

MARIANA MONTEIRO USHIZIMA

Pareto iterated local search: uma aplicação ao sequenciamento de
linhas de montagem de modelos mistos

Mariana Monteiro Ushizima

Pareto iterated local search: uma aplicação ao sequenciamento de
linhas de montagem de modelos mistos

Dissertação apresentada à Faculdade de
Engenharia do Campus de Guaratinguetá,
Universidade Estadual Paulista, para obtenção
do título de Mestre em Engenharia Mecânica,
na área de Gestão e Otimização.

Orientadora: Prof^a. Dr^a. Marcela Aparecida
Guerreiro Machado de Freitas
Coorientador: Prof. Dr. Fernando Augusto
Silva Marins

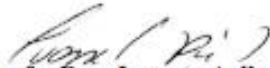
U85p	Ushizima, Mariana Monteiro Pareto Iterated Local Search: uma aplicação ao sequenciamento de linhas de montagem de modelos mistos / Mariana Monteiro Ushizima – Guaratinguetá, 2021. 105 f : il. Bibliografia: f. 84-95 Dissertação (Mestrado) – Universidade Estadual Paulista, Faculdade de Engenharia de Guaratinguetá, 2021. Orientadora: Prof^a Dr^a: Marcela Aparecida Guerreiro M. de Freitas Coorientador: Prof. Dr. Fernando Augusto Silva Marins 1. Inteligência artificial. 2. Processo decisório por critério múltiplo. 3. Pesquisa operacional. 4. Indústria automobilística. I. Título. CDU 007.52(043)
------	---

MARIANA MONTEIRO USHIZIMA

**ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE
“MESTRE EM ENGENHARIA MECÂNICA”**

**PROGRAMA: ENGENHARIA MECÂNICA
CURSO: MESTRADO**

APROVADA EM SUA FORMA FINAL PELO PROGRAMA DE PÓS-GRADUAÇÃO


Prof. Dra. Ivonete Avila
Coordenadora

BANCA EXAMINADORA:



PROF. DR. FERNANDO AUGUSTO SILVA MARINS

Coorientador / UNESP/FEG

participou por videoconferência



PROF. DR. ANEIRSON FRANCISCO DA SILVA

UNESP/FEG

participou por videoconferência



PROF. DR. FABRÍCIO MACIEL GOMES

USP

participou por videoconferência

Fevereiro de 2021

DADOS CURRICULARES

MARIANA MONTEIRO USHIZIMA

NASCIMENTO	27.05.1991 – Cruzeiro/SP
FILIAÇÃO	Carlos Alberto Ushizima Maria Clara Serapião Monteiro Ushizima
2009/2014	Curso de Graduação em Engenharia de Produção Mecânica Faculdade de Engenharia de Guaratinguetá Universidade Estadual Paulista “Júlio de Mesquita Filho”
2012/2013	Período sanduíche no Institut National Polytechnique de Grenoble – Filial Génie Industriel Grenoble - França

dedico este trabalho aos meus pais

AGRADECIMENTOS

Agradeço à minha família pelo apoio,
aos meus amigos pelo incentivo,
aos meus orientadores e à banca pelas contribuições,
e aos colegas e demais docentes e funcionários da Faculdade de Engenharia do Campus de Guaratinguetá que fizeram parte dessa jornada.

RESUMO

Em um contexto de Indústria 4.0, em que a transformação do mercado de consumo exige cada vez mais que as indústrias repensem e reformulem seus modelos de negócio para lidar com as constantes mudanças nas demandas dos clientes, sistemas de produção capazes de serem flexíveis para produzirem diversos modelos de forma eficiente ganham destaque. Além disso, neste mesmo contexto, a aplicação de inteligência artificial está inserida no conjunto de tecnologias que vão permitir as indústrias trilharem uma trajetória rumo a um sistema inteligente de manufatura. Neste trabalho aborda-se o Problema de Sequenciamento de Linhas de Montagem de Modelos Mistos – linhas destinadas à produção de modelos diversificados, suportada por diferentes fornecedores. Em situações em que existe um alto número de modelos a serem produzidos, o sequenciamento é uma ferramenta importante para obter benefícios tais como nivelar estoques intermediários e carga de trabalho, evitar paradas de linha, e o desbalanceamento de carga do fornecedor e minimizar a variação da taxa de produção. Neste trabalho, os objetivos do sequenciamento multiobjetivo considerados são minimizar a distância entre o consumo real do consumo ideal de componentes ao longo dos instantes e o número de setups, simultaneamente. Para resolver o problema, foi aplicado de forma inédita ao sequenciamento, a meta-heurística *Pareto Iterated Local Search*, escolhido devido sua simplicidade e eficiência. O método foi testado utilizando instâncias da literatura, se mostrando capaz de fornecer boas soluções, com tempo computacional competitivo, quando comparado a outros métodos. Posteriormente, o modelo foi aplicado a um caso real de montagem de estruturas em uma indústria do ramo automotivo, resultando em um conjunto de boas soluções, que melhoram o sequenciamento atual, em um tempo adequado, para o decisor optar.

PALAVRAS-CHAVE: Sequenciamento. Multiobjetivo. Linhas de montagem de modelos mistos. Meta-heurísticas. *Pareto iterated local search*.

ABSTRACT

In a context of Industry 4.0, in which the transformation of the consumer market increasingly requires industries to rethink and reformulate their business models to deal with the constant changes in customer demands, production systems capable of being flexible to produce diverse models efficiently gain prominence. In addition, in this same context, use of artificial intelligence is inserted in the set of technologies that will allow the industries to follow a path towards an intelligent manufacturing system. This work addresses the Sequencing Problem of Mixed Model Assembly Lines - lines designed to produce various models and supported by different suppliers. In cases where there is a high number of models to be produced, sequencing is an important tool to obtain benefits such as leveling intermediate inventory and workload, avoiding line stops, and unbalancing the supplier's load and minimizing variation in production rate. In this work, the multiobjective sequencing objectives considered are minimizing the distance between the actual consumption of the ideal consumption of components over time and the number of setups, simultaneously. To solve the problem, the metaheuristic Pareto Iterated Local Search was applied in an unprecedented way to sequencing, chosen due to its simplicity and efficiency. The method was tested using the literature, showing itself capable of providing good solutions, with competitive computational time, when compared to other methods. Subsequently, the model was applied in a real case of assembling structures in an automotive industry, producing a set of good solutions, which improve the current sequencing, in a suitable time, for the decision maker to choose from, according to his priorities.

KEYWORDS: Sequencing. Multiobjective. Mixed model assembly lines. Metaheuristics. Pareto iterated local search.

LISTA DE ILUSTRAÇÕES

Figura 1 - Publicações por ano: " <i>Mixed-Model Assembly Line</i> "	18
Figura 2 - Percentual das publicações com a palavra-chave " <i>Mixed-Model Assembly Line</i> " que também contém " <i>Sequencing</i> "	18
Figura 3 - <i>Clusters</i> mapa de palavras-chave <i>Sequencing</i> e <i>Multi-Objective</i> e <i>Mixed-Model Assembly Line</i>	19
Figura 4 - Número de publicações com as palavras-chave " <i>Sequencing</i> " e " <i>Multi-Objective</i> " e percentual destas com a palavra-chave " <i>Mixed-Model Assembly Line</i> "	20
Figura 5 - Número de publicações com as palavras-chave " <i>Multi-Objective</i> " e " <i>Iterated Local Search</i> "	20
Figura 6 - Áreas do conhecimento com aplicação do ILS multiobjetivo	21
Figura 7 - Cluster mapa de palavras-chave <i>Iterated Local Search</i> and <i>Multi-Objective</i>	22
Figura 8 - Publicações por ano: " <i>Industry 4.0</i> "	22
Figura 9 - Cluster mapa de palavras-chave " <i>Industry 4.0</i> "	23
Figura 10 - Fluxograma para aplicação da abordagem proposta.....	26
Figura 11 - Classificação das referências utilizadas: Ano de publicação.....	30
Figura 12 - Tipos de documento utilizados no trabalho + Classificação dos periódicos lidos na Área Engenharias III na WebQualis.....	31
Figura 13 – Período e inovações tecnológicas das quatro revoluções industriais	32
Figura 14 - Exemplo de uma Linha de Montagem de Modelos Mistos	34
Figura 15 - Etapas de planejamento da produção de modelos mistos.....	37
Figura 16 - Consumo real <i>versus</i> consumo ideal de determinado componente a_j na linha.....	40
Figura 17 - Gráfico do número de soluções possíveis por problema do exemplo da Tabela 3	51
Figura 18 - Exemplo de Fronteira Pareto-ótima.....	55
Figura 19 - Ilustração do ILS.....	59
Figura 20 - Pseudocódigo da meta-heurística ILS	59
Figura 21 - Pseudocódigo da meta-heurística VNS	60
Figura 22 - Representação de uma solução do sequenciamento de MMMAL.....	65
Figura 23 - Fronteira Eficiente do exemplo.....	66
Figura 24 - Exemplo do Movimento <i>Swap</i>	68
Figura 25 - Exemplo do Movimento <i>Shift</i> para frente.....	68
Figura 26 - Exemplo do Movimento <i>Shift</i> para trás	69
Figura 27 - Exemplo do Movimento de Perturbação	69

Figura 28 - Pseudocódigo da meta-heurística PILS	70
Figura 29 - Consumo do componente a_1 atualmente, PILS com 12 e 18 setups.....	80
Figura 30 - Consumo do componente a_3 atualmente, PILS com 12 e 18 setups.....	80
Figura 31 - Consumo do componente a_{12} atualmente, PILS com 12 e 18 setups.....	80
Figura 32 - Consumo do componente a_{16} atualmente, PILS com 12 e 18 setups.....	81
Quadro 1 - Comparação entre indústria automotiva tradicional e futura	33
Quadro 2 - Classificação das MMAL.....	35

LISTA DE TABELAS

Tabela 1 - Relatório do número de publicações e citações resultante da pesquisa	17
Tabela 2 - Periódicos mais utilizados	31
Tabela 3 - Exemplo do número de sequências possíveis por problema	51
Tabela 4 – Exemplo de consumo de componentes por produto	66
Tabela 5 - Exemplo de sequencias possíveis e número de setups e consumo resultantes.....	67
Tabela 6 - Set 1: demanda (Q_i) de cada tipo de produto (A_i) por problema, com $Q = 10$	72
Tabela 7 - Set 2: demanda (Q_i) de cada tipo de produto (A_i) por problema, com $Q = 12$	73
Tabela 8 – Consumo de componentes (a_j) para cada produto (A_i)	73
Tabela 9 - Métricas de desempenho para o Set 1	74
Tabela 10 - Métricas de desempenho para o Set 2	75
Tabela 11 - Matriz de Demandas (Q_i) do Caso Real por Produto (A_i)	76
Tabela 12 - Matriz de componentes (a_j) do Caso Real.....	77
Tabela 13 - Comparação entre os níveis de iterações para o Caso Real	78
Tabela 14 - Melhores sequências obtidas pelo PILS	79

SUMÁRIO

1	INTRODUÇÃO	13
1.1	CONTEXTUALIZAÇÃO DA PESQUISA	13
1.2	JUSTIFICATIVA E CONTRIBUIÇÕES	15
1.3	QUESTÕES DA PESQUISA E OBJETIVOS	24
1.4	PROCEDIMENTO METODOLÓGICO E MATERIAIS	25
1.5	DELIMITAÇÃO DA PESQUISA	28
1.6	ESTRUTURA DO TRABALHO	29
2	FUNDAMENTAÇÃO TEÓRICA.....	30
2.1	INDÚSTRIA 4.0	32
2.2	LINHAS DE MONTAGEM DE MODELOS MISTOS	34
2.3	PROBLEMA DE SEQUENCIAMENTO DE linhas de montagem de modelos mistos	37
2.3.1	Problemas com um único objetivo	38
2.3.1.1	Minimizar ineficiências utilizando penalizações	38
2.3.1.2	Minimizar comprimento total da linha.....	38
2.3.1.4	Minimizar a variação de consumo de componentes.....	40
2.3.1.5	Minimizar a variação da taxa de produção de cada modelo.....	43
2.3.1.6	Minimizar o tempo de parada da linha.....	44
2.3.2	Problemas com múltiplos objetivos	44
2.3.2.1	Minimizar variação na carga de trabalho e a variação de componentes consumidos	45
2.3.2.2	Minimizar variação de componentes consumidos e paradas de linha.....	45
2.3.2.3	Minimizar a variação da taxa de produção de cada modelo e número de setups.....	46
2.3.2.4	Minimizar variação da carga de trabalho e setups	47
2.3.2.5	Minimizar variação do consumo de componentes, minimizar setup e minimizar tempo extra/tempo ocioso	47
2.3.2.6	Minimizar a variação da taxa de produção de cada modelo, minimizar setup e minimizar tempo extra/tempo ocioso	49
2.4	SEQUENCIAMENTO DE MMAL E OTIMIZAÇÃO.....	49
2.4.1	Otimização Combinatória	50
2.4.2	Otimização Multiobjetivo	52
2.4.2.1	Métodos <i>a priori</i>	53
2.4.2.2	Métodos <i>a posteriori</i>	54

2.4.2.3	Métodos iterativos	55
2.5	META-HEURÍSTICAS	55
2.6	<i>PARETO ITERATED LOCAL SEARCH</i>	57
2.6.1	Busca Local Iterativa	58
2.6.2	Pesquisa em Vizinhança Variável	59
2.6.3	<i>Pareto Iterated Local Search</i>	61
3	PILS APLICADO AO PROBLEMA DE SEQUENCIAMENTO DE MMAL ..	64
3.1	MODELAGEM DO PROBELMA	64
3.2	REPRESENTAÇÃO DA SOLUÇÃO.....	65
3.3	AVALIAÇÃO DA SOLUÇÃO.....	65
3.4	GERAÇÃO DA SOLUÇÃO INICIAL	67
3.5	ESTRUTURAS DE VIZINHANÇA.....	68
3.6	PERTURBAÇÃO.....	69
3.7	ALGORITMO E PSEUDOCÓDIGO	69
4	RESULTADOS	72
4.1	COMPARAÇÃO COM RESULTADOS DA LITERATURA.....	72
4.2	APLICAÇÃO EM UM CASO REAL.....	76
5	CONCLUSÃO	82
5.1	RESPOSTAS ÀS QUESTÕES DE PESQUISA E OBJETIVOS	82
5.2	SUGESTÕES PARA TRABALHOS FUTUROS	83
	REFERÊNCIAS	84
	APÊNDICE A - Código em C++	96

1 INTRODUÇÃO

Apresentam-se neste capítulo as considerações iniciais, a justificativa, as questões e os objetivos da pesquisa, sua delimitação, os materiais e métodos utilizados e, finalmente, a estrutura desta dissertação.

1.1 CONTEXTUALIZAÇÃO DA PESQUISA

O ambiente industrial vive, desde o início dos anos 2000, sua quarta revolução industrial (BORTOLINI *et al.*, 2017). Este movimento também é chamado de Indústria 4.0, termo que foi proposto pela primeira vez em 2011 pelo governo alemão, como parte de um plano estratégico para colocar o país em um novo patamar de industrialização (KAGERMANN *et al.*, 2013).

Esta nova era, com a transformação digital do mercado de consumo, tem exigido que as indústrias repensem e reformulem seus modelos de negócio para lidar com as constantes mudanças nas demandas dos clientes e turbulências do mercado (FATHI; GHOBAKHLOO, 2020). Para superar os desafios de produzir produtos cada vez mais individualizados, a Indústria 4.0 aposta na maior flexibilidade na fabricação, buscando também obter os produtos no tempo e qualidade esperados pelo cliente, de maneira eficiente (ZHONG *et al.*, 2017).

A busca por sistemas produtivos eficientes não é uma novidade. No início do século XX, Henry Ford criou o conceito de linhas de produção, em um contexto de produção em massa de um único produto, tendo entre os objetivos diminuir o tempo e reduzir custos de produção. Neste momento, no entanto, a pressão competitiva tem forçado os fabricantes a projetar diferentes variantes dos seus produtos, transformando a tendência de produção em massa, em customização em massa (SAIF *et al.*, 2019).

No contexto automotivo, por exemplo, as variantes podem ser desde estruturais, como o estilo de carroceria, passando pelo número de subconjuntos necessários, como número de combinações de motor / transmissão, até variantes periféricas, que são as opcionais que não afetam o design do núcleo, como assentos, teto solar e espelhos elétricos (LYONS; UM; SHARIFI, 2020). Neste setor, a tendência contínua de customização em massa levou a um aumento importante no número de variantes nos últimos anos (BALLER *et al.*, 2020). A fabricante de automóveis AUDI, por exemplo, produzia apenas quatro modelos de automóveis diferentes em 1996, 14 em 2004 e 53 em 2014 (WÖRNER, 2015).

Quanto mais opcionais, mais modelos distintos possíveis de serem montados. À título de exemplo, considerando um veículo que tenha 30 opcionais, mais de um bilhão de combinações ($2^{30} = 1.073.741.824$) seriam possíveis.

Para atender esta necessidade de flexibilização oriunda dos desejos dos consumidores, sistemas produtivos atingem um nível cada vez mais alto de complexidade. Por esta e outras razões, os sistemas que ganham importância e que mais tem se desenvolvido são as Linhas de Montagem de Modelos Mistos (MMAL, do inglês *Mixed-Model Assembly Lines*), nos quais é possível produzir vários modelos de um mesmo produto em uma mesma linha, fazendo com que as técnicas de produção em massa possam ser usadas para uma família de modelos (KRENCZYK; DZIKI, 2020, ZHANG; XU; ZHANG, 2021).

Para encurtar o tempo de resposta à demanda, muitos estudos são focados na utilização eficaz das MMAL. Para Saif *et al.* (2019), para tal, dois problemas devem ser solucionados:

- 1) Designação das diferentes tarefas às estações na linha, para distribuir igualmente a carga de trabalho entre elas, chamado de problema de balanceamento;
- 2) Determinação da sequência de produção para diferentes modelos na linha, chamado de problema de sequenciamento.

Apesar de quase qualquer sequência ser tecnicamente viável, as MMAL podem utilizar de determinadas sequências para a produção de modelos diferentes de um mesmo produto, para obter benefícios. Segundo Zandieh e Moradi (2017), entre os objetivos de sequenciamento encontrados na literatura estão minimizar a variação da taxa de produção, minimizar a duração de paradas de linha, minimizar sobrecarga de trabalho, minimizar o custo de tempo útil e ocioso, minimizar o custo/tempo de setup, e nivelar carga de trabalho.

Do aspecto matemático, o problema de sequenciamento pode ser visto como um problema de Otimização Combinatória com variáveis discretas, onde o número de possíveis soluções cresce exponencialmente com o número de produtos a serem sequenciados e com a demanda da linha no horizonte planejado, caracterizando um problema complexo (McMULLEN, 2001b).

Este problema é, especialmente na indústria automotiva, onde número elevado de combinações pode ser possível, considerado intratável por métodos exatos de otimização (ex.: *Branch and Bound*, Programação Dinâmica...). Por isso, resultados de pesquisas publicadas nesta área são baseados em métodos de Inteligência Artificial de busca de soluções quase-ótimas (KRENCZYK; DZIKI, 2020). No contexto de Indústria 4.0, a Inteligência Artificial é um dos benefícios trazidos à indústria de manufatura (AZIZ, 2020), desempenhando um papel essencial e auxiliando, como neste caso, na tomada de decisão (ZHONG *et al.*, 2017).

Estes métodos de busca de soluções quase-ótimas, chamados de heurísticas e meta-heurísticas, surgiram dentro da Pesquisa Operacional para resolver problemas de otimização de alta complexidade computacional (BELFIORE; FÁVERO, 2013).

Os algoritmos meta-heurísticos são genéricos, ou seja, são feitos para resolver diferentes problemas, e não um problema específico. A ideia principal é partir de uma solução arbitrária e depois melhorá-la progressivamente. Assim, eles são capazes de fornecer boas soluções, sem garantia de atingir a ótima, em um tempo relativamente mais curto, comparado às técnicas tradicionais de otimização (XIONG *et al.*, 2015).

Entre estas meta-heurísticas está a Busca Local Iterativa (ILS, do inglês *Iterated Local Search*), proposta por Lourenço, Martin e Stutzle (2003). Devido à sua simplicidade e, ao mesmo tempo, poder de busca, o ILS tem sido amplamente aplicado a diversos problemas de otimização combinatória de objetivo único.

Com a motivação de criar um algoritmo simples e eficaz para resolver problemas com múltiplos objetivos, Geiger (2006) propôs uma adaptação do ILS, o *Pareto Iterated Local Search* (PILS). Uma abordagem que, ao contrário de muitas meta-heurísticas desenvolvidas para resolver problemas multiobjetivo, depende pouco da configuração de parâmetros na sua implementação. Além disso, o ILS é robusto, eficaz e fácil de implementar (ASSIS *et al.*, 2013).

Este trabalho apresenta a aplicação da meta-heurística PILS ao problema de Sequenciamento Multiobjetivo de MMAL, com objetivo de minimizar, simultaneamente, a variação da taxa de consumo de componentes e o número de setups. Apesar de muitos estudos serem focados na minimização da variação da taxa de consumo de componentes, no levantamento bibliográfico realizado, não foram encontrados trabalhos que consideram este objetivo e a minimização de setups simultaneamente. Tampouco foram encontrados trabalhos que apliquem o PILS ao Sequenciamento de MMAL. Por esta razão, aliado ao fato de ser um método simples e eficiente, esta meta-heurística foi selecionada para este trabalho.

