2 remains applicable. Since DTA operates in continuous time, constraint 3 needs to consider time as a continuous variable rather than discretized intervals. Constraint 4, on the other hand, remains entirely relevant to the DTA.

In the basic Inventory Routing Problem (IRP), the solution to the problem involves determining the following (COELHO; CORDEAU; LAPORTE, 2014):

1. Which customers to serve in each period.
2. Which vehicles belonging to the supplier to utilize.
3. The quantity of goods to deliver to each customer that is visited.

4. The specific delivery routes to take for each vehicle.

On DTA, the time is continuous and not discrete, as in the basic IRP. However, it is not planning more than one period at once. The basic IRP is NP-hard because it subsumes the classical VRP (COELHO; CORDEAU; LAPORTE, 2014).

According to Coelho et al. (2014), the basic IRP is defined on a graph $G = (V, A)$, where $V = \{0, \dots, n\}$ is the vertex set and $A = \{(i, j): i, j \in V, i \neq j\}$ is the arc set. Vertex 0 represents the supplier, and the vertices of $V' = V \backslash \{0\}$ represent customers.

Both the supplier and customers face unit inventory-holding costs (h_i per period). Additionally, each customer has a maximum inventory capacity (C_i). The planning horizon spans periods, and in each period $t \in T = \{1, \dots, p\}$, the supplier r^t makes available a quantity of product. We assume that the supplier has sufficient inventory to meet all demand throughout the planning horizon and that inventories cannot be negative. The variables I_0^t and I_i^t represent the inventory levels at the end of period t for the supplier and customer i, respectively. At the beginning of the planning horizon, the decision-maker possesses knowledge of the current inventory levels of the supplier and all customers (I_0^0 and I_i^0 for $i \in V'$) and is fully aware of the demand d_i^t from each customer i for each period t. A set of vehicles $K = \{1, \dots, K\}$ with capacities Q_k is available. Each vehicle can execute one route per period to deliver products from the supplier to a subset of customers. A routing cost c_{ij} is associated with the arc $(i, j) \in A$.

For this study, an adaptation and expansion of the basic model of the IRP was made to represent the particularities of the DTA. The set of period time $t \in T = \{1, \dots, p\}$ was eliminated, and the variables' dependents on this set. The IRP for DTA is defined as a complete graph $G(N, E)$, where the node set $N = \{0, 1, \dots, n\}$ consists of the depot (node 0) and the set of n clients/points of sale $C = N \backslash \{0\}$, while the set $E = \{(i, j): i, j \in N, i \neq j\}$ represents the arcs between nodes. A heterogeneous fleet of vehicles is positioned at the depot with different vehicle capacities to serve (QIN et al., 2021). The set $K = \{1, \dots, l\}$ represents the l distinct types of vehicles available at the depot, each with capacity Q_k, an associated fixed cost f_k and a cost c_k for the unit distance traveled. For each arc $(i, j) \in E$, a symmetric travel distance $d_{ij} = d_{ji}$, is provided. $R^k = (r_1, r_2, \dots, r_{|R|})$ is the route for a $k \in K$ with $r_1 = r_{|R|} = 0$, where $R = (R^1, R^2, \dots, R^l)$ is the set of routes. Each route R^k starts and ends at the depot, and the remaining components define the sequence of points of sale visited by vehicle k. The cost of route R^k is the sum of the travel costs $\sum_{r_i=1}^{|R|-1} \sum_{p \in P} d_{r_i r_{(i+1)}} \cdot c^k$, referred to as the route variable cost, and the fixed cost f_k of associated vehicle $k \in K$, as expressed in Equation 1.

$$R^k_{total\ cost} = \left[\sum_{r_i=1}^{|R|-1} \sum_{p \in P} d_{r_i r_{(i+1)}} \cdot c^k\right] + f_k \tag{1}$$

The set $P = \{1, \ldots, 10\}$ represents the 10 distinct types of products available at each node $i \in N$, with Q_p being the capacity required to transport one item of the product $p \in P$. Each pair of type of product $p \in P$ and node $i \in N$, has an associated stock buffer B_p^i, a target level tl_p^i, a holding cost ct_p^i, a price of sale pr_p^i, an inventory level s_p^i (on hand plus in transit), a buffer status bs_p^i, and a ready rate r_p^i. For all products $p \in P$ the price of sale pr_p^i in depot is 0. The holding cost of the points of sale is the value paid by the depot for the sale of products. Points of sale do not involve buying products. Stock buffer B_p^i is composed of tl_p^i, s_p^i, r_p^i and bs_p^i. The buffer status of a distinct B_p^i is given by Equation 2.

$$bs_p^i = \frac{tl_p^i - s_p^i}{tl_p^i} \tag{2}$$

The first decision in the replenishment process concerns the number of items D_p^i to replenish in each stock buffer B_p^i. This is a typical product-mix selection problem. Following Schragenheim, Dettmer, and Patterson's (2009) definition, the buffer status bs_p^i determines the selection of stock buffers for replenishment. The higher the buffer status bs_p^i of the stock buffer B_p^i, the higher the priority. However, this study aims to propose a solution for the policy of replenishment stock buffers that guarantees the availability of products for demand and business profitability. Profit is the difference between the revenue from the replenished items and the costs of holding and transport. The profit from one item of stock buffer B_p^i is calculated as the difference between the sale price and the inventory holding costs, represented by Equation 3:

$$\varphi_p^i = pr_p^i - ct_p^0 - ct_p^i \tag{3}$$

where ct_p^0 is the inventory holding cost at the depot and ct_p^i is the inventory holding cost of the product $p \in P$ at the point of sale $i \in C$.

To balance availability and profitability, we propose applying the buffer status bs_p^i and the inverse of the ready rate r_p^i as weight on item profit φ_p^i, as represented by Equation 4.

$$\gamma_p^i = \varphi_p^i \cdot \left(1 + \left(1 - r_p^i\right)\right) \cdot bs_p^i \tag{4}$$

Thus, stock buffers B_p^i with lower inventory levels (buffer status in black or red region) and high profitability may receive higher priority. The bigger the stock buffer bs_p^i and the inverse of ready rate $\left(1 + \left(1 - r_p^i\right)\right)$ for a stock buffer B_p^i, the higher the item profit weighted by buffer status γ_p^i.

The number of items to be replenished is determined by the Z_1 MIP model, which is composed of Equations (5) – (9).

$$\max \sum_{i \in C} \sum_{p \in P} D_p^i \cdot \gamma_p^i \tag{5}$$

Subject to

$$\sum_{i \in C} \sum_{p \in P} D_p^i \cdot Q_p \leq \sum_{k \in K} Q_k \tag{6}$$

$$D_p^i \leq tl_p^i - s_p^i \quad , \forall\, i \in C;\ b \in U \tag{7}$$

$$\sum_{i \in C} D_p^i \leq s_p^0 \quad , \forall\, b \in U \tag{8}$$

$$D_p^i \in \mathbb{Z}_+ \ \ \forall\, p \in P;\ i \in C \tag{9}$$

The objective function (5) expresses the maximization of the total item profit weighted by the buffer status and inverse of the ready rate. Constraint (6) imposes the total vehicle capacity $\sum_{k \in K} Q_k$ as an upper bound to the necessary capacity $\sum_{i \in C} \sum_{p \in P} D_p^i \cdot Q_p$ to transport the items. Constraint (7) imposes the number of missing items $(tl_p^i - s_p^i)$ on the stock buffer B_p^i as an upper bound to D_p^i for all $i \in C$ and $p \in P$. Constraint (8) prevents the total number of items for replenishing the product $p \in P$ from exceeding the number of items s_p^0 available in the depot.

The number of items D_p^i returned by model Z_1 is a parameter used for taking the second decision, which is fleet routing for delivery.

The cost of each route R^k is limited to a predefined percentage t of the total profit of the route to control transport costs and avoid routes without financial viability, as represented in Equation 10:

$$t \cdot \left(\sum_{r_i = 2}^{|R|-1} \sum_{p \in P} \varphi_p^i \cdot D_{r_i p}^k \right) \leq R_{total\ cost}^k \tag{10}$$

where parameter t is the maximum percentage of the $R^k_{total\ cost}$ over the total profit, and $D^k_{r_i p}$ denotes the number of items delivered by vehicle $k \in K$ to a given stock buffer B^i_p. We assume that more than one vehicle $k \in K$ can visit a point of sale $i \in C$.

The fleet routing is given by MIP model Z2, composed of Equations (11)– (24), adapted from Fachini and Armentano (2020).

$$\max \sum_{k \in \mathcal{K}} \sum_{i \in C} \sum_{p \in P} D^k_{ip} \cdot \gamma^i_p - \sum_{k \in \mathcal{K}} \sum_{ij \in \mathcal{E}} x^k_{ij}.c_k.d_{ij} - \sum_{k \in \mathcal{K}} (f_k \cdot y^k_0) \tag{11}$$

$Subject\ to$

$$\sum_{i \in C} \sum_{p \in P} D^k_{ip} \cdot Q_p \leq Q_k \cdot y^k_0;\ \forall\, k \in \mathcal{K} \tag{12}$$

$$y^k_0 - \sum_{i \in C} \sum_{p \in P} D^k_{ip} \leq 0\ ,\ \forall\, k \in \mathcal{K} \tag{13}$$

$$M \cdot y^k_i - \sum_{p \in P} D^k_{ip} \leq 0\ ,\ \forall\, k \in \mathcal{K};\ i \in C \tag{14}$$

$$\sum_{k \in \mathcal{K}} D^k_{ip} \leq D_{ip},\ \forall\, i \in C;\ p \in P \tag{15}$$

$$\sum_{i \in \mathcal{N}} x^k_{ij} - y^k_j = 0,\quad \forall\, k \in \mathcal{K};\ j \in \mathcal{N};\ i \neq j \tag{16}$$

$$\sum_{j \in \mathcal{N}} x^k_{ij} - y^k_i = 0,\quad \forall\, k \in \mathcal{K};\ i \in \mathcal{N};\ i \neq j \tag{17}$$

$$\sum_{j \in \mathcal{N}} x^k_{ji} - \sum_{j \in \mathcal{N}} x^k_{ij} = 0,\quad \forall\, k \in \mathcal{K};\ i \in \mathcal{N};\ i \neq j \tag{18}$$

$$\sum_{i \in S} \sum_{j \in S} x^k_{ij} \leq |S| - 1,\quad \forall\, k \in \mathcal{K}; S \subset C;\ i \neq j \tag{19}$$

$$\left(t \cdot \sum_{i \in C} \sum_{p \in P} \varphi^i_p \cdot D^k_{ib}\right) - \sum_{k \in \mathcal{K}} \sum_{ij \in \mathcal{E}} x^k_{ij}.c_k.d_{ij} - \sum_{k \in \mathcal{K}} (f_k \cdot y^k_0) \geq 0\ \ \forall\, k \in \mathcal{K} \tag{20}$$

$$D^k_{ip} \in \mathbb{Z}_+\ \ \forall\ \ p \in P;\ \ i \in C;\ k \in K \tag{21}$$

$$x^k_{ij}\ \in \{0,1\}\ \forall\ \ ij \in \mathcal{E}; k \in K; i \neq j \tag{22}$$

$$y^k_i \in\ \{0,1\}\ \forall\ \ i \in N;\ k \in K \tag{23}$$

$$0\ \leq t \leq\ 1 \tag{24}$$

The objective function (11) maximizes the weighted net profit, which is the difference between the total item profit weighted by buffer status $\sum_{k \in \mathcal{K}} \sum_{i \in C} \sum_{p \in P} D^k_{ip} \cdot \gamma^i_p$, total variable transport cost $\sum_{k \in \mathcal{K}} \sum_{ij \in \mathcal{E}} x^k_{ij}.c_k.d_{ij}$ and total fixed transport cost $\sum_{k \in \mathcal{K}} (f_k \cdot y^k_0)$.

Constraint (12) avoids that capacity of a vehicle $k \in \mathcal{K}$ is exceed by the total capacity $\sum_{i \in C} \sum_{p \in P} D_{ip}^{k} \cdot Q_p$ necessary to transport the items. If a vehicle $k \in \mathcal{K}$ is designated to a delivery $\sum_{i \in C} \sum_{p \in P} D_{ip}^{k}$, constraint (13) activated it at depot y_0^k.

For each vehicle $k \in \mathcal{K}$ and each point of sale $i \in C$, constraint (14) designates a vehicle $k \in \mathcal{K}$ for a given point of sale $i \in C$ if there are items to deliver. For each $i \in C$ and each $p \in P$, constraint (15) limits the quantity of a product items to delivery to a point of sale according to determining the model Z1. Constraint (16) and (17) creates an arc x_{ij}^k for each $k \in \mathcal{K}$ and $j \in \mathcal{N}$, when $y_j^k = 1$ or $y_i^k = 1$. Constraint (18) ensures that a node $i \in \mathcal{N}$ receive a vehicle $k \in \mathcal{K}$ and send to another node $j \in \mathcal{N}$, for each $k \in \mathcal{K}$, $i \in \mathcal{N}$ and $i \neq j$. Whenever a route R^k designated to a vehicle $k \in \mathcal{K}$ yields a subtour, constraint (19) is a feasibility cut for Z_2, for each $k \in \mathcal{K}$, each subtour $S \subset C$ and $i \neq j$. This constraint was implemented in Cplex and Python using lazy constraint callback (see the code in Appendix A, Figure 1). The validity of cut (19) is easily seen because it corresponds to a subtour elimination constraint (DANTZIG; FULKERSON; JOHNSON, 1954). Similarly, whenever route R^k designated to a vehicle $k \in \mathcal{K}$ yields an impracticable cost, constraint (20) is a feasibility cut for Z_2. Constraint (20) corresponds to Equation 3, which limits the total transport cost of a route for each vehicle $k \in \mathcal{K}$. Equations (21) – (24) indicate the domain of the model variables.

The complete declaration of the mathematical notation of the problem description and the following sections are in Appendix A, section 1.

6.4 A Hybrid Evolutionary Algorithm solution

Koç et al. (2015) introduced a hybrid evolutionary algorithm (HEA) designed to solving vehicle routing problems with time windows involving heterogeneous fleets. The primary objective of their approach is to minimize both the fixed vehicle costs and the distribution costs, which can be defined in terms of en-route time or distance. The HEA demonstrates success by integrating various metaheuristics and introducing advanced and efficient procedures specifically tailored to address the complexities associated with heterogeneous fleets. To incorporate the particularities of the DTA we adapted the HEA.

