

# Modelos matemáticos para programação do sistema *cross docking* com múltiplas docas

## *Mathematical models for multi-dock cross docking system programming*

Versão do autor aceita publicada online: 30 out. 2021

Publicado online: 23 nov. 2021

**Como citar esse artigo - American Psychological Association (APA):** Santos, L. G., & Fuchigami, H. Y. (2021). Modelos matemáticos para programação do sistema *cross docking* com múltiplas docas. *Exacta*. DOI: <https://doi.org/10.5585/exactaep.2021.19561>.

**Lorrany Guilherme Santos**

[lorrany00@hotmail.com](mailto:lorrany00@hotmail.com)

<https://orcid.org/0000-0001-6770-4393>

Universidade Federal de Goiás

Pesquisadora da Área de Pesquisa Operacional

**Hélio Yochihiro Fuchigami**

[heliofuchigami@ufg.br](mailto:heliofuchigami@ufg.br)

Universidade Federal de Goiás

Pesquisa Operacional

**Resumo:** Este trabalho foi motivado pela necessidade de contribuir com a distribuição eficiente de produtos ao longo da cadeia de suprimentos e propor uma ferramenta de gestão capaz de reduzir o elevado tempo de fluxo dos veículos nos centros de distribuição das empresas. Portanto, esse artigo se propõe principalmente a criar modelos que sequenciem de forma eficiente a entrada e saída de veículos do centro *cross docking*, consolidando as principais restrições não consideradas de forma conjunta anteriormente e minimizando o resultado de um dos principais indicadores nas organizações logísticas: o tempo de permanência dos veículos. Com o intuito de melhorar a eficiência computacional dos modelos propostos, realiza-se de forma complementar o estudo das variáveis e parâmetros envolvidos na programação linear inteira mista propondo opções para a redução do tempo de execução. Os modelos apresentados

são baseados em variáveis de alocação considerando restrições de múltiplas docas. A experimentação computacional foi realizada por meio de um conjunto de 160 problemas-testes e os resultados evidenciaram que a diferença entre os veículos de carga e descarga alocados no dia consegue melhor explicar a variação dos resultados do que as atribuições de variáveis isoladas.

**Palavras chave:** *Cross docking. Flow shop* híbrido. Sequenciamento de caminhões. Programação linear inteira mista, scheduling.

**Abstract:** *This work was motivated by the need to contribute to the efficient distribution of products throughout the supply chain and to propose a management tool capable of reducing the high flow time of vehicles in the companies' distribution centers. Therefore, this article mainly proposes to create models that efficiently sequence the entry and exit of vehicles from the cross docking center, consolidating the main restrictions not considered together previously and minimizing the result of one of the main indicators in logistic organizations: o time of permanence of vehicles. In order to improve the computational efficiency of the proposed models, the study of variables and parameters involved in mixed integer linear programming is carried out, proposing options to reduce execution time. The models presented are based on allocation variables considering multiple dock constraints. The computational experimentation was performed using a set of 160 test-problems and the results showed that the difference between loading and unloading vehicles allocated on the day can better explain the variation in the results than the attributions of isolated variables.*

**Palavras chave:** *Cross docking. Hybrid flow shop. Truck sequencing. Mixed integer linear programming, scheduling.*

## 1 Introdução

Em décadas anteriores, o tempo que uma empresa necessitava para processar e entregar um pedido variava de 25 a 30 dias e os erros nas entregas eram frequentes, o que gerava uma maior necessidade de estoque em todos os elos da cadeia de suprimentos: fabricantes, atacadistas, distribuidores e varejistas (Bowersox, Closs & Cooper, 2006).

Já hoje, a exigência no desempenho das entregas está cada vez mais acirrada, portanto os gerentes da cadeia de suprimentos em todo o mundo estão constantemente buscando métodos promissores para melhorar a eficiência da distribuição de produtos e atender a alguns objetivos importantes das cadeias de suprimentos, incluindo a minimização de custos; melhoria da satisfação do cliente; utilização eficiente de recursos; maximização de receita; e criação de valor

(Dulebenets, 2018a; Felfel, Ayadi, & Masmoudi, 2016; Luo, Yang, & Wang, 2019; Nogueira, Coutinho, Ribeiro, & Ravetti, 2020).

Além das pressões por agilidade, redução do tempo ocioso, do custo com operações fiscais, de custos com deslocamento, estadia e diária dos transportadores, existe a pressão legal, como no caso da Lei 13.103/2015, que estabelece que o prazo de carga e descarga deve ser de no máximo 5 horas, contados da chegada do veículo ao seu endereço de destino (Brasil, 2015). A indenização pelo excesso de tempo em carga/descarga atualmente é de R\$ 1,60 por tonelada/hora da capacidade total do veículo.

Portanto, faz-se necessário desenvolver métodos inovadores para chegar aos resultados esperados, uma vez que os modelos tradicionais de armazenagem e distribuição não são satisfatórios com relação ao tempo de entrega e aos custos operacionais. Deste modo, uma estratégia de armazenagem que tem um grande potencial para controlar os custos logísticos e de distribuição, mantendo ao mesmo tempo um nível de serviço satisfatório ao cliente é o *cross docking* (Apte & Viswanathan, 2000).

O *cross docking* consiste em transferir imediatamente a carga recebida, sem nenhum tipo de armazenagem ou *picking* (separação dos pedidos). Desta forma, a redução dos estoques diminui os custos operacionais e o espaço utilizado no centro de distribuição (CD). Além disso, a inexistência de estoque gera entregas mais frequentes dos produtos, que satisfaz os clientes que também precisarão de estoque reduzido dos seus produtos, pois depositam confiança nas entregas constantes (Yu & Egbelu, 2008).

A implementação do *cross docking* é uma alternativa viável para a maior parte das empresas que já trabalham com os CDs tradicionais, isso porque normalmente não é necessário realizar grandes investimentos e nem alterações de infraestruturas na adaptação. Ao invés disso, envolve apenas um investimento em tempo e energia na melhoria e mudança dos processos.

Apesar das inúmeras vantagens oferecidas pelo sistema, a inexistência de estoque revela os erros associados às operações, por esse motivo é necessário ter uma precisão na programação dos veículos que são carregados e descarregados no CD. Diante disso, o objetivo geral da pesquisa foi elaborar dois modelos matemáticos de sequenciamento de veículos, considerando a atividade de carga e descarga de mercadorias em múltiplas docas.

Optou-se por modelar o sistema como um problema de *flow shop* híbrido com dois estágios de produção e  $m$  máquinas paralelas em cada um desses estágios ( $m > 1$ ). No entanto, busca-se extrapolar as barreiras dos métodos tradicionais e do sequenciamento das atividades de forma

isolada para preencher as lacunas do conhecimento aplicado. Por esse motivo, o intuito passa a ser: compreender como as definições clássicas podem ser aplicadas não apenas nas áreas da manufatura, onde a maior parte dos trabalhos nesse sentido são desenvolvidos, mas também na distribuição de mercadorias, ou seja, nas áreas logísticas das organizações.