1.2 JUSTIFICATIVA E CONTRIBUIÇÕES

Nesta seção descreve-se a importância do tema pesquisado, que é identificado como uma lacuna de conhecimento obtida por meio de uma revisão da literatura. Deste fato percebe-se a relevância deste trabalho como uma contribuição científica. Além disso, pelo novo modelo proposto, pode-se caracterizá-lo como inovador. Finalmente, por atender a uma necessidade real da empresa ao qual foi aplicado traz notadamente uma contribuição e novos

conhecimentos para a organização, que estão entre os requisitos para seu avanço, no contexto da Indústria 4.0.

É bem conhecido que a Indústria 4.0 tem um importante impacto estratégico de longo prazo no desenvolvimento industrial global, para alcançar maior eficiência, competência e competitividade. Devido à importância deste assunto, tem havido uma demanda crescente por pesquisas que forneçam *insights* sobre as questões, desafios e soluções relacionadas ao design, implementação e gestão da Indústria 4.0 (XU; XU; LI, 2018).

Neste trabalho é abordado o problema de sequenciamento multiobjetivo de MMAL, utilizando a meta-heurística PILS. As MMAL são importantes pois refletem a estratégia do mercado voltada para a diversificação. E o estudo dessas linhas produzindo diversos modelos tem sido de especial interesse para a indústria moderna (CAMUSSI; CERDÁ; CAFARO, 2021).

Devido à sua grande relevância prática, pesquisas têm tido como foco o sequenciamento de curto-prazo de produtos em nessas linhas desde os anos 1960 (BOYSEN; FLIEDNER; SCHOLL, 2009a), buscando obter benefícios, tais como: a taxa de utilização constante de componentes críticos consumidos em uma linha, a redução da sobrecarga dos postos de trabalho e das paradas para preparação ou ajuste de máquinas, e a diminuição dos estoques intermediários (GHINATO, 2000).

O sequenciamento envolvendo múltiplos objetivos torna-se relevante por se assemelhar mais à realidade encontrada nos sistemas produtivos. Apesar de existirem muitas pesquisas envolvendo MMAL, incluindo os trabalhos que lidavam com o sequenciamento multiobjetivo, alguns autores indicam que ainda há oportunidades para trabalhos científicos neste campo, principalmente os relacionados a aplicação das técnicas desenvolvidas na academia aos problemas práticos (SILVA; TUBINO; SEIBEL, 2015; SAIF *et al.*, 2019).

Como o sequenciamento multiobjetivo de MMAL se trata de um problema complexo do ponto de vista matemático, dado que o número de possíveis soluções cresce exponencialmente com o número de produtos e a demanda a ser sequenciada, o método escolhido para resolvê-lo trata-se de uma meta-heurística. Com os resultados do modelo, pretende-se suportar a tomada de decisão, oferecendo soluções robustas e alinhadas com os objetivos da organização.

Segundo Soares *et al.* (2016), a bibliometria pode auxiliar na identificação de tendências de crescimento do conhecimento em determinada disciplina, dispersão e obsolescências de campos científicos. Assim sendo, para evidenciar a relevância científica do tema estudado, foram realizadas pesquisas na base Scopus®, em busca de Artigos e *Reviews* publicados entre os anos de 1999 e 2020.

As palavras-chave “*Mixed-Model Assembly Line*”, “*Sequencing*”, “*Multi-Objective*”, “*Iterated Local Search*” e “*Industry 4.0*” foram buscadas, de forma isolada ou combinada, em “*Title, Abstract and Keywords*”. Para limitar os resultados às áreas de interesse da pesquisa, os documentos encontrados foram filtrados pelas seguintes áreas de conhecimento: *Engineering*; *Computer Science*; *Decision Sciences*; *Business, Management and Accounting* e *Mathematics*.

A Tabela 1 foi construída com base no número de publicações e de citações apresentadas no relatório base Scopus®, de acordo com cada palavra-chave e suas combinações.

Tabela 1 - Relatório do número de publicações e citações resultante da pesquisa

Palavras-chave e combinações pesquisadas	Publicações	Citações
“ <i>Mixed-Model Assembly Line</i> ”	447	8.001
“ <i>Sequencing</i> ”	20.807	>680.000
“ <i>Industry 4.0</i> ”	3.871	>28.000
“ <i>Mixed-Model Assembly Line</i> ” e “ <i>Sequencing</i> ”	197	3.743
“ <i>Industry 4.0</i> ” e “ <i>Mixed-Model Assembly Line</i> ”	7	19
“ <i>Industry 4.0</i> ” e “ <i>Mixed-Model Assembly Line</i> ” e “ <i>Sequencing</i> ”	-	-
“ <i>Sequencing</i> ” e “ <i>Multi-Objective</i> ”	220	3.882
“ <i>Mixed-Model Assembly Line</i> ”, “ <i>Sequencing</i> ” e “ <i>Multi-Objective</i> ”	43	648
“ <i>Multi-Objective</i> ” e “ <i>Iterated Local Search</i> ”	27	306
“ <i>Sequencing</i> ”, “ <i>Multi-Objective</i> ” e “ <i>Iterated Local Search</i> ”	-	-
“ <i>Mixed-Model Assembly Line</i> ”, “ <i>Multi-Objective</i> ” e “ <i>Iterated Local Search</i> ”	-	-

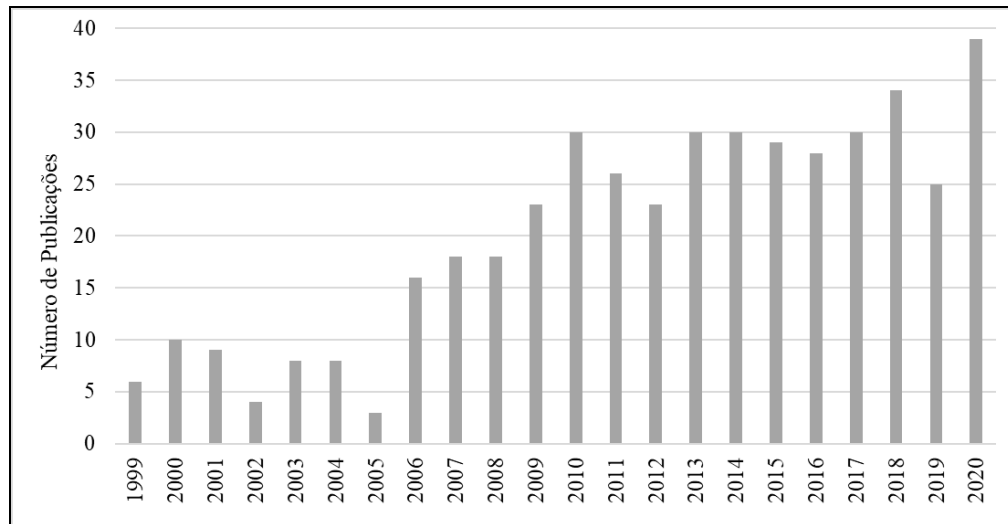
Fonte: Scopus (2020).

Analisando a Tabela 1, nota-se que as palavras-chave, quando pesquisados separadamente, têm numerosas publicações e citações. A primeira palavra-chave buscada, “*Mixed-Model Assembly Line*”, retornou 447 publicações, com 8.001 citações, a segunda, “*Sequencing*”, 20.807 publicações, com mais de 680 mil citações. Entretanto, “*Sequencing*” é um termo utilizado em muitas áreas de conhecimento e, mesmo com todos os filtros feitos, apareceram muitos artigos não relacionados com o tipo de sequenciamento tratado aqui. E a terceira, “*Industry 4.0*”, retornou 3.871 publicações, com mais de 28.000 citações.

A Figura 1 apresenta a evolução do número de publicações ao longo dos anos da palavra-chave “*Mixed-Model Assembly Line*”. Nela pode-se constatar a contemporaneidade do assunto, que teve seus estudos intensificados em meados dos anos 2000, sendo que 72%

dos trabalhos foram publicados depois de 2010 e destes 72%, 57% foram publicados a partir de 2015.

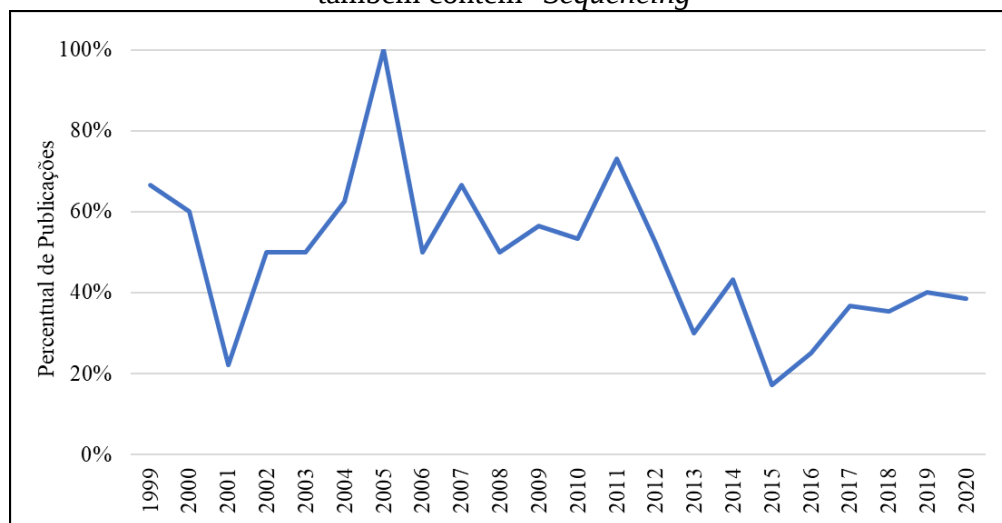
Figura 1 - Publicações por ano: "*Mixed-Model Assembly Line*"



Fonte: Scopus (2020).

Na Figura 2 é mostrado a evolução ao longo dos anos do percentual do total de artigos encontrados com a palavra-chave "*Mixed-Model Assembly Line*" que foram encontrados também na pesquisa combinada com a palavra-chave "*Sequencing*". Percebe-se que os trabalhos relacionados ao sequenciamento de MMAL, que totalizam 197, tiveram uma queda entre os anos de 2013 e 2015, mas voltaram a ganhar força a partir de 2016.

Figura 2 - Percentual das publicações com a palavra-chave "*Mixed-Model Assembly Line*" que também contém "*Sequencing*"

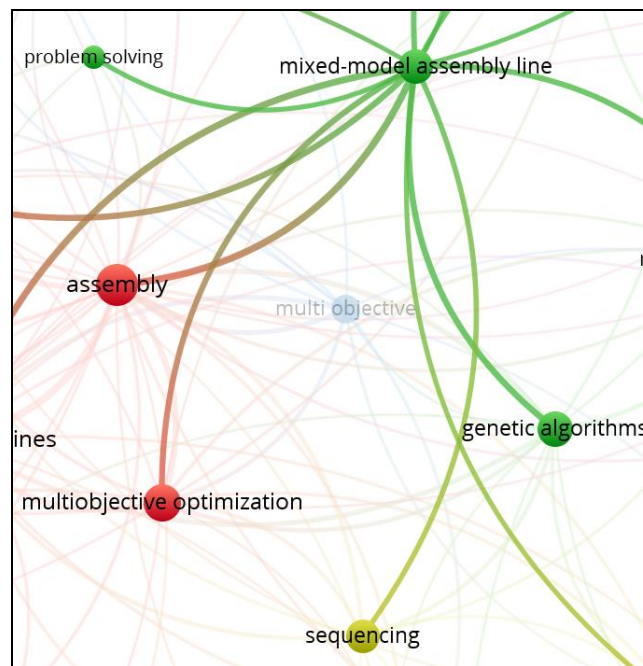


Fonte: Scopus (2020).

Apenas 43 das 220 publicações encontradas com as palavras-chave “*Sequencing*” e “*Multi-Objective*”, também continham “*Mixed-Model Assembly Line*” no seu título, resumo ou palavras-chave, como mostrado na Tabela 1. Estas aplicações de sequenciamento multiobjetivo a outras estruturas indicam que pode haver evoluções no estudo de sequenciamento que ainda não foram aplicadas as MMAL.

Usando o *software* livre VOSViewer foram contabilizadas as palavras-chave que ocorreram com uma frequência maior ou igual a 3 nesses artigos. A Figura 3 mostra uma fraca relação entre os três temas (*Sequencing* e *Multi-Objective* e *Mixed-Model Assembly Line*), o que mostra que existe uma boa oportunidade de pesquisa nesse campo.

Figura 3 - *Clusters* mapa de palavras-chave *Sequencing* e *Multi-Objective* e *Mixed-Model Assembly Line*

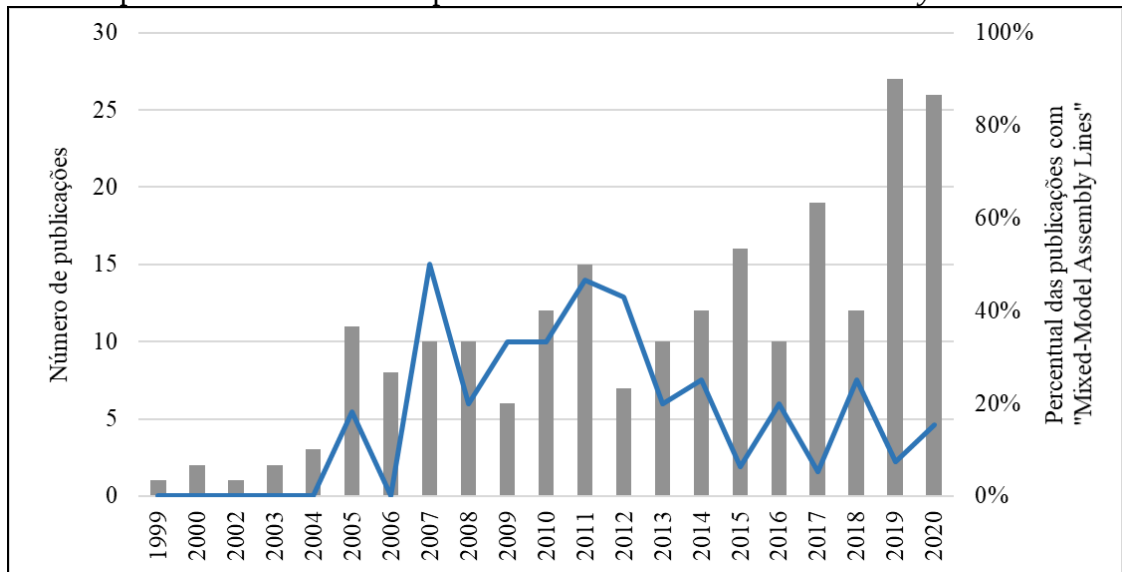


Fonte: VOSViewer (2020).

Um fato que ressalta a originalidade deste trabalho é que na leitura das 43 publicações encontradas, não foram identificados trabalhos que considerem a minimização da variação da taxa de consumo de componentes e a minimização de setups de forma simultânea, objetivos estudados neste trabalho.

Para ilustrar estas publicações ao longo dos anos, as barras da Figura 4 representam as 220 com “*Sequencing*” e “*Multi-Objective*”, e a linha, o percentual destas que também contém “*Mixed-Model Assembly Line*”. Nota-se que as publicações acerca do tema voltaram a crescer, tendo em 2019 o valor máximo histórico.

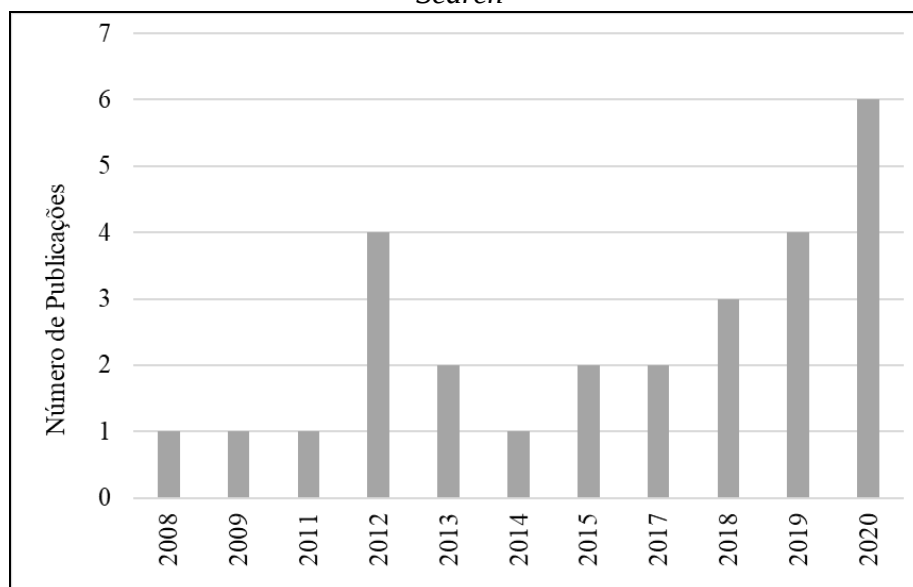
Figura 4 - Número de publicações com as palavras-chave "Sequencing" e "Multi-Objective" e percentual destas com a palavra-chave "Mixed-Model Assembly Line"



Fonte: Scopus (2020).

Na Tabela 1, pode-se observar ainda que o resultado da pesquisa utilizando “*Multi-Objective*” e “*Iterated Local Search*” foi de 27 publicações, com 306 citações. A quantidade destas publicações ao longo dos anos é mostrada na Figura 5. Das 27 publicações encontradas, 93% foram publicadas depois de 2010 e destas 93%, 68% foram publicadas a partir de 2015, ilustrando um aumento no interesse de pesquisadores neste assunto.

Figura 5 - Número de publicações com as palavras-chave “*Multi-Objective*” e “*Iterated Local Search*”

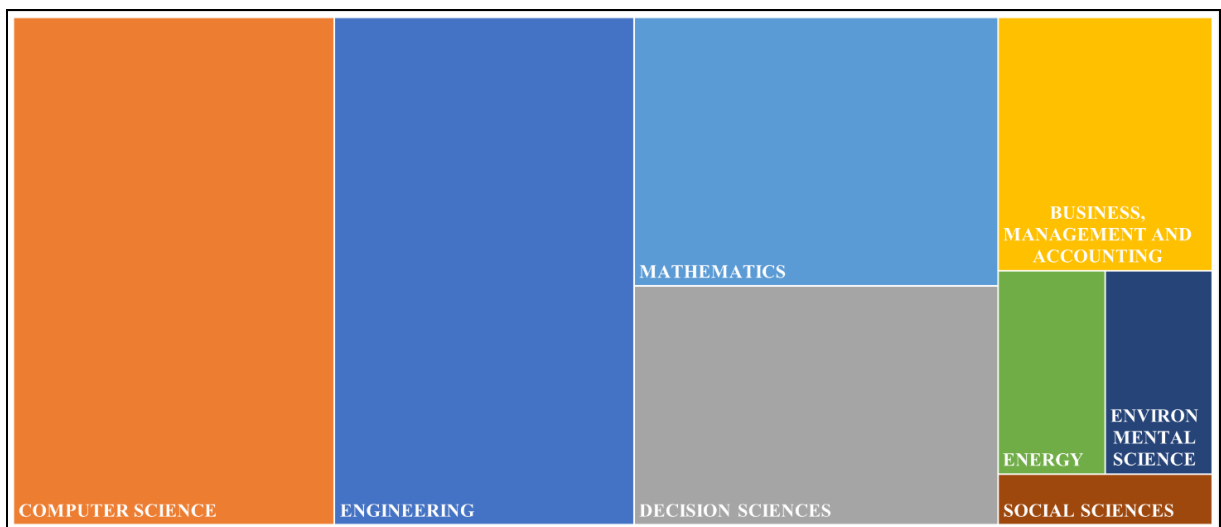


Fonte: Scopus (2020).

Quando apresentada em Geiger (2006), a aplicação do PILS a problemas de sequenciamento de fluxos de trabalho multiobjetivo gerou bons resultados, apesar da simplicidade da abordagem do algoritmo, fato este que a torna uma ferramenta promissora para resolução eficiente de outros problemas de otimização multiobjetivo.

A Figura 6 apresenta as aplicações do ILS multiobjetivo pelas várias áreas de conhecimento como: Ciência da Computação, Engenharia e Matemática, sendo que este trabalho também contempla grande parte destas áreas de pesquisa.