6.4.1 The general framework of the HEA

The hybrid evolutionary algorithm (HEA) combines two state-of-the-art metaheuristic concepts and a clustering algorithm that has proven highly successful for a variety of vehicle routing problem: adaptive large neighborhood search (ALNS) (BREKKÅ et al., 2022; FONTAINE, 2022; MALLADI et al., 2022), population-based search (CHAGAS et al., 2022; KOÇ et al., 2015; VIDAL, 2022; WANG et al., 2022) and K-means (FATEMI-ANARAKI et al., 2022; WANG et al., 2021, 2022).

The ALNS heuristic aims to derive best new solutions by iteratively removing and reinserting ordered nodes in a solution by destroying and repairing operators (BREKKÅ et al., 2022). The K-means algorithm is a typical clustering method for grouping data into k clusters and minimizing inter-cluster similarities (WANG et al., 2021).

Algorithm 1 presents the general structure of HEAs. The K-means algorithm and ALNS operators are combined to generate the initial population (Line 1). Two parents are selected (Line 4) through a binary tournament, following which the crossover operation (Line 5) generates a new offspring, OFF. The advanced SPLIT algorithm is applied to the offspring OFF, which segments the giant tour G (the sequence of all points of sale to be visited) by choosing the vehicle type for each route. If offspring OFF is infeasible, the education procedure (Line 8) uses ALNS operators to make offspring OFF feasible. The insertion and removal operators used by ALNS are explained in Section 2 of Appendix A. The adaptive weight adjustment procedure (AWAP) updates the probabilities associated with the operators and penalty parameters used in the education procedure (Line 9). AWAP is explained in Section 3 of Appendix A. Elite solutions are subjected to an intensification procedure based on the ALNS algorithm (Line 10) to improve their quality.

If, at any iteration, the population size n_a reaches $n_p + n_o$, then a survivor selection mechanism is applied (Line 11). The population size, shown by n_a, changes during the algorithm as new offspring are added, but is limited by $n_p + n_o$, where n_p is a constant denoting the size of the population initialized at the beginning of the algorithm and n_o is a constant indicating the maximum allowable number of offspring that can be inserted into the population.

The mutation is applied to a randomly selected individual from the population with probability p_m at each iteration of the algorithm. If there are no improvements in the best-known solution for several consecutive iterations it_r, the entire population undergoes regeneration (Line 12). The population POP is ranked by weighted net profit (see Equation 11) (Line 13), and the first individual is set to the best solution BESTSOL. HEA terminated when the elapsed time reached the time limit (Line 15).

Algorithm 1 - The general framework of the HEA

Initialization: initialize the population POP (Algorithm 1 in Appendix A).
$\text{BESTSOL} = \emptyset$;
While elapsed time < limit time **do:**
    Parent selection: Select parent solutions P1 and P2 from POP by binary tournament.
    Crossover: Apply two-point crossover from P1 and P2 to giant tour G and fleet F to generate offspring OFF;
    Split tour: Apply algorithm 3 (Appendix A) to OFF to generate an individual IND_i and add it to POP.
    Education: Educate IND_i applying ALNS algorithm 4 (Appendix A);
    AWAP: Update probabilities of the ALNS operators.
    Intensification: Intensify elite solutions applying ALNS algorithm 5 (Appendix A).
    Survivor selection: If the population size n_a reaches $n_p + n_o$, then select survivors.
    Diversification: Diversify the population with MUTATION or REGENERATION procedures.
    Best solution: Ranking population POP by weighted net profit (Equation 11) and setting the best individual IND_1 to the best solution BESTSOL.
End while.
Return best solution BESTSOL

6.4.2 Initialization

Two steps divide the procedure to generate the initial population, as shown in Algorithm 1, explained in Section 4 of Appendix A. Step 1 obtains the product mix and creates routes for clusters of points of sale using K-means algorithm 2, as explained in Section 4 of Appendix A. The outcome is a viable solution that has not yet been optimized. In Step 2, the ALNS algorithm generates the population from the initial solution given in Step 1.

6.4.3 Split Tour

The split tour procedure aims to create delivery routes from the product mix, giant tours, and vehicle fleets. The split tour logic is represented by Algorithm 3 (Appendix A) which is like the K-means algorithm (Algorithm 2 in Appendix A) and allows more than one vehicle to visit a point of sale. Until the capacity limit is reached, or there are no more products to allocate, each iteration allocates the products of the points of sale on the giant tour to a fleet vehicle. For more details, see Section 5 in Appendix A.

6.4.4 Education

The education procedure is applied to each offspring to make it feasible. The ALNS algorithm educates HEA by applying destroy and repair operators and removing some nodes in each iteration. According to equation (20), a feasible solution is one that is financially viable. For more details, refer to Algorithm Education in Section 6 of Appendix A.

6.4.5 Parent Selection

To yield offspring, the HEA selects two parents through a binary tournament. The selection process randomly chooses two individuals from the population and maintains one with the best weighted net profit (Equation 11).

6.4.6 Crossover

The OX crossover operator is well-suited for cyclic permutations, and the giant tour encoding, and vehicle fleet sequence allow recycling crossovers designed for the vehicle routing problem. Initially, two positions, i and j are randomly selected in the first parent, P_1. Subsequently, substrings $(i, \ldots, j)$ are copied to the first offspring O_1 at the same positions. The second parent P_2 is swept cyclically from position $j + 1$ onwards to fill the empty positions in O_1. This procedure is like giant tour and fleet vehicle sequences. Two offspring are obtained from the original OX version. However, in the HEA group, we randomly selected one offspring.

6.4.7 Intensification

We introduced a two-phase aggressive intensification procedure to improve the quality of elite individuals. This procedure intensifies the search for promising regions in the solution space. Section 7 of Appendix A details the pseudocode for this method.

6.4.8 Survivor Selection

Avoiding premature convergence is a key challenge in population-based metaheuristics. Ensuring the diversity of the population, in other words, searching for a different location in the solution space during the algorithm in the hope of being closer to the best known or optimal solutions, constitutes a significant trade-off between solutions in a population. This procedure eliminates individual clones from the population and individuals with the worst weighted net profits (Equation 11). Only n_p individuals remain, so that the offspring can be generated from the best solutions.

6.4.9 Diversification

Efficient management of feasible solutions plays a significant role in population diversity. After education, the mutation procedure improved HEA performance.

Over iterations, individuals tend to become more similar, making it challenging to avoid premature convergence. To overcome this difficulty, we introduce a new scheme to increase population diversity. The diversification stage includes two procedures: regeneration and mutation, as explained in Section 8 of Appendix A.

6.5 Computational Experiments

This section presents the setup and results of the computational experiments performed to assess the performance of the MIP and HEA. The HEA implementation was in Python 3.8 and that of the MIP was in Cplex 20.1.0 on a computer with 8 gigabytes of RAM and a CPU Intel Core i5 2.40 GHz processor. First, we describe the parameters and instances used in the solutions. The results are presented below.

6.5.1 Data Sets and Experimental Settings

The localization datasets for the point-of-sale were adapted from the R1 instance of Solomon (1987), while the vehicle fleet dataset was adjusted based on the Liu and Shen (1999) instances. The R1 dataset consists of random Cartesian coordinates, where the depot is located at the center and customers are scattered throughout. The specific coordinates can be found in the Supplementary Material files. Liu and Shen (1999) introduced five types of vehicles with varying costs and capacities. We have incorporated the variable costs, which represent the costs per unit distance traveled. In Appendix A, Table 2, the vehicle parameters are summarized. Additionally, we have included ten types of products with different transport capacities and unit holding costs at the depot, as indicated in Table 3 of Appendix A.

The experiment ran into a 3600 s limit time. For instances which do not return a solution into this limit, it was extended to 28,800 s. Cplex was parameterized as follows: workmen = 4000, tree memory = 10,000, strategy files = 2. For the HEA, we initially used the parameters suggested by Koç (2015) which were adjusted in experiments of test to improve solution performance. All instances used the following parameter values: $it_t = 500,\ it_r = 50,\ it_w = 100,\ n_e = 5,\ p_m \in [0.4,0.6]$, $[b_l^i, b_u^i] = [0.3,0.8]$, $[b_l^e, b_u^e] = [0.1,0.16]$, $[b_l^m, b_u^m] = [0.1,0.16]$, $\sigma_1 = 3$, $\sigma_2 = 2$, $\sigma_3 = 0$, $r_p = 0.1$, $t = 0.05$. The values of parameters n_p and n_o equal the number of instance of points of sale.

6.5.2 Instances

The instances used in this study are characterized by the number of points of sale, either 25 or 50, and the inventory level available at the depot. Additionally, the total capacity of the fleet is classified as low, balanced, or high. The low level allows for replenishment of up to 50% of the required items to complete stock buffers, the balanced level enables replenishment of up to 100% of the items, and the high-level permits replenishment of up to 200% of the items. These different levels were chosen to evaluate solutions under varying resource constraints. The supplementary files provide detailed product data for the points of sale, including target levels, inventory levels, ready rates, holding costs, and prices. For instances with 75 and 100 points of sale, the MIP (mixed integer programming) approach did not yield a viable solution within the timeout limit of 28,800 seconds. As a result, the MIP approach was ruled out for these instances. Appendix A, Table 4, summarizes these instances. The fleet available at the depot for transportation is heterogeneous and consisted of five types of vehicles, namely A, B, C, D, and E. The composition of the fleet for each instance is provided in Table 5 of Appendix A.

Table 6.1 shows the distribution of the buffer regions by the number of points of sale. The column "gross profit" is the profit (price minus holding costs) if all items to achieve the target level are replenished and sold.

Table 6.1 - The initial buffers regions and gross profit

	Buffer regions				
#Points of sale	**Black**	**Green**	**Yellow**	**Red**	**Gross profit**
25	77	58	56	59	2,638,684.36
50	171	108	113	108	5,313,452.67
75	245	179	165	161	7,764,848.21
100	320	250	226	204	10,106,613.92

6.6 Outcomes and Discussions

6.6.1 Analysis of Computational Performance

The computational performance of the MIP and HEA is evaluated using CPU time as the measure. MIP outperforms HEA in instances 1-10, achieving faster computational times. However, for instances with 50 points of sale (11-18), the CPU time required by MIP exceeds the limit of 3600 seconds, whereas HEA successfully finds a solution within the time limit. Figure 6.3 illustrates the disparities between the solutions obtained by MIP and HEA. Additionally, Figure 6.4 makes this difference even more evident with the % deviation from the HEA to the MIP, according to equation 7.

$$PD = \frac{(HEA - MIP)}{MIP} \times 100 \tag{7}$$

Figure 6.3 - CPU time (s)

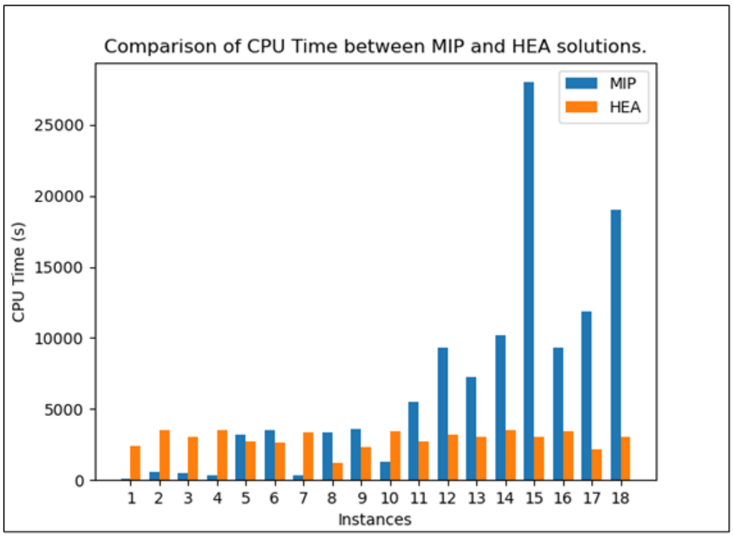
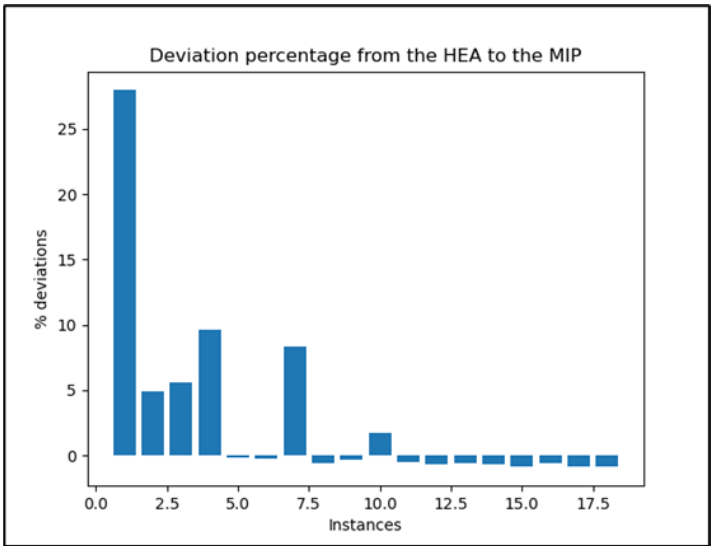


Figure 6.4 – % Deviation



Instances 1 to 10 show a positive percentage deviation, indicating that MIP outperforms HEA in these cases. Specifically, instances 1, 2, 3, 4, 7, and 10 demonstrate an advantage for MIP, with the HEA requiring 7% to 27% more time to find the optimal solution. However, instances 11 to 18 reveal a different scenario, with HEA surpassing MIP. In these instances, HEA exhibits significantly better performance, completing the optimization process with 51% to 89% less time than MIP. These results indicate that for larger instances, HEA outperforms MIP in terms of computational efficiency.

6.6.2 Comparative Analysis

The performance evaluation of MIP and HEA was conducted based on three key aspects: the selection of buffer regions for replenishment, transport outcomes, and financial considerations. Table 6.2 presents the percentage of selected buffers categorized by the regions outlined in Table 6.1. Notably, the solutions yielded similar results as both utilized the Z_1 model.

Across the buffer regions, the black region had the highest average percentage of selected buffers for replenishment, followed by the red, yellow, and green regions. In instances where product availability was low (1, 2, 3, 10, 11, 12), all selected buffers belonged to the most critical regions (black and red). Surprisingly, the choice of buffer was not influenced by the fleet capacity.