Na comparação entre os problemas, os dois estágios do *flow shop* híbrido são análogos no *cross docking*. Os dois estágios de produção correspondem às etapas de recebimento de mercadorias vindas de grandes indústrias e a expedição para o centro urbano. Já as máquinas disponíveis em cada estágio se trata do número de docas para realizar cada uma dessas operações. Por fim, o número de tarefas é representado pela quantidade de veículos a serem carregados e descarregados. Outros parâmetros não correlacionados ao modelo de *flow shop* serão definidos posteriormente.

A partir do modelo matemático desenvolvido, realizou-se um estudo da eficiência computacional do modelo apresentado. Essa análise se deu-se primeiramente através do número de variáveis e restrições envolvidas nas propostas, da qual esperava-se constatar que problemas menores, ou seja, com menos variáveis e menos restrições tenha melhor eficiência computacional, no entanto essa hipótese foi testada experimentalmente através de uma extensa base de dados que varia conforme o número de caminhões de entrada e saída, número de docas, tipos de produtos e distribuição de demanda nos caminhões, afinal, cada modelo tem sua peculiaridade e outras condições de complexidade podem ter um peso maior sobre o resultado final. Na sequência, o trabalho foi organizado da seguinte forma: a seção 2 contém a revisão da literatura, subdividida em publicações abordando *cross docking* e análise bibliográfica quantitativa de trabalhos desenvolvidos nos últimos anos. A seção 3 apresenta a descrição dos problemas e as formulações propostas. A seção 4, descreve a experimentação computacional e os resultados obtidos. Por fim, a seção 5 traz as conclusões do trabalho, com a análise crítica do que foi proposto e os próximos passos da pesquisa.

## **2 Revisão da literatura**

### *2.1 Publicações abordando cross docking*

Nesta seção, estão presentes os principais resultados que fornecem a fundamentação teórica no assunto e quais são as lacunas no conhecimento e, principalmente, qual a contribuição teórica da pesquisa proposta. Esse embasamento é guiado pela combinação da busca automática e manual de trabalhos com alto nível de contribuição para o desenvolvimento de novas

ferramentas e metodologias que solucionem o problema proposto de sequenciamento de caminhões no centro *cross docking*.

O trabalho de Yu e Egbelu (2008) é uma referência para a compreensão da logística e gestão da cadeia de abastecimento, eles consideram que a melhor maneira de reduzir custos e melhorar a eficiência não é simplesmente aperfeiçoar uma operação, mas eliminá-la, se possível. Tendo o *cross docking* esse potencial de eliminar as operações mais caras, justificam-se os esforços para encontrar a melhor sequência de agendamento para caminhões responsáveis pela entrada e saída de produtos, visando minimizar o tempo total de operação. Apesar de ter grande contribuição no desenvolvimento de técnicas de modelagem para o *cross docking*, o cenário da pesquisa de Yu e Egbelu (2008) não considera operações internas do armazém, como a classificação e distribuição de mercadorias. Não considerar esse tipo de restrição representa um importante desafio para a construção de um cenário mais realista.

Posteriormente, Yan (2014) desenvolve um trabalho que apresenta semelhanças com o trabalho de Yu e Egbelu (2008) com relação ao conceito estabelecido de *cross docking* e a opção de não considerar operações internas do armazém, mas vai além ao explorar quatro algoritmos heurísticos focados em solucionar os problemas mais complexos, em que não se obtém a solução ótima, porém encontram soluções de alta qualidade. No entanto, a maior limitação desse trabalho para torná-lo mais realista é não considerar múltiplas docas de recebimento e expedição.

Seguindo uma sequência cronológica de trabalho relevantes para o caso, é válido mencionar o trabalho de Fonseca, Nogueira e Ravetti (2019), cujo problema foi inicialmente modelado como um *flow shop* de duas máquinas com restrições de precedência, com o objetivo de minimizar o *makespan*, e foi posteriormente generalizado para o caso de *cross docking*. A formulação proposta foi baseada na combinação do problema de Chen e Lee (2009) e Lima (2014). A contribuição de Fonseca, Nogueira e Ravetti (2019) está na metodologia utilizada para aumentar a eficiência de problemas com grandes conjuntos de dados, um método híbrido baseado em uma técnica de relaxação Lagrangiana através do algoritmo de volume. Apesar do centro *cross docking* considerado pelos autores adotar, assim como no trabalho de Yan (2014), apenas uma doca exclusiva para cada etapa de carga e descarga, o procedimento merece destaque por gerar boas soluções viáveis para o cenário traçado.

No ano seguinte, Shahmardan e Sajadieh (2020) utilizaram-se da programação linear inteira mista (PLIM) para encontrar a doca ideal com o objetivo de reduzir o *makespan*. O cenário

abordado nesse trabalho permite que os caminhões utilizados para as operações de carregamento sejam os mesmos que os caminhões de descarregamento, com isso não é necessária a descarga total desse veículo. Essa restrição é fundamentalmente importante para centros de distribuição menores, que contam frequentemente com essa limitação de disponibilidade de veículos. Apesar dessas considerações tornarem mais complexa a resolução, Shahmardan e Sajadieh (2020) conseguiram mostrar a vantagem para a função objetivo proposta com a possibilidade de descarga parcial. No entanto, ao considerar que o número máximo de caminhões é o mesmo que o número de docas, inviabiliza a utilização desses resultados para grandes centros logísticos.

Neste mesmo ano, Zheng, et al (2020) realizaram um trabalho que insere os fatores de capacidade limitada de armazenamento, de caminhões e de áreas de operacionalização interna do armazém. Além disso, esse trabalho considera como função objetivo custos operacionais totais, incluindo multa por desrespeito às janelas de tempo dos veículos e multa por atraso nas entregas de produtos. Quanto às limitações dessa pesquisa, pode-se citar o fato de não existirem restrições que objetivem diretamente reduzir o estoque intermediário, uma vez que, além de ser um dos principais objetivos do *cross docking*, pode ainda ser preocupante para armazéns de produtos perecíveis, como é o caso da cadeia de frios em que foi desenvolvido o trabalho de Zheng et al (2020).

Por fim, os trabalhos mais recentes, como Gaudioso et al (2021) e Gunawan et al (2021) utilizam estratégias heurísticas para melhorar a eficiência computacional dos modelos de programação linear inteira. Gaudioso et al (2021) utiliza a técnica de decomposição Lagrangeana para o Sistema *cross docking* com única doca e Gunawan et al (2021) utiliza uma meta-heurística de geração de colunas para minimizar os custos de transportes e operacionais de um problema que considera a integração entre o roteamento de veículos e o sequenciamento *cross-docking*.

Portanto nesse trabalho, os autores focaram esforços para desenvolver formulações que consideram um cenário com múltiplas docas, que incluem restrições para redução do estoque intermediário e as restrições relacionadas as operações internas do armazém, como a classificação e distribuição de mercadorias. Desta forma, consolidando as limitações existentes nas principais obras que abordam o *cross docking* e que foram detalhadas nessa seção.