Figura 6 - Áreas do conhecimento com aplicação do ILS multiobjetivo



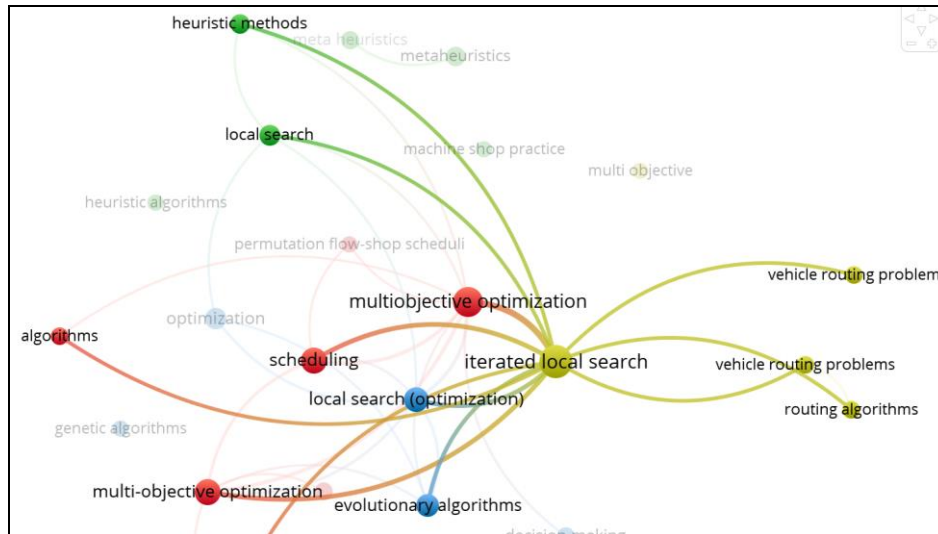
Fonte: Scopus (2020).

Como consta na Tabela 1, a busca da combinação “*Multi-Objective*” e “*Iterated Local Search*” com “*Sequencing*” ou “*Mixed-Model Assembly Lines*” não retornou resultados. Com isto, é possível identificar uma lacuna de pesquisa, para qual esta pesquisa pode trazer uma contribuição científica.

Analisando a Figura 7 revela-se a formação de vários clusters associados às palavras-chave representativas e pode-se inferir que existem artigos que utilizam o ILS aplicado à problemas multiobjetivo. Ou seja, esses resultados mostram o interesse científico e prático em combinar essas técnicas. Entretanto, como visto nos resultados da análise anterior dos 27 artigos publicados relacionados às palavras-chave citadas, nenhum deles aplicou o método ao sequenciamento de MMAL, confirmando a originalidade desta dissertação.

Sendo assim, como contribuição teórica, pode-se destacar a inédita aplicação do PILS ao problema de sequenciamento de MMAL, a consideração dos objetivos de minimizar simultaneamente a variação da taxa de componentes e o número de setups.

Figura 7 - Cluster mapa de palavras-chave *Iterated Local Search and Multi-Objective*

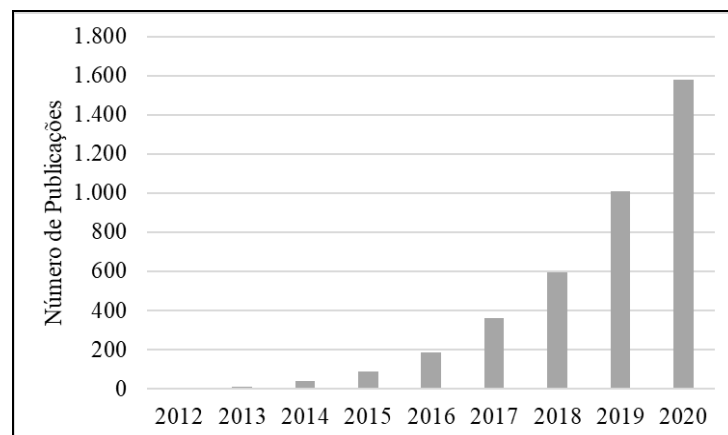


Fonte: VOSViewer (2020).

No contexto da Indústria 4.0, novas formas e métodos para transformar o sistema de manufatura tradicional em um sistema inteligente são fundamentais (ZHONG *et al.*, 2017). Em termos de inovação, a quarta revolução industrial traz para a indústria de manufatura os benefícios de muitos dos paradigmas tecnológicos recentes, incluindo Internet das Coisas, *Big Data*, Computação em Nuvem e Inteligência Artificial (AZIZ, 2020).

A Figura 8 apresenta a evolução do número de publicações ao longo dos anos da palavra-chave “*Industry 4.0*”. As publicações, que começaram a surgir no ano de 2012, vem crescendo ano a ano. Nos três últimos anos, a média de crescimento do número de publicações foi de mais de 60% ao ano, confirmando o crescente interesse dos pesquisadores para esse tópico.

Figura 8 - Publicações por ano: “*Industry 4.0*”



Fonte: Scopus (2020).

Apenas sete das 447 publicações encontradas com as palavras-chave “*Mixed-Model Assembly Line*”, também continham “*Industry 4.0*” no seu título, resumo ou palavras-chave, como mostrado na Tabela 1. A primeira publicação encontrada é do ano de 2016, e quatro publicações são de 2020.

Do aspecto prático, além de contribuir com uma interface que possibilite a obtenção de boas soluções por meio da aplicação do método de resolução proposto, para suportar um processo de decisão real e diário; pretende-se contribuir com o avanço rumo a um sistema de manufatura inteligente.

1.3 QUESTÕES DA PESQUISA E OBJETIVOS

Expostas a contextualização, justificativa e contribuições científicas, surgem as seguintes questões de pesquisa:

- a) O PILS é eficiente no sequenciamento de MMAL?
- b) Os resultados obtidos no caso real melhoram o desempenho da linha, em relação ao que é praticado?

O foco deste trabalho é apresentar uma abordagem simples e eficiente para a resolução do problema de sequenciamento de linhas de montagem de modelos mistos multiobjetivo. Sendo assim, o objetivo geral deste estudo é obter um sequenciamento eficiente de MMAL, considerando simultaneamente manter taxa constante de consumo de componentes e minimizar o número de setups, por meio da aplicação da meta-heurística multiobjetivo PILS, capaz de reduzir a variabilidade do consumo de componentes ao longo dos períodos.

A seguir são apresentados os objetivos específicos que se atendidos, juntamente com as repostas às questões de pesquisa, fica caracterizada a consecução do objetivo geral. Como objetivos específicos, podem-se considerar:

- Implementar o PILS para solução deste problema, sem garantir a obtenção da solução ótima, mas uma solução viável e eficiente;
- Realizar experimentos com o algoritmo implementado;
- Aplicar o algoritmo em um caso real.

1.4 PROCEDIMENTO METODOLÓGICO E MATERIAIS

Neste trabalho é tratado o problema do sequenciamento de MMAL multiobjetivo, em que se busca otimizar, simultaneamente, a distância entre o consumo real e o ideal de componentes, e o número de setups.

Este trabalho é classificado como uma pesquisa aplicada, com objetivos empíricos normativos, pelo fato de que os resultados do modelo de otimização, têm a finalidade de desenvolver políticas e estratégias que melhorem uma situação atual. A abordagem do problema é quantitativa, sendo o meio a pesquisa de campo, e o método modelagem e simulação (BERTRAND; FRANSOO, 2002).

A base de dados deste trabalho é constituída pela quantidade de cada modelo que deve ser sequenciada em uma linha, e a lista de componentes consumidos por cada modelo. Duas fontes para obtenção dessas bases de dados foram utilizadas: a literatura, para que fosse feita a validação da solução proposta; e o ERP (ERP, do inglês *Enterprise Resource Planning*) da empresa, para obtenção dos dados utilizados para aplicação ao caso real. Para a extração dos dados do ERP foi utilizado o MS-Query.

Para a implementação da meta-heurística aplicada ao sequenciamento, foi utilizado o software Dev-C++. Ele se trata de um Ambiente de Desenvolvimento Integrado (IDE, do inglês *Integrated Development Environment*) disponibilizado pela BloodshedSoftware e de acesso livre. Ele é utilizado para compilar programas escritos nas linguagens C/C++ e é aplicado neste trabalho, para a implementação do algoritmo para o sequenciamento. A versão utilizada foi a 5.11.

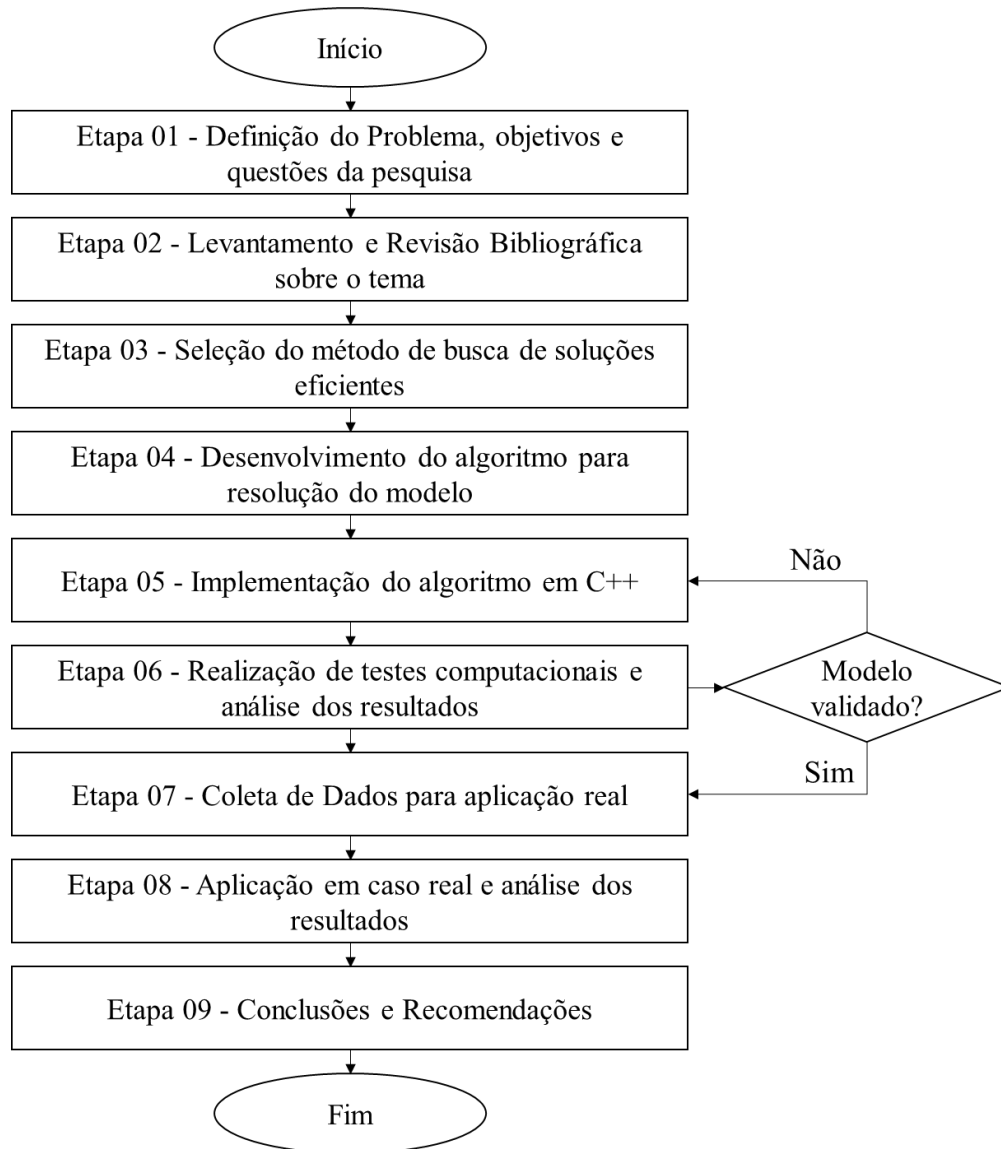
Para a análise dos resultados obtidos, foi utilizado o Microsoft® Excel® for Office 365. O Excel possibilita a organização dos dados, a execução de cálculos, e a análise visual de resultados de modo prático e eficaz.

Para a execução das etapas deste trabalho, que são detalhadas a seguir, utilizou-se como referência o fluxograma exibido na Figura 10.

Etapa 01 – Definição do problema, objetivos e questões da pesquisa

Identifica-se o problema a ser tratado pelo modelo proposto, o sequenciamento de MMAL, as perguntas que se pretende responder e o que se pretende alcançar com o modelo – neste caso, a otimização simultânea da distância entre o consumo real e o ideal de componentes, e o número de setups, que no contexto da linha real estudada, é o objetivo buscado.

Figura 10 - Fluxograma para aplicação da abordagem proposta



Fonte: Elaborada pela autora.

Etapa 02 – Levantamento e Revisão Bibliográfica sobre o tema

Analisa-se as características dos modelos de sequenciamento de MMAL existentes na literatura como ponto de partida para construir o modelo proposto. Apesar da combinação dos objetivos propostos neste trabalho ser inédita, outros trabalhos considerando estes objetivos individualmente, ou com outros objetivos, podem ser encontrados.

Etapa 03 – Seleção do método de busca de soluções eficientes

A partir da leitura de publicações, seleciona-se um método de busca de soluções que seja aderente ao problema, que tenha apresentado boas soluções em outras aplicações, e que

seja de baixa complexidade, para que seja mais bem aceito pelos usuários e de fácil manutenção posterior.

Etapa 04 – Desenvolvimento do algoritmo para resolução do modelo

Desenvolve-se o algoritmo definindo parâmetros, variáveis e as funções objetivos utilizadas para obtenção dos resultados esperados, e os componentes necessários para aplicação do método solucionado, incluindo representação da solução, procedimento de busca e explorações de soluções.

Etapa 05 – Implementação do algoritmo em C++

Implementa-se o algoritmo definido na linguagem escolhida, C++. Nesta etapa, faz-se o uso de uma instância simples (e que, portanto, seja possível obter todos os sequenciamentos possíveis de forma manual), gerada de forma aleatória, para checar se a implementação do algoritmo está gerando os resultados esperados.

Etapa 06 – Realização de testes computacionais e análise dos resultados

Realiza-se testes com instâncias disponíveis na literatura, para que seja possível comparar os resultados obtidos pelo algoritmo implementado com aqueles obtidos por outros autores utilizando outros métodos, assim como seus tempos de execução, e verificar a eficácia e eficiência da solução proposta. Uma vez verificada a eficácia e eficiência, prossegue-se para a próxima etapa.

Etapa 07 – Coleta de dados para aplicação real

Coleta-se os dados reais a partir do sistema ERP da empresa que incluem: os modelos a serem sequenciados, a demanda de cada um deles ao longo de um período (um turno) e a lista de materiais de cada um deles.

Etapa 08 – Aplicação em caso real e análise dos resultados

Aplica-se o modelo a um caso real e verifica-se se o objetivo proposto é atingido.

Etapa 09 – Conclusões e recomendações

A partir dos resultados, conclusões sobre o modelo proposto e recomendações para trabalhos futuros podem ser geradas.

1.5 DELIMITAÇÃO DA PESQUISA

Dentre as diferentes resoluções de problemas de sequenciamento de MMAL, dois objetivos gerais podem ser identificados: a minimização da sobrecarga de trabalho e os objetivos Just-in-Time. Segundo Boysen, Fliedner e Scholl (2009a), esses dois objetivos foram adotados em três diferentes abordagens do sequenciamento:

- Sequenciamento de modelos mistos (*Mixed-model sequencing*): visa minimizar a sobrecarga de trabalho dependente da sequência, considerando tempo de operação, movimentos do operador, e outras características operacionais;
- Sequenciamento de automóveis (*Car sequencing*): visa minimizar a sobrecarga de trabalho dependente da sequência de maneira implícita. Isso é possível formulando um conjunto de restrições e verificando se possíveis sequências de produtos inseridos na linha respeitam ou minimizam a violação de tais restrições.
- Programação nivelada (*Level scheduling*): enquanto as duas primeiras abordagens visam minimizar violações das restrições de capacidade, esta última busca encontrar sequências que estão alinhadas com a filosofia Just-in-Time. Para isto, uma taxa ideal de produção é definida, e os modelos são sequenciados de modo que as variações entre as taxas reais e ideais são minimizadas.

Este trabalho está inserido na terceira abordagem, aliando a ela o objetivo de minimização de setups.

Além disso, algumas premissas são consideradas na modelagem:

- Trabalho em linha reta, fluxo contínuo, sem estoques intermediários;
- Sequenciamento período único: o horizonte pode ser um dia ou um turno de trabalho;
- Não é considerado o comprimento da linha de montagem, ou seja, assume-se que quando um modelo é montado, todos os seus componentes são consumidos no instante que este modelo entra na linha;
- Os produtos mantêm a sequência por todos os postos de trabalho;
- Os modelos são sequenciados buscando o nivelamento de componentes apenas de primeiro nível, ou seja, utilizados diretamente na montagem final, não considerando, portanto, subconjuntos e matérias-primas;
- Os tempos de montagem dos diferentes modelos nas estações estão balanceados.

Quanto à aplicação do modelo desenvolvido a um caso real, o objeto de estudo limita-se a uma linha de montagem de importante empresa multinacional brasileira do ramo metalúrgico do segmento de autopeças.

Nesta linha de montagem, diversos modelos de chassis são montados por dia. Estes modelos possuem componentes comuns, mas também componentes distintos. O sequenciamento da linha é feito de forma que toda a demanda de modelos iguais seja produzida na sequência, em cada turno.

1.6 ESTRUTURA DO TRABALHO

Este trabalho está organizado em mais quatro capítulos, além deste, de Introdução. No Capítulo 2, apresenta-se a fundamentação teórica relacionada as Linhas de Montagem de Modelos Mistos, as meta-heurísticas, ao Problema de Sequenciamento de MMAL, e o *Pareto Iterated Local Search*. O Capítulo 3 apresenta a descrição da aplicação do PILS ao problema de sequenciamento de MMAL e, na sequência, o Capítulo 4, a os resultados e análises dos experimentos computacionais realizados. Por fim, no Capítulo 5 apresentam-se as conclusões e sugestões para estudos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

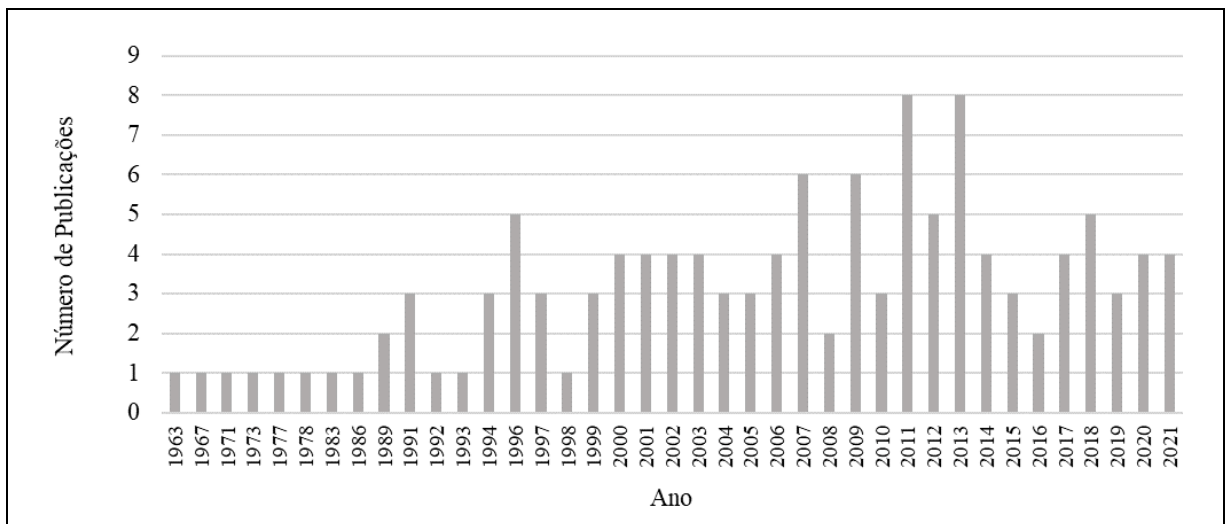
O objetivo da revisão sistemática é fornecer percepções coletivas por meio da síntese teórica de algum tema. Para os acadêmicos, o processo de revisão aumenta o rigor metodológico. Para profissionais, ajuda a desenvolver uma base de conhecimento confiável, acumulado de uma série de estudos (TRANFIELD; DENYER; SMART, 2003).

Para fundamentar esta pesquisa, partiu-se da busca de publicações na Base Scopus®, descrita na Seção 1.2 deste trabalho. Depois de um refinamento por idioma (inglês, português e francês), e de uma leitura dos resumos para identificar os trabalhos de maior afinidade com o tema, foram selecionados os artigos para serem lidos integralmente.

Após aprofundamento na leitura, sentiu-se necessidade de buscar mais informações, portanto, também foram utilizados outros documentos relevantes citados nos trabalhos identificados nas buscas e fruto de pesquisa em outras bases.

Na Figura 11, pode-se observar o ano de publicação dos documentos utilizados nesta pesquisa.