Table 6.2 – Buffers selected percentage

# Instances	POS	Black	Red	Yellow	Green
1	25	90,79%	37,70%	0,00%	0,00%
2	25	90,79%	37,70%	0,00%	0,00%
3	25	90,79%	37,70%	0,00%	0,00%
4	25	86,84%	83,61%	43,86%	12,50%
5	25	100,00%	100,00%	100,00%	100,00%
6	25	100,00%	100,00%	100,00%	98,21%
7	25	86,84%	83,61%	43,86%	12,50%
8	25	100,00%	100,00%	100,00%	100,00%
9	25	100,00%	100,00%	100,00%	98,21%
10	50	89,35%	26,13%	0,00%	0,00%
11	50	89,35%	26,13%	0,00%	0,00%
12	50	89,35%	26,13%	0,00%	0,00%
13	50	84,62%	69,37%	49,15%	13,73%
14	50	100,00%	100,00%	100,00%	99,02%
15	50	100,00%	100,00%	100,00%	99,02%
16	50	84,62%	72,07%	49,15%	13,73%
17	50	100,00%	100,00%	100,00%	99,02%
18	50	100,00%	100,00%	100,00%	99,02%
	Mean	**93,52%**	**72,23%**	**54,78%**	**46,94%**

In instances with balanced and high transport capacity (5, 6, 14, and 15) and balanced product availability (4-6 and 13-15), both MIP and HEA solutions successfully served 100% of the instances. For instances with low transport capacity (4 and 13), buffers were selected from all four regions, representing approximately 85% for the black region, 69% for the red region, 49% for the yellow region, and 14% for the green region. Similar patterns were observed for instances with high product availability (7-9 and 16-18). Table 6.3 provides a visual representation of the percentage of buffer regions receiving replenishment. These percentages were calculated based on the quantities of items provided by the Z_1 model. Notably, no buffers remained in the black region, while only a few were present in the red or yellow regions. Most of the buffers were in the green region.

Instances with low product availability and 25 points of sale (1-3) exhibited the highest percentage in the red region and the lowest percentage in the yellow and green regions. Similar trends were observed for instances with 50 points of sale (10-12). Interestingly, the transport capacity did not significantly impact these instances, as the results remained consistent across all three capacity levels.

Table 6.3 – Buffer region position with the replenish

# Instances	POS	Black	Red	Yellow	Green
1	25	0,00%	5,43%	1,09%	93,48%
2	25	0,00%	5,43%	1,09%	93,48%
3	25	0,00%	5,43%	1,09%	93,48%
4	25	0,00%	0,00%	0,67%	99,33%
5	25	0,00%	0,00%	0,00%	100,00%
6	25	0,00%	0,00%	0,00%	100,00%
7	25	0,00%	0,00%	0,67%	99,33%
8	25	0,00%	0,00%	0,00%	100,00%
9	25	0,00%	0,00%	0,00%	100,00%
10	50	0,00%	2,22%	1,67%	96,11%
11	50	0,00%	2,22%	1,67%	96,11%
12	50	0,00%	2,22%	1,67%	96,11%
13	50	0,00%	0,00%	0,68%	99,32%
14	50	0,00%	0,00%	0,00%	100,00%
15	50	0,00%	0,00%	0,00%	100,00%
16	50	0,00%	0,34%	0,68%	98,98%
17	50	0,00%	0,00%	0,00%	100,00%
18	50	0,00%	0,00%	0,00%	100,00%

In instances with balanced availability (4-6 and 13-15), the HEA solution maintained less than 1% of the buffers in the red region of instance 4, where the transport capacity level was low. There was no buffer in the red region for the other instances of this group in either solution. For instances with low transport capacity (4 and 13), less than 1% of the buffers remained in the yellow region. In instances where transport capacity was balanced or high (5, 6, 14, and 15), 100% of the buffers were in the green region, and more than 99% in instances 4 and 13. The instances with high availability (7- 9 and 16-18) had similar results, except for 16, in which less than 1% of the buffers remained in the red region. Table 6 in Appendix A lists the absolute routing results.

Table 6 of Appendix A shows that MIP and HEA achieved very different routing performances. In Table 6.4, the percentage deviation in the number of vehicles and route distance indicates the deviation of the HEA from the MIP. In five instances (4, 5, 7, 13, and 16), MIP and HEA used the same number of vehicles: four with a low transport capacity, and one with a balanced level. In another five instances (3, 6, 11, 12, and 18), MIP used fewer vehicles than HEA, four of which (3, 6, 12, and 18) had a high level of transport capacity and one (11) had a balanced level. In eight instances (1, 2, 8, 9, 10, 14, 15, and 17), HEA used fewer vehicles, with two (1 and 10) having low transport capacity, four (2, 8, 14, and 17) having balanced capacity, and only two (9 and 15) having high capacity.

Table 6.4 – Routing results

	% Deviation		Fleet utilization		Distance mean per vehicle	
# Instances	**Number of vehicles**	**Route distance**	**MIP**	**HEA**	**MIP**	**HEA**
1	-30,00%	0,06%	100%	70%	56,00	80,04
2	-8,33%	37,93%	60%	55%	50,32	75,71
3	56,25%	65,71%	42%	66%	47,37	50,24
4	0,00%	21,70%	100%	100%	56,42	68,66
5	0,00%	0,00%	90%	90%	57,11	57,11
6	20,83%	25,10%	63%	76%	51,14	52,94
7	0,00%	19,57%	100%	100%	56,42	67,46
8	-11,11%	1,49%	90%	80%	56,87	64,94
9	-4,17%	10,52%	63%	61%	50,20	57,89
10	-5,56%	-45,39%	95%	89%	133,57	77,24
11	13,04%	-48,47%	61%	68%	124,23	56,63
12	29,03%	-44,50%	42%	55%	127,66	54,91
13	0,00%	-81,69%	100%	100%	410,53	75,16
14	-2,94%	-51,60%	89%	87%	131,41	65,53
15	-23,91%	-64,75%	61%	47%	116,70	54,06
16	0,00%	-67,98%	100%	100%	226,85	72,63
17	-2,86%	-59,24%	88%	85%	158,14	66,35
18	10,87%	-50,21%	61%	68%	121,64	54,63

Regarding the route distance, HEA resulted in greater distances for eight of them, and one with a distance equal to the MIP in instances with 25 POS. In instances with 50 POS, the MIP presented a greater distance for all of them. The same was true for the average distance traveled per vehicle.

Table 7 in Appendix A presents the financial results, and Table 5 shows the percentage deviation based on the results. In Table 6.5, the weighted net profit % deviation is less than 1%, with a slight advantage for MIP. The gross profit of the MIP has an advantage of 6.84% and 3.53% in two instances with 25 points of sale (3-6), and a more significant advantage in four instances with 50 points of sale: of 30.15% (11), 17.74% (12), 46.72% (15), and 16.69% (18). The performance of MIP on total cost transport was superior for instances with 25 points of sale. In instances with 50 points of sale, the HEA solution was superior. The difference in net profit was approximately 1% in 12 instances. In six instances, MIP exhibited a superior performance between 3% and 47%.

Table 6.5 – Financial results deviation

# Instances	Weighted Net Profit (%Dev)	Gross Profit (%Dev)	Transport Total Cost (%Dev)	Product Total Cost (%Dev)	Net Profit (%Dev)
1	-0,0006	-0,0002	0,0988	-0,0002	-0,0008
2	-0,0011	-0,0002	0,2611	-0,0002	-0,0014
3	-0,0088	-0,0684	1,3534	-0,0822	-0,0742
4	-0,0013	-0,0002	0,3253	-0,0002	-0,0013
5	-0,0007	0	0	0	0
6	-0,0052	-0,0353	0,7067	-0,037	-0,0385
7	-0,0013	-0,0002	0,3217	-0,0002	-0,0013
8	-0,0007	-0,0002	0,0969	-0,0002	-0,0007
9	-0,0009	-0,0003	0,172	-0,0002	-0,001
10	0,0019	-0,0002	-0,3153	-0,0002	0,0027
11	-0,0007	-0,3015	-0,2294	-0,2982	-0,3021
12	-0,0045	-0,1774	-0,1203	-0,1832	-0,1779
13	0,0097	-0,0002	-0,7045	-0,0002	0,011
14	0,0031	-0,0002	-0,3633	-0,0002	0,0032
15	0,0015	-0,4672	-0,651	-0,4713	-0,4656
16	0,0043	-0,0002	-0,5183	-0,0002	0,0049
17	0,0039	-0,0002	-0,4205	-0,0002	0,0041
18	-0,0005	-0,1669	-0,2811	-0,1649	-0,1659

6.7 Conclusions

This study aims to propose a solution to the S-DBR/DTA capable of planning the replenishment of stocks to guarantee the products' availability and the profitability of the business. We propose two solutions to fill this gap: one based on MIP model and the other based on a hybrid evolutionary algorithm (HEA).

The first stage of the MIP solution (model Z_1) determines the number of items to be replenished, respecting the fleet transport capacity to maximize the total item profit weighted by the buffer status and inverse of the ready rate. The second stage (model Z_2) determines financially viable routes that maximize the weighted net profit - the difference between the total item profit weighted by buffer status, total variable transport cost, and total fixed transport cost.

The first stage of the HEA determines the number of items to be replenished using model Z_1. In the second stage, to determine the financial viability of fleet routing, an initial solution is established using the K-means clustering algorithm, and, posteriorly, the solution is optimized by a hybrid algorithm using ALNS and Genetic Algorithm.

The experiments showed that model Z_1 selected the black region with a higher average percentage of buffers for replenishment, followed by the red, yellow, and green regions. In instances with low product availability and low, balanced, or high fleet capacity (1, 2, 3 and 10, 11, 12), only buffers from the most critical regions (black and red) were selected. Therefore, the model prioritizes the buffers that most need replacement, regardless of fleet capacity. The Z_1 will meet approximately 100% of buffers of the instances where product availability is balanced and high if the transport capacity is also balanced and high (5, 6, 8, 9, 14, 15, 17, and 18).

For instances with a low level of transport capacity and balanced or high levels of products availability (4, 7, 13, and 16), the model indicated the replenishment of buffers in the four regions, being that in the green region, the percentage was the lowest and in red the biggest. In this case, the model balanced the need for replacement, failing to replace the buffers with lower profitability, even in the red region.

More than 90% of the selected buffers will raise their level to the green region if they receive the number of items suggested by the model. Therefore, it tries to reach the target level by determining the amounts to be replenished.

In the second stage, MIP and HEA use the same number of vehicles, for instances with low transport capacity (4, 5, 7, 13, and 16). For instances with high transport capacity (3, 6, 11, 12, and 18), MIP used fewer vehicles than HEA. In instances (1, 2, 8, 9, 10, 14, 15, and 17), HEA used fewer vehicles, with two instances (1 and 10) having low transport capacity, four (2, 8, 14, and 17) having balanced capacity, and only two (9 and 15) having high capacity. The vehicle utilization performance of both was the same, 78%. The HEA outperformed the MIP in the average distance traveled per vehicle. In MIP, vehicles travel on average 112 km, and in HEA, the average distance traveled is 64 km, 43% less. Therefore, HEA can result in shorter travel times and more trips as vehicles return to the warehouse sooner.

Regarding the financial results, the difference in weighted net profit % deviation was less than 1%, with a slight advantage for MIP. It can be assumed that the objective function's result is the same for both. For other indicators, HEA achieved 7% less than MIP in gross profit, 1.5% less in total transportation cost, and 7% less in total product cost and net profit. HEA was unable to allocate all products to vehicles, so it had a drop in net income.

Overall, the MIP achieved better computational performance in instances with 25 POS and HEA in instances with 50 POS. MIP also outperforms HEA performance in terms of routing and financial results, but not computational performance, which can provide the HEA some advantage in practice.

This study contributes to the literature of the S-DBR/DTA and practitioners by proposing methods for distribution planning and alternative solutions for goods distribution. Other studies could repeat the experiments using larger instances or develop improvements and adaptations for specific problems. We also suggest comparative studies with consolidated stock replenishment policies.

7 FINAL CONSIDERATIONS

This research was motivated by a few studies and deficiencies in the operating mode of the Simplified Drum-Buffer-Rope. Furthermore, empirical assumptions and principles extracted from the DBR and the Theory of Constraints without validation by scientific methods supported the emergence of the S-DBR. The main objective of the research was to propose improvements to the MTA and DTA methods of S-DBR. For the MTA, solutions were proposed for prioritizing production orders in environments with and without dependent setup time. For the DTA, solutions were proposed for planning the replacement of stock buffers.

To answer the first research question, a computational experiment was conducted, simulating a flow shop production environment, to evaluate the impact of replacing and combining the prioritization by buffer status rule with previously known rules. Table 3.4 in chapter 3 shows that the results indicate a performance improvement when combining the prioritization by buffer status with the STP and SRPT rules, denominated PSP-SPT, and PSP-SRPT. Replacing the prioritization by buffer status rule with the SPT and SRPT rules also yielded promising results, although the service level was lower than the PSP-SPT and PSP-SRPT combinations. However, the average level of inventory in the system was lower. In the prioritization by buffer status, the downstream work in process was removed from its calculation, generating the PSP1 rule, which provoked exciting results in combination with the SPT rule (PSP1-SPT). In practice, the advantage of this adaptation is that we don't need to collect information from production orders in advance.

Regarding the second research question, we can say that for flow shop environments, the SRPT rule proved to be the most efficient in terms of inventory units per service level, which means it needs less inventory to guarantee product availability. If the objective is to increase the level of service, and if a higher inventory cost is accepted, then the PSP1-SPT rule is the best option. The prioritization by buffer status rule only obtained the third worst result regarding the rate of inventory units per percentage of service level. An important finding is that prioritizing production orders by looking only at buffer status can lead to undesired results. Other factors are also essential, such as the moment the order arrives on the production line, the processing time, and the remaining time to be completed. Furthermore, the use of information from the production orders in advance did not improve availability, as seen in the results of rule PSP1.

To answer the third research question, a dispatch method was proposed, which was based on the Particle Swarm Optimization metaheuristic, to minimize the average flow time and total setup time, called PSO for Sequence (PSO-S) which was presented in chapter 4. Results showed that PSO-S can overcome prioritization by buffer status in flow shop environments with time-dependent setup.