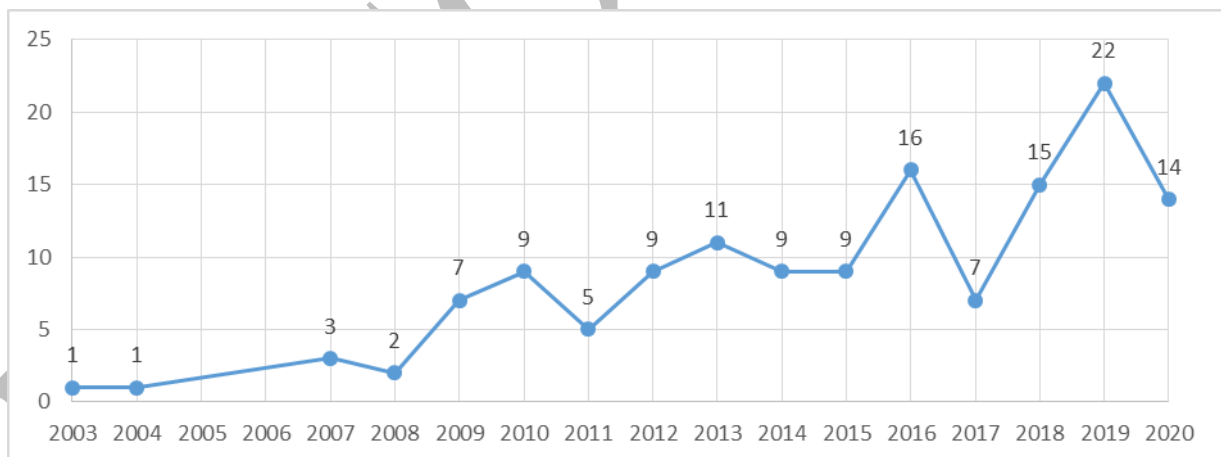
## 2.2 Análise bibliográfica quantitativa

A fim de realizar uma pesquisa quantitativa na literatura para mensurar o desenvolvimento de trabalhos correlatos, foi realizada uma busca dos termos “*cross docking*” e “*scheduling*” na plataforma *Scopus*, obtendo-se um aglomerado de informações em relação a como o estudo vem se desenvolvendo com referência ao tema proposto. Na amostra de 190 trabalhos encontrados inicialmente, foram utilizados filtros para retirada de trabalhos como: duplicados; resumos de congressos; revisões de literatura, problemas de *cross docking* ligados a roteamento de veículos e outros arquivos não relacionados ao escopo geral do projeto. Foram mantidos apenas artigos com modelagem, algoritmos e heurísticas de problemas aplicados

Com essa filtragem chegou-se ao número de 140 artigos, e com eles é possível iniciar uma análise de dados, a qual percebe-se que a quantidade de trabalhos encontrados em revistas corresponde a 74% do total e os trabalhos de conferências são apenas 26%.

Além disso, a Figura 1, destaca na amostra o número de publicações desenvolvidas em decorrer os anos, tendo-se o primeiro artigo publicado no ano de 2003. A partir de então, nota-se uma tímida expressão de autores na área, contudo, o tema mostrou ascendência expressiva a partir do ano de 2009, seguida de uma relevante queda em 2011 e 2017, chegando ao ápice em 2019 com 22 publicações.

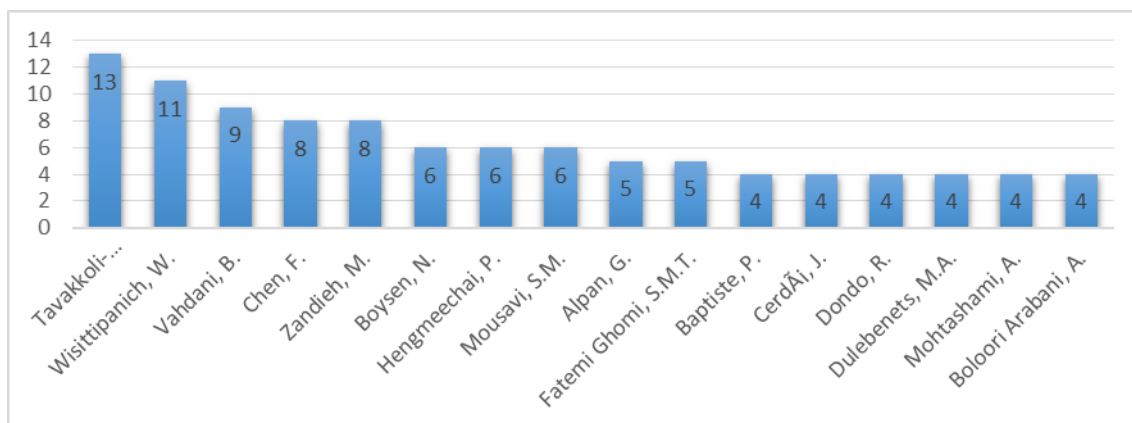
Figura 1 – Número de publicações ao longo dos anos



Fonte: elaborado pelos autores

Ao mencionar o número de publicações, faz-se necessário apresentar ainda os autores com maior contribuição na área, ou seja, com maior número de artigos publicados. Nesta análise, apresentam-se os autores com pelo menos quatro trabalhos publicados, não necessariamente o primeiro autor na publicação. Dito isso, foi possível construir a Figura 2, destacando Tavakkoli-Moghaddam, R. com 13 publicações, sendo o principal contribuinte na área.

Figura 2 – Número de artigos por autor

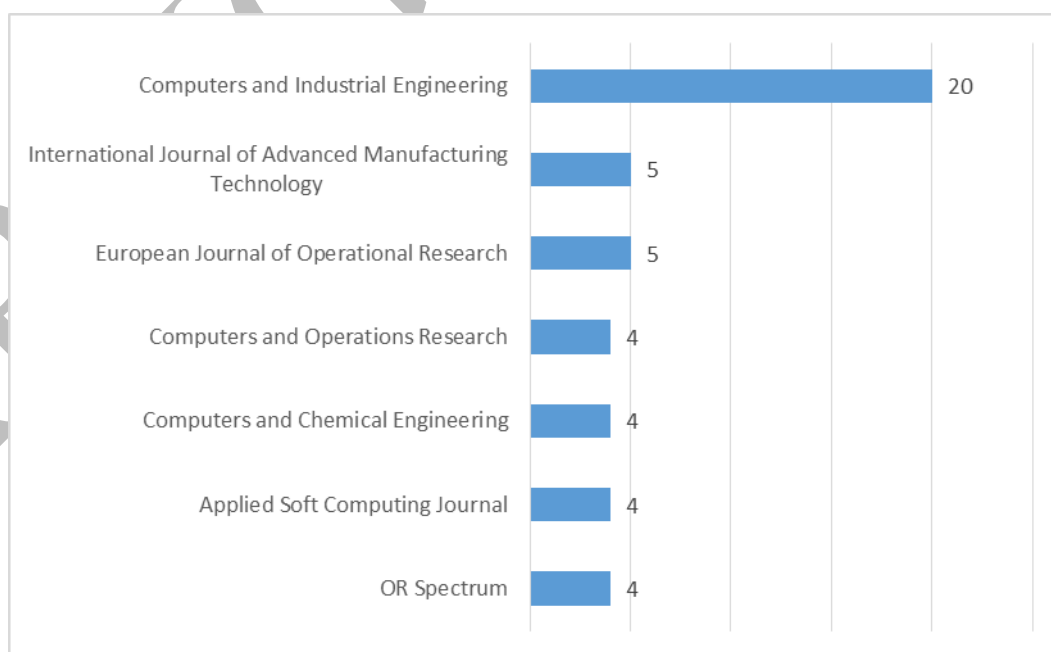


Fonte: elaborado pelos autores

Em relação aos países que possuem publicações pertinentes ao assunto, destaca-se o Iran com 38 trabalhos publicados, China e Estados Unidos com 24 publicações cada um deles, França com 14, Tailândia com 11.