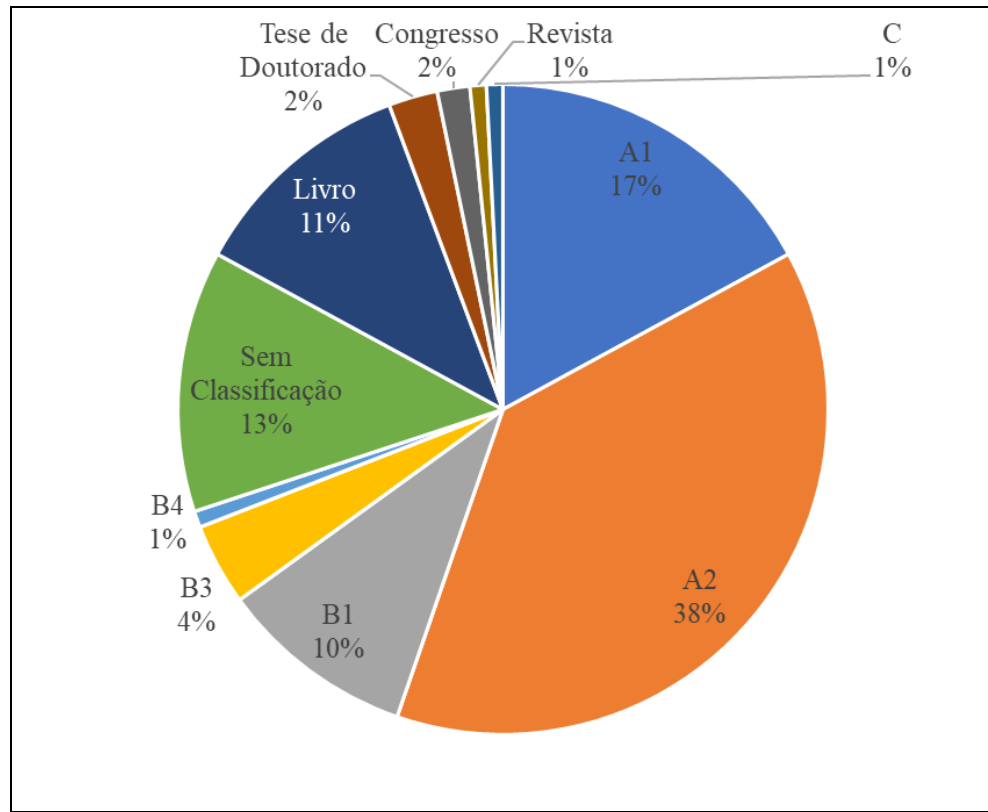
Figura 11 - Classificação das referências utilizadas: Ano de publicação



Fonte: Elaborada pela autora.

As publicações lidas são, em sua maioria, de periódicos, seguidas por livros e teses de doutorado. Do total destas publicações, 71% têm avaliação na WebQualis da CAPES, na área Engenharias III. No gráfico da Figura 12, pode-se observar que a maioria dos artigos tem avaliação A1-A2, o que indica que o trabalho foi fundamentado nas publicações com alta confiabilidade e expressão.

Figura 12 - Tipos de documento utilizados no trabalho + Classificação dos periódicos lidos na Área Engenharias III na WebQualis



Fonte: Elaborada pela autora.

Os periódicos mais utilizados para fundamentar este trabalho, assim com suas avaliações na WebQualis da CAPES, na área Engenharias III, são apresentados na Tabela 2.

Tabela 2 - Periódicos mais utilizados

Periódico	Avaliação	Quantidade
International Journal of Production Research	A2	15
Computers & Industrial Engineering	A2	12
European Journal of Operational Research	A1	8
Computers & Operations Research	A2	6

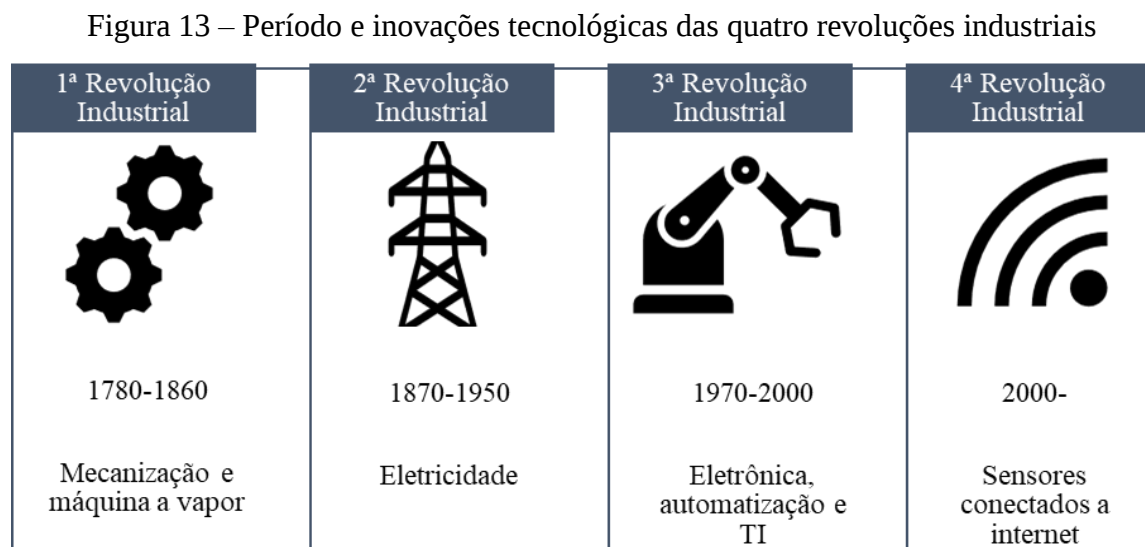
Fonte: Elaborada pela autora.

Nas próximas seções será apresentada a fundamentação teórica da Indústria 4.0, das Linhas de Montagem de Modelos Mistos e do Problema de Sequenciamento, das características do modelo de otimização, de Meta-heurísticas, e do PILS.

2.1 INDÚSTRIA 4.0

O ambiente industrial vive, desde o início dos anos 2000, sua quarta revolução industrial (BORTOLINI *et al.*, 2017). Este movimento também é chamado de Indústria 4.0, termo que foi proposto pela primeira vez em 2011 pelo governo alemão, como parte de um plano estratégico para colocar o país em um novo patamar de industrialização (KAGERMANN *et al.*, 2013).

As três primeiras revoluções industriais surgiram como resultado da mecanização, eletricidade e da Tecnologia da Informação (ZHONG *et al.*, 2017) e agora, a quarta revolução industrial traz para a indústria de manufatura os benefícios de muitos dos paradigmas tecnológicos, incluindo sistemas ciber-físicos, Internet das Coisas, *Big Data*, Computação em Nuvem e Inteligência Artificial (AZIZ, 2020). A Figura 13 mostra o período e as inovações tecnológicas de cada uma das revoluções.



Fonte: Adaptado de Bortolini *et al.* (2017).

É bem conhecido que a Indústria 4.0 tem um importante impacto estratégico de longo prazo no desenvolvimento industrial global, para alcançar maior eficiência, competência e competitividade (XU; XU; LI, 2018). Neste contexto, novos meios para transformar o sistema de manufatura tradicional em um sistema inteligente são fundamentais (ZHONG *et al.*, 2017).

Benotmane e Dudás (2020) descreveram como a transformação do paradigma tradicional para o inteligente, baseado no conceito da Indústria 4.0, pode ser alcançada. Nela, apresentaram uma comparação entre a indústria automotiva atual e seu futuro, que pode ser vista no Quadro 1.

Quadro 1 – Comparação entre indústria automotiva tradicional e futura

Indústria automotiva tradicional	Indústria automotiva futura
- Os recursos são limitados e não sustentáveis por muito tempo - Linhas de produção fixas e máquinas não-reconfiguráveis, que tornam a flexibilidade limitada - Produtos limitados (modelos de carros proposto pelas montadoras) - Máquinas tem tempos de parada e a manutenção mais difícil de prever antes de quebras - Intervenção de operador humano em caso de problema ou manutenção - Produto (carro) baseado em combustível diesel causa poluição ao meio ambiente - Produtividades mais baixas comparadas à indústria futura com grande <i>lead time</i>, resultando em baixa competitividade - Falta de uso dos conceitos de Internet das Coisas para coletar dados analíticos para processos e alcançar a segurança cibernética da empresa - Uma força de trabalho maior deve ser aplicada, resultando em custos de mão-de-obra mais elevados - Uso de materiais pesados com custos mais altos	- Produção sustentável devido ao uso de recursos eficientes - Aplicação de linhas de produção e sistemas logísticos flexíveis e reconfiguráveis - Diversidade de produtos, que podem satisfazer as demandas dos clientes - Uso de realidade aumentada e ferramentas de simulação para prever erros e mal funcionamentos - Uso do conceito cooperativo de máquina-máquina ou máquina-humano com aplicação de Inteligência Artificial - Produto (carro) é elétrico e autônomo, emitindo menos poluentes e barulho para o ambiente - Produtividade mais alta com boa qualidade, baixo <i>lead time</i>, resultando em empresas mais competitivas - Aplicação eficiente de Internet das Coisas e segurança cibernética para garantir melhor operação e segurança dos dados armazenados - Instalação de aparelhos mais tecnológicos e uso de mão-de-obra mais qualificada para operar estes sistemas inteligentes - Uso crescente de materiais leves em componentes de fabricação e estrutura do veículo

Fonte: Adaptado de Benotsmane; Dudás (2020).

Analisando o Quadro 1, pode-se constatar que tanto a aplicação de Inteligência Artificial; quanto a aplicação de linhas de produção flexíveis, para permitir uma maior diversidade de produtos para satisfazer as demandas dos clientes, conceitos abordados neste trabalho, estão inseridas na visão de indústria automotiva futura.

As técnicas de Inteligência Artificial têm sido usadas em vários segmentos das empresas, assumindo um papel proeminente para integrar pessoas, informações e produtos, incluindo o gerenciamento de várias operações de manufatura, logística e varejo (BENOTSMANE; DUDÁS, 2020). Dentro de um sistema inteligente, ela desempenha um papel essencial,

auxiliando, entre outras coisas, na tomada de decisão, incluindo dos problemas de planejamento e sequenciamento de produção (ZHONG *et al.*, 2017).

Já para atender a flexibilidade necessária para produzir produtos cada vez mais individualizados, os sistemas que ganham importância e que mais tem se desenvolvido são as Linhas de Montagem de Modelos Mistos (MMAL, do inglês *Mixed-Model Assembly Lines*) (KRENCZYK; DZIKI, 2020, ZHANG; XU; ZHANG, 2021).

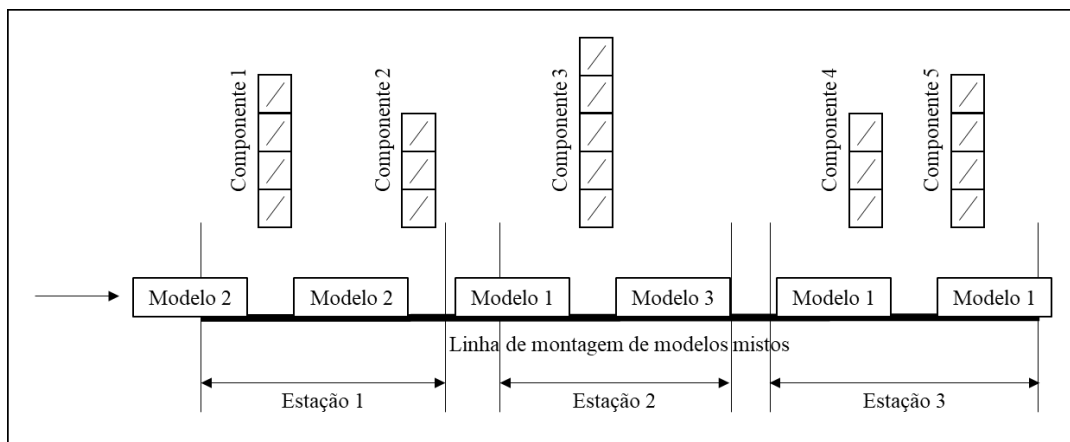
2.2 LINHAS DE MONTAGEM DE MODELOS MISTOS

Na busca por sistemas produtivos mais eficientes, em um contexto de flexibilização da produção para atender as exigências do mercado e de redução de desperdícios, surgem as Linhas de Montagem de Modelos Mistos (MMAL, do inglês *Mixed-Model Assembly Lines*), que possibilitam a produção de vários modelos de um mesmo produto em uma mesma linha.

A indústria automotiva, onde produções de grandes volumes devem ser combinadas com variadas modificações em pequenas quantidades exigidas pelos clientes, é um bom exemplo da aplicação de tais linhas (DUPLAGA; HAHN; HUR, 1996).

Nessas linhas de montagem, o processo é dividido em elementos menores, chamados de tarefas, que são designadas às estações que compõe a linha. Os produtos passam pelas estações, e as tarefas são realizadas obedecendo as relações de precedência entre elas (GÜDEN; MERAL, 2016). Um exemplo de MMAL é mostrado na Figura 14.

Figura 14 - Exemplo de uma Linha de Montagem de Modelos Mistos



Fonte: Adaptado de Xiaobo *et al.* (2006).

Além disso, outras características podem definir o design das MMAL, como as mostradas no Quadro 2.

Quadro 2 - Classificação das MMAL

Classificação	Descrição
Quanto ao sistema de transporte	Esteira em velocidade constante.
	Sistema estacionário , em que o produto permanece estacionário em cada posto de trabalho, e, após conclusão da tarefa atual: produto é transferido para o posto seguinte ou produto permanece no posto e os operadores da próxima operação substituem os operadores atuais.
Quanto a relação do produto com o sistema de transporte	Produtos fixos : produtos que não podem ser movidos independentemente do transportador. Podem ser girados em qualquer direção, mas não podem ser removidos de sua posição no transportador. Nesta configuração, normalmente, não há previsão para estoque regulador entre postos da linha de montagem, então os operadores geralmente trabalham em condições compassadas.
	Produtos móveis : produtos com movimentos independentes. Este tipo de produto na linha permite o acúmulo de estoque regulador entre os postos, tendo como consequência o descompasso do trabalho dos operadores.
Quanto as estações de trabalho	Fechados : é indesejável, ou mesmo impossível, que os operadores de estações adjacentes violem os limites de uma estação fechada. Portanto, os tempos disponíveis para completar as tarefas são limitados.
	Abertos : os limites de estações abertas podem ser cruzados, porém a extensão do cruzamento pode, ou não, ser restringida. Em ambos os casos, o requisito é que não ocorra interferência entre operadores adjacentes. Neste caso, há flexibilidade nos tempos disponíveis para completar tarefas alocadas em postos abertos.
Quanto à disciplina de entrada dos produtos na linha	Taxa fixa : o período de lançamento é a média ponderada do tempo total de montagem para todos os produtos a serem montados, durante todas as estações.
	Taxa variável : o período de lançamento é o tempo de processo do último produto lançado no primeiro posto, ou seja, concluindo os trabalhos sobre o produto atual, o operador do primeiro posto pode começar imediatamente a trabalhar no próximo produto.
Quanto ao tempo de processo	Determinístico : a natureza estocástica dos tempos das tarefas pode ser negligenciada, de modo que os tempos são considerados estáticos e determinísticos. A premissa é justificada se as variações são pequenas.
	Estocástico : assume-se que os tempos de processo seguem uma distribuição conhecida (ou até mesmo desconhecida).
Quanto ao layout da linha	Linha reta : as estações são dispostas em série ao longo do fluxo da linha.
	Linha em U : a linha em U permite que os operadores trabalhem em mais de um modelo por ciclo porque tem acesso a mais de uma estação simultaneamente, o que influencia o sequenciamento consideravelmente.
	Linhas de alimentação : uma ou mais linhas de alimentação fluem para a linha principal. Operadores podem realizar tarefas em ambas as linhas.

Fonte: Adaptado de Dar-El (1978); Boysen; Flidner; Scholl (2009a).

Além do design, segundo Monden (1983), para utilizar as MMAL de forma efetiva, o procedimento para projetá-las deve considerar:

1. Determinação do tempo de ciclo da linha;
2. Cálculo de um número mínimo de processos;
3. Elaboração de um diagrama de relações de precedência integradas entre as operações;
4. Balanceamento da linha;
5. Determinação da sequência de programação para a introdução dos produtos na linha;
6. Determinação da duração das operações de cada processo.

Desde que se começou a estudar a utilização efetiva de MMAL, muitos trabalhos têm focado no balanceamento e sequenciamento dessas linhas, inclusive simultaneamente.

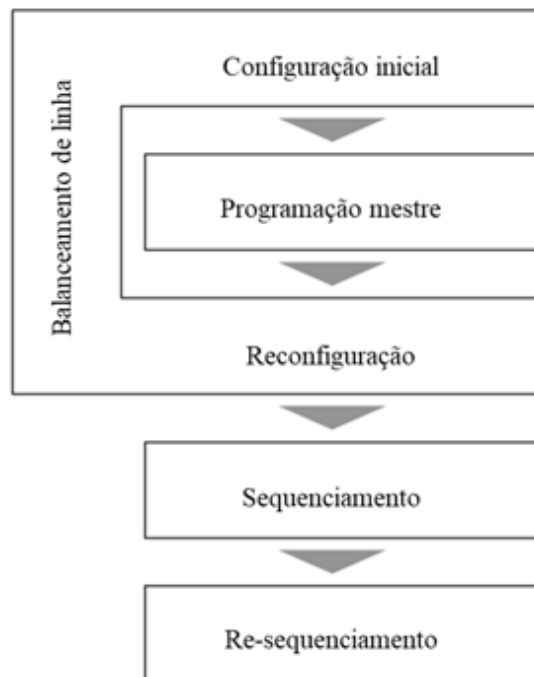
Entre os trabalhos que consideraram o balanceamento e o sequenciamento de forma simultânea está o de Kim, Kim e Kim (2000), que apresentou um novo método utilizando algoritmo coevolutivo, com objetivo de minimizar o trabalho extra; e o de Saif *et al.* (2019) que aborda o problema com múltiplos objetivos, buscando minimizar a variação de consumo de componentes, o tempo total de montagem e o custo de penalidade de atraso de entrega simultaneamente, aplicando o método proposto Colônia de Abelhas Artificial Multiobjetivo (MABC, do inglês *Multi-objective Artificial Bee Colony*).

Boysen, Fliedner e Scholl (2009b) chamam atenção para o fato de que o principal problema deste tipo de abordagem decorre dos momentos diferentes destas duas decisões. Enquanto o balanceamento está no planejamento de médio prazo, quando os planos de produção reais para cada turno são normalmente desconhecidos; as decisões sobre o sequenciamento da produção são tomadas no âmbito do curto prazo. Portanto, parece importante decompor o problema de decisão total em etapas de planejamento separadas, porém coordenadas, como mostrado na Figura 15.

Inseridos nestas etapas de planejamento de produção de MMAL, vários estudos relevantes podem ser encontrados na literatura, como: design de MMAL (BUKCHIN; DAR-EL; RUBINOVITZ, 2002), modelagem da complexidade de manufatura em MMAL (BUKCHIN; RABINOWITCH, 2006), balanceamento de MMAL (SIMARIA; VILARINHO, 2009), sequenciamento de linhas em U (LI; GAO; SUN, 2012; LIAN *et al.*, 2012; HAMZADAYI; YILDIZ, 2013), sequenciamento de linhas paralelas (KUCUKKOC; ZHANG, 2014), sequenciamento do abastecimento de componentes de MMAL em

supermercados integrados na linha (BOYSEN; EMDE, 2014). O restante desta revisão, contudo, focará apenas no problema de sequenciamento de MMAL em linha única e reta, como abordado neste estudo.

Figura 15 - Etapas de planejamento da produção de modelos mistos



Fonte: Adaptado de Boysen; Fliedner; Scholl (2009b).

2.3 PROBLEMA DE SEQUENCIAMENTO DE LINHAS DE MONTAGEM DE MODELOS MISTOS

Desde os anos 60, as MMAL já eram utilizadas em indústrias que desejavam manter vários modelos em produção, ao invés de produzir lotes para estocar.

Muitos estudos tratando da utilização eficaz das MMAL se concentram no sequenciamento de tais linhas, a fim de garantir a sua flexibilidade para produzir diferentes modelos em dadas sequências e obter benefícios, tais como: a taxa de utilização constante de componentes críticos consumidos em uma linha, a redução da sobrecarga dos postos de trabalho e das paradas para preparação ou ajuste de máquinas, e a diminuição dos estoques intermediários (GHINATO, 2000).

Nos trabalhos desenvolvidos, também foram feitas análises combinando dois ou mais objetivos. A revisão destes trabalhos foi segmentada entre os com um único objetivo e os com múltiplos objetivos.

2.3.1 Problemas com um único objetivo

2.3.1.1 Minimizar ineficiências utilizando penalizações

Thomopoulos (1967) deu um passo importante na tentativa de buscar soluções eficientes de sequenciamento, com a criação um procedimento para sequenciar uma MMAL com objetivo de otimizar a utilização dos operadores da linha de montagem, buscando minimizar as ineficiências (neste caso, a ociosidade, a falta ou sobrecarga de trabalho e mão de obra extra). O procedimento é baseado no cálculo de custo de penalização das ineficiências resultantes do sequenciamento de vários modelos.

2.3.1.2 Minimizar comprimento total da linha

Dar-El e Cucuy (1977) apontam que a fraqueza do método proposto por Thomopoulos (1967) é a natureza arbitrária das penalidades utilizadas, e propõem, o que definem como, a primeira tentativa bem-sucedida de resolver o sequenciamento de MMAL de forma ótima. O algoritmo proposto tem como restrição que as linhas já estejam balanceadas, e minimiza o comprimento total da linha, para zerar o tempo ocioso de cada estação. Ele é destinado a ser usado para o design e planejamento de novas MMAL.

A importância deste objetivo é minimizar o custo de investimento da linha, que pode aumentar proporcionalmente ao seu comprimento (BOYSEN; FLIEDNER; SCHOLL, 2009a).

Trazendo dois elementos diferentes para a análise: MMAL híbridas que comportam tanto estações abertas, como fechadas, e tempos de setup dependentes da sequência, Kim, Hyun e Kim (1996) apresentaram um novo algoritmo genético que conseguiu atingir soluções quase-ótimas em um tempo computacional razoável.