Although both have achieved the same level of service, the PSO-S requires less inventory because it reduces the average throughput time. Furthermore, PSO-S reduced the standard deviation in all performance measures showing that it can keep the production system more stable and predictable. Once again, there is evidence that prioritizing production orders by looking only at buffer status can lead to undesired results. In this case, it is essential to adjust the prioritization of orders according to the characteristics of the environment.

In response to the fourth research question, a solution was developed for planning the stock buffers replenishment located in the goods distribution network, according to chapter 5. The concern was to propose a solution that sought to ensure the high availability of products and profitability of the business when planning the replenishment of point-of-sale stock buffers from a distribution center. The solutions proposed were a MIP model and a hybrid evolutionary algorithm (HEA). MIP outperformed HEA in computational performance on instances with 25 points of sale. In instances with 50 points of sale, the MIP performance dropped considerably, while the HEA performance was maintained.

Regarding availability, the two solutions had very similar results, as well as financial results. The most significant difference is in the routing, where the HEA surpassed the MIP in the indicator of average distance traveled by a vehicle. That is, HEA is better than MIP for creating delivery routes. When looking more closely at the results, it was possible to see that HEA presents better results in instances with 50 points of sale. Both MIP and HEA can be adopted to manage the replenishment of buffers in the distribution network. The main factor in deciding which solution to adopt is the network's number of nodes and the vehicle fleet's size, which impacts computational performance. The results generated were promising. Therefore, we believe this presents a necessity for further research that suggests other improvements to S-DBR.

Fulfilling the research objective contributes to the advancement of S-DBR and expands the scientific literature on it. For companies and practitioners, the research points to solutions for practical problems that can help improve bottom-line results while maintaining or increasing service levels. There is a limitation in the literature review in chapter 2, which only covered studies that proposed changes to the DBR and S-DBR systems. Exploring the literature of all articles related to these systems may be more interesting from a scientific point of view and may increase the stimulus for further research.

Regarding the study evaluating dispatch rules for the MTA in chapter 3, we cannot generalize the results to other manufacturing arrangements than flow shop. Therefore, further studies are needed that address, for example, arrangements of the job shop type, cellular manufacturing, and assembly points. In the study that points out a solution to the MTA problem with dependent setup times, in chapter 4, the limitation refers to comparing the PSO-S only with the buffer status prioritization rule and the FIFO rule. Other rules can be applied to validate the results that indicate PSO-S is a good solution. In addition, other production environments can be studied, e.g., when there are tasks with time-dependent setup, time-independent setup, and no setup. A possible shortcoming of the PSO-S is the lack of consideration of the time that a task is waiting in the queue or inside the system, which can generate unavailability of products since production orders can stay for a long time stopped at a workstation.

A suggestion for future studies related to MTA is to investigate its behavior or propose solutions to common problems in production environments, such as dynamic scaling of lots of production and transfer, and resource constraints, namely raw material, storage space, time, people, and cost control. Another suggestion is to investigate environments with a hybrid strategy, MTA and MTO, in which the challenge is to complete the orders on the promised date and ensure product availability.

Two problems pointed out to the MTA by the literature also can motivate an investigation: the existence of multiple Capacity-Constrained Resources (CCRs) and wandering bottlenecks. The S-DBR literature points both out as challenging for the MTA. Concerning the DTA, further research is needed that proposes improvements and shows that it can bring good results to companies through its mechanisms that seek to guarantee high availability.

To evaluate the DTA solutions in chapter 5, instances with only 25 and 50 points of sale were adopted. However, solutions need to be tested with more significant instances, a greater number of products, different types of transport modes, and a more significant number of distribution centers. Other solutions can be proposed, not only for planning the replenishment of stock buffers, but also for managing the dynamic buffer. Finally, the integration between MTA and DTA can also be the subject of study.

REFERENCES

ALI, K. BEN; TELMOUDI, A. J.; GATTOUFI, S. Adopted rescheduling strategy for solving the dynamic job shop using GA based Local Search. **Proceedings of International Conference on Advanced Systems and Emergent Technologies, IC_ASET 2019**, p. 68–73, 2019.

ALJANABI, A. R. A.; GHAFOUR, K. M. Supply chain management and market responsiveness: a simulation study. **Journal of Business and Industrial Marketing**, v. 36, n. 1, p. 150–163, 2021.

AL-TAHAT, M. D.; DALALAH, D.; BARGHASH, M. A. Dynamic programming model for multi-stage single-product Kanban-controlled serial production line. **Journal of Intelligent Manufacturing**, v. 23, n. 1, p. 37–48, 2012.

BAKER, K. R.; TRIETSCH, D. **Principles of Sequencing and Scheduling**. Hoboken, NJ, USA: John Wiley and Sons, 2009.

BAKER, K. R.; TRIETSCH, D. **Principles of Sequencing and Scheduling**. Hoboken, NJ, USA: John Wiley & Sons, Inc., 2018.

BARCO, C. F.; FILHO, M. G. The effect of transfer lot size on manufacturing lead time: A stochastic analysis. **European Journal of Industrial Engineering**, v. 15, n. 6, p. 825–851, 2021.

BECCHETTI, L.; LEONARDI, S. Non-clairvoyant scheduling to minimize the average flow time on single and parallel machines. **Conference Proceedings of the Annual ACM Symposium on Theory of Computing**, p. 94–103, 2001.

BEKTUR, G.; SARAÇ, T. A mathematical model and heuristic algorithms for an unrelated parallel machine scheduling problem with sequence-dependent setup times, machine eligibility restrictions and a common server. **Computers & Operations Research**, v. 103, p. 46–63, 2019.

BLACKSTONE, J. H.; COX, J. F. Designing unbalanced lines - understanding protective capacity and protective inventory. **Production Planning & Control**, v. 13, n. 4, p. 416–423, 2002.

BONNEY, M. Reflections on production planning and control (PPC). **Gestão & Produção**, v. 7, n. 3, p. 181–207, 2000.

BREKKÅ, I. et al. The Fish Feed Production Routing Problem. **Computers & Operations Research**, v. 144, p. 105806, 2022.

BUENO, A.; GODINHO FILHO, M.; FRANK, A. G. Smart production planning and control in the Industry 4.0 context: A systematic literature review. **Computers and Industrial Engineering**, v. 149, p. 106774, 2020.

CAMPBELL, A. et al. The Inventory Routing Problem. In: **Fleet Management and Logistics**. Boston, MA: Springer US, 1998. p. 95–113.

CARDÓS, M.; GARCÍA-SABATER, J. P. Designing a consumer products retail chain inventory replenishment policy with the consideration of transportation costs. **International Journal of Production Economics**, v. 104, n. 2, p. 525–535, 2006.

CASTRO, R. F.; GODINHO-FILHO, M.; TAVARES-NETO, R. F. Dispatching method based on particle swarm optimization for make-to-availability. **Journal of Intelligent Manufacturing**, v. 1, p. 1–10, 2020.

CHAGAS, J. B. C. et al. The double traveling salesman problem with partial last-in-first-out loading constraints. **International Transactions in Operational Research**, v. 29, n. 4, p. 2346–2373, 2022.

CHANG, Y. C.; HUANG, W. T. A modification of simplified drum-buffer-rope for re-entrant flow shop scheduling. **Information Technology Journal**, v. 10, n. 1, p. 40–50, 2011.

CHANG, Y.-C.; HUANG, W.-T. An enhanced model for SDBR in a random reentrant flow shop environment. **International Journal of Production Research**, v. 52, n. 6, p. 1808–1826, 2013.

CHEN, Y. C.; LEE, C. E. A bottleneck-based group scheduling procedure for job-shop cells. **Journal of the Chinese Institute of Industrial Engineers**, v. 18, n. 5, p. 1–12, 2001.

CHEN, Y. et al. Particle swarm optimizer with crossover operation. **Engineering Applications of Artificial Intelligence**, v. 70, p. 159–169, 2018.

CHOPARD, B.; TOMASSINI, M. Particle Swarm Optimization. In: **Natural Computing Series**. [s.l.] IEEE, 2018. v. 4p. 97–102.

CHOUDHARY, D.; SHANKAR, R. Joint decision of procurement lot-size, supplier selection, and carrier selection. **Journal of Purchasing and Supply Management**, v. 19, n. 1, p. 16–26, 2013.

COELHO, L. C.; CORDEAU, J.-F.; LAPORTE, Gilbert. Thirty years of inventory routing. **Transportation Science**, v. 48, n. 1, p. 1-19, 2014.

COX, J. F.; SCHLEIER, J. Theory of Constraints Handbook. 2010.

DANIEL, V.; GUIDE, R. Scheduling with priority dispatching rules and drum-buffer-rope in a recoverable manufacturing system. **International Journal of Production Economics**, v. 53, n. 1, p. 101–116, 1997.

DANTZIG, G.; FULKERSON, R.; JOHNSON, S. Solution of a Large-Scale Traveling-Salesman Problem. **Journal of the Operations Research Society of America**, v. 2, n. 4, p. 393–410, 1954.

DARLINGTON, J. et al. Design and implementation of a Drum-Buffer-Rope pull-system. **Production Planning & Control**, v. 26, n. 6, p. 489–504, 2015.

DARZI RAMANDI, M.; KHAKZAR BAFRUEI, M. Effects of government's policy on supply chain coordination with a periodic review inventory system to reduce greenhouse gas emissions. **Computers and Industrial Engineering**, v. 148, p. 106756, 2020.

DISNEY, S. M.; PONTE, B.; WANG, X. Exploring the nonlinear dynamics of the lost-sales order-up-to policy. **International Journal of Production Research**, v. 59, n. 19, p. 5809–5830, 2021.

DREXL, A.; KIMMS, A. Lot sizing and scheduling — Survey and extensions. **European Journal of Operational Research**, v. 99, n. 2, p. 221–235, 1997.

EL-BOURI, A.; NAIRY, S. An investigation of cooperative dispatching for minimising mean flowtime in a finite-buffer-capacity dynamic flowshop. **International Journal of Production Research**, v. 49, n. 6, p. 1785–1800, 2011.

FACHINI, R. F.; ARMENTANO, V. A. Logic-based Benders decomposition for the heterogeneous fixed fleet vehicle routing problem with time windows. **Computers & Industrial Engineering**, v. 148, p. 106641, 2020.

FALLAH, M. et al. Production planning under variable buffer size. **Kuwait Journal of Science and Engineering**, v. 38, n. 1 B, p. 211–226, 2011.

FATEMI-ANARAKI, S. et al. A hybrid of K-means and genetic algorithm to solve a bi-objective green delivery and pick-up problem. **Journal of Industrial and Production Engineering**, v. 39, n. 2, p. 146–157, 2022.

FONTAINE, P. The vehicle routing problem with load-dependent travel times for cargo bicycles. **European Journal of Operational Research**, v. 300, n. 3, p. 1005–1016, 2022.

FRAMINAN, J. M.; FERNANDEZ-VIAGAS, V.; PEREZ-GONZALEZ, P. Using real-time information to reschedule jobs in a flowshop with variable processing times. **Computers and Industrial Engineering**, v. 129, p. 113–125, 2019.

GAURY, E. G. A.; PIERREVAL, H.; KLEIJNEN, J. P. C. An evolutionary approach to select a pull system among Kanban, Conwip and Hybrid. **Journal of Intelligent Manufacturing**, v. 11, n. 2, p. 157–167, 2000.

GEORGIADIS, P.; POLITOU, A. Dynamic Drum-Buffer-Rope approach for production planning and control in capacitated flow-shop manufacturing systems. **Computers and Industrial Engineering**, v. 65, n. 4, p. 689–703, 2013.

GHORBANI, M. et al. Developing a logical model for cellular manufacturing systems by theory of constraints thinking process approach. **International Journal of Logistics Systems and Management**, v. 18, n. 2, p. 270–282, 2014.

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

GOLDRATT, E. M. **Moving from Make to Stock (MTS) to Make to Availability (MTA) - GST MTA**. Goldratt Webcast Series, 2009. Available in: <https://www.toc-goldratt.com/en/product/the-goldratt-strategy-and-tactic-on-moving-from-make-to-stock-mts-to-make-to-availability-mta>

GOLDRATT, E. M. **The race**. [s.l.] Croton-on-Hudson NY: North river press, 1986.

GOLDRATT, E. M.; COX, J. **The Goal: A Process of Ongoing Improvement**. [s.l.] North River Press, 1984.

GOLDRATT, E.; COX, J. The Goal: Excellence in manufacturing. **European Journal of Operational Research**, v. 26, n. 3, p. 412–413, 2003.

GOLHAR, D. Y.; STAMM, C. L. The just-in-time philosophy: A literature review. **International Journal of Production Research**, v. 29, n. 4, p. 657–676, 1991.

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

GOVINDAN, K. Vendor-managed inventory: a review based on dimensions. **International Journal of Production Research**, v. 51, n. 13, p. 3808–3835, 2013.

GOVONI, H. et al. Analysis of production resources improvement strategies in make-to-stock environments managed by the simplified drum-buffer-rope system. **Gestão & Produção**, v. 28, n. 4, 2021.

GUAN, Z. et al. Operation and control of flow manufacturing based on constraints management for high-mix/low-volume production. **Frontiers of Mechanical Engineering in China**, v. 3, n. 4, p. 454–461, 2008.

GUPTA, M.; KO, H.-J.; MIN, H. TOC-based performance measures and five focusing steps in a job-shop manufacturing environment. **International Journal of Production Research**, v. 40, n. 4,

p. 907–930, 2002.

HABIB ZAHMANI, M. Extraction of Dispatching Rules for Single Machine Total Weighted Tardiness using a Modified Genetic Algorithm and Data Mining. **International Journal of Manufacturing Research**, v. 13, n. 1, p. 1, 2017.

HADAS, L.; CYPLIK, P.; FERTSCH, M. Method of buffering critical resources in make-to-order shop floor control in manufacturing complex products. **International Journal of Production Research**, v. 47, n. 8, p. 2125–2139, 2009.

HEGER, J. et al. Dynamic adjustment of dispatching rule parameters in flow shops with sequence-dependent set-up times. **International Journal of Production Research**, v. 54, n. 22, p. 6812–6824, 2016.

HOLLMANN, R. L.; SCAVARDA, L. F.; THOMÉ, A. M. T. Collaborative planning, forecasting and replenishment: a literature review. **International Journal of Productivity and Performance Management**, v. 64, n. 7, p. 971–993, 2015.