Ainda, pode-se apresentar na Figura 3 as principais revistas e que, portanto, apresentaram maior número de publicações sobre o tema desta pesquisa. Destacando a revista *Computers and Industrial Engineering*, que apresentou 20 artigos na área, seguida das revistas *International Journal of Advanced Manufacturing Technology* e *European Journal of Operational Research*, com cinco artigos cada.

Figura 3 – Número de publicações por revista



Fonte: elaborado pelos autores



Por último, considerando as referências presentes no contexto da amostra, foca-se no montante de citações desenvolvidas ao longo dos anos, desde a data de publicação, de cada trabalho. O número de citações é um indicador de quão bem aceito e relevante tem sido o trabalho para o meio acadêmico, desta forma, pode-se destacar o artigo de Yu e Egbelu (2008), com 207 citações, sendo responsável por quase vinte e cinco por cento do montante de citações apresentadas na Tabela 1.

Tabela 1 – Número de citações por publicação

Autor	Ano	Citações	Principal abordagem
Yu, W.; Egbelu, P.J.	2008	207	Os autores realizaram um estudo para encontrar o melhor sequenciamento de veículos nas docas. Para isso propuseram um modelo matemático para redução do <i>makespan</i> e um algoritmo heurístico que diminuísse a quantidade de produtos que passam pelo armazenamento temporário.
Vahdani, B.; Zandieh, M.	2010	130	Esse trabalho baseou-se e superou os resultados do artigo de Yu e Egbelu (2008) aplicando 5 algoritmos meta-heurísticos para programar os caminhões nas docas, com objetivo de reduzir a quantidade de produtos do armazenamento temporário.
Boysen, N.; Fliedner, M.;Scholl, A.	2010	113	Considera uma simplificação do cenário <i>cross docking</i> para propor o sequenciamento de caminhões, visando alcançar melhorias para o cenário real.
Boysen, N.	2010	106	Esse trabalho propõe soluções exatas e heurísticas para o sequenciamento de caminhões com múltiplos objetivos: tempo de fluxo, o tempo de processamento e o atraso dos caminhões de saída
Chen, F.; Lee, C.-Y.	2009	105	Este artigo estuda o <i>cross docking</i> , fazendo referência e analogia ao <i>flow shop</i> de duas máquinas.
Chen, F.; Song, K.	2009	88	Um algoritmo de programação inteira mista para sequenciar caminhões <i>cross docking</i> em instâncias de pequeno porte e quatro heurísticas para instâncias de médio e grande porte.
Arabani, A.B.; Ghomi, S.F.; Zandieh, M.	2011	87	Os autores propõem cinco meta-heurísticas para serem comparadas com as propostas de minimização de <i>makespan</i> de Yu e Egbelu (2008).
Soltani, R.; Sadjadi, S.J.	2010	69	Apresenta duas meta-heurísticas para programar caminhões do sistema <i>cross docking</i> visando minimizar o tempo total de fluxo do sistema. Além disso, é considerado que todos os caminhões estão disponíveis para programação no instante zero e caminhões de saída tem capacidade infinita.

Larbi, R.; Alpan, G.; Baptiste, P.;Penz, B.	2011	61	O artigo exhibe algoritmos de sequenciamento dos caminhões <i>cross docking</i> de forma a minimizar a soma dos custos operacionais, considerando 3 diferentes políticas de informações sobre a ordem de chegadas e o conteúdo de todos os caminhões de descarga (completa, parcial e nenhuma).
Forouharfard, S.; Zandieh, M.	2010	58	O trabalho utiliza do algoritmo competitivo imperialista para sequenciar veículos no sistema <i>cross docking</i> de forma a minimizar o estoque temporário.

Fonte: elaborado pelos autores

### 3 Descrição do problema e formulação

O problema tratado pode ser definido genericamente da seguinte forma. Considere um conjunto de  $n$  caminhões de entrada e  $m$  caminhões de saída como atividades que devem ser processadas nas  $2k$  docas, sendo  $k$  o número de docas dedicadas para as atividades de descarga dos  $n$  caminhões de entrada e outras  $k$  docas exclusivas para as atividades de carregamento dos  $m$  caminhões de saída. Cada caminhão de entrada descarrega no centro *cross docking*  $p$  diferentes tipos de produtos e cada caminhão de saída solicita para o carregamento outros  $p$  diferentes tipos de produtos.

Os caminhões de entrada  $i$  abastecem o sistema *cross docking* com uma quantidade  $A_{pi}$  de produtos do tipo  $p$  e os caminhões de saída  $o$  têm uma demanda por  $B_{po}$  produtos do tipo  $p$ . O intervalo de tempo destinado para realizar as operações de carga ou descarga, somado ao *setup* entre as atividades de qualquer caminhão e independentemente da doca é denominado *slot t*. As atividades não podem ser interrompidas e todos os caminhões de entrada estão disponíveis para processamento no instante zero da programação. Já os caminhões de saída apenas podem ser processados quando todos os produtos  $B_{po}$  solicitados estiverem disponíveis no centro *cross docking*, ou seja, foram previamente descarregados dos caminhões de entrada. A solução consiste na elaboração de uma programação que minimize o tempo total de fluxo, ou seja, a soma dos *slots t* atribuído aos caminhões de saída (aqueles que deixaram o sistema *cross docking*).

Para a solução do problema foram propostos dois modelos matemáticos: ambos baseados em variáveis de alocação, sendo que o primeiro conta com dois tipos de variáveis de decisão e o segundo adiciona um terceiro tipo, com o objetivo de reduzir a relaxação do modelo. A relaxação do modelo ocorre pelo não conhecimento prévio do *slot t* máximo, portanto em busca

de um melhor resultado, o valor de  $t$  foi considerado através da situação mais pessimista em que a demanda do primeiro caminhão de saída fosse suprida apenas pelo último caminhão de entrada, nessa situação primeiramente todos os caminhões de entrada seriam descarregados, utilizando um número  $n/k$  slots inteiros para essa operação. Posteriormente, os caminhões de saída seriam carregados demandando outros  $n/k$  slots de tempo. Esse e outros parâmetros, as variáveis e os modelos propostos são apresentados a seguir.