2.3.1.3 Minimizar a sobrecarga de trabalho

Mais tarde, muitos trabalhos passaram a focar no sequenciamento de MMAL com o objetivo de nivelar a carga (tempo total de operação) em cada estação de trabalho, e de alinhar produção e demanda, ou seja, aproximar o consumo real ao consumo ideal de componentes.

Um dos pioneiros nestes estudos foi Monden (1983), que propôs uma abordagem para o sequenciamento das MMAL, onde é considerada uma linha abastecida por vários *kanbans*, e o consumo de componentes acontece simultaneamente. Segundo ele, o sequenciamento tem

basicamente duas metas.

A primeira meta é o nivelamento da carga (tempo total de montagem) em cada estação (l) da linha. Esta meta reconhece que os produtos podem não ter o mesmo tempo de operação em cada estação da linha, e podem até exceder o tempo de ciclo pré-determinado. Portanto, impõe que o sequenciamento deve respeitar a condição (1):

$$\max (l) \left\{ \frac{\sum_{i=1}^I Q_i T_{il}}{\sum_{i=1}^I Q_i} \right\} \leq c \quad (1)$$

sendo:

Q_i = quantidade planejada de produção do produto A_i ($i=1 \dots, I$);

T_{il} = tempo de operação do produto A_i no processo l ;

C = tempo de ciclo (*takt time*) dado por (2):

$$C = \frac{\text{Tempo total de operação por dia}}{\sum_{i=1}^I Q_i} \quad (2)$$

A MMAL deve ser capaz de lidar com esta condição, por exemplo, evitando o sequenciamento sucessivo de produtos com tempos de operação relativamente mais longos, para prevenir atrasos ou paradas de linha (ZHUQI; SHUSAKU, 1994).

No trabalho de Monden (1983) esta primeira meta é incorporada ao algoritmo de solução que considera principalmente a segunda meta.

Posteriormente, outros trabalhos consideraram a minimização da sobrecarga total de trabalho – ou seja, o tempo de processamento do trabalho que não é completado dentro do tempo do tempo de ciclo, como a heurística de Yano e Rachamadugu (1991) e os procedimentos propostos por Bautista e Cano (2008).

Aroui, Alpan e Frein (2017) propuseram três meta-heurísticas: Algoritmos Genéticos (GA, do inglês *Genetic Algorithms*), Recozimento Simulado (SA, do inglês *Simulated Annealing*), e uma combinação destas duas (GA-SA), para obter um sequenciamento que minimize a sobrecarga de trabalho. Elas foram testadas utilizando dados de uma linha de montagem de caminhões. Os resultados foram comparados com soluções obtidas utilizando programação linear inteira mista (MILP, do inglês *Mixed Integer Linerar Programming*), e mostram que o GA-SA teve a melhor performance, melhorando a melhor solução viável encontrada pelo MILP depois de três horas, em menos de 20 minutos.

O sequenciamento de modelos para minimizar o trabalho total inacabado dentro da estação de trabalho foi discutido em Zandieh e Moradi (2019). Um novo algoritmo imperialista competitivo (ICA, do inglês *Imperialist Competitive Algorithm*) foi desenvolvido para resolver problemas de todos os tamanhos. Resultados comparados com outros algoritmos da literatura mostraram que o ICA fornece boas soluções com baixos custos computacionais, superando os demais quanto a qualidade das soluções.

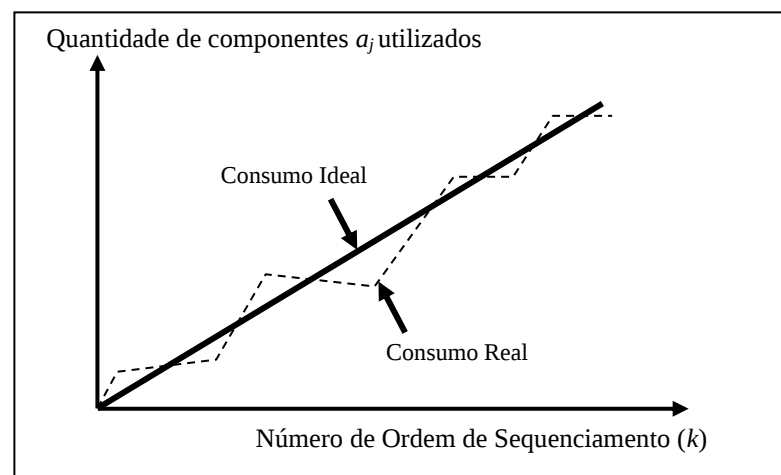
Como a sobrecarga de trabalho geralmente é evitada pela designação de trabalhadores extra, este objetivo também pode ser chamado de minimização do trabalho extra total (BOYSEN; FLIEDNER; SCHOLL, 2009a), que também aparece em uma série de trabalhos, especialmente com múltiplos objetivos.

2.3.1.4 Minimizar a variação de consumo de componentes

Este tipo de problema é classificado por Kubiak (1993) como Variação da Taxa de Saída (ORV, do inglês *Output Rate Variation*) e se aplica ao caso em que as saídas requeridas para cada modelo são distintas. O ORV tem como objetivo minimizar a variação na taxa de consumo de todos os componentes durante a fabricação dos modelos, que é o caso apresentado por Monden (1983).

A segunda meta do trabalho de Monden (1983) é manter uma taxa constante de utilização dos componentes usados na linha e, para resolvê-la, foi proposto o Algoritmo *Goal Chasing* (GC). Neste modelo, o problema consiste em minimizar a distância entre o consumo ideal e o consumo real de componentes, como é mostrado na Figura 16.

Figura 16 - Consumo real versus consumo ideal de determinado componente a_j na linha



Fonte: Sanches; Marins; Montevechi (2010).

Uma crítica ao algoritmo proposto por Monden (1983) é que ele considera apenas os componentes de Nível 1, ou seja, componentes utilizados diretamente na linha de montagem, deixando de lado partes que componham estes componentes, matéria-prima, etc. (LEU; MATHESON; REES, 1996). Apesar disso, ele, que pode ser considerado satisfatório, abre caminho para a evolução de trabalhos mais complexos.

Preenchendo esta lacuna, Miltenburg e Sinnamon (1989) apresentaram um modelo multinível para o sequenciamento de MMAL. Nele, considera-se as variações de consumos no nível de produto, subconjunto, componente e matéria-prima. Eles propuseram duas heurísticas para resolvê-lo e mostraram que o GC é um caso especial de uma dessas heurísticas.

Além disso, outra característica do GC é que, a cada instante, é escolhido para entrar no sequenciamento o produto que vai minimizar a distância entre o consumo ideal e o consumo real de componentes neste instante, não considerando, portanto, o efeito que isto causará no restante do sequenciamento e, conseqüentemente, nas demais distâncias. Por este motivo, outra crítica a este algoritmo é que ele é considerado míope (BAUTISTA; COMPANYS; COROMINAS, 1996).

Com objetivo de atenuar esta característica míope do GC, Bautista, Companys e Corominas (1996) propuseram heurísticas para resolver o problema de sequenciamento com o mesmo objetivo e conseguiram melhorar os resultados anteriores.

Os autores testaram três tipos de modificação para melhorar o procedimento de sequenciamento: simetria, construindo o caminho pela união dois segmentos, um começando do primeiro instante, e outro do último; horizonte, olhando a cada instante dois instantes além do atual, ao invés de apenas o próximo; e preservação da taxa dos modelos, evitando que haja um empobrecimento da composição do restante das unidades a serem sequenciadas, ou seja, que diferenciem muito da composição inicial.

Uma comparação com os resultados de um procedimento exato proposto, a Programação Dinâmica Limitada, mostrou que a modificação quanto ao horizonte foi a que aproximou mais o resultado ao ótimo.

A Programação Dinâmica é uma técnica que busca uma solução para um problema, combinando a solução de subproblemas, que podem apresentar soluções dependentes, ou seja, afetar a próxima solução (CORMEN; LEISERSON; RIVEST, 2002). Um algoritmo de Programação Dinâmica resolve cada subproblema uma vez e memoriza sua resposta, evitando assim recalcular a resposta quando o subproblema é encontrado.

Apesar do desenvolvimento tecnológico na área computacional, a desvantagem encontrada na aplicação da Programação Dinâmica encontra-se no volume de restrições de

demanda, ou seja, na grande dependência das soluções dos subproblemas, exigindo abordagens específicas e funções complexas, como observado em Celano, Costa e Fichera (2004).

Por esta razão, com o avanço nas aplicações de meta-heurísticas, diversos autores propuseram resoluções aplicando-as ao problema de sequenciamento de MMAL com objetivo de minimizar a variação no consumo de componentes.

Leu, Matheson e Rees (1996) introduziram a aplicação de GA, gerando uma solução em questão de segundos que melhora a solução do GC. Ao contrário do GC, o procedimento proposto permite que todas as etapas do sequenciamento sejam avaliadas em conjunto.

Mais tarde, Zhu e Zhang (2011) transformaram o problema de sequenciamento em um modelo de grafos para realizar testes com a Otimização por Colônia de Formigas (ACO, do inglês *Ant Colony Optimisation*) e a Otimização por Colônia de Formigas com Formiga Elitista (ACOE, do inglês *Ant Colony Optimisation with Elitist Ant*) e obtiveram resultados que superaram tanto o GC quanto o GA.

Retomando uma outra linha, com o objetivo de aprimorar o procedimento de Miltenburg e Sinnamon (1989) para múltiplos níveis de componentes, investigando vários caminhos para achar a solução no espaço de busca explorado, ao invés de um único como no original, Leu, Huang e Russel (1997) aplicaram a Busca em Feixe (BS, do inglês *Beam Search*) e apresentaram resultados de melhor qualidade que o anterior, sem grandes aumentos no custo computacional, para diversas instâncias. Com o mesmo objetivo, Ponnambalam, Aravindan e Rao (2003) aplicaram o GA, que também obteve bons resultados.

Ainda com objetivo de minimizar a variação do consumo de componentes, Xiaobo e Zhou (1999) propuseram um sequenciamento de MMAL com múltiplos postos de trabalho. A diferença com relação aos trabalhos anteriores é que o algoritmo proposto, denominado *Goal Chasing* Modificado (MGC, do inglês *Modified Goal Chasing*), não considera o consumo de todos os componentes em um único instante, mas, sim, o acumulado de componentes nos diferentes postos no momento que são usados, para determinar a melhor sequência de modelos para entrar na linha de montagem. Além disso, também aplicaram o SA ao problema. Em geral, o MGC pôde gerar boas soluções, porém não ótimas. Por outro lado, para exemplos de pequenas instâncias, o SA conseguiu atingir os valores ótimos. Para instâncias maiores, o algoritmo MGC pôde gerar boas soluções com baixos tempos computacionais, enquanto o SA pôde obter melhores soluções com maiores, porém aceitáveis, tempos computacionais. Conclui-se, portanto, que para instâncias grandes, pode-se haver um *trade off* entre a precisão e o tempo computacional para escolher um dos métodos.

Celano, Costa e Fichera (2004) também aplicaram o SA ao problema de sequenciamento e compararam ao GC e MGC. As análises mostraram que na maioria dos casos, o SA superou o desempenho das heurísticas.

Apesar de haver muitos estudos sobre o sequenciamento MMAL, são poucos os que mostram sua aplicação em empresas. Com o objetivo aqui discutido, o trabalho que se destaca é o de Duplaga, Hahn e Hur (1996), que descreveram a abordagem utilizada no sequenciamento de carros na Hyundai Motor Company. Eles criaram um procedimento heurístico baseado no GC. Para simplificar os cálculos, ao invés de considerar todos os componentes no processo, para aqueles que tinham somente duas opções, apenas um dos componentes era inserido no modelo pois, desta forma, o outro seria consequentemente nivelado. Para eles, esta versão foi capaz de gerar bons resultados em um tempo conveniente.

2.3.1.5 Minimizar a variação da taxa de produção de cada modelo

Esses problemas são classificados por Kubiak (1993) como Variação da Taxa de Produto (PRV, do inglês *Product Rate Variation*), consistem em sequenciamentos que minimizam a variação da taxa com que diferentes modelos são produzidos na linha, e não necessariamente minimizam a variação na demanda de componentes de processos que abastecem a linha.

Essa abordagem, que deriva da exposta na subseção anterior, surge no trabalho de Miltenburg (1989). Nela, supõe-se que todos os produtos requerem aproximadamente o mesmo número e *mix* de componentes e, portanto, considera-se apenas a taxa de demanda dos produtos, ignorando as taxas de demanda dos componentes. Ou seja, o objetivo é sequenciar a linha de forma que a proporção dos produtos produzidos durante um determinado período é a mais próxima possível da taxa de demanda destes produtos no período todo.

Nesta circunstância, duas heurísticas são propostas. A primeira se assemelha ao GC e, assim como esta, é míope, por não considerar o efeito posterior do sequenciamento de um produto em um dado instante. Já na segunda, a decisão do produto a ser sequenciado é baseada em dois instantes – o atual e o próximo. Aproxima-se a variabilidade ao longo destes dois instantes, e sequencia-se de modo a que a variabilidade seja a menor possível.

Miltenburg (1989) ainda alerta que, em muitos casos, nem todos os produtos têm aproximadamente o mesmo número e *mix* de componentes. Neste caso, os componentes de outros níveis devem ser considerados no sequenciamento da MMAL. Ele afirma que o método proposto também se aplica a este problema, por meio do ajuste da função objetivo – acrescentan-

do os termos de diferença entre o consumo ideal e real dos componentes de outros níveis.

Inman e Bulfin (1991), com as mesmas suposições de Miltenburg (1989), propõe um algoritmo com objetivo de manter a produção o mais próximo da demanda o possível, seguindo a Regra da Data Mais Cedo (EDD, do inglês *Earliest Due Date*), que consiste em minimizar a diferença entre o tempo que o produto é produzido e o tempo em que ele é requisitado. O EDD alcançou resultados tão bons quanto os do trabalho anterior, porém se destacando em termos de custo computacional, sendo relativamente rápido.

Kubiak e Sethi (1991), baseados na formulação do problema do sequenciamento proposta por Miltenburg (1989), propõe um problema de designação equivalente, que visa minimizar o custo de designar certa unidade de um produto em um dado instante, atribuindo custos de excesso de estoque, se o produto é produzido antes do tempo, e de falta, caso o produto seja produzido tarde demais, em relação a uma posição ideal calculada.

2.3.1.6 Minimizar o tempo de parada da linha

O conceito de Automação, no Sistema Toyota de Produção, admite que os trabalhadores parem a linha sempre que falharem em completar suas operações dentro dos limites de suas estações. Portanto, a paralisação da linha se torna um critério crucial em problemas de sequenciamento (XIAOBO; OHNO, 2000). Com objetivo de achar uma sequência ótima que minimiza o tempo total de parada de linha, Xiaobo e Ohno (1994) propuseram um *Branch and Bound* para resolução deste problema, com base nos limites inferior e superior do tempo total de parada e tempo ocioso.

Os mesmos autores também propuseram uma aplicação do SA para encontrar boas soluções sub-ótimas em problemas de grande escala. Os exemplos numéricos mostram que a qualidade das soluções obtidas pelo SA e o tempo computacional depende dos parâmetros de resfriamento, e para problemas de grande escala, boas soluções puderam ser obtidas em um tempo computacional aceitável (XIAOBO e OHNO, 1997).

Anos depois, considerando, desta vez, os tempos de caminhada dos trabalhadores, Xiaobo e Ohno (2000) propuseram um algoritmo heurístico para resolver problema de sequenciamento com o objetivo de minimizar o tempo total de parada da linha.

2.3.2 Problemas com múltiplos objetivos

Nos trabalhos de sequenciamento multiobjetivo de MMAL pode-se encontrar aplicações

usando tanto a abordagem da transformação dos objetivos em uma única função de avaliação, quanto a de buscar soluções Pareto-ótimas, como será mostrado a seguir. Eles serão divididos, novamente, pelos objetivos do sequenciamento.

2.3.2.1 Minimizar variação na carga de trabalho e a variação de componentes consumidos

Leu, Huang e Russel (1997) aplicaram BS para resolver um problema de sequenciamento de MMAL com uma função biobjetivo, apresentando uma fronteira de boas soluções de *trade-off*. O problema, que visa minimizar a variação de componentes consumidos e minimizar a variação na carga de trabalho, foi resolvido achando um conjunto de sequencias viáveis que satisfaçam um objetivo em um nível pré-determinado e, selecionado um sequenciamento final do conjunto de soluções para minimizar o outro objetivo.

Com os mesmos objetivos, Tamura, Long e Ohno (1999) propuseram dois algoritmos heurísticos, um baseado no GC e outro na Busca Tabu (TS, do inglês *Tabu Search*), e um algoritmo baseado em Programação Dinâmica, para resolver o problema de uma linha que conta com uma sublinha de apoio, e pode absorver variações da linha de montagem. A função avaliada neste caso foi a soma ponderada dos dois objetivos. Nos testes, o TS forneceu soluções melhores, enquanto o GC apresentou mais vantagem no tempo computacional.

Korkmazel e Meral (2001) mostram que o problema também pode ser formulado como um problema de designação, e combinam os objetivos usando pesos. Além disso, eles buscam entender em que momento vale a pena considerar o objetivo de minimizar a variação na carga de trabalho, uma vez que o outro é priorizado, e concluíram que isso ocorre especialmente nos casos em que a linha de montagem não é longa e/ou a variabilidade nos tempos de processo dos modelos é alta.

Buergin *et al.* (2018) propuseram um modelo com estes objetivos, e validaram em uma aplicação na indústria aeronáutica. A abordagem multiobjetivo neste caso envolveu a monetização dos objetivos. Os resultados mostraram melhora significativa em termos de menos sobrecarga de trabalho e maior proximidade entre demanda e produção.

2.3.2.2 Minimizar variação de componentes consumidos e paradas de linha

O trabalho de Kotani, Ito e Ohno (2004) fornece uma nova formulação para o problema de sequenciamento com dois objetivos, considerando minimizar o tempo total de parada de

linha com trabalhadores auxiliares, tendo como restrição manter uma taxa constante de utilização dos componentes. Para resolvê-la, eles propõem um algoritmo de duas fases que adota o GC, na primeira fase, e o SA, na segunda.

Al-e-hashem, Aryanezhad e Jabbarzadeh (2011) desenvolveram um algoritmo híbrido, unindo algoritmos genéticos e procedimento baseado em eventos para resolver o problema, com os mesmos objetivos, porém considerando uma linha de apoio. Para agregar os objetivos, como eles não tem a mesma dimensão, eles foram convertidos em termos de custos. A eficiência do algoritmo proposto é demonstrada por meio de vários testes e comparando os resultados com soluções ótimas, obtidos por enumeração completa.

2.3.2.3 Minimizar a variação da taxa de produção de cada modelo e número de setups

McMullen e Frazier (2000) utilizaram o SA para resolver um problema de sequenciamento de MMAL visando minimizar o número de setups e manter a taxa de produção de cada modelo constante. Os pesos foram definidos de modo que os dois objetivos tivessem a mesma contribuição, e depois foram ajustados para dar mais ênfase a cada um. Os resultados foram comparados com soluções obtidas por Busca Tabu (TS, do inglês *Tabu Search*) da literatura. As comparações mostraram que o SA fornece resultados superiores aos da TS, em termos de valor da função objetivo, e fornece soluções quase-ótimas para problemas menores.

McMullen também aplicou SA, TS e GA (McMULLEN, 2001c), Mapa Auto-Organizável de Kohonen (McMULLEN, 2001a) e ACO (McMULLEN, 2001b) para buscar fronteiras-Pareto para o problema de sequenciamento com objetivo de minimizar simultaneamente a variação das taxas de produção e do número de setups. Os resultados dos experimentos realizados mostram que as abordagens alcançam soluções quase-ótimas, sendo que para maiores instâncias, o ACO se destaca em termos de desempenho e tempo computacional.

Mansouri (2005) aplicou o Algoritmo Genético multiobjetivo (MOGA, do inglês *Multi-Objective Genetic Algorithm*) buscando resolver o mesmo problema. Comprado com a enumeração completa em problemas pequenos, o MOGA se mostrou comparável, com tempos computacionais consideravelmente menores. Na comparação com outros algoritmos existentes, o algoritmo proposto superou a qualidade das soluções.

Depois, Moradi, Zandieh e Mahdavi (2011) apresentaram um método chamado Algoritmo Genético de Classificação Não Dominada (NRGA, do inglês *Non-Dominated Ranking Genetic Algorithm*) para resolver o problema de sequenciamento e compararam com

os resultados do MOGA, mostrando qualidade superior.

Com os mesmos objetivos, Alpay (2009) utilizou o Procedimento de Busca Guloso, Aleatório e Adaptativo (GRASP, do inglês *Greedy Randomized Adaptive Search Procedure*) e comparou com resultados do SA, TS e GA da literatura. Para pequenas instâncias, o GRASP se mostrou melhor, ou ao menos competitivo com as demais heurísticas. Para instâncias maiores, o GRASP obteve resultados bem melhores do que as outras heurísticas, porém foi inferior em termos de esforço computacional.