HOPP, W. J.; SPEARMAN, M. L. **Factory Physics**. [s.l: s.n.].

HUANG, Y. S.; CHEN, H. W. A mixed dispatching rule for semiconductor wafer fabrication. **International Journal of Systems Science: Operations and Logistics**, v. 5, n. 3, p. 195–203, 2018.

HUNG, Y. F.; CHANG, C. BIN. Dispatching rules using flow time predictions for semiconductor wafer fabrications. **Journal of the Chinese Institute of Industrial Engineers**, v. 19, n. 1, p. 67–75, 2002.

HUNG, Y. H.; HUANG, K. C.; YANG, T. A simulation study on CONWIP system design for bicycle chain manufacturing. **European J. of Industrial Engineering**, v. 16, n. 1, p. 31, 2022.

IKEZIRI, L. M. et al. Pulling the distribution in supply chains: simulation and analysis of Dynamic Buffer Management approach. **International Journal of Systems Science: Operations & Logistics**, p. 1–22, 2021.

IKEZIRI, L. M. et al. Theory of constraints: review and bibliometric analysis. **International Journal of Production Research**, v. 57, n. 15–16, p. 5068–5102, 2019.

JANS, R.; DEGRAEVE, Z. Modeling industrial lot sizing problems: a review. **International Journal of Production Research**, v. 46, n. 6, p. 1619–1643, 2008.

KABADURMUS, O.; DURMUSOGLU, M. B. Design of pull production control systems using axiomatic design principles. **Journal of Manufacturing Technology Management**, v. 31, n. 3, p. 620–647, 2019.

KASEMSET, C.; KACHITVICHYANUKUL, V. Bi-level multi-objective mathematical model for job-shop scheduling: The application of theory of constraints. **International Journal of Production Research**, v. 48, n. 20, p. 6137–6154, 2010.

KEGENBEKOV, Z.; JACKSON, I. Adaptive Supply Chain: Demand–Supply Synchronization Using Deep Reinforcement Learning. **Algorithms**, v. 14, n. 8, p. 240, 2021.

KHALIL, R. A.; STOCKTON, D. J.; FRESCO, J. A. Managing the effects of variability in drum-buffer-rope systems. **International Journal of Industrial and Systems Engineering**, v. 3, n. 1, p. 53, 2008.

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

KOÇ, Ç. et al. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. **Computers & Operations Research**, v. 64, p. 11–27, 2015.

KORYTKOWSKI, P.; WIŚNIEWSKI, T.; RYMASZEWSKI, S. An evolutionary simulation-based optimization approach for dispatching scheduling. **Simulation Modelling Practice and Theory**, v. 35, p. 69–85, 2013.

KOUKI, C. et al. Analysis of base-stock perishable inventory systems with general lifetime and lead-time. **European Journal of Operational Research**, v. 287, n. 3, p. 901–915, 2020.

KOULAMAS, C.; KYPARISIS, G. J. Single-machine and two-machine flowshop scheduling with general learning functions. **European Journal of Operational Research**, v. 178, n. 2, p. 402–407, 2007.

LADJ, A.; TAYEB, F. B. S.; VARNIER, C. Hybrid of metaheuristic approaches and fuzzy logic for the integrated flowshop scheduling with predictive maintenance problem under uncertainties. **European J. of Industrial Engineering**, v. 15, n. 5, p. 675, 2021.

LEE, C.-J.; RIM, S.-C. A Mathematical Safety Stock Model for DDMRP Inventory Replenishment. **Mathematical Problems in Engineering**, v. 2019, p. 1–10, 2019.

LEE, J. H. et al. Research on enhancement of TOC Simplified Drum-Buffer-Rope system using novel generic procedures. **Expert Systems with Applications**, v. 37, n. 5, p. 3747–3754, 2010.

LEE, J. H.; YU, J. M.; LEE, D. H. A tabu search algorithm for unrelated parallel machine scheduling with sequence- and machine-dependent setups: Minimizing total tardiness. **International Journal of Advanced Manufacturing Technology**, v. 69, n. 9–12, p. 2081–2089, 2013.

LEE, J.-H. et al. Research on enhancement of TOC Simplified Drum-Buffer-Rope system using novel generic procedures. **Expert Systems with Applications**, v. 37, n. 5, p. 3747–3754, 2010.

LIANG, Y.-C.; CUEVAS JUAREZ, J. R. A novel metaheuristic for continuous optimization problems: Virus optimization algorithm. **Engineering Optimization**, v. 48, n. 1, p. 73–93, 2016.

LIANG, Y.-C.; CUEVAS JUAREZ, J. R. A novel metaheuristic for continuous optimization problems: Virus optimization algorithm. **Engineering Optimization**, v. 48, n. 1, p. 73–93, 2016.

LIN, J.; SPIEGLER, V. L. M.; NAIM, M. M. Dynamic analysis and design of a semiconductor supply chain: a control engineering approach. **International Journal of Production Research**, v. 56, n. 13, p. 4585–4611, 2018.

LITTLE, J. D. C. A Proof for the Queuing Formula: $L = \lambda W$. **Operations Research**, v. 9, n. 3, p. 383–387, 1961.

LIU, F.-H.; SHEN, S.-Y. The fleet size and mix vehicle routing problem with time windows. **Journal of the Operational Research Society**, v. 50, n. 7, p. 721–732, 1999.

LIU, H. et al. A modified particle swarm optimization for large-scale numerical optimizations and engineering design problems. **Journal of Intelligent Manufacturing**, v. 30, n. 6, p. 2407–2433, 2019.

LIU, Y.; TIAN, Y.; SAWARAGI, T. A TOC-based heuristic algorithm for solving a two-row pattern container loading problem. **International Journal of Services Operations and**

Informatics, v. 2, n. 4, p. 339, 2007.

LIZARRALDE-AIASTUI, A.; DE EULATE, U. A. P.; MEDIAVILLA-GUISASOLA, M. A strategic approach for bottleneck identification in make-to-order environments: A drum-buffer-rope action research based case study. **Journal of Industrial Engineering and Management**, v. 13, n. 1, p. 18–37, 2020.

LOUW, L.; PAGE, D. C. Queuing network analysis approach for estimating the sizes of the time buffers in Theory of Constraints-controlled production systems. **International Journal of Production Research**, v. 42, n. 6, p. 1207–1226, 2004.

MALLADI, S. S. et al. Stochastic fleet mix optimization: Evaluating electromobility in urban logistics. **Transportation Research Part E: Logistics and Transportation Review**, v. 158, p. 102554, 2022.

MARICHELVAM, M. K.; GEETHA, M.; TOSUN, Ö. An improved particle swarm optimization algorithm to solve hybrid flowshop scheduling problems with the effect of human factors – A case study. **Computers & Operations Research**, v. 114, p. 104812, 2020.

MOSCA, A.; VIDYARTHI, N.; SATIR, A. Integrated transportation – inventory models: A review. **Operations Research Perspectives**, v. 6, p. 100101, 2019.

NASIRI, M. M.; YAZDANPARAST, R.; JOLAI, F. A simulation optimisation approach for real-time scheduling in an open shop environment using a composite dispatching rule. **International Journal of Computer Integrated Manufacturing**, v. 30, n. 12, p. 1239–1252, 2017.

NGUYEN, S. et al. Automatic programming via iterated local search for dynamic job shop scheduling. **IEEE Transactions on Cybernetics**, v. 45, n. 1, p. 1–14, 2015.

NGUYEN, S. **Optimization, dispatching rules and hyper-heuristics: A comparison in dynamic single machine scheduling**. IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society. **Annals**...Toronto, Canadian: IEEE, out. 2017. Available in: <http://ieeexplore.ieee.org/document/8216827/>

NGUYEN, S.; ZHANG, M. **A PSO-based hyper-heuristic for evolving dispatching rules in job shop scheduling**. 2017 IEEE Congress on Evolutionary Computation (CEC). **Annals**...IEEE, jun. 2017b. Available in: <https://www.researchgate.net/publication/318297382>. Acesso em: 12 ago. 2020

NOUIRI, M. et al. An effective and distributed particle swarm optimization algorithm for flexible job-shop scheduling problem. **Journal of Intelligent Manufacturing**, v. 29, n. 3, p. 603–615, 5 mar. 2018.

NOWOTARSKI, P.; PASLAWSKI, J. **Industry 4.0 Concept Introduction into Construction SMEs**. (M. M. Y. I. C. E. S. A. Drusa M. Rybak J., Ed.)IOP Conference Series: Materials Science and Engineering. **Annals**...Institute of Physics Publishing, 2017. Disponível em: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85033675852&doi=10.1088%2F1757-899X%2F245%2F5%2F052043&partnerID=40&md5=16be1e85b9bafeff52411b402c7bf505>

ONWUBOLU, G. C. Tabu search-based algorithm for the TOC product mix decision. **International Journal of Production Research**, v. 39, n. 10, p. 2065–2076, 2001.

PANIZZOLO, R. Theory of constraints (TOC) production and manufacturing performance. **International Journal of Industrial Engineering and Management**, v. 7, n. 1, p. 15–23, 2016.

PASS, S.; RONEN, B. Management by the market constraint in the Hi-Tech industry. **International Journal of Production Research**, v. 41, n. 4, p. 713–724, 2003.

PEETERS, K.; VAN OOIJEN, H. Hybrid make-to-stock and make-to-order systems: a taxonomic review. **International Journal of Production Research**, v. 58, n. 15, p. 4659–4688, 2020.

POORMOAIED, S.; ATAN, Z. A continuous review policy for two complementary products with interrelated demand. **Computers and Industrial Engineering**, v. 150, p. 106980, 2020.

PRATA, B. DE A.; DE ABREU, L. R.; LIMA, J. Y. F. Heuristic methods for the single-machine scheduling problem with periodical resource constraints. **TOP**, p. 1–23, 2020.

QIAO, F.; WU, Q. Layered drum-buffer-rope-based scheduling of reentrant manufacturing systems. **IEEE Transactions on Semiconductor Manufacturing**, v. 26, n. 2, p. 178–187, 2013.

QIN, W. et al. A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem. **Computers and Industrial Engineering**, v. 156, n. March, p. 107252, 2021.

QIU, J. et al. **Integrated Inventory Placement and Transportation Vehicle Selection using Neural Network**. 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE). **Annals**...IEEE, 2022.

RADOVILSKY, Z.; FRANKEL, M. Identifying buffer size in front of capacity-constrained resource with parallel processes. **International Journal of Business Research**, v. 13, n. 4, 2013.

RAJENDRAN, C.; HOLTHAUS, O. Comparative study of dispatching rules in dynamic flowshops and jobshops. **European Journal of Operational Research**, v. 116, n. 1, p. 156–170, 1999.

RHEE, S. H.; CHO, N. W.; BAE, H. Increasing the efficiency of business processes using a theory of constraints. **Information Systems Frontiers**, v. 12, n. 4, p. 443–455, 2010.

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

ROLF, B. et al. Assigning dispatching rules using a genetic algorithm to solve a hybrid flow shop scheduling problem. **Procedia Manufacturing**, v. 42, p. 442–449, 2020.

SAIF, U. et al. Drum buffer rope-based heuristic for multi-level rolling horizon planning in mixed model production. **International Journal of Production Research**, v. 57, n. 12, p. 3864–3891, 2019.

SÁNCHEZ, D. G. et al. A Clustering Approach for the Optimal Siting of Recharging Stations in the Electric Vehicle Routing Problem with Time Windows. **Energies**, v. 15, n. 7, p. 2372, 2022.

SARKAR, B. et al. Optimal production delivery policies for supplier and manufacturer in a constrained closed-loop supply chain for returnable transport packaging through metaheuristic approach. **Computers & Industrial Engineering**, v. 135, p. 987–1003, 2019.

SCHRAGENHEIM, E. Make-to-Stock under Drum-Buffer-Rope and Buffer Management Methodology. **International Conference Proceedings, APICS - The Educational Society for Resource Management**, p. 1–5, 2002.

SCHRAGENHEIM, E. Managing Make-to-Stock and the Concept of Make-to-Availability. In: MCGRAW-HILL (Ed.). . **Theory of Constraints Handbook**. New York: [s.n.].

SCHRAGENHEIM, E.; COX, J.; RONEN, B. Process flow industry—scheduling and control using theory of constraints. **International Journal of Production Research**, v. 32, n. 8, p. 1867–1877, 1994.

SCHRAGENHEIM, E.; DETTMER, H. W. **Manufacturing at Warp Speed: Optimizing Supply**

Chain Financial Performance. [s.l.] CRC Press, 2000.

SCHRAGENHEIM, E.; DETTMER, H. W.; PATTERSON, J. W. **Supply Chain Management at Warp Speed**. [s.l.] CRC Press, 2009.

SHAHZAD, A.; MEBARKI, N. Learning Dispatching Rules for Scheduling: A Synergistic View Comprising Decision Trees. **Tabu Search and Simulation.(5) p**, v. 3, 2016.

SHIMIZU, Y.; IKEDA, M. **A parallel hybrid binary PSO for capacitated logistics network optimization***. Journal of Advanced Mechanical Design, Systems and Manufacturing. **Anais**...2010. Available in: <http://dx.doi.org/10.1299/jamdsm.4.616>

SHIMIZU, Y.; SAKAGUCHI, T.; MIURA, T. Parallel computing for huge scale logistics optimization through binary PSO associated with topological comparison. **Journal of Advanced Mechanical Design, Systems, and Manufacturing**, v. 8, n. 1, 2014.

SIMONS, J. V; SIMPSON, W. P. An exposition of multiple constraint scheduling as implemented in the goal system (formerly disastertm). **Production and Operations Management**, v. 6, n. 1, p. 3–22, 2009.

SIRIKRAI, V.; YENRADEE, P. Modified drum-buffer-rope scheduling mechanism for a non-identical parallel machine flow shop with processing-time variation. **International Journal of Production Research**, v. 44, n. 17, p. 3509–3531, 2006.

SOLOMON, M. M. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. **Operations Research**, v. 35, n. 2, p. 254–265, abr. 1987.

SRIKUN, I.; SAWETSAKULANOND, B. **A solution for stochastic optimal power flow with integrated wind power generation using a modified cultural-based bee algorithm**. 19th International Conference on Electrical Machines and Systems, ICEMS 2016. **Anais**...Chiba, Japan: 2016.