- **Parâmetros:**

$i$ : índice do caminhão de entrada	$i = 0, \dots, n$
$o$ : índice do caminhão de saída	$o = 0, \dots, m$
$k$ : índice das docas	$k = 1, \dots, K$
$t$ : índice do slot de tempo	$t = 1, \dots, T$
$A_{pi}$ : quantidade de produtos do tipo $p$ no caminhão de entrada $i$	$p = 1, \dots, P,$ $i = 1, \dots, n$
$B_{po}$ : quantidade de produtos do tipo $p$ no caminhão de saída $o$	$p = 1, \dots, P,$ $o = 1, \dots, m$
$n$ : número de caminhões de entrada	
$m$ : número de caminhões de saída	
$P$ : quantidade de tipos de produtos	
$K$ : número de docas	
$T$ : número máximos de slots, considerado $\frac{(n+m)}{K}+1$	

### 3.1 Modelo baseado em variáveis de alocação

- **Variáveis de decisão:**

$$X_{ikt} \begin{cases} 1 \text{ se o caminhão de entrada } i \text{ está alocado na doca } k \text{ no slot } t \\ 0, \text{ caso contrario} \end{cases}$$

$$Y_{okt} \begin{cases} 1 \text{ se o caminhão de saída } o \text{ está alocado na doca } k \text{ no slot } t \\ 0, \text{ caso contrario} \end{cases}$$

$$C_o \geq 0$$

$$o = 0, \dots, m$$

• **Modelo matemático:**

$$\text{Min } Z = \sum_{o=1}^m C_o \quad (1)$$

$$C_o = \sum_{k=1}^K \sum_{t=1}^T (Y_{okt}) \cdot t \quad o = 1, \dots, m \quad (2)$$

$$\sum_{k=1}^K \sum_{t=1}^T X_{ikt} = 1 \quad i = 1, \dots, n \quad (3)$$

$$\sum_{k=1}^K \sum_{t=1}^T Y_{okt} = 1 \quad o = 1, \dots, m \quad (4)$$

$$\sum_{i=1}^n X_{ikt} \leq 1 \quad k = 1, \dots, K, t = 1, \dots, T \quad (5)$$

$$\sum_{o=1}^m Y_{okt} \leq 1 \quad k = 1, \dots, K, t = 1, \dots, T \quad (6)$$

$$\sum_{h=1}^t \sum_{k=1}^K \sum_{i=1}^n (X_{ikh}) \cdot (A_{pi}) \geq \sum_{h=1}^t \sum_{k=1}^K \sum_{o=1}^m (Y_{okh}) \cdot (B_{po}) \quad t = 1, \dots, T, p = 1, \dots, P \quad (7)$$

$$X_{ikt} \in \{0,1\} \quad i = 1, \dots, n, k = 1, \dots, K, t = 1, \dots, T \quad (8)$$

$$Y_{okt} \in \{0,1\} \quad o = 1, \dots, m, k = 1, \dots, K, t = 1, \dots, T \quad (9)$$

A função objetivo (1) minimiza o tempo de fluxo, ou seja, a soma dos instantes de saída dos caminhões. As expressões (2) calculam o valor do tempo de fluxo. As restrições (3) e (4) asseguram que cada caminhão deve ser obrigatoriamente alocado em um *slot* e em uma doca. As restrições (5) e (6) reafirmam que em cada *slot* e em cada doca só pode existir um caminhão alocado. As restrições (7) garantem que um caminhão de saída só pode ser atribuído a um *slot*  $t$  se toda a demanda exigida desse caminhão de saída pode ser satisfeita pelos produtos remanescentes no estoque temporário, que são todos os produtos enviados por caminhões de entrada anteriores reduzindo os produtos que já foram entregues pelos caminhões de saída. As expressões (8) e (9) definem o domínio das variáveis do modelo.

O tamanho problema em relação ao modelo 1, em termos do número e tipo das variáveis e das restrições, é apresentado na Tabela 2.

Tabela 2- Tamanho do problema em relação ao modelo em termos do número de variáveis e de restrições

Variáveis	Binárias	$KT(m+n)$
	Inteiras	$m$
	<b>Total</b>	<b><math>KT(n+m)+m</math></b>
Restrições	(2)	$m$
	(3)	$n$
	(4)	$m$
	(5)	$KT$
	(6)	$KT$
	(7)	$TP$
	<b>Total</b>	<b><math>2KT + TP + 2m + n</math></b>

Fonte: elaborado pelos autores

O modelo proposto possui  $KT(m+n)$  variáveis binárias,  $m$  variáveis inteiras e  $KT(n+m)+m$  restrições. Assim, uma base de dados com  $n= 5$  caminhões de entrada e  $m = 7$  caminhões de saída,  $K = 3$  docas,  $T = 4$  e  $P = 5$  terá 151 variáveis e 63 restrições; já uma instância  $n = 20$  caminhões de entrada e  $m = 20$  caminhões de saída,  $K = 5$  docas,  $T=11$  e  $P= 10$  terá 2220 variáveis e 280 restrições

### 3.2 Modelo baseado em variáveis de alocação adaptado

- **Variáveis de decisão:**

$$X_{ikt} \begin{cases} 1 \text{ se o caminhão de entrada } i \text{ está alocado na doca } k \text{ no slot } t \\ 0, \text{ caso contrario} \end{cases}$$

$$Y_{okt} \begin{cases} 1 \text{ se o caminhão de saída } o \text{ está alocado na doca } k \text{ no slot } t \\ 0, \text{ caso contrario} \end{cases}$$

$C_o \geq 0$ , término da operação de carregamento do caminhão de saída  $o$ .  $o = 0, \dots, m$

$g_{itlp} \geq 0$ , quantidade de produtos do tipo  $p$  descarregado do caminhão  $i$ , no slot tempo  $t$  e transferido para o portão de saída no slot de tempo  $l$ .

$$i= 1, \dots, n, \quad t = 1, \dots, T, \quad l= 1, \dots, T, \quad p=1, \dots, P$$

- **Modelo matemático:**

$$\text{Min } Z = \sum_{o=1}^m C_o \quad (10)$$

$$C_o = \sum_{k=1}^K \sum_{t=1}^T (Y_{okt}) \cdot t \quad o = 1, \dots, m \quad (11)$$

$$\sum_{k=1}^K \sum_{t=1}^T X_{ikt} = 1 \quad i = 1, \dots, n \quad (12)$$

$$\sum_{k=1}^K \sum_{t=1}^T Y_{okt} = 1 \quad o = 1, \dots, m \quad (13)$$

$$\sum_{i=1}^n X_{ikt} \leq 1 \quad k = 1, \dots, K, t = 1, \dots, T \quad (14)$$

$$\sum_{o=1}^m Y_{okt} \leq 1 \quad k = 1, \dots, K, t = 1, \dots, T \quad (15)$$

$$\sum_{l=t}^T g_{itlp} \geq \sum_{k=1}^K (X_{ikt}) \cdot (A_{pi}) \quad i = 1, \dots, n, t = 1, \dots, T, p = 1, \dots, P \quad (16)$$

$$\sum_{i=1}^n \sum_{t=1}^l g_{itlp} \geq \sum_{k=1}^K (Y_{okl}) \cdot (B_{po}) \quad o = 1, \dots, m, l = 1, \dots, T, p = 1, \dots, P \quad (17)$$

$$X_{ikt} \in \{0,1\} \quad i = 1, \dots, n, k = 1, \dots, K, t = 1, \dots, T \quad (18)$$

$$Y_{okt} \in \{0,1\} \quad o = 1, \dots, m, k = 1, \dots, K, t = 1, \dots, T \quad (19)$$