Chutima e Pinkoompee (2009) estudaram a performance de buscas locais embutidas em Algoritmos Meméticos (MA, do inglês *Memetic Algorithms*) para obter soluções Pareto-ótimas para o sequenciamento de MMAL. Os resultados demonstraram que MA combinados com buscas locais apropriadas superaram outras meta-heurísticas na busca de soluções Pareto-ótimas.

2.3.2.4 Minimizar variação da carga de trabalho e setups

Bolat, Savsar e Al-Fawzan (1994) apresentam uma formulação generalizada para o problema de sequenciamento. Um procedimento *Branch and Bound*, para ser aplicado em problemas pequenos, e duas heurísticas, para quaisquer problemas de tamanho real, são desenvolvidas para minimizar o custo de setup e de trabalho extra simultaneamente. Os algoritmos testados em dados reais mostraram ser aplicáveis ao sequenciamento de MMAL em tempo real.

Liu *et al.* (2014) desenvolveram um modelo matemático para minimizar simultaneamente o custo de horas extras, tempo ocioso e de setup. Para resolvê-lo, propuseram a Busca Dispersa Avançada (ASS, do inglês *Advanced Scatter Search*), utilizando um novo método de geração de diversificação baseado em algoritmos genéticos para gerar um conjunto de soluções iniciais potencialmente diversas e de alta qualidade. O modelo e o algoritmo propostos foram aplicados e validados em uma MMAL em uma fabricante de ar-condicionado. Os resultados do ASS foram superiores ao SS, e GA em termos de qualidade e estabilidade das soluções.

2.3.2.5 Minimizar variação do consumo de componentes, minimizar setup e minimizar tempo extra/tempo ocioso

Um dos primeiros trabalhos investigando objetivos conflitantes foi o de Hyun, Kim e

Kim (1998), que considera três simultaneamente: minimizar o trabalho extra total, manter uma taxa constante consumo de componentes e minimizar custo de setup. Eles desenvolveram um novo algoritmo genético, o PS-NC GA, do inglês *Pareto Stratum-Niche Cubicle Genetic Algorithm*, que enfatiza a busca por um conjunto de diversas soluções não dominadas. O algoritmo proposto foi comparado com outros GAs existentes para várias instancias, e os resultados revelaram que o PS-NC GA teve um desempenho melhor que os anteriores, especialmente para problemas maiores e que envolvem maior variação no custo de setup.

Com os mesmos objetivos, porém adaptado para quatro níveis de componentes, Ponnambalam, Aravindan e Rao (2003) compararam o PS-NC GA com um GA utilizando a soma ponderada dos objetivos. Eles chegaram à conclusão que o primeiro performou melhor e que a abordagem convencional utilizando pesos para combinar os diferentes critérios pode levar o decisor a tomar uma decisão errada.

Akgündüz e Tunalı (2010), na resolução deste mesmo problema, concluíram que apesar do GA ter sido bem-sucedido no tratamento de diversos problemas de otimização, a escolha correta de seus parâmetros é um trabalho árduo. Além disso, diferentes valores dos parâmetros podem ser ótimos para diferentes fases do processo de busca, o que significa que o uso de valores de parâmetros fixos pode levar a um desempenho inferior. Assim, a fim de resolver o problema do sequenciamento de MMAL multiobjetivo de forma eficiente, propuseram uma abordagem adaptável baseada em GA e obtiveram soluções de melhor qualidade.

Tavakkoli-Moghaddam e Rahimi-Vahed (2006) buscaram diminuir os custos do trabalho extra, da variação da taxa de produção total e do setup, unindo os objetivos e atribuindo um peso a cada um e aplicaram o MA. Os resultados computacionais mostraram que o MA proposto tem a capacidade de competir com um software de modelagem e otimização, principalmente para instâncias maiores. Rabbani, Rahimi-Vahed e Torabi (2008) aplicaram o MA e o GA na resolução do sequenciamento de MMAL com a soma ponderada dos mesmos três objetivos, e novamente o MA mostrou um desempenho melhor.

Fattahi *et al.* (2012) compararam o desempenho de três algoritmos, o Algoritmo de Monden desenvolvido, o SA e Otimização simples de Colônia de Formigas (SACO, do inglês *Simple Ant Colony Optimization*). A função objetivo considerada foi a minimização do custo de tempo ocioso e do tempo extra e o custo da variação da entrega de componentes. Os resultados mostraram que o SA gerou soluções de melhor qualidade que os outros dois algoritmos, e que o SA tem mais destaque no custo total, custo de tempo ocioso e extra do que os outros dois. O Algoritmo de Monden desenvolvido no critério da entrega dos componentes,

alcança melhores resultados.

Salehi *et al.* (2013) compararam o desempenho do software Lingo, que utiliza o método *Branch and Bound*, com o SA na resolução do sequenciamento de MMAL sincronizadas, que são linhas onde as peças em processo ficam nas estações até serem todas processadas e depois são simultaneamente movidas para as próximas estações. Os objetivos do sequenciamento foram: minimizar o custo de tempo ocioso, minimizar o custo da variação da taxa de produção e o custo de setup. Para instâncias pequenas, o SA alcançou os resultados do Lingo em um menor tempo. Para as instâncias grandes, o SA gerou boas soluções em um tempo de execução razoável, apresentando melhor relação custo-benefício.

2.3.2.6 Minimizar a variação da taxa de produção de cada modelo, minimizar setup e minimizar tempo extra/tempo ocioso

Tendo a mesma base de objetivos apresentada na subseção anterior de minimizar o trabalho extra total, e o custo de setup, porém minimizando a variação da taxa de produção dos modelos na linha, ao invés do consumo de componentes, algumas publicações que buscaram obter uma fronteira Pareto-ótima para o problema podem ser encontradas na literatura como a aplicação de um algoritmo híbrido baseado em Algoritmo de Salto de Sapo Embaralhado (SFLA, do inglês *Suffled Frog-Leaping Algorithm*) e Otimização de Bactérias (BO, do inglês *Bacteria Optimization*) (RAHIMI-VAHED; MIRZAEI, 2007); a aplicação de um algoritmo híbrido com Otimização por Enxame de Partículas (PSO, do inglês *Particle Swarm Optimization*) (RAHIMI-VAHED; MIRGHORBANI; RABBANI, 2007); a aplicação de Busca Dispersa Multiobjetivo (MOSS, do inglês *Multi-Objective Scatter Search*) (RAHIMI-VAHED *et al.*, 2007).

Os resultados destes trabalhos foram comparados com três GAs da literatura e demonstraram superioridade em termos de qualidade de soluções, especialmente para problemas grandes. Comparados entre si, o método que obteve melhores resultados foi o MOSS. Uma desvantagem que pode ser observada nos três trabalhos, entretanto, é a quantidade de parâmetros que devem ser definidos nos algoritmos. Ambos os trabalhos utilizaram diversos experimentos para calibrar os algoritmos, escolhendo entre eles os que foram efetivos em termos de qualidade e diversidade de soluções.

2.4 SEQUENCIAMENTO DE MMAL E OTIMIZAÇÃO

2.4.1 Otimização Combinatória

Muitos problemas de otimização, tanto de importância prática quanto teórica, consistem na busca da “melhor” configuração de um conjunto de variáveis para atingir objetivos. Estes problemas parecem se dividir naturalmente em duas categorias: aqueles em que as variáveis têm valores reais e aqueles em que as variáveis são discretas. Entre estes últimos, encontra-se uma classe de problemas chamados de Otimização Combinatória (BLUM; ROLI, 2003), que podem ser definidos por:

- Um conjunto de variáveis $X = \{x_1, \dots, x_n\}$;
- Domínio das variáveis: $D_1, \dots, D_n$;
- Restrições;
- Função objetivo f a ser minimizada (ou maximizada), onde $f: D_1 \times \dots \times D_n \rightarrow \mathbb{R}^+$.

O conjunto de todas as possibilidades viáveis é: $S = \{s = \{(x_1, v_1), \dots, (x_n, v_n)\} \mid v_i \in D_i, s \text{ satisfaz todas as restrições}\}$. S é chamado de espaço de busca, dado que cada elemento do conjunto pode ser visto como uma solução candidata. Para resolver um problema de otimização combinatória, deve-se encontrar uma solução $s^* \in S$ que minimize (ou maximize) o valor da função objetivo.

Do aspecto matemático, o problema de sequenciamento pode ser visto como um problema de Otimização Combinatória.

Para resolver esta importante classe de problemas, muitos algoritmos têm sido desenvolvidos, e podem ser classificados em exatos ou aproximados. Os algoritmos exatos têm a garantia de encontrar, para toda instância de tamanho finito, uma solução ideal em tempo limitado. Porém, existem problemas de Otimização Combinatória para os quais não existe algoritmo capaz de obter a solução em tempo polinomial. Neste caso, o uso de métodos aproximados, onde a garantia de encontrar a solução ótima é sacrificada em função da obtenção de boas soluções em um tempo bem menor, é uma alternativa (BLUM; ROLI, 2003).

No sequenciamento de MMAL, o número de soluções possíveis cresce exponencialmente com o número de produtos a serem sequenciados e com a demanda da linha no horizonte planejado. Problemas deste tipo são classificados como NP-difícil (HYUN; KIM; KIM, 1998), o qual, geralmente, não é possível resolver por métodos exatos dentro de um tempo aceitável.

Considerando i produtos, $i = \{1, 2, \dots, I\}$, e Q_i a demanda de cada produto, o número total de sequências possíveis no sequenciamento de MMAL (McMULLEN, 2001b) pode ser

dado pela Equação (3):

$$\text{Total Sequências Possíveis} = \frac{\left(\sum_{i=1}^I q_i\right)!}{\prod_{i=1}^I (q_i!)} \quad (3)$$

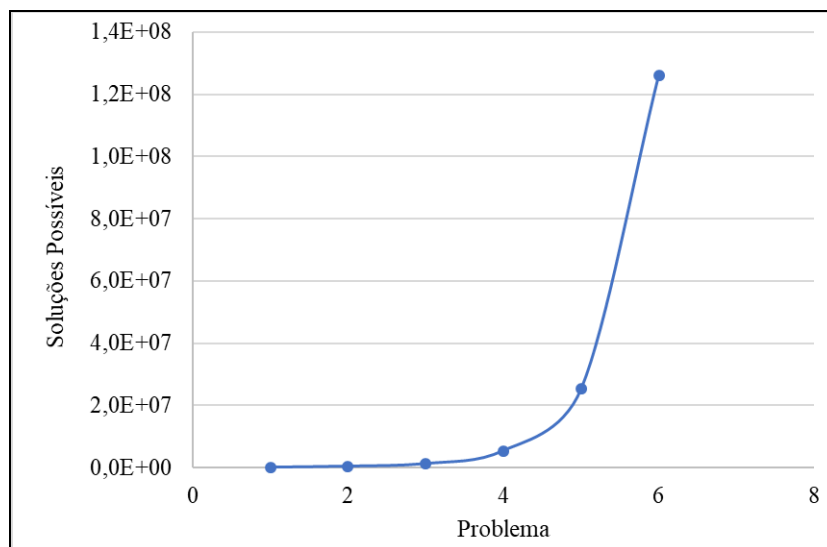
A Tabela 3 mostra, como exemplo, a demanda e o número de soluções possíveis para cinco problemas. Na Figura 17, é possível ver como, mesmo a demanda aumentando em uma unidade a cada problema, o número de soluções cresce exponencialmente.

Tabela 3 - Exemplo do número de sequências possíveis por problema

Problema	Demanda por modelo	Demanda total	Sequências Possíveis
1	(2,2,2,3,1)	10	$7,56 \times 10^4$
2	(2,2,2,3,2)	11	$4,16 \times 10^5$
3	(2,2,2,4,2)	12	$1,25 \times 10^6$
4	(2,2,3,4,2)	13	$5,41 \times 10^6$
5	(2,3,3,4,2)	14	$2,52 \times 10^7$
6	(3,3,3,4,2)	15	$1,26 \times 10^8$

Fonte: Elaborada pela autora.

Figura 17 - Gráfico do número de soluções possíveis por problema do exemplo da Tabela 3



Fonte: Elaborada pela autora.

Sendo assim, para a resolução destes problemas mais complexos, a aplicação de métodos exatos dá lugar aos algoritmos aproximados. Segundo Arroyo (2002), os métodos aproximados podem ser divididos em três classes que diferem basicamente na forma como exploram o espaço de soluções dos problemas:

- Heurísticas construtivas: constroem uma solução do zero, adicionando componentes por meio de regras específicas até que uma solução seja completa. especializadas para um dado problema;
- Heurística de Busca Local: inicia como uma solução completa do problema e de maneira iterativa tenta substituir a solução atual por uma solução melhor, proveniente de sua vizinhança, que é constituída todas as soluções alcançáveis por meio de uma regra de movimento que modifica a solução inicial;
- Meta-heurísticas: métodos inteligentes flexíveis, que tentam combinar métodos heurísticos básicos em uma estrutura que explora de forma eficiente e eficaz um espaço de busca.

As heurísticas construtivas são geralmente os métodos aproximados mais rápidos, porém frequentemente geram soluções de qualidade inferior quando comparada a heurísticas de busca local. Já as meta-heurísticas se destacam por ter a possibilidade de flexibilizar a função objetivo e obter soluções de boa qualidade, relativamente ao tempo de processamento necessária.

2.4.2 Otimização Multiobjetivo

Muitos problemas de otimização do mundo real envolvem múltiplos objetivos, podendo ser encontrados em áreas como internet, finanças, biomedicina, gestão, teoria dos jogos e engenharia (CHINCHULUUN; PARDALOS, 2007).

Um problema de otimização multiobjetivo, geralmente, consiste em minimizar (ou maximizar) simultaneamente um conjunto de objetivos satisfazendo um conjunto de restrições. Diferente da otimização mono-objetivo, para este tipo de otimização não existe uma única solução que otimize cada um dos objetivos, mas sim um conjunto de soluções eficientes no qual nenhuma solução é melhor que outra solução para todos os objetivos. O decisor é o responsável pela escolha de uma solução eficiente particular que pondere os objetivos globais do problema (ARROYO, 2002).

Um problema de otimização multiobjetivo pode ser matematicamente formulado como:

$$\begin{array}{ll} \text{Minimizar} & F(x) = (f_1(x), \dots, f_m(x))^T \\ \text{sujeito a} & x \in \Omega, \end{array} \quad (5)$$

onde Ω é o espaço de decisão e $x \in \Omega$ é um vetor de decisão, $F(x)$ consiste em m funções objetivos $f_i : \Omega \rightarrow R, i = 1, \dots, m$, onde R^m é o espaço objetivo.

Os métodos de otimização multiobjetivo podem ser classificados em três categorias (ARROYO, 2002):

- Método *a priori*: tomada de decisão antes da busca;
- Método *a posteriori*: tomada de decisão depois da busca;
- Método iterativa: tomada de decisão durante a busca.

2.4.2.1 Métodos *a priori*

Para reduzir a complexidade de um problema de otimização multiobjetivo, é possível convertê-lo para um problema mono-objetivo, de modo que apenas uma única função seja avaliada. Outra abordagem onde a contribuição do decisor acontece *a priori*, é a classificação ordinal de preferências. Neste caso o problema é resolvido considerando apenas o primeiro objetivo na ordem de preferências predefinidas sem se considerar os demais objetivos. A seguir, o problema é resolvido para o segundo objetivo sujeito à solução encontrada para o objetivo anterior. Esse processo é repetido até que se esgotem todos os objetivos.

O método tradicional mais utilizado na otimização multiobjetivo é o da Soma Ponderada, onde cada objetivo é associado a um peso, definido de acordo com sua importância pelo decisor (ZADEH, 1963). Considerando um peso $w_i \geq 0$ associado a cada função objetivo, estes pesos são normalizados pela restrição:

$$\sum_{i=1}^m w_i = 1 \quad (6)$$

O problema de otimização multiobjetivo se torna um problema de otimização com valor escalar:

$$\begin{array}{ll} \text{Minimizar} & \sum_{i=1}^m w_i f_i(x) \\ \text{sujeito a} & x \in \Omega, \end{array} \quad (7)$$

A escolha dos fatores de ponderação, no entanto, pode levar a tomadas de decisões errôneas, dado que as unidades de medidas dos objetivos normalmente não são iguais, o que torna difícil combiná-los e a avaliação da importância relativa deles não é uma tarefa fácil (HYUN; KIM; KIM, 1998; MAKAROUNI; PSARRAS; SISKOS, 2015).

Outro método utilizado é o Método ε -restrito, que consiste na otimização do objetivo mais importante, sujeitando-se às restrições dos outros objetivos (HAIMES; LASDON; WISMER, 1971). O problema de otimização multiobjetivo também se torna um problema de otimização com valor escalar:

$$\begin{aligned} &\text{Minimizar } f_j \\ &\text{sujeito a } f_i(x) \leq \varepsilon_i, \text{ para todo } i = 1, 2, \dots, m, i \neq j \\ &x \in \Omega, \end{aligned} \tag{8}$$

2.4.2.2 Métodos *a posteriori*

Os objetivos em (5) frequentemente conflitam entre si. Portanto, a melhoria de um objetivo pode levar a deterioração de outro. Neste caso, uma única solução que pode otimizar todos os objetivos simultaneamente não existe. Por isso, ao invés de avaliar uma única solução, uma alternativa interessante para um tomador de decisão pode ser avaliar as melhores soluções de *trade-off*, chamadas de soluções Pareto-ótimas (ZHOU *et al.*, 2011). Neste tipo de abordagem, a ação do decisor ocorre *a posteriori*.

Alguns conceitos de otimização multiobjetivo (COELLO COELLO; LAMONT; VAN VELDHUIZEN, 2007) auxiliam no entendimento deste trabalho:

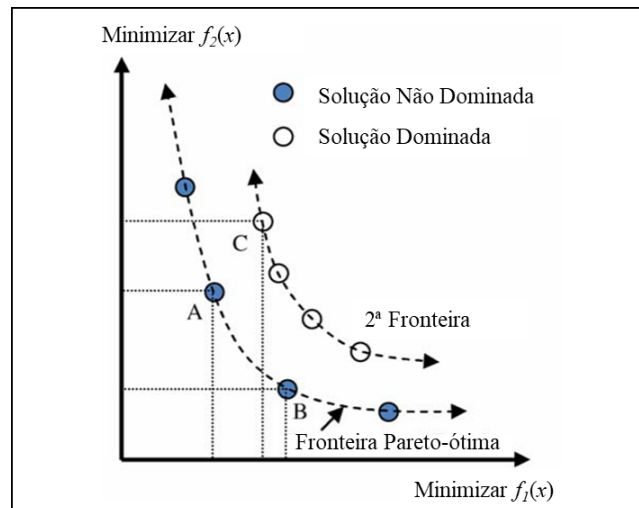
Definição 1 (Dominância de Pareto): Dado o vetor $F(x)$, com m funções objetivo para minimização apresentado em (5). Uma solução x domina uma outra solução x' se, e somente se, para todos os objetivos $f_i(x) \leq f_i(x')$ com $i = 1, \dots, m$ e, existe pelo menos um objetivo $f_i(x) < f_i(x')$. A dominância de x sobre x' é simbolizada por $x \prec x'$.

Definição 2 (Solução Pareto-ótima): Uma solução x^* é chamada de Pareto-ótima se não existe qualquer outra solução x tal que $x \prec x^*$, ou seja, x^* não é dominada por nenhuma outra solução.

Definição 3 (Conjunto Pareto-ótimo): O conjunto de todas as soluções Pareto-ótimas é chamado de conjunto Pareto P . O conjunto-imagem associado ao conjunto Pareto P é dominado fronteira Pareto-ótima.

Uma fronteira Pareto-ótima é um conjunto de pontos em uma curva que possuem um conjunto de atributos que dominam todos os outros pontos fora da fronteira (ALPAY, 2009). A Figura 18 ilustra um exemplo de soluções Pareto-ótimas para um problema de minimização de dois objetivos. Apesar de, na prática, não ser possível encontrar uma fórmula analítica para alcançar a curva que contém as soluções não dominadas, é viável obter estas soluções e seus objetivos por meio de uma busca sistemática (CHUTIMA; JITMETTA, 2013).

Figura 18 - Exemplo de Fronteira Pareto-ótima



Fonte: Adaptado de Chutima; Jitmetta (2013).

2.4.2.3 Métodos iterativos

Nestes métodos, o responsável pela decisão intervém durante o processo de otimização, inserindo suas preferências e guiando a busca para regiões do espaço de busca onde exista soluções de interesse.

2.5 META-HEURÍSTICAS

Para resolver a maioria dos problemas de otimização combinatória, o trabalho de Lin e Kernighan (1973) já indicava que as heurísticas pareciam ser a única linha de ataque viável.

Do ponto de vista teórico, as soluções obtidas eram satisfatórias; e do ponto de vista prático, importava que boas respostas fossem obtidas em tempos factíveis.

O termo “heurística” deriva da palavra grega “*heuriskein*”, que significa encontrar ou descobrir. É usado no campo de otimização para caracterizar um certo tipo de método de resolução de problemas que tenta atingir uma boa solução, não necessariamente ótima. Característica que o diferencia dos métodos exatos, que garantem uma solução ótima do problema (MARTÍ; REINELT, 2011).

No caso do sequenciamento de MMAL, as heurísticas foram investigadas por muitos pesquisadores, pois o tempo computacional é um fator crítico na escolha de um método para resolver este problema (LEU; HUANG; RUSSEL, 1997).