SULLIVAN, T.; REID, R. A.; CARTIER, B. The TOCICO Dictionary. 2007.

TASGETIREN, M. F. et al. A particle swarm optimization algorithm for makespan and total flowtime minimization in the permutation flowshop sequencing problem. **European Journal of Operational Research**, v. 177, n. 3, p. 1930–1947, 2007.

TEPPAN, E. C.; DA COL, G. **Genetic Algorithms for Creating Large Job Shop Dispatching Rules**. Smart Innovation, Systems and Technologies. **Annals**...Springer, 2020.

THÜRER, M. et al. Drum-buffer-rope and workload control in High-variety flow and job shops with bottlenecks: An assessment by simulation. **International Journal of Production Economics**, v. 188, p. 116–127, 2017.

THÜRER, M.; STEVENSON, M. Bottleneck-oriented order release with shifting bottlenecks: An assessment by simulation. **International Journal of Production Economics**, v. 197, p. 275–282, 2018a.

THÜRER, M.; STEVENSON, M. On the beat of the drum: improving the flow shop performance of the Drum–Buffer–Rope scheduling mechanism. **International Journal of Production Research**, v. 56, n. 9, p. 3294–3305, 3 maio 2018b.

TIAN, D.; SHI, Z. MPSO: Modified particle swarm optimization and its applications. **Swarm and Evolutionary Computation**, v. 41, n. August 2017, p. 49–68, 2018.

TRANFIELD, D.; DENYER, D.; SMART, P. Towards a Methodology for Developing Evidence-

Informed Management Knowledge by Means of Systematic Review. **British Journal of Management**, v. 14, n. 3, p. 207–222, 2003.

TURKENSTEEN, M.; VAN DEN HEUVEL, W. The trade-off between costs and carbon emissions from economic lot-sizing decisions. **INFOR: Information Systems and Operational Research**, v. 61, n. 2, p. 169–198, 2023.

UMBLE, M.; UMBLE, E.; MURAKAMI, S. Implementing theory of constraints in a traditional Japanese manufacturing environment: The case of Hitachi Tool Engineering. **International Journal of Production Research**, v. 44, n. 10, p. 1863–1880, 2006.

URBAN, W. TOC implementation in a medium-scale manufacturing system with diverse product rooting. **Production & Manufacturing Research**, v. 7, n. 1, p. 178–194, 2019.

URBAN, W.; ROGOWSKA, P. Methodology for bottleneck identification in a production system when implementing TOC. **Engineering Management in Production and Services**, v. 12, n. 2, p. 74–82, 2020.

VAN DER WALT, A.; BEAN, W. L. Inventory management for the in-flight catering industry: A case of uncertain demand and product substitutability. **Computers and Industrial Engineering**, v. 165, p. 107914, 2022.

VAN DONSELAAR, K.; BROEKMEULEN, R.; DE KOK, T. Heuristics for setting reorder levels in periodic review inventory systems with an aggregate service constraint. **International Journal of Production Economics**, v. 237, p. 108137, 2021.

VIDAL, T. Hybrid genetic search for the CVRP: Open-source implementation and SWAP* neighborhood. **Computers and Operations Research**, v. 140, n. November 2020, p. 105643, 2022.

VITAL-SOTO, A.; AZAB, A.; BAKI, M. F. Mathematical modeling and a hybridized bacterial foraging optimization algorithm for the flexible job-shop scheduling problem with sequencing flexibility. **Journal of Manufacturing Systems**, v. 54, p. 74–93, 2020.

WALKER, W. T. Practical application of drum-buffer-rope to synchronize a two-stage supply chain. **Production and Inventory Management Journal**, v. 43, n. 3/4, p. 13–23, 2002.

WANG, C.; GHENNIWA, H.; SHEN, W. Real time distributed shop floor scheduling using an agent-based service-oriented architecture. **International Journal of Production Research**, v. 46, n. 9, p. 2433–2452, 2008.

WANG, J. Q. et al. Schedule-based execution bottleneck identification in a job shop. **Computers and Industrial Engineering**, v. 98, p. 308–322, 2016.

WANG, L. C. et al. Demand-pull replenishment model for hospital inventory management: A dynamic buffer-adjustment approach. **International Journal of Production Research**, v. 53, n. 24, p. 7533–7546, 2015.

WANG, Y. et al. Collaborative multicenter vehicle routing problem with time windows and mixed deliveries and pickups. **Expert Systems with Applications**, v. 197, p. 116690, 2022.

WANG, Y. et al. Two-echelon multi-period location routing problem with shared transportation resource. **Knowledge-Based Systems**, v. 226, p. 107168, 2021.

WANG, Z.-W.; LI, S.-S. Effective channels of breaking the phonon bottleneck: A lattice relaxation approach. **Journal of Applied Physics**, v. 110, n. 4, p. 043512, 2011.

WATSON, K. J.; PATTI, A. A comparison of JIT and TOC buffering philosophies on system performance with unplanned machine downtime. **International Journal of Production Research**, v. 46, n. 7, p. 1869–1885, 2008.

WOO, K.; PARK, S.; FUJIMURA, S. The real-time buffer management for DBR scheduling. **Production**, v. 16, n. 1–2, 2009.

WU, H. H. et al. Simulation and scheduling implementation study of TFT-LCD Cell plants using Drum-Buffer-Rope system. **Expert Systems with Applications**, v. 37, n. 12, p. 8127–8133, 2010.

WU, H. H.; LIU, J. Y. A capacity available-to-promise model for drum-buffer-rope systems. **International Journal of Production Research**, v. 46, n. 8, p. 2255–2274, 2008.

WU, H. H.; YEH, M. L. A DBR scheduling method for manufacturing environments with bottleneck re-entrant flows. **International Journal of Production Research**, v. 44, n. 5, p. 883–902, 2006.

XIA, X. et al. A fitness-based multi-role particle swarm optimization. **Swarm and Evolutionary Computation**, v. 44, p. 349–364, 2019.

XU, L. et al. Estimation of tool wear and optimization of cutting parameters based on novel ANFIS-PSO method toward intelligent machining. **Journal of Intelligent Manufacturing**, 2020.

XU, X. W.; FANG, L.; ZHU, X. L. The Study of MTA Production Model and Production Processes. **Applied Mechanics and Materials**, v. 268–270, n. PART 1, p. 2062–2065, 2012.

YANG, M.-L.; TSAI, T.-C. Enhancement of scheduling reliability in building project using theory of constraint. **Journal of the Operations Research Society of Japan**, v. 51, n. 4, p. 284–298, 2008.

YUE, L. et al. Order Releasing and Scheduling for a Multi-Item MTO Industry: An Efficient Heuristic Based on Drum Buffer Rope. **Applied Sciences 2022, Vol. 12, Page 1925**, v. 12, n. 4, p. 1925, 2022.

YUNG-CHIA CHANG. Using simplified drum-buffer-rope for re-entrant flow shop scheduling in a random environment. **AFRICAN JOURNAL OF BUSINESS MANAGEMENT**, v. 5, n. 26, p. 10796–10810, 2011.

ZHANG, H.; ROY, U. A semantics-based dispatching rule selection approach for job shop scheduling. **Journal of Intelligent Manufacturing**, v. 30, n. 7, p. 2759–2779, 2019.

APPENDIX A – CHAPTER 6 SUPPLEMENTARY MATERIAL

1. Mathematical Notation

Table 1 – Mathematical notation

	Symbol	Meaning
Sets	N	Set of nodes $\{0, 1, \ldots, n\}$ encompassing depot (node 0) and n clients/points of sale.
	C	Set of $N\backslash\{0\}$ nodes encompassing n clients/points of sale.
	E	Set of edges $\{(i,j): i, j \in N, i \neq j\}$.
	$G(N,E)$	Graph associated to distribution network.
	$C_i(x_i, y_i)$	Cartesian coordinate for a node $i \in N$.
	Θ	Set of nodes coordinates $\{C_0, C_1, \ldots, C_n\}$.
	K	Set of $\{1, \ldots, l\}$ distinct l types of vehicles that compose the fleet available at the depot.
	G	The route of all points of sale to be visited in each solution, named giant tour.
	C_{visits}	Points of sale $\{i: i \in C\}$ to be visit given to the initial solution of the Hybrid Heuristic Algorithm.
	C_{visits}^k	Cluster composed by points of sale $i \in C_{visits}$ to be visited by vehicle $k \in K$
	D	Set of number of items $\{D_p^i: i \in C, p \in P, D_p^i > 0\}$ given to the initial solution of the Hybrid Heuristic Algorithm, where D_p^i is the decision variable for the of number of items destinated to a stock buffer B_p^i.
	F	Fleet of vehicles for a give solution of the Hybrid Heuristic Algorithm.
	IND_i	An individual of a population of the Hybrid Heuristic Algorithm, composed by set of number of items D, giant tour G, set of routes R and fleet F.
	POP	Population composed by individual IND_i.
	A	Set of centroids $\{\tau_k: k = \{1,2, \ldots, \|F\|\}\}$ for K-means algorithm, τ_k is a centroid associated to a vehicle $k \in F$.
Parameters	d_{ij}	Euclidian distance between nodes $i \in N$ and $j \in N$.
	Q_k	Capacity for each $k \in K$.
	f_k	Fixed cost f_k for each $k \in K$.
	c_k	Cost by one unit distance traveled for each $k \in K$.
	R^k	Route of a vehicle $k \in K$ composed by sequence of visited nodes $(r_1, r_2, \ldots, r_{\|R\|})$, starting and ending at depot $(r_1 = r_{\|R\|} = 0)$.
	Φ_k	Centroid associated to a vehicle $k \in K$.
	R	Set of routes $(R^1, R^2, \ldots, R^l)$.
	P	Represents the set $\{1, \ldots, 10\}$ of 10 distinct types of products available at each node $i \in N$.
	Q_p	Capacity required to transport an item of a product $p \in P$.
	B_p^i	Stock buffer for a product $p \in P$ warehoused in a node $i \in N$.
	tl_p^i	Target level for a stock buffer B_p^i.
	ct_p^i	Holding cost for a product $p \in P$ warehoused at node $i \in N$.
	pr_p^i	Price of sale for a product $p \in P$ warehoused at node $i \in N$.
	s_p^i	Inventory level (on hand plus in transit) for a stock buffer B_p^i.
	bs_p^i	Buffer status for a stock buffer B_p^i.
	r_p^i	Ready rate measure for a stock buffer B_p^i.
	$\tau_k(x,y)$	Coordinate (x, y) associated for each centroid $\tau_k \in A$.
	n_a	Size of a population at Hybrid Heuristic Algorithm.
	n_p	Number of initial individuals in a population for Hybrid Heuristic Algorithm.
	n_o	Maximum number of offspring in a population for Hybrid Heuristic Algorithm.
	$[b_l^i, b_u^i]$	A lower and an upper bound of removed nodes as percentage of the total number of nodes in an instance, used in initialize Hybrid Heuristic Algorithm.
	$[b_l^e, b_u^e]$	A lower and an upper bound of removed nodes as percentage of the total number of nodes in an instance, used in education procedure of the Hybrid Heuristic Algorithm.
	$[b_l^m, b_u^m]$	A lower and an upper bound of removed nodes as percentage of the total number of nodes in an instance, used in diversification procedure of the Hybrid Heuristic Algorithm.
	t	Percentual limit of transport cost of a route R^kover total profit of the route.
Decision Variables	D_p^i	Positive integer decision variable for the number of items destinated to a stock buffer B_p^i.
	D	Set $\{D_p^i: i \in C, p \in P;\ D_p^i \geq 0\}$ containing optimal values of the decision variable D_p^i.
	D_{ip}^k	Positive integer decision variable for the number of items to delivery by vehicle $k \in K$ to a stock buffer B_p^i.
	y_i^k	This binary decision variable is set to 1 if the vehicle $k \in \mathcal{K}$ is designated to visit point of sale $i \in C$, otherwise, is set to 0.
	x_{ij}^k	This binary decision variable is set to 1 if vehicle $k \in \mathcal{K}$ is designated to edge $(i,j) \in \mathcal{E}$, otherwise, is set to 0.

2. ALNS Operators

2.1 Removal Operators

The destroy phase of the educational procedure uses three removal operators, as described in detail below.

1. Random removal (RR): The RR operator randomly selects node $j \in N\backslash\{0\}\backslash L_r$ and removes it from the solution.
2. Worst distance removal (WDR): The purpose of the WDR operator is to choose the number of expensive nodes according to their distance-based cost. The cost of node $j \in N\backslash\{0\}\backslash L_r$ is dependent on its distance from its predecessor i and its distance from its successor k. The WDR operator iteratively removes node j^* from the solution, where $j^* = \arg\max_{j\in N\backslash\{0\}\backslash L_r}\{d_{ij} + d_{jk}\}$.
3. Neighborhood removal (NR): In a given solution with a set R of routes, the NR operator calculates an average distance $\bar{d}(R) = \sum_{(i,j)\in R} d_{ij}/|R|$ for each route $R^k \in R$, and selects a node $j^* = \arg\max_{(R^k\in R; j\in R^k)}\{\bar{d}(R) - d_{R\backslash\{j\}}\}$, where $d_{R\backslash\{j\}}$ denotes the average distance of route R excluding node j.

2.1 Insertion Operators

The repair phase of the education procedure uses two insertion operators.

1. Greedy insertion (GI): The aim of this operator is to find the best possible position for insertion for all the nodes in L_r. For node $i \in N\backslash L_r$ that succeeded in the solution destroyed by $N\backslash\{0\}\backslash L_r$, and node $j \in L_r$ we define $\gamma(i,j) = d_{ij} + d_{jk} - d_{ik}$. We find the least-cost insertion position for $j \in L_r$ using $i^* = \arg\min_{i\in N\backslash L_r}\{\gamma(i,j)\}$. This process is iteratively applied to all the nodes in L_r.
2. Greedy insertion with noise function (GINF): The GINF operator is based on the GI operator but extends it by allowing a degree of freedom in selecting the best place for a node. This is done by calculating the noise cost $v(i,j) = \gamma(i,j) + d_{max}p_{n^\epsilon}$ where d_{max} is the maximum distance between any two nodes, p_n is a noise parameter used for diversification and is set equal to 0.1, and ϵ is a random number $[-1,1]$.

3. Adaptive Weight Adjustment Procedure (AWAP)

Each removal and insertion operator has a certain probability of being chosen in each iteration. A roulette-wheel mechanism based on the historical performance of every operator calibrated the selection criterion.