As equações de (10) a (15) são equivalentes às equações de (1) a (6). As restrições (16) e (17) substituem a equação (7). O conjunto de restrições (16) garantem que todos os produtos descarregados na doca  $k$  no  $slot$  de tempo  $t$  serão encaminhados para os portões de saída nos instantes  $l$  posterior ao instante  $t$ . O conjunto de restrições (17) por sua vez asseguram que todos os produtos que foram carregados nos caminhos de saída foram previamente encaminhados para os portões de saída. As expressões (18) e (19) definem o domínio das variáveis do modelo.

O tamanho problema em relação ao modelo 2, em termos do número e tipo das variáveis e das restrições, é apresentado na Tabela 3.

Tabela 3- Tamanho do problema em relação ao modelo em termos do número de variáveis e de restrições

Variáveis	Binárias	$KT(m+n)$
	Inteiras	$nT^2P+m$
	<b>Total</b>	<b><math>KT(m+n) + nT^2P+m</math></b>
	(11)	$m$
	(12)	$n$
	(13)	$m$
Restrições	(14)	$KT$
	(15)	$KT$
	(16)	$nTP$
	(17)	$mTP$
	<b>Total</b>	<b><math>TP(n+m)+2m+2KT+n</math></b>

Fonte: elaborado pelos autores

O modelo adaptado não possui alterações no número de variáveis binárias, continuando com  $KT(m+n)$  variáveis desse tipo. No entanto, o número de variáveis inteiras passa para  $nT^2P+m$  e  $TP(n+m)+2m+2KT+n$  restrições. Assim, uma base de dados com  $n = 5$  caminhões de entrada e  $m = 7$  caminhões de saída,  $K = 3$  docas,  $T = 4$  e  $P = 5$  terá 551 variáveis e 283 restrições; já uma instância  $n = 20$  caminhões de entrada e  $m = 20$  caminhões de saída,  $K = 5$  docas,  $T = 11$  e  $P = 10$  terá 26420 variáveis e 4570 restrições

#### 4 Experimentação computacional

##### 4.1 Definição da amostragem

Para efeito de análise de desempenho do modelo proposto, avaliaram-se na experimentação computacional 160 problemas-testes, formados pela combinação do número de caminhões entrada, número de caminhões de saída, quantidade de docas, número de tipos diferentes de produtos, uma matriz de distribuição de ofertas de produtos de cada tipo em cada caminhão de entrada e outra matriz de demanda de cada tipo de produto em cada caminhão de saída. Assim, 2 (alternativas de número de caminhões de entrada) x 2 (alternativas de caminhões de saída) x 2 (alternativa de quantidade de docas) x 2 (alternativa de quantidade de tipos de produtos) x 10 (réplicas para cada combinação) = 160 problemas.

Os valores para o número de caminhões de entrada e saída, número de docas, e tipos de produtos respectivamente, foram:  $n = \{20,25\}$ ,  $m = \{20,30\}$ ,  $k = \{5,10\}$  e  $P = \{10,15\}$ . As quantidades demandadas e ofertadas por cada caminhão foram geradas no intervalo uniforme  $U[1,150]$ .

O modelo gerador de exemplares, desenvolvido pelos próprios autores está descrito abaixo:

$$Aaux_{pi} = U [1;100] \quad p = 1, \dots, P \quad i = 1, \dots, n \quad (18)$$

$$Baux_{po} = U [1;100] \quad p = 1, \dots, P \quad o = 1, \dots, m \quad (19)$$

$$Asum_p = \sum_{i=1}^n Aaux_{pi} \quad p = 1, \dots, P \quad (20)$$

$$Bsum_p = \sum_{o=1}^m Baux_{po} \quad p = 1, \dots, P \quad (21)$$

$$\text{Se } Asum_p > Bsum_p$$

$$B_{po} = Baux_{po} + \left( \frac{Asum_p - Bsum_p}{m} \right) \quad o = 1, \dots, m \quad p = 1, \dots, P \quad (22)$$

$$A_{pi} = Aaux_{pi} \quad i = 1, \dots, n \quad p = 1, \dots, P \quad (23)$$

$$\text{Se } Bsum_p > Asum_p$$

$$A_{pi} = Aaux_{pi} + \left( \frac{Bsum_p - Asum_p}{n} \right) \quad i = 1, \dots, n \quad p=1, \dots, P \quad (24)$$

$$B_{po} = Baux_{po} \quad o = 1, \dots, m \quad p=1, \dots, P \quad (25)$$

As equações (18) e (19) são responsáveis por realizar o sorteio de valores seguindo uma distribuição uniforme, com números variando de 0 a 100, para cada elemento temporário das matrizes  $Aaux_{pi}$  e  $Baux_{po}$ . O grupo de equações (20) realiza a soma de todas as linhas da matriz  $Aaux_{pi}$ , ou seja, soma a quantidade total de produtos ofertados em todos os caminhões de entrada. O conjunto de equações (21) realiza a soma de todas as linhas da matriz  $Baux_{po}$ , a soma da quantidade total de produtos demandados por todos os caminhões de saída.

As equações (22) e (23) apenas acontecerão se a condicional de que  $Asum_p > Bsum_p$  for verdadeira, significando que existem mais produtos de um tipo  $p$  sendo ofertado pelos caminhões de entrada do que demandados pelos caminhões de saída, neste caso a matriz  $B_{po}$  definitiva será definida pela equação (22), onde cada elemento da nova matriz será a soma do elemento equivalente na matriz  $Baux_{po}$  com a média da diferença entre as quantidades ofertadas e demandadas de cada produto nas matrizes  $Aaux_{pi}$  e  $Baux_{po}$ .

Já a matriz  $A_{pi}$  definitiva é definida pela equação (23), sendo composta pelos mesmos elementos da matriz  $Aaux_{pi}$ . Para a condicional contrária,  $Bsum_p > Asum_p$ , a matriz  $A_{pi}$  definitiva será definida pela equação (24), onde cada elemento da nova matriz será a soma do elemento equivalente na matriz  $Aaux_{pi}$  com a média da diferença entre as quantidades ofertadas e demandadas de cada produto nas matrizes  $Aaux_{pi}$  e  $Baux_{po}$ . Já a matriz  $B_{po}$  definitiva é definida pela equação (24), sendo composta pelos mesmos elementos da matriz  $Baux_{po}$ .