A partir da década de 1980, intensificaram-se os estudos em estratégias que poderiam ser adaptadas para resolver diversos problemas efetivamente, o que segundo Glover (1986) seria a onda do futuro na resolução de problemas, graças a reunião de conceitos das áreas de Otimização e Inteligência Artificial. Foi neste mesmo trabalho que o termo meta-heurística foi introduzido. O prefixo “*metá*”, também do grego, exprime a ideia de “nível superior, maior generalidade” (CHAVES, 2009).

As meta-heurísticas representam uma combinação de procedimentos de busca com estratégias de mais alto nível, incluindo intensificação e diversificação, buscando escapar de ótimos locais e com intuito de encontrar soluções muito próximas do ótimo global, porém sem garantir a otimalidade (BELFIORE; FÁVERO, 2013). Além disto, elas são flexíveis, genéricas e, portanto, adaptáveis a muitas aplicações, sem necessidade de grandes adaptações a cada problema, ao contrário das heurísticas, específicas para cada problema (OLIVEIRA, 2005).

Apesar de existirem algumas definições para o termo meta-heurística, Blum e Roli (2003) listaram as propriedades fundamentais que as caracterizam:

- Meta-heurísticas são estratégias que “guiam” o processo de busca;
- O objetivo é explorar de forma eficiente o espaço de busca, a fim de encontrar soluções (quase-) ótimas;
- As técnicas que constituem as meta-heurísticas variam de procedimentos simples de busca local a processos complexos de aprendizagem;
- Algoritmos meta-heurísticos são aproximados e geralmente não-determinísticos;

- Eles podem incorporar mecanismos para evitar ficar presos em ótimos locais do espaço de busca;
- Meta-heurísticas não são específicas para um determinado problema;
- Meta-heurísticas podem fazer uso de conhecimento específico do problema por meio de heurísticas que são controladas por estratégias superiores;
- Meta-heurísticas mais avançadas usam a experiência obtida durante a busca (incorporado em alguma forma de memória) para guiar a busca.

Uma meta-heurística terá sucesso em um problema de otimização se puder fornecer um equilíbrio entre a diversificação (*exploration*) e a intensificação (*exploitation*). A diversificação é necessária para identificar partes do espaço de busca com soluções de alta qualidade. A intensificação é importante para intensificar a busca nas regiões promissoras encontradas. As principais diferenças entre as meta-heurísticas existentes tratam do modo particular com que elas tentam atingir este equilíbrio (BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Como exemplos de meta-heurísticas pode-se citar: Busca Tabu, Recozimento Simulado, Algoritmos Genéticos, Pesquisa em Vizinhança Variável, Otimização por Colônia de Formigas, Algoritmos Meméticos e a Busca Local Iterativa.

Com a crescente complexidade de problemas de otimização da vida real, a demanda por otimizadores robustos, rápidos e precisos está aumentando entre pesquisadores de várias áreas. Neste contexto, as meta-heurísticas têm sido aplicadas de forma eficaz em inúmeros problemas, com diferentes estruturas, se consolidando com uma forte técnica de otimização (SOTOUDEH-ANVARI; HAFEZALKOTOB, 2018). Estes problemas incluem problemas da indústria e serviços, em áreas vão desde finanças, a gerenciamento da produção e engenharia (BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Lourenço (2005) enfatiza as oportunidades de aplicação de meta-heurísticas no campo do Gerenciamento Logístico Integrado, para resolver problemas como transporte e roteamento de veículos, localização de instalações e desenho da malha logística, gerenciamento de estoques, alocação de recursos e sequenciamento da produção.

2.6 PARETO ITERATED LOCAL SEARCH

O PILS (GEIGER, 2006) foi desenvolvido para a resolução de problemas de otimização multiobjetivo, motivada pela demanda de heurísticas simples, porém efetivas, para solucionar complexos problemas.

O PILS é caracterizado pela busca intensificada e diversificada em um único algoritmo. Para tal fim, ele utiliza a estrutura da Busca Local Iterativa (ILS, do inglês *Iterated Local Search*) (LOURENÇO; MARTIN; STUTZLE, 2003), que traz a ideia de perturbações para superar ótimos locais e continuar a busca em áreas promissoras do espaço de busca; combinada com a Pesquisa em Vizinhança Variável (VNS, do inglês *Variable Neighborhood Search*) (MLADENOVIC; HANSEN, 1997), que tem múltiplos operadores de vizinhança em um único algoritmo, para evitar ótimos locais.

Nas próximas subseções, o ILS e o VNS serão descritos. E na sequência, o PILS, algoritmo multiobjetivo que utiliza estas duas meta-heurísticas, também será apresentado.

2.6.1 Busca Local Iterativa

A meta-heurística Busca Local Iterativa (ILS, do inglês *Iterated Local Search*) (LOURENÇO; MARTIN; STUTZLE, 2003) procura focar a busca não no espaço completo de soluções, mas em um subespaço S^* definido por soluções que são ótimos locais de determinado procedimento de otimização.

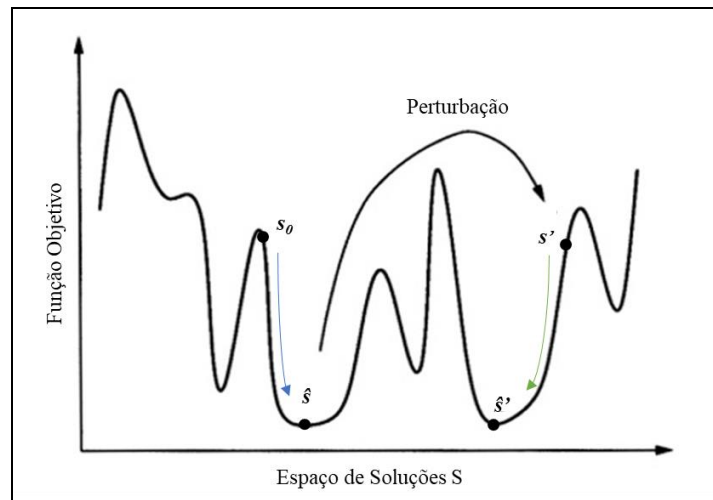
O ILS consiste em um processo iterativo, no qual uma solução é perturbada aleatoriamente, gerando novas soluções de partida para um método de busca local. A perturbação tem uma importância muito grande. Uma perturbação pequena pode fazer com que o algoritmo não consiga escapar da região de atração de um ótimo local já pesquisado. Porém, com uma perturbação muito grande o algoritmo tem um reinício aleatório.

O algoritmo ILS é inicializado gerando-se uma solução inicial qualquer para o problema (s_0). Em seguida, é aplicada uma heurística de busca local, obtendo uma solução ótima local ($\hat{s}$), que é a melhor solução entre todas as analisadas. A heurística termina quando todas as trocas foram investigadas, e se há melhora em relação a solução em que foi aplicada a busca.

Baseado nessa solução, tem início um processo iterativo onde, a cada iteração, perturba-se aleatoriamente a solução corrente, produzindo uma nova solução (s'). A intensidade da perturbação, ou seja, o número de movimentos feitos, varia a cada iteração. Outra busca local é aplicada sobre s' , encontrando a solução ótima local desta região ($\hat{s}'$).

A Figura 19 ilustra um exemplo do funcionamento do ILS.

Figura 19 - Ilustração do ILS



Fonte: Adaptado de Lourenço; Martin; Stutzle (2003).

O critério de aceitação é utilizado para decidir em qual solução a próxima perturbação será aplicada, comparando a solução $\hat{s}'$ com a solução $\hat{s}$ e aceita a melhor como a nova solução corrente.

O processo é repetido até que o critério de parada seja satisfeito.

A Figura 20 apresenta o pseudocódigo do ILS.

Figura 20 - Pseudocódigo da meta-heurística ILS

algoritmo ILS

gere uma solução inicial s_0

$\hat{s} \leftarrow \text{BuscaLocal} (s_0)$

enquanto (critério de parada não for satisfeito) **faça**

$s' \leftarrow \text{Perturbação} (\hat{s}, \text{histórico})$

$\hat{s}' \leftarrow \text{BuscaLocal} (s')$

$\hat{s} \leftarrow \text{CritériodeAceitação} (\hat{s}, \hat{s}', \text{histórico})$

fim-enquanto

fim-algoritmo

Fonte: Adaptado de Lourenço; Martin; Stutzle (2003).

2.6.2 Pesquisa em Vizinhança Variável

A meta-heurística Pesquisa em Vizinhança Variável (VNS, do inglês *Variable Neighborhood Search*) (MLADENOVIC; HANSEN, 1997) combina busca local com mudanças sistemáticas de vizinhança, procurando escapar de ótimos locais.

Ao contrário de outros métodos de busca local, o VNS não segue uma trajetória, mas explora vizinhanças cada vez mais distantes da solução corrente, gerando vizinhos aleatoriamente e movendo-se para a nova solução se e somente se uma melhoria for identificada.

Além disso, um método de busca local é aplicado repetidamente para transformar a solução vizinha em um ótimo local. Portanto, o VNS é um método que combina mudanças de vizinhança estocástica com mudanças de vizinhança determinística.

O pseudocódigo do VNS é apresentado na Figura 21.

Figura 21 - Pseudocódigo da meta-heurística VNS

algoritmo VNS

gere uma solução inicial s

seja w_{max} o número de estruturas diferentes de vizinhança

enquanto (critério de parada não for satisfeito) **faça**

$w \leftarrow 1$

enquanto ($w \leq w_{max}$) **faça**

 gere um vizinho s' aleatoriamente da w -ésima vizinhança de s ($s' \in N^w(s)$)

 aplique uma busca local em s' obtendo um ótimo local ($\hat{s}$)

se ($f(\hat{s}) < f(s)$) **então**

$s \leftarrow \hat{s}$

$w \leftarrow 1$

Senão

$w \leftarrow w + 1$

fim-se

fim-enquanto

fim-enquanto

fim-algoritmo

Fonte: Adaptado de Mladenovic; Hansen (1997).

O VNS parte de um conjunto pré-definido de estruturas de vizinhança N^w , $w = \{1, \dots, w_{max}\}$, sendo $N^w(s)$ o conjunto de soluções da w -ésima vizinhança da solução corrente s . O VNS é inicializado com uma solução inicial qualquer e a cada iteração é gerado aleatoriamente um vizinho, s' , dentro da vizinhança N^w da solução corrente ($s' \in N^w(s)$). Enquanto $w \leq w_{max}$, um novo vizinho é gerado.

Então, uma heurística de busca local é implementada para obter um ótimo local ($\hat{s}$). Se o

ótimo local for melhor que a solução corrente ($f(\hat{s}) < f(s)$), a busca continua a partir deste, recomeçando da primeira estrutura de vizinhança ($w = 1$). Caso contrário, continua-se a busca a partir da próxima estrutura de vizinhança, N^{w+1} .

2.6.3 *Pareto Iterated Local Search*

O ILS, originalmente, é uma meta-heurística baseada em uma solução única, ou seja, ao invés de aplicar repetidamente um processo de busca local a soluções iniciais geradas aleatoriamente, ele gera a solução inicial para a próxima iteração perturbando o ótimo local encontrado na iteração atual (BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Os problemas de otimização multiobjetivo envolvem dois ou mais objetivos, normalmente conflitantes. Sendo assim, quando todos os objetivos são de igual importância, existe um conjunto de soluções ótimas, denominadas soluções Pareto-ótimas, que representam o *trade-off* entre estes objetivos; ao invés de uma única solução que otimize simultaneamente todos os objetivos, como na otimização com um único objetivo (ARROYO, 2002).

Portanto, no PILS proposto por Geiger (2006) para resolver problemas multiobjetivo, o ILS foi adaptado para lidar não apenas com uma única, mas com um conjunto de alternativas de uma só vez. Na sequência, a descrição deste algoritmo.

O PILS parte de uma solução inicial x_1 . Então, é realizada uma busca intensificada de aprimoramento até que um conjunto de soluções ótimas locais seja identificado, armazenado em um conjunto P^{approx} , que representa a aproximação do verdadeiro conjunto Pareto P . Neste passo inicial, um conjunto de vizinhanças garante que todas as alternativas identificadas sejam localmente ótimas, não apenas para um único, mas para um conjunto de vizinhanças. Este princípio, vindo do VNS, promete levar a melhores resultados, já que todos os ótimos globais são localmente ótimos com relação a todas suas possíveis vizinhanças, enquanto esse não é necessariamente o caso para ótimos locais.

Depois de identificar um conjunto localmente ótimo, um passo de diversificação é realizado em uma solução x_2 , usando um operador que perturba a solução, continuando a busca a partir da solução perturbada x_3 . O operador deve ser significativamente diferente das vizinhanças usadas na intensificação, caso contrário a busca retornaria à solução anterior. Por outro lado, a perturbação não deve destruir inteiramente as características da alternativa, pois isto resultaria em um reinício aleatório da pesquisa, sem manter atributos promissores das soluções. O algoritmo termina quando o critério de parada é satisfeito.

O pseudocódigo do PILS será apresentado na próxima seção.

Geiger (2006 e 2011) aplicou o PILS em problema de sequenciamento *Flow Shop* multiobjetivo, testando-o em um conjunto de instâncias da literatura. Foram obtidos resultados encorajantes em termos de qualidade, apesar da simplicidade da abordagem do algoritmo.

Gomes, Neves e Souza (2014) aplicaram o PILS para resolver o problema de cronograma de projeto com restrição de recursos e relações de precedência. O PILS apresentou bons resultados e comparado aos outros algoritmos testados, se destacou com a menor taxa de erro, dada pelo percentual das soluções encontradas que não pertencem a Fronteiro Pareto-Ótima.

Na literatura, podem-se encontrar outros trabalhos que propuseram versões para o ILS multiobjetivo, com algumas variações em relação ao PILS, como exposto a seguir, porém nenhum deles aborda o problema de sequenciamento de MMAL.

Para resolver o problema de programação de tarefas em um sistema *No-Wait Flowshop*, com objetivo de otimizar o fluxo total e o atraso total das tarefas com relação às suas datas de entrega, Oliveira Junior, Arroyo e Souza (2010) propuseram uma adaptação do ILS, chamada ILS-Multi, que apresentou resultados superiores aos obtidos com a heurística GRASP multiobjetivo.

A este mesmo problema outra versão do ILS multiobjetivo, que conta com três tipos de busca local, foi aplicada (NADERI *et al.*, 2012). Desta vez, a heurística proposta também superou outros métodos encontrados na literatura.

Assis *et al.* (2013) propuseram um ILS multiobjetivo (MOILS, do inglês *Multi-Objective Iterated Local Search*) para resolver o problema de roteamento de veículos, com entregas fixas e coletas opcionais, minimizando simultaneamente o custo e a demanda de coleta não cumprida. Os resultados obtidos mostraram que o MOILS foi capaz de ter um desempenho significativamente superior que outros três métodos da literatura.

Com objetivo de resolver uma variante do problema de roteamento de veículos, considerando múltiplos depósitos, Galindres-Guancha, Toro-Ocampo e Gallego-Rendón (2018), apresentaram uma adaptação do método de Geiger, para minimizar o custo da operação e balanceamento das rotas. Os resultados obtidos foram de muito boa qualidade.

Outra abordagem para resolver problema de roteamento de veículos, com janela de tempo e com tempos de viagem e serviço estocásticos utilizando MOILS foi proposta por Miranda, Branke e Conceição (2018). Os experimentos mostraram a eficácia do algoritmo proposto.

Xu *et al.* (2017) desenvolveram um outro MOILS, que conta com uma fase de busca com profundidade variável baseada em Pareto, que dinamicamente ajusta a profundidade de busca durante o processo, para equilibrar exploração e diversificação, para resolver o problema de sequenciamento em um sistema *Flowshop* permutacional considerando tempos de setup. Os resultados do MOILS foram comparados com diversos algoritmos evolutivos e se mostraram melhores.

Schulz, Neufeld e Buscher (2019) criaram um algoritmo ILS com múltiplas fases para determinar uma Fronteira Pareto tridimensional para o sequenciamento de fluxo de produção com consciência energética, considerando o tempo total de produção, o custo total de energia e a carga no pico. O ILS se provou adequado na pesquisa do espaço de soluções, o que permite o suporte as decisões práticas.

Com os estudo apresentados até aqui, pode-se perceber que o ILS, mesmo em sua versão multiobjetivo, apresenta resultados satisfatórios, com uma implementação simples, quando comparado a outras abordagens, o que suporta a escolha deste método para aplicação neste trabalho. No próximo capítulo, será detalhada a aplicação do PILS ao problema de sequenciamento de MMAL.

3 PILS APLICADO AO PROBLEMA DE SEQUENCIAMENTO DE MMAL

Neste capítulo serão detalhados a modelagem do problema e o algoritmo multiobjetivo utilizado no estudo, o PILS.

3.1 MODELAGEM DO PROBELMA

O problema de sequenciamento de MMAL considerando de forma simultânea a minimização da variação do consumo de componentes na linha e do número de setups é modelado como mostrado a seguir:

Índice / Conjuntos

- I Conjunto de modelos ($i = 1, \dots, I$)
- J Conjunto de componentes ($j = 1, \dots, J$)
- K Conjunto de instantes (tempo) ($k = 1, \dots, K$)

Parâmetros

- θ_k número de instantes decorridos ($\theta_k = k$)
- Q_i quantidade de produção de cada produto A_i , $\sum_{i=1}^I Q_i = Q$
- b_{ij} quantidade necessária do componente a_j ($j = 1 \dots J$) para produzir uma unidade do produto A_i ($i = 1 \dots I$)
- N_j quantidade total necessária do componente a_j ($j = 1 \dots J$) a ser consumido para produzir todos os produtos A_i ($i = 1, \dots, I$);

Variáveis

- $X_{j,k}$ quantidade total consumida do componente a_j ($j = 1 \dots J$) até o instante k ;
- s_k necessidade de setup no momento k . $s_k = 1$ se o setup é necessário (ou seja, se o produto em k é diferente do produto de $k-1$) ou 0, se o setup não é necessário. Neste caso, assume-se que um setup inicial é requerido, independente da sequência, e que os tempos de setup não dependem da sequência.

Modelo Matemático

O primeiro objetivo deste trabalho foi definido em Monden (1983) e pode ser representado por (9):

$$\text{Minimizar } D = \sum_{k=1}^K \sqrt{\sum_{j=1}^J \left(\frac{\theta_k \cdot N_j}{Q} - X_{j,k-1} - b_{ij} \right)^2} \quad (9)$$

Sujeito a $X_{j,k} \leq N_j$ para $j = 1, \dots, J$ e $k = 1, \dots, K-1$

$X_{j,K} = N_j$ para $j = 1, \dots, J$

$X_{j,k} \in \mathbb{Z}^+$ para $j = 1, \dots, J$ e $k = 1, \dots, K$

sendo:

$D_{k,i}$ = distância entre o consumo real e ideal de componentes no instante k do produto i .

O segundo objetivo, formulado por McMullen e Frazier (2000), para minimização do número de setups (S) é apresentada em (10):

$$\text{Minimizar} \quad S = 1 + \sum_{k=2}^Q s_k \quad (10)$$

Sujeito a $s_k = \{0,1\}$ para $k = 2, \dots, K$

3.2 REPRESENTAÇÃO DA SOLUÇÃO

Para aplicar o algoritmo PILS, faz-se necessário desenvolver uma representação para possíveis soluções. Uma solução do sequenciamento de MMAL pode ser representada por um vetor, em que a posição indica a ordem com que o produto deve ser inserido na linha.

Um exemplo de solução, não necessariamente ótima, do sequenciamento de três modelos ($A_i = 1, 2$ e 3), que totalizam uma demanda (Q) de cinco unidades ($Q_1 = 2, Q_2 = 2, Q_3 = 1$), pode ser visto na Figura 22.

Figura 22 - Representação de uma solução do sequenciamento de MMAL

1	1	2	2	3
---	---	---	---	---

Fonte: Elaborado pela autora.

3.3 AVALIAÇÃO DA SOLUÇÃO

Neste trabalho são considerados dois objetivos na otimização. O primeiro deles é a minimização da variação do consumo de componentes na linha (nas tabelas, referida como Consumo), dada pela Equação (9); e o segundo, a minimização do número de setups, dada

pela Equação (10). Estes dois objetivos são conflitantes, portanto, a minimização de um não significa a minimização de outro.

Para avaliar as soluções, ao invés de criar uma função para unir estes dois números que possuem grandezas diferentes, optou-se por utilizar a avaliação da Fronteira Eficiente. Neste caso, o resultado do algoritmo será um conjunto de soluções, classificadas como eficientes, que compõe uma curva, e dominam as demais soluções. Na prática, para o sequenciamento de MMAL, para cada número de setups requeridos (indicado no eixo x do gráfico), estará associada a mínima variação do consumo de componentes encontrada (no eixo y do gráfico).

Considere que os modelos apresentados na seção anterior têm o consumo dos componentes a_1 , a_2 e a_3 , conforme mostrado na Tabela 4. Utilizando como exemplo a mesma configuração de demanda, também da seção anterior, a Tabela 5 apresenta todas as soluções possíveis, assim como o número resultante de setups e da variação do consumo de componentes.