The probability is recalculated for each operator after it_w iterations of roulette wheel segmentation according to its total score. Initially, the probabilities of each removal and insertion operator are equal. Let p_i^t be the probability of operator i in the last it_w iterations, $p_i^{t+1} = p_i^t\left(1 - r_p\pi_i/\tau_i\right)$, where r_p is the roulette wheel probability for operator i; in the last segment, π_i is the score, and τ_i is the number of times. The scores of all operators were zero at the start of each segment. The scores are changed by σ_1 if a best new solution is found, by σ_2 if the new solution is better than the current solution, and by σ_3 if the new solution is worse than the current solution.

4. Initialization

The procedure used to generate the initial population consists of two steps, as shown in Algorithm 1. Step 1 obtains the product mix and creates routes for clusters of points of sale using the K-means algorithm. In Step 2, the ALNS algorithm generates the population from the initial solution given in Step 1.

In Step 1, model Z_1 establishes the buffers that will be replenished (Line 2). With this, the points of sale that will be visited are defined. K-means uses these data to create clusters using Algorithm 3 (Line 4). Each cluster has a set of information: vehicle, group of points of sale to be visited, and number of items of each product to be delivered (Line 5). A *cluster* is a travelling salesman problem routed through the 2-opt heuristic to minimize route distance (Line 7). A list of the sequence of visits of all vehicles, named the giant tour, is used along with the HEA to build combinations of routes (Line 9). The output of step 1 is the initial solution composed of the number of products destined for points of sale, a set of routes, a giant tour, and a fleet of vehicles (Line 10).

The ALNS algorithm generates a population from the initial solution given in Step 1. New individuals are created by applying the initial solution operators based on random removals and greedy insertions with a noise function (see Sections 4.3.1 and 4.3.2) to the giant tour to alternate points of sale positions (Lines 16-18), until the initial population size reaches n_p. We select these two operators to diversify the initial population. Several nodes, calculated as a percentage of the total number of nodes in an instance, are removed randomly, from the initialization interval $\left[b_l^i, b_u^i\right]$ which is defined by a lower and upper bound. Algorithm 4 performs the split tour of the updated giant tour in new routes with split deliveries (Line 20). The split-tour procedure is explained in Section 4.2. The individual is added to the initial population if the routes are feasible; otherwise, the position of the vehicle responsible for the infeasible route is changed with that of another vehicle in the fleet. In this case, the individual is again subjected to ALNS until the algorithm achieves a feasible solution (Lines 21-27). A feasible solution in financial terms is given according to Equation (20).

The algorithm ends when the population achieved a size n_p (Line 31). Finally, the algorithm returns the initial population (Line 32).

Algorithm 1 – Initialize population based on K-means and ALNS algorithm

Step 1 – Initial solution

1 *Buffer selection:* get product mix $D = \left\{D_p^i : i \in C, p \in P, D_p^i > 0\right\}$ and points of sale to be visited $C_{visits} =$
2 $\{i : i \in C\}$ from model Z_1;
3 *Clusters of points of sales:* Points of sale C_{visits} are clustering through K-means algorithm (Algorithm 3).
4 Let C_{visits}^k be the cluster composed of points of sale $i \in C_{visits}$ to be visited by vehicle $k \in K$ and D_{ip}^k the number
5 of items of a product $p \in P$ to be delivered by a vehicle $k \in K$ to a point of sale $i \in C_{visits}$;
6 *Routes:* A cluster C_{visits}^k is a travelling salesman problem, which is improved by a 2-opt heuristic to establish the
7 set of routes $R = \{R^k : k \in F\}$ with minimal financial costs.
8 *Giant tour:* Create giant tour G through routes R;
9 *Initial solution:* Let $\mathrm{S}(D, R, G, F)$ be the initial solution;
10 ***Step 2 – Generates the population from initial solution***
11 Let $IND_1 = \mathrm{S}(D, R, G, F)$ be the first individual of population POP:
12 $POP = \emptyset$
13 **While** population POP size $\leq n_p$ **do:**
14 $L_r = \emptyset$;
15 Apply a random removal operator to IND_i giant tour G to remove a set of nodes and add them to L_r;
16 Apply greedy insertion noise operator to IND_i of the partially destroyed giant tour G to insert the nodes
17 of L_r;
18 Let $\hat{G}$ be the new giant tour obtained by applying the insertion operator;
19 Apply split tour to $D, \hat{G}, F$ through Algorithm 3;
20 Let $\hat{D}$ and $\hat{R}$ new split deliveries and set of routes;
21 **If** routes $\hat{R}$ are feasible **then:**
22 $IND_i =$ new solution $S\left(\hat{D}, \hat{R}, \hat{G}, F\right)$;
23 Add new feasible solution IND_i into population POP;
24 **Else:**
25 For each infeasible route $R^k \in \hat{R}$ do:
26 Change the position of vehicle k associated with route R^k on individual IND_i
27 fleet F with another vehicle at a random position.
28 Let $\hat{F}$ be the new fleet;
29 $IND_i = \left(\hat{D}, \hat{R}, \hat{G}, \hat{F}\right)$;
30 **End if;**
31 **End while**;
32 Return initial population POP;

The clustering of points of sales (Line 3 of Algorithm 2) is given by K-means, as in Algorithm 3. The K-means algorithm is a typical clustering method for grouping data into k different clusters connected to centroids and minimizing inter-cluster similarities while maximizing intra-cluster similarities (WANG et al., 2021). The objective function minimizes the sum distance between the points of sale and clustering center. In the classical K-means algorithm, distance is a standard metric for similarity estimation (WANG et al., 2021).

K-means clustering consists of two separate phases. The first phase randomly defines the centroids, and the second phase assigns each element to the nearest centroid according to the Euclidean distance. This process is carried out until the function criterion (e.g., distance) reaches a minimum (SÁNCHEZ et al., 2022). The K-means algorithm randomly places the centroids and then moves them at each stage to minimize distance.

Each vehicle is a centroid, and the objective of the K-means algorithm is to allocate a group of points of sale to each centroid. Deliveries are divided among vehicles when one vehicle cannot service a point of sale alone. Therefore, a point of sale may belong to more than one centroid. For this, we designed algorithm 2 using a two-step algorithm.

Algorithm 2 – Clustering points of sale by K-means and split delivers

Input: A, C_{visits}, D, F;
***Step 1* - Clustering points of sale by K-means**
Let the set of centroids $A = \{\tau_k : k = \{1,2,\dots,|F|\}\}$, $\tau_k(x, y)$ be the coordinate associated with each $\tau_k \in A$
randomly selected from points of sale coordinates Θ;
While coordinate positions (x, y) of the centroids $\tau_k \in A$ are changed **do:**
    Calculate the Euclidean distance from each point of sale $i \in C_{visits}$ to each centroid $\tau_k \in A$;
    Connect each $D_p^i \in D$ to the closest centroid τ_k that has sufficient free capacity at associated vehicle
    $k \in F$ to transport all items, respecting the limit capacity; otherwise, include D_p^i to $\widehat{D}$;
    Let $\overline{C_{visits}}$ be the set of clusters C_{visits}^k;
    Let $\overline{D} = \{D_{ip}^k : p \in P, k \in F, i \in C_{visits}^k\}$ be the set of number of items D_{ip}^k of products $p \in P$ to be
    delivered by a vehicle $k \in F$ to points of sale $i \in C_{visits}^k$;
    For each centroid, $\tau_k \in A$ define a new coordinate (x, y) by the arithmetic mean of the point-of-sale
    coordinates associated with the respective cluster $\{C_i(x_i, y_i) : i \in C_{visits}^k\}$;
End while;
***Step 2 – Split* $\boldsymbol{D_p^i} \in \boldsymbol{\widehat{D}}$ between the remaining capacity of the vehicles $\boldsymbol{k} \in \boldsymbol{F}$.**
For each $D_p^i \in \widehat{D}$ **do:**
    Rank A centroids from closest to farthest from the D_p^i point of sale;
    For each centroid $\tau_k \in A$ **do**:
        Allocate the maximum number of items D_p^i in the free capacity of vehicle $k \in F$ associated
        with τ_k;
        Subtract from D_p^i the number of items allocated to the vehicle of the τ_k;
        Set D_p^i the number of items allocated to D_{ip}^k;
        Include D_{ip}^k in $\overline{D}$;
        Include i in C_{visits}^k;
        If $D_p^i = 0$ **then:**
            Exit For loop;
        End if;
    End for;
Return $\overline{D}$, $\overline{C_{visits}}$

Step 1 of Algorithm 3 starts by defining for each centroid the initial coordinate (x, y), extracted randomly from the coordinates of the points of sale (Line 3). At each iteration (Line 5), the Euclidean distance between each point of sale and centroids is calculated. The points of sale are then incorporated into the nearest centroid (Lines 6-7). The incorporation procedure allocates products from each point of sale to the vehicle's capacity limit (Lines 6-7). All items are allocated to a vehicle if they have a sufficient capacity to receive them. Otherwise, the product is left out to be divided between the vehicles in the second step of the algorithm. Next, for each centroid, the new coordinate is defined using the average coordinates of the incorporated points of sale (Line 14). The first phase ends when none of the centroid coordinates change from one iteration to another (Line 5).

In the second phase, the product items are divided among vehicles with a free capacity if not incorporated into centroids, going from the closest to the furthest centroid (Lines 18-29). The output of the algorithm is the mix of products that each vehicle must deliver and the delivery routes (Line 30).

5. Split Tour

The split tour procedure aims to create delivery routes from the product mix, giant tours, and vehicle fleets. The split tour logic is represented by Algorithm 3, which is like the K-means algorithm (Algorithm 2) and allows more than one vehicle to visit a point of sale. Each iteration allocates the products of the points of sale on the giant tour to a fleet vehicle until the limit capacity is reached or until there are no more products to allocate (Lines 4-5). If the vehicle has the capacity for all product items, the algorithm allocates it (Lines 6-10); otherwise, it allocates enough items to occupy all free capacity (Lines 12-14). The point of sale is added to the route when the vehicle no longer has free capacity (Lines 16-18). There is no change to another product or to the next point of sale until all the items of a product are allocated (Lines 20-29). After this procedure, the vehicle route is added to the route set (Line 31), and the algorithm moves to the following vehicle (Line 4). The output of the algorithm is a mix of products that each vehicle must deliver and the delivery routes (Line 34).

Algorithm 3 – Split Tour

Input: D, G, F
Let $ACTUAL\_i$ be the first point of sale of giant tour G;
Let $ACTUAL\_D_p^i$ be the number of items of the first product from point of sale $ACTUAL\_i$;
For each $ACTUAL\_k \in F$ **do:**
  While $ACTUAL\_k$ has free capacity and ACTUAL_$D_p^i \geq 1$ **do**:
    If vehicle $ACTUAL\_k$ has the capacity to accept all items ACTUAL_D_p^i **then:**
      $\widehat{D_{ip}^k} = ACTUAL\_D_p^i$;
      Allocate $ACTUAL\_D_p^i$ at $ACTUAL\_k$ and reduce its free
      capacity;
      $ACTUAL\_D_p^i = 0$;
    Else:
      Let $\widehat{D_{ip}^k}$ be the number of items from $ACTUAL\_D_p^i$ enough to occupy the $ACTUAL\_k$
      free capacity.
      $ACTUAL\_D_p^i = ACTUAL\_D_p^i - \widehat{D_{ip}^k}$
    End if;
    Add $\widehat{D_{ip}^k}$ to set $\widehat{D}$.
    If $ACTUAL\_k$ has no free capacity **then:**
      Include $ACTUAL\_i$ in route R^k;
    End if;

    If $ACTUAL\_D_p^i = 0$ **then:**
      If ACTUAL_D_p^i is not the last product from $ACTUAL\_i$ **then:**
        Let ACTUAL_D_p^i the number of items of the next product p
        from the point of sale $ACTUAL\_i$;
      Else if $ACTUAL\_i$ is not the last one point of sale in G:
        Let $ACTUAL\_i$ be the next point of sale $i \in G$;
        Let $ACTUAL\_D_p^i$ be the number of items of the first product
        from the point of sale $ACTUAL\_i$;
      End if;
    End if;
  End while;
  Include R^k in $\hat{R}$;
End for;
Return $\widehat{D}, \hat{R}$;

6. Education

The education procedure is applied to each offspring to make it feasible. The ALNS algorithm educates HEA by removing insertion operators and changing the positions of some nodes in each iteration.

Algorithm 4 details the educational procedure. The removal procedure (Line 4) runs for n' iterations, removes n' customers from the solution, and adds them to the removal list L_r, where n' is in the interval of removable nodes $n'[b_l^e, b_u^e]$. An insertion operator is selected to iteratively insert the nodes into the partially destroyed solution, starting from the first customer of L_r, until L_r is empty (Line 5). The removal and insertion operators were randomly selected according to past performance and a certain probability, as explained in Section 4.3.3. The vehicles responsible for infeasible routes change positions with another vehicle in the fleet (Lines 8-11).

A solution is feasible if it is financially viable according to Equation (20). The split tour procedure is applied to the product mix, new giant tour, and new fleet sequence (Line 13). The individual is updated if the split tour generates feasible routes; otherwise, the individual is again subjected to iteration (Line 2). This procedure is repeated when there is an infeasible route (Line 19).

Algorithm 4 – Education through ALNS algorithm

1 **Input:** $IND_i = \{D, R, G, F\}$;
2 **While** there are infeasible routes in R **do:**
3 $\quad L_r = \emptyset$;
4 $\quad$ A removal operator is selected and applied to individual IND_i giant tour G to remove a set of nodes and
5 $\quad$ add them to L_r;
6 $\quad$ Select an insertion operator and apply it to individual IND_i partially destroyed giant tour G to
7 $\quad$ insert the nodes of L_r;
8 $\quad$ Let $\hat{G}$ be the new giant tour obtained by applying the removal and insertion operator;
9 $\quad$ **For** each infeasible $R^k \in R$ **do:**
10 $\quad\quad$ Change the position of vehicle k associated with route R^k with another
11 $\quad$ vehicle in a random position in fleet F;
12 $\quad$ **End for;**
13 $\quad$ Let $\hat{F}$ be the new fleet;
14 $\quad$ Apply split tour to $D, \hat{G}, \hat{F}$ (Algorithm 4);
15 $\quad$ Let $\widehat{D_{ip}^k}$ and $\hat{R}$ be the new split deliveries and set of routes, respectively;
16 $\quad$ **If** routes in $\hat{R}$ are feasible **then:**
17 $\quad\quad$ Update $\widehat{D_{ip}^k}$, $\hat{R}, \hat{G}$ and $\hat{F}$ for individual IND_i;
18 $\quad$ **End if;**
19 **End while;**
20 Return feasible solution IND_i

7. Intensification

We introduced a two-phase aggressive intensification procedure to improve the quality of elite individuals. This procedure intensifies the search for promising regions in the solution space. Algorithm 5 details the pseudocode of this method.