O conjunto de restrições (22), (23), (24) e (25) asseguram que o total de produtos ofertados nos caminhões de entrada e os demandados nos caminhões de saída sejam iguais, respeitando a definição básica de *cross docking* de que nenhum produto fica acumulado no sistema.

#### 4.2 Implementação computacional

Os modelos foram implementados em JuMP (*Julia for Mathematical Programming*), uma linguagem de modelagem embutida na linguagem de programação Julia, escolhida para este trabalho por ter um código fácil, moderno, com IDE gratuito e eficiente para problemas de programação linear inteira mista. Por meio desta linguagem de modelagem, é possível realizar a execução sequencial de vários exemplares; além disso, a JuMP já inclui vários resolvedores



também gratuitos. O resolvidor utilizado foi o CPLEX e todos os testes foram conduzidos em um processador Intel®Core™i7 com 2.4 GHz, 8.0 GB de memória RAM e sistema operacional Windows. Os 160 problemas-testes foram resolvidos até um tempo máximo de 3600 segundos pelo algoritmo *branch-and-cut* incluído no CPLEX. As medidas analisadas para cada um dos modelos foram: o tempo médio de computação (CPU) em segundos e *gap* de otimalidade.

## 5 Resultados e Discussão

A Tabela 4 a seguir apresenta os resultados numéricos da experimentação computacional com os modelos propostos, sendo o modelo 1 baseado em variáveis de alocação enquanto o modelo 2 trata-se do modelo adaptado.

Tabela 4- Comparativo da eficiência computacional média por porte do problema

<i>n</i>	<i>m</i>	<i>P</i>	<i>K</i>	Tempo de CPU (s)		Gap de otimalidade	
				Modelo 1	Modelo 2	Modelo 1	Modelo 2
20	20	10	5	2,5668	2,7839	0	0
			10	3,5606	4,0357	0	0
		15	5	8,251	8,5614	0	0
			10	6,4264	6,6131	0	0
	30	10	5	2,2138	2,4288	0	0
			10	3,9801	4,1172	0	0
		15	5	3,3648	3,4213	0	0
			10	5,6208	5,9023	0	0
25	20	10	5	3060,474	3057,802	0,0322	0,0323
			10	1132,997	1157,414	0,0059	0,0057
		15	5	3628,581	3605,092	0,0415	0,0404
			10	497,9219	508,7734	0,0028	0,0028
	30	10	5	2,8378	3,0622	0	0
			10	5,5383	5,8682	0	0
		15	5	4,2364	4,636	0	0
			10	6,9184	4,4382	0	0

Fonte: elaborado pelos autores

Conforme os dados da Tabela 4, para as duas formulações desenvolvidas, grande parte dos problemas testes foram solucionados até a otimalidade em menos de 6 segundos, exceto os exemplares com número de caminhões de entrada maior do que saída, como os exemplares com 25 caminhões de entrada e 20 caminhões de saída, conforme destacado (em rosa) na Tabela 4. Ainda, segundo essa mesma Tabela, dos problemas-testes com maior dificuldade de execução, pode-se perceber que a maioria daqueles com 10 docas conseguem ainda chegar à solução ótima

no tempo limitado. Mas aquelas com 5 docas não chegam a solução ótima apesar de apresentar bons resultados, com *gap* de otimalidade médio de 3,66%.

Percebe-se através da Tabela 5 que se tratando primeiramente da análise por caminhões de entrada, o tempo de processamento de CPU médio para problemas-testes com 20 caminhões foi de 4,5 segundos no modelo 1, enquanto no modelo 2 esse tempo foi de 4,7 segundos, ambos com *gap* igual a zero (otimalidade). Ao realizar a mudança para 25 caminhões de entrada o tempo de CPU médio aumentou significativamente no modelo 1, tendo um resultado de 1042,4 segundos, e se manteve na ordem de 1043,8 segundos para o modelo 2.

A próxima análise esteve relacionada com o número de caminhões de saída. Ao fixar esse campo em 20 unidades, o tempo de CPU consumido pelo modelo 1 foi de 1042,6 segundos e pelo modelo 2 foi 1043,8 segundos. Alterando essa variável para 30, encontra-se um tempo de 4,3 segundos para o modelo 1 e 4,6 segundos para o segundo modelo.

A terceira configuração dos parâmetros é baseada na alternância do número das docas entre 5 e 10 unidades. Percebe-se que ao aumentar o número de docas, acontece uma redução do tempo médio de CPU de 839,1 para 207,9 segundos no modelo 1 e de 836,0 para 212,5 segundos no modelo 2.

Por fim, é analisada a quantidade de tipos de produtos. Percebe-se a relação decrescente entre o aumento do número de tipos de produtos e o tempo médio computacional, de forma que, variou-se de 523,8 para 520,2 no modelo 1 e 529,7 para 518,8 no modelo 2.

Tabela 5 – Tempos médio e *gap* de otimalidade por parâmetro dos problema-testes

		Modelo 1		Modelo 2	
		CPU (s)	GAP	CPU (s)	GAP
<i>n</i>	20	4,5018	0	4,7329	0
	25	1042,438	0,0103	1043,761	0,0101
<i>m</i>	20	1042,601	0,0103	1043,884	0,0101
	30	4,3388	0	4,6092	0
<i>k</i>	5	839,0694	0,0092	835,9734	0,0091
	10	207,8704	0,0011	212,5202	0,0011
<i>p</i>	10	526,771	0,0048	529,689	0,0047
	15	520,1688	0,0055	518,8046	0,0054

Fonte: elaborado pelos autores

Afim de mensurar a eficiência dos resultados alcançados deveria ser realizada uma análise comparativa dos modelos publicados no meio acadêmico. Contudo existe imensa dificuldade relacionada ao estabelecimento dos mesmos parâmetros nas formulações existentes que

inviabiliza temporariamente a comparação. Os recursos computacionais distintos, como a capacidade de processamento, armazenamento e memória são empecilhos para esse objetivo. No entanto, o principal fator que torna infactível a comparação é a divergência nos exemplares utilizados nos testes, que para essa pesquisa foi gerada pelos próprios autores por não ser possível encontrar na literatura exemplares padrões que atendessem a todos os requisitos da proposta em questão.

## 6 Conclusões

A proposta deste estudo foi examinar a derivação prática e logística de um problema de programação em *flow shop* híbrido comumente aplicado em problemas da manufatura. Os modelos desenvolvidos visam contribuir com um problema corriqueiro na logística brasileira, os altos custos gastos com armazenagem e *picking* (separação) dos produtos. Neste cenário, encontrou-se na modelagem do *cross docking* a solução para a redução de custos. Outra motivação para a adoção da modelagem desse sistema foi a dificuldade de controle encontrada no gerenciamento das cargas a serem carregadas e descarregadas. A ausência de estoque torna visível problemas no mal gerenciamento desses recursos.

Diante disso, a contribuição desse trabalho foi propor um modelo matemático de programação inteira mista, organizando o fluxo de carga e descarga no centro *cross docking* de forma a reduzir o tempo total de fluxo dos veículos, garantindo o atendimento do cliente em quantidades e tempo solicitado. Outra consequência da melhoria no fluxo é a redução do custo com diárias e taxas que protegem os motoristas que ficam ociosos aguardando as operações de carga e descarga de seus veículos. Além disso, as formulações desenvolvidas pelos autores contribuem para sanar as lacunas do conhecimento ao consolidar nesse trabalho os aspectos mais relevantes para alcançar um modelo mais próximo da realidade. Dentre esses aspectos, pode-se citar a inclusão de restrições que consideram múltiplas docas e as operações internas do armazém, como a classificação e distribuição de mercadorias.

Desta forma, para atingir o objetivo proposto, foram elaboradas e avaliadas duas formulações baseadas em variáveis de alocação, baseadas no acréscimo de variáveis visando reduzir a relaxação do modelo e consequentemente o tempo computacional. Para o diagnóstico dos métodos de solução, foram feitas comparações teóricas por meio da análise do tamanho do problema em relação a cada formulação, e também a comparação prática, através da implementação da experimentação computacional. Foram resolvidos na experimentação computacional 160 problemas-testes elaborados respeitando as restrições básicas de um sistema

*cross docking* real e com os resultados avaliou-se o tempo médio de CPU e o *gap* de otimalidade para cada grupo de parâmetro variado.

Os resultados dos modelos foram bastante satisfatórios, atingindo a otimalidade na maior parte dos problemas-testes. As únicas exceções foram aquelas instâncias com número de caminhões de entrada maior do que os caminhões de saída, não garantindo pelo *branch and cut* uma solução ótima no tempo delimitado.

O modelo 2 objetivou reduzir a relaxação do modelo 1 relacionada ao valor de  $t$ , ou seja, buscou reduzir o espaço de soluções viáveis e conseqüentemente o tempo computacional para encontrar a solução ótima. No entanto, os resultados mostraram que o comportamento não foi como esperado, já que não foi suficiente para melhorar consideravelmente a qualidade da solução. Alguns exemplares chegaram a obter um tempo médio de CPU até maior do que no modelo 1. Um dos possíveis motivos para este aumento está pautado no aumento do número de variáveis e restrições do modelo 2 quando comparado ao modelo 1, já que a primeira formulação apresentava apenas dois tipos de variáveis de decisão e seis grupos de restrições, enquanto a formulação 2 emprega três diferentes tipos de variáveis de decisão e sete grupos de restrição.

Como proposta para trabalhos futuros, sugere-se a criação de novos problemas-testes visando agregar nas análises de sensibilidade com relação a variação no número dos parâmetros e conseqüentemente aumentar a confiabilidade dos resultados encontrados. Além disso, poderiam ser exploradas novas formulações baseadas em diferentes paradigmas e que adicionem restrições que tornem o modelo mais realístico e aplicável em diferentes tipos de organização, como por exemplo, as restrições de capacidade dos veículos.

## Referências

- Apte, U. M., & Viswanathan, S. (2000). Effective cross docking for improving distribution efficiencies. *International Journal of Logistics*, 3(3), 291-302.
- Arabani, A. B., Ghomi, S. F., & Zandieh, M. (2011). Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert systems with Applications*, 38(3), 1964-1979.
- Bowersox, D. J., Closs, D. J., Cooper, M. B., & Bowersox, J. C. (2013). Gestão logística da cadeia de suprimentos. *AMGH Editora*.
- Boysen, N. (2010). Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research*, 37(1), 32-41.

Boysen, N., Fliedner, M., & Scholl, A. (2010). Scheduling inbound and outbound trucks at cross docking terminals. *OR spectrum*, 32(1), 135-161.

BRASIL, Lei 13.103/15. Lei do motorista. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2015/lei/113103.htm](http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/113103.htm)>. Acesso em 08 fev.2020

Chen, F., & Lee, C. Y. (2009). Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research*, 193(1), 59-72.

Chen, F., & Song, K. (2009). Minimizing makespan in two-stage hybrid cross docking scheduling problem. *Computers & Operations Research*, 36(6), 2066-2073.

Conceição, S. V., & Quintão, R. T. (2004). Evaluation of the logistic performance of Brazil's soft drink supply chain. *Gestão & Produção*, 11(3), 441-453.

Dulebenets, M. A. (2018). A comprehensive evaluation of weak and strong mutation mechanisms in evolutionary algorithms for truck scheduling at cross-docking terminals. *IEEE Access*, 6, 65635–65650.

Felfel, H., Ayadi, O., & Masmoudi, F. (2016). Multi-objective stochastic multi-site supply chain planning under demand uncertainty considering downside risk. *Computers & Industrial Engineering*, 102, 268–279

Fonseca, G. B.; Nogueira, T. H.; Ravetti, M. G. (2019). A hybrid Lagrangian metaheuristic for the cross-docking flow shop scheduling problem. *European Journal of Operational Research*, v. 275, n. 1, p. 139-154.

Forouharfard, S., & Zandieh, M. (2010). An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems. *The International Journal of Advanced Manufacturing Technology*, 51(9-12), 1179-1193.

Gaudioso, M., Monaco, M. F., & Sammarra, M. (2021). A Lagrangian heuristics for the truck scheduling problem in multi-door, multi-product Cross-Docking with constant processing time. *Omega*, 101, 102255.

Larbi, R., Alpan, G., Baptiste, P., & Penz, B. (2011). Scheduling cross docking operations under full, partial and no information on inbound arrivals. *Computers & Operations Research*, 38(6), 889-900.

Luo, H., Yang, X., & Wang, K. (2019). Synchronized scheduling of make to order plant and cross-docking warehouse. *Computers & Industrial Engineering*, 138, Article 106108.

Nogueira, T. H., Coutinho, F. P., Ribeiro, R. P., & Ravetti, M. G. (2020). Parallel-machine scheduling methodology for a multi-dock truck sequencing problem in a crossdocking center. *Computers & Industrial Engineering*, Article 106391.

- Shahmardan, A., & Sajadieh, M. S. (2020). Truck scheduling in a multi-door cross-docking center with partial unloading–Reinforcement learning-based simulated annealing approaches. *Computers & Industrial Engineering*, 139, 106134.
- Soltani, R., & Sadjadi, S. J. (2010). Scheduling trucks in cross-docking systems: A robust meta-heuristics approach. *Transportation Research Part E: Logistics and Transportation Review*, 46(5), 650-666.
- Vahdani, B., & Zandieh, M. (2010). Scheduling trucks in cross-docking systems: Robust meta-heuristics. *Computers & Industrial Engineering*, 58(1), 12-24.
- Yan, W. (2014). *Heuristics for Truck Scheduling at Cross Docking Terminals* (Doctoral dissertation, Concordia University).
- Yu, W., & Egbelu, P. J. (2008) Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research*, 184(1), 377-396.
- Zheng, F., Pang, Y., Xu, Y., & Liu, M. (2020). Heuristic algorithms for truck scheduling of cross-docking operations in cold-chain logistics. *International Journal of Production Research*, 1-22.