Algorithm 5 – Intensification by ALNS algorithm

Initialize: $L_e = \{IND_i : IND_i \in POP, i = \{1,2,\dots,n_e\}\}$; $i = 1$;
While $i \le n_e$ do:
  Step 1
  While the IND_i weighted net profit is improved and routes R of IND_i are not infeasible **do**:
    $L_r = \emptyset$;
    Select a removal operator and apply to individual IND_i giant tour G to remove a set of nodes and add them to L_r;
    Select an insertion operator and apply it to individual IND_i partially destroyed giant tour G to insert the nodes of L_r;
    Let $\hat{G}$ be the new giant tour obtained by applying the insertion operator;
    Apply split tour to $D_p^i, \hat{G}, F$ (Algorithm 3);
    Let $\widehat{D_{ip}^k}$ and $\hat{R}$ be the new split deliveries and set of routes, respectively;

    For each infeasible $R^k \in \hat{R}$ **do:**
      Change the position of vehicle k associated with route R^k in individual IND_i fleet F to another vehicle at a random position.
    End for;
    Let $\hat{F}$ be the new fleet;
    If IND_i weighted net profit is improvement and routes $\hat{R}$ are feasible **then:**
      Update $\widehat{D_{ip}^k}$, $\hat{R}, \hat{G}$ and $\hat{F}$ for individual IND_i;
    End if;
  End while;
  Step 2
  While the IND_i weighted net profit is improved and routes R of IND_i are not infeasible **do**:
    $L_r = \emptyset$;
    Apply a random removal operator to individual IND giant tour G to remove a set of nodes and add them to L_r;
    Apply a greedy insertion operator to an individual IND partially destroyed giant tour G to insert the nodes of L_r;
    Let $\hat{G}$ be the new giant tour obtained by applying insertion operator;
    Apply split tour to $D_p^i, \hat{G}, F$ (Algorithm 3);
    Let $\widehat{D_{ip}^k}$ and $\hat{R}$ be the new split deliveries and set of routes, respectively;
    For each infeasible $R^k \in \hat{R}$ **do:**
      Change the position of vehicle k associated with route R^k in the individual IND fleet F with another vehicle at a random position.
    End for;
    Let $\hat{F}$ be the new fleet;
    If IND_i weighted net profit is improved and the routes $\hat{R}$ are feasible **then:**
      Update $\widehat{D_{ip}^k}$, $\hat{R}, \hat{G}$ and $\hat{F}$ for individual IND_i;
    End if;
  End while;
  $i = i + 1$
End while

The algorithm starts with an elite list of solutions L_e, which takes the best n_e individuals from the main population as measured by the weighted net profit (Equation 11).

Step 1 is like the primary education procedure (Section 4.3). Step 2 explores different regions of the search space using the RR operator, intensifying this area by applying the GI operator to a partially destroyed solution, like Step 2 of Algorithm 1.

8. Diversification

Efficient management of feasible solutions plays a significant role in population diversity. The performance of the HEA was improved by applying a mutation after the educational procedure. Over iterations, individuals tend to become more similar, making it challenging to avoid premature convergence. To overcome this difficulty, we introduced a new scheme to increase population diversity. The diversification stage includes two steps: regeneration and mutation.

Regeneration occurs when the maximum allowable number of iterations for regeneration it_r is reached without improvement in the value of the best solution. In this procedure, the n_e elite individuals are preserved and transferred to the next generation. The remaining $n_p - n_e$ individuals, which are ranked according to their weighted net profit (Equation 11), are subjected to the RR and GINF operators to create new individuals, like Step 2 of Algorithm 2. Only n_p new individuals are kept in the population at the end of the procedure.

The mutation procedure is applied with a probability p_m. This procedure randomly selects an individual IND_i that differs from the best solution. Two randomized structure-based ALNS operators, RR and GINF, are then used to change the positions of a specific number of nodes, which are chosen from the interval $[b_l^m, b_u^m]$ of removable nodes in the mutation procedure.

9. Tables

Table 2 – Vehicles parameters

Vehicles types	Capacity	Fixed cost	Variable cost
A	30	50	1.5
B	50	80	2.5
C	80	140	4
D	120	250	6
E	100	300	5

Table 3 – Products parameters

Products types	Capacity needed for transport	Unit holding cost in depot
P1	0.009	4.60
P2	0.005	9.10
P3	0.011	0.32
P4	0.010	8.37
P5	0.009	5.55
P6	0.014	6.08
P7	0.026	9.80
P8	0.001	5.81
P9	0.009	9.41
P10	0.0022	8.25

Table 4 – Instances for experiment

#	Points of sale	Level of products availability in depot*	Level of fleet total capacity *
1	25	Low	Low
2			Balanced
3			High
4		Balanced	Low
5			Balanced
6			High
7		High	Low
8			Balanced
9			High
10	50	Low	Low
11			Balanced
12			High
13		Balanced	Low
14			Balanced
15			High
16		High	Low
17			Balanced
18			High

* Necessary to full stock buffers

Table 5 – Fleet composition

Instances	Fleet
1, 4, 7	A^2, B^2, C^2, D^2, E^2
2, 5, 8	A^4, B^4, C^4, D^4, E^4
3, 6, 9	A^8, B^8, C^8, D^7, E^7
10, 13, 16	A^4, B^4, C^4, D^4, E^3
11, 14, 17	A^7, B^7, C^8, D^8, E^8
12, 15, 18	$A^{14}, B^{14}, C^{15}, D^{15}, E^{15}$

Table 6 – Routing results

	Fleet*			**Number of Vehicles used**		**Routes distance**	
Instance	**MIP**	**HEA**	**Vehicles available**	**MIP**	**HEA**	**MIP**	**HEA**
1	A^2, B^2, C^2, D^2, E^2	A^0, B^2, C^2, D^2, E^1	10	10	7	559,95	560,26
2	A^4, B^4, C^4, D^0, E^0	A^3, B^4, C^4, D^0, E^0	20	12	11	603,78	832,82
3	A^8, B^8, C^0, D^0, E^0	A^8, B^8, C^8, D^1, E^0	38	16	25	757,91	1255,9
4	A^2, B^2, C^2, D^2, E^2	A^2, B^2, C^2, D^2, E^2	10	10	10	564,2	686,63
5	A^4, B^4, C^4, D^4, E^2	A^4, B^4, C^4, D^4, E^2	20	18	18	1027,98	1027,98
6	A^8, B^8, C^8, D^0, E^0	A^8, B^8, C^8, D^5, E^0	38	24	29	1227,36	1535,37
7	A^2, B^2, C^2, D^2, E^2	A^2, B^2, C^2, D^2, E^2	10	10	10	564,2	674,62
8	A^4, B^4, C^4, D^4, E^2	A^3, B^4, C^4, D^4, E^1	20	18	16	1023,73	1038,96
9	A^8, B^8, C^8, D^0, E^0	A^8, B^8, C^7, D^0, E^0	38	24	23	1204,73	1331,51
10	A^4, B^4, C^4, D^4, E^2	A^4, B^4, C^3, D^4, E^2	19	18	17	2404,23	1313,01
11	A^7, B^7, C^8, D^1, E^0	A^7, B^7, C^8, D^4, E^0	38	23	26	2857,37	1472,32
12	$A^{14}, B^{14}, C^3, D^0, E^0$	$A^{14}, B^{14}, C^{12}, D^0, E$	73	31	40	3957,41	2196,42
13	A^4, B^4, C^4, D^4, E^3	A^4, B^4, C^4, D^4, E^3	19	19	19	7800	1428,05
14	A^7, B^7, C^7, D^8, E^5	A^7, B^7, C^7, D^8, E^4	38	34	33	4467,96	2162,38
15	$A^{15}, B^{13}, C^{15}, D^3, E$	$A^{15}, B^{15}, C^5, D^0, E^0$	75	46	35	5368,12	1892,15
16	A^3, B^3, C^4, D^4, E^4	A^3, B^3, C^4, D^4, E^4	18	18	18	4083,23	1307,3
17	A^8, B^8, C^8, D^8, E^3	A^7, B^8, C^8, D^8, E^3	40	35	34	5534,96	2255,98
18	$A^{15}, B^{14}, C^{15}, D^2, E$	$A^{15}, B^{15}, C^{15}, D^6, E$	75	46	51	5595,61	2786,11

*The letters represent the type of vehicle with the number of vehicles at the exponent

Table 7 – Finantial results

	Weighted Net Profit		Gross Profit		Transport Total Cost		Products Total Cost		Net Profit	
Instances	**MIP**	**HEA**	**MIP**	**HEA**	**MIP**	**HEA**	**MIP**	**HEA**	**MIP**	**HEA**
1	788703,2	788192,06	603467,35	603341,83	3410,32	3747,3	739809,83	739661,89	600057,03	599594,52
2	789484,48	788629,36	603467,35	603339,6	2629,03	3315,5	739809,83	739652,75	600838,31	600024,1
3	789653,86	782725,02	603467,35	562179,57	2459,65	5788,49	739809,83	679030,75	601007,7	556391,08
4	1057301,31	1055968,91	961369,72	961191,07	3363	4457,05	1227629,45	1227413,99	958006,72	956734,02
5	1160555,37	1159764,76	1151480,61	1151480,61	5825,41	5825,42	1487153,98	1487153,98	1145655,19	1145655,2
6	1161455,48	1155380,22	1151454,18	1110828,47	4922,23	8400,56	1487125,42	1432156,55	1146531,95	1102427,9
7	1057301,31	1055966,21	961369,72	961184,09	3363	4444,88	1227629,45	1227403,65	958006,72	956739,2
8	1160555,39	1159763,91	1151474,52	1151220,23	5825,39	6390,04	1487143,4	1486841,59	1145649,13	1144830,19
9	1161493,92	1160398,95	1151484,72	1151183,78	4886,87	5727,34	1487161,96	1486796,02	1146597,86	1145456,44
10	1630001,03	1633141,18	1229260,13	1228987,31	11129	7619,84	1505613,21	1505291,2	1218131,1	1221367,47
11	1630963,57	1629895,78	1229257,85	858676,77	10166,5	7834,28	1505603,38	1056697,38	1219091,36	850842,49
12	1630722,11	1623303,68	1229253,55	1011164,75	10408	9156,38	1505606,07	1229813,65	1218845,6	1002008,37
13	2060004,58	2079984,65	1846954,85	1846624,99	28960,9	8556,9	2357955,01	2357560,95	1817994	1838068,09
14	2376200,88	2383585,09	2309291,29	2308827,6	21513,4	13698,43	3004033,88	3003441,29	2287777,93	2295129,16
15	2377517,55	2381016,07	2309274,84	1230280,33	20198,2	7049,65	3004005,53	1588095,75	2289076,66	1223230,68
16	2081543,81	2090533,9	1860169,82	1859820,05	18206	8769,81	2372903,22	2372494,72	1841963,78	1851050,24
17	2374351,83	2383711,2	2309265,99	2308804,01	23363,9	13539,12	3003997,9	3003395,14	2285902,09	2295264,9
18	2377744,75	2376609,07	2309316,15	1923991,26	19971	14356,65	3004064,01	2508597,53	2289345,17	1909634,61

Table 8 – MIP gaps

#	Points of sale	Level of products availability in depot*	Level of fleet total capacity *	**Gap (%)**
1	25	Low	Low	0.01
2			Balanced	0.01
3			High	0.01
4		Balanced	Low	0.01
5			Balanced	0.01
6			High	0.02
7		High	Low	0.01
8			Balanced	0.01
9			High	0.01
10	50	Low	Low	0.01
11			Balanced	0.01
12			High	0.01
13		Balanced	Low	0.84
14			Balanced	0.01
15			High	0.13*
16		High	Low	0.01*
17			Balanced	0.02*
18			High	0.01

Figure 1 – Lazy constraint call-back for subtour elimination using Dantzig method

```
class noSubtourCallBack(cplex.callbacks.LazyConstraintCallback) :
    def __call__(self): #Funcao chamada pelo CPLEX
        for k in range(0, self.mp.m_vehicles):
            aux_k_route = np.full(shape=[self.mp.n_dc_clients], fill_value=-1, dtype=np.intc)
            k_route = np.full(shape=[self.mp.n_dc_clients + 1], fill_value=-1, dtype=np.intc)
            are_there_route = False
            for i in self.customers_:
                for j in self.customers_:
                    if i != j:
                        if int(0.5 + self.get_values("x_{0}_{1}_{2}".format(k, i, j))) == 1:
                            aux_k_route[i] = j
                            are_there_route = True
            nex = -1
            k_route = []
            for i in self.customers_:
                if len(k_route) == 0:
                    if aux_k_route[i] > -1:
                        while True:
                            if nex == -1:
                                nex = aux_k_route[i]
                            else:
                                nex = aux_k_route[nex]

                            if nex == 0:
                                k_route = []
                                nex = -1
                                break
                            elif nex in k_route:
                                break
                            else:
                                k_route.append(nex)
                else:
                    break

            if len(k_route) > 1:
                print('sub tour', time.time())
                SN = [i for i in aux_k_route if i not in k_route and i != -1]

                _var = ["x_{0}_{1}_{2}".format(k, i, j) for i in k_route for j in SN if i != j and i != -1 and j != -1]
                _coef = [1 for i in k_route for j in SN if i != j and i != -1 and j != -1]
                self.add(cplex.SparsePair(ind=_var, val=_coef), rhs=1.0, sense="G")

                _var = ["x_{0}_{1}_{2}".format(k, i, j) for i in k_route for j in k_route if i != j]
                _coef = [1 for i in k_route for j in k_route if i != j]
                # Dantzig subtour constraint
                self.add(cplex.SparsePair(ind=_var, val=_coef), rhs=len(k_route) - 1, sense="L")
```