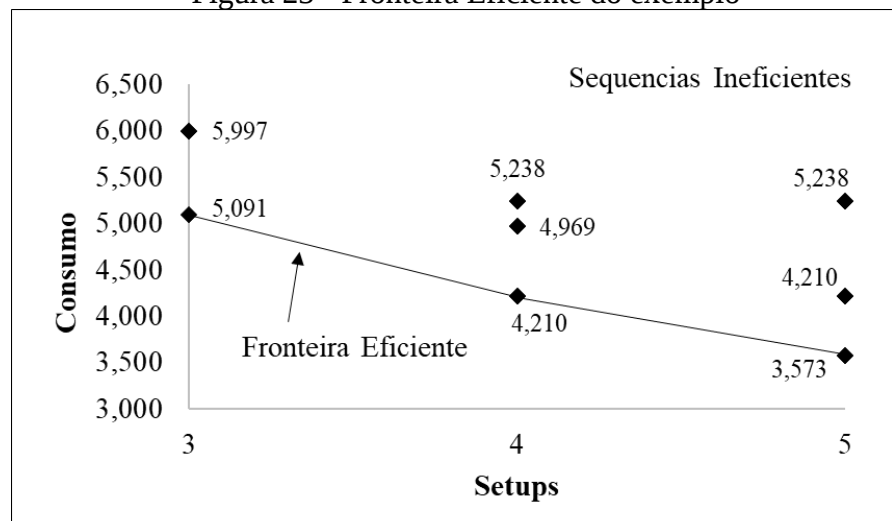
Tabela 4 – Exemplo de consumo de componentes por produto

Produto (A_i)	a_1	a_2	a_3
1	2	1	0
2	3	1	1
3	1	0	2

Fonte: Elaborada pela autora.

A partir das soluções da Tabela 5 é possível construir a Fronteira Eficiente, associando cada número requerido de setups com a mínima variação de consumo. A fronteira do exemplo é mostrada na Figura 23.

Figura 23 - Fronteira Eficiente do exemplo



Fonte: Elaborado pela autora.

Tabela 5 - Exemplo de sequencias possíveis e número de setups e consumo resultantes

# Solução	Solução	Setups	Consumo
1	[1,1,2,2,3]	3	5,997
2	[1,1,2,3,2]	4	4,969
3	[1,1,3,2,2]	3	5,091
4	[1,2,1,2,3]	5	5,238
5	[1,2,1,3,2]	5	4,210
6	[1,2,2,1,3]	4	5,238
7	[1,2,2,3,1]	4	4,210
8	[1,2,3,1,2]	5	3,573
9	[1,2,3,2,1]	5	3,573
10	[1,3,1,2,2]	4	4,969
11	[1,3,2,1,2]	5	4,210
12	[1,3,2,2,1]	4	4,210
13	[2,1,1,2,3]	4	5,238
14	[2,1,1,3,2]	4	4,210
15	[2,1,2,1,3]	5	5,238
16	[2,1,2,3,1]	5	4,210
17	[2,1,3,1,2]	5	3,573
18	[2,1,3,2,1]	5	3,573
19	[2,2,1,1,3]	3	5,997
20	[2,2,1,3,1]	4	4,969
21	[2,2,3,1,1]	3	5,091
22	[2,3,1,1,2]	4	4,210
23	[2,3,1,2,1]	5	4,210
24	[2,3,2,1,1]	4	4,969
25\	[3,1,1,2,2]	3	5,997
26	[3,1,2,1,2]	5	5,238
27	[3,1,2,2,1]	4	5,238
28	[3,2,1,1,2]	4	5,238
29	[3,2,1,2,1]	5	5,238
30	[3,2,2,1,1]	3	5,997

Fonte: Elaborada pela autora.

3.4 GERAÇÃO DA SOLUÇÃO INICIAL

O PILS parte de uma solução inicial que é gerada de modo aleatório, conforme procedimento descrito abaixo.

Em um processo iterativo, começando na primeira posição da linha, um valor entre 1 e I é gerado aleatoriamente, sendo I o número de produtos a serem sequenciados. Se o número de

produtos I sequenciados até o momento não tiver atingido sua demanda total, ele é alocado na posição corrente, senão, outro valor é gerado aleatoriamente, até achar um modelo que não tenha sido totalmente sequenciado. O processo é repetido para alocação na posição adjacente e continua até que toda a demanda da linha tenha sido sequenciada.

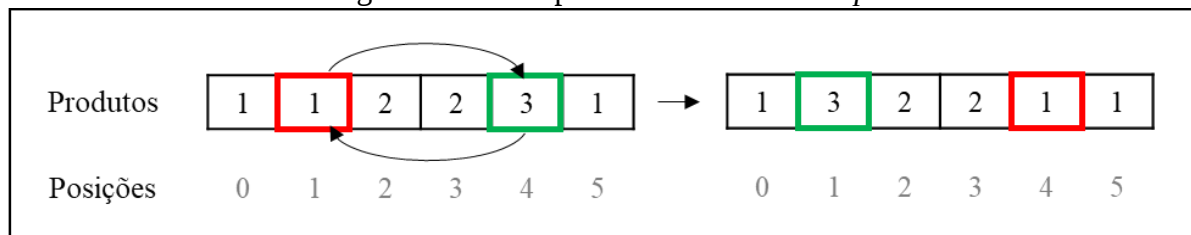
3.5 ESTRUTURAS DE VIZINHANÇA

Para explorar o espaço de soluções dos problemas, são aplicados três movimentos no sequenciamento dos modelos, conforme descrito a seguir.

O primeiro movimento, *Swap*, consiste em trocar a posição de dois produtos (um assume a posição do outro).

Na Figura 24, exemplifica-se o movimento *Swap* entre as posições 1 e 4 do vetor.

Figura 24 - Exemplo do Movimento *Swap*

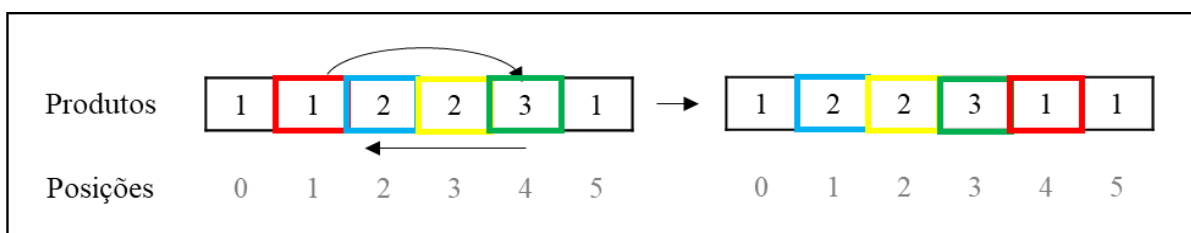


Fonte: Elaborado pela autora.

O segundo movimento, *Shift* para frente, consiste em tirar um produto de uma posição a reinseri-lo na posição b , com $a < b$. Na prática, há o “arraste” de todos os produtos entre as posições a e b , inclusive, para a esquerda no vetor.

Um exemplo de movimento entre as posições 1 e 4 é mostrado na Figura 25.

Figura 25 - Exemplo do Movimento *Shift* para frente



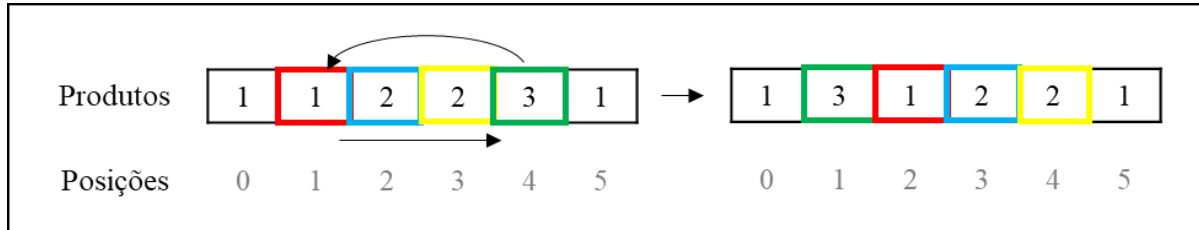
Fonte: Elaborado pela autora.

Finalmente, o terceiro movimento, *Shift* para trás, consiste em tirar um produto de uma posição a reinseri-lo na posição b , com $a > b$. Na prática, há o “arraste” de todos os produtos

entre as posições a e b , inclusive, para a direita no vetor.

Um exemplo de movimento entre as posições 4 e 1 é mostrado na Figura 26.

Figura 26 - Exemplo do Movimento *Shift* para trás



Fonte: Elaborado pela autora.

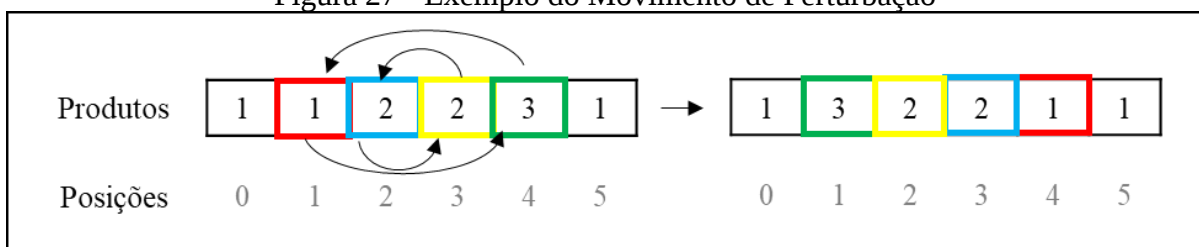
3.6 PERTURBAÇÃO

Uma das etapas do PILS é a perturbação, que tem como objetivo explorar outras regiões do espaço de busca.

A estratégia de perturbação proposta por Geiger (2006) começa com a escolha aleatória de uma posição $b \leq Q - 4$. O movimento consiste em trocar os produtos das posições b e $b + 3$, assim como das posições $b + 1$ e $b + 2$, mantendo a ordem dos produtos antes de b , e depois de $b + 3$ igual.

Um exemplo de movimento considerando $b = 1$ é mostrado na Figura 27.

Figura 27 - Exemplo do Movimento de Perturbação



Fonte: Elaborado pela autora.

3.7 ALGORITMO E PSEUDOCÓDIGO

O algoritmo PILS foi desenvolvido seguindo o pseudocódigo da Figura 28.

A geração da solução inicial é feita conforme método descrito na Seção 3.3. E então, o conjunto P^{approx} é atualizado com esta solução.

Na sequência, tem início a fase de intensificação, onde explora-se toda a vizinhança da solução. Para isso são utilizadas três estrutura de vizinhança. Cada estrutura de vizinhança

consiste em fazer todas as trocas possíveis entre os produtos, utilizando os movimentos detalhados na Seção 3.4.

Figura 28 - Pseudocódigo da meta-heurística PILS

algoritmo PILS

inicialize parâmetros: defina as vizinhanças $N_1, \dots, N_k$

$i \leftarrow 1$

gere uma solução inicial x

$P^{approx} \leftarrow \{x\}$

enquanto (critério de parada não for satisfeito) **faça**

enquanto (x localmente ótimo em relação a $N_1, \dots, N_k$, portanto $i > k$) **faça**

 Calcule $N_i(x)$

 Avalie $N_i(x)$

 Atualize P^{approx} com $N_i(x)$

se ($\exists x' \in N_i(x) \mid x' < x$) **então**

$x \leftarrow x'$

$i \leftarrow 1$

 Reorganize as vizinhanças $N_1, \dots, N_k$ em alguma ordem aleatória

Senão

$i \leftarrow i + 1$

fim-se

fim-enquanto

 Vizinhanças de $x \leftarrow$ ‘investigadas’

$i \leftarrow 1$

se ($\exists x' \in P^{approx} \mid$ vizinhanças não tem sido investigadas ainda) **então**

$x \leftarrow x'$

Senão

 Selecione $x' \in P^{approx}$

$x'' \leftarrow N_{perturb}(x')$

$x \leftarrow x''$

fim-se

fim-enquanto

fim-algoritmo

Primeiro, escolhe-se aleatoriamente uma das estruturas para que sejam gerados todos os vizinhos da solução corrente. Se um dos vizinhos for dominante, ele se torna a solução corrente, as vizinhanças são reordenadas aleatoriamente e o procedimento recomeça; caso contrário, a busca continua com a próxima estrutura. Após visitar toda a vizinhança da solução corrente marca-se como visitada.

Caso existam soluções de P^{approx} não visitadas, uma delas é selecionada aleatoriamente como a solução corrente. Caso contrário, se todas as soluções já tiverem sido visitadas, seleciona-se aleatoriamente uma solução de P^{approx} e a ela é aplicada a perturbação descrita na Seção 3.5. O resultado da perturbação passa a ser a nova solução corrente.

O procedimento é repetido enquanto o critério de parada não for satisfeito. O critério de parada adotado neste trabalho foi o número máximo de iterações do algoritmo.

4 RESULTADOS

O método heurístico proposto foi codificado em C++ e os testes computacionais foram executados em um computador com processador Intel® Core™ i7-8550U CPU @ 1.80 GHz a 4.00GHz e 16GB de RAM.

Os experimentos foram realizados para dois conjuntos de instâncias da literatura, e uma instância baseada em um caso real. Na sequência, os resultados obtidos são apresentados.

4.1 COMPARAÇÃO COM RESULTADOS DA LITERATURA

Como não foram encontradas publicações anteriores que resolvessem simultaneamente o nivelamento do consumo de componentes e a redução do número de setups, as instâncias escolhidas para este teste foram de trabalhos que consideraram como objetivos o nivelamento da produção de produtos e a redução do número de setups.

Os dois conjuntos testados, *Set 1* e *Set 2*, foram obtidos em McMullen (2001b), que os utilizou para comprar o desempenho dos métodos ACO (com diferentes estratégias), GA, SA e TS, na resolução do problema de sequenciamento para nivelar a produção de produtos e a reduzir o número de setups.

As Tabelas 6 e 7 mostram os problemas do *Set 1* e *2*, respectivamente. Para cada problema, tem-se a demanda (Q_i) de cada produto (A_i), assim como o número de soluções possíveis para este sequenciamento.

Os problemas são caracterizados por configurações diferentes de demanda de cinco produtos. O *Set 1* tem sete problemas, todos com 10 unidades de demanda. E *Set 2*, nove problemas, com 12 unidades de demanda cada.

Tabela 6 - *Set 1*: demanda (Q_i) de cada tipo de produto (A_i) por problema, com $Q = 10$

Problema	A_0	A_1	A_2	A_3	A_4	Número de Soluções
B	6	1	1	1	1	5.040
C	5	2	1	1	1	15.120
D	4	2	2	1	1	37.800
E	4	3	1	1	1	25.200
F	3	3	2	1	1	50.400
G	3	2	2	2	1	75.600
H	2	2	2	2	2	113.400

Fonte: Elaborado pela autora.

Tabela 7 - Set 2: demanda (Q_i) de cada tipo de produto (A_i) por problema, com $Q = 12$

Problema	A_0	A_1	A_2	A_3	A_4	Número de Soluções
B	8	1	1	1	1	11.880
C	7	2	1	1	1	47.520
D	6	3	1	1	1	110.880
E	6	2	2	1	1	166.320
F	5	3	2	1	1	332.640
G	5	2	2	2	1	498.960
H	4	3	2	2	1	831.600
I	4	4	2	1	1	415.800
J	3	3	2	2	2	1.663.200

Fonte: Elaborado pela autora.

Para conseguir comparar o desempenho do modelo aqui proposto com os do trabalho citado, foi necessária uma adaptação nestas instâncias. Esta adaptação consiste em considerar que cada produto consome uma unidade de um componente diferente. Na Tabela 8 é mostrado o consumo dos componentes (a_j) para cada produto (A_i).

Tabela 8 – Consumo de componentes (a_j) para cada produto (A_i)

Produto (A_i) Componente (a_j)	A_0	A_1	A_2	A_3	A_4
a_0	1	0	0	0	0
a_1	0	1	0	0	0
a_2	0	0	1	0	0
a_3	0	0	0	1	0
a_4	0	0	0	0	1

Fonte: Elaborado pela autora.

Os resultados utilizando as demais abordagens foram obtidos na literatura, e não, implementando novamente os algoritmos. Portanto, as métricas para avaliar o desempenho das abordagens são as mesmas apresentadas por McMullen (2001b). As três métricas são: taxa média de inferioridade, percentil e CPU *ratio*.

A taxa de inferioridade é dada pela diferença percentual, para cada valor de setup, entre o resultado da função objetivo obtido pela meta-heurística, e a solução ótima, obtida por enumeração completa. A taxa média de inferioridade (referida a partir daqui como taxa de inferioridade) é, portanto, a média entre as taxas de inferioridade para cada número de setups. Para esta métrica, quanto mais baixo o valor, melhor.

O percentil é dado pelo número de soluções da enumeração completa superiores as obtidas pela meta-heurística (função objetivo menor), na quantidade associada de setups. Por

exemplo, se das 37.800 soluções obtidas via enumeração completa, 12 delas forem superiores às soluções obtidas pela meta-heurística, quer dizer que a meta-heurística está no 99,9683º percentil, se comparada a fronteira ótima.

Outra maneira de expressar o percentil é em número de soluções inferiores para 1.000.000 de soluções, que no exemplo seria 317,46. Esta segunda maneira será utilizada daqui em diante, para expressar melhor as diferenças entre as abordagens. Assim como a taxa de inferioridade, quanto menor o valor do percentil, melhor.

Finalmente, o *CPU ratio* é dado pela divisão entre o tempo de CPU necessário para obter a Fronteira-Pareto por enumeração completa e o tempo de CPU da meta-heurística aplicada. Neste caso, quanto maior o valor, melhor.

Vale ressaltar que a comparação do *CPU ratio* é válida, dado que representa a relação entre os tempos da meta-heurística e enumeração completa obtidas em uma mesma máquina.

Na Tabela 9 são apresentados os resultados de cada métrica para o *Set 1*, e na Tabela 10, para o *Set 2*. Após o valor de cada métrica, entre parênteses, está o desvio padrão entre os resultados de cada um dos problemas dentro do conjunto.

O parâmetro que pode ser variado no PILS é o número de iterações da busca. Para o *Set 1*, o PILS foi testado com três níveis de parâmetro diferentes. O PILS-10 com 10 iterações, o PILS-20 com 20 iterações e o PILS-30 com 30 iterações. O PILS foi executado 10 vezes para cada instância, e as métricas foram calculadas com a média dos resultados destas rodadas.

Tabela 9 - Métricas de desempenho para o *Set 1*

Abordagem	Taxa de Inferioridade (DP)	Percentil (DP)	CPU ratio (DP)
ACO-1	0,68 (0,87)	49,13 (74,32)	1,20 (0,00)
ACO-2	0,85 (0,99)	42,52 (45,16)	1,20 (0,00)
ACO-3	1,60 (2,32)	137,63 (168,20)	1,20 (0,00)
ACO-4	3,15 (2,84)	142,04 (124,46)	1,20 (0,00)
ACO-5	2,10 (1,87)	90,39 (70,86)	1,20 (0,00)
ACO-6	2,72 (2,02)	111,49 (97,76)	1,20 (0,00)
GA	4,31 (2,52)	222,03 (164,23)	0,71 (0,00)
SA	0,69 (1,03)	102,04 (113,71)	1,20 (0,00)
TS	1,15 (1,39)	83,14 (104,35)	1,20 (0,00)
PILS-10	0,31 (0,46)	219,20 (359,41)	13,25 (7,99)
PILS-20	0,09 (0,23)	45,35 (119,99)	9,06 (6,24)
PILS-30	0,06 (0,15)	15,12 (40,00)	5,18 (2,80)

Fonte: Elaborado pela autora.

Analisando os resultados obtidos (Tabela 9), pode-se observar que o PILS, em todos os níveis de iterações testados, apresenta bom desempenho quanto à inferioridade, sendo melhor que todos os outros métodos testados. Para 10 e 20 iterações, o desempenho em percentil é menos competitivo, em contrapartida, para 30, o melhor dentre todos as abordagens. O PILS também se mostra muito forte em relação ao tempo de CPU requerido, apresentando um CPU *ratio* mais de dez vezes maior do que os outros métodos.

Para o Set 2, o PILS foi testado com três níveis de parâmetro diferentes. O PILS-12 com 12, o PILS-24 com 24 e o PILS-36 com 36 iterações. O PILS foi executado 10 vezes para cada instância, e as métricas foram calculadas com a média dos resultados destas rodadas.

Tabela 10 - Métricas de desempenho para o Set 2

Abordagem	Taxa de Inferioridade (DP)	Percentil (DP)	CPU <i>ratio</i> (DP)
ACO-1	6,46 (6,31)	105,40 (73,65)	11,07 (12,69)
ACO-2	6,26 (6,59)	107,46 (101,20)	11,07 (12,69)
ACO-3	7,16 (5,13)	47,50 (36,20)	11,07 (12,69)
ACO-4	12,67 (9,85)	83,42 (51,93)	11,07 (12,69)
ACO-5	7,82 (6,22)	65,79 (39,44)	11,07 (12,69)
ACO-6	7,01 (5,01)	74,41 (42,47)	11,07 (12,69)
GA	1,13 (1,14)	53,98 (58,59)	5,23 (4,44)
SA	0,65 (0,64)	26,99 (28,14)	8,80 (7,46)
TS	0,79 (0,94)	32,33 (35,86)	8,80 (7,46)
PILS-12	0,89 (0,86)	333,17 (638,75)	88,78 (87,83)
PILS-24	0,43 (0,40)	29,66 (27,05)	51,87 (53,98)
PILS-36	0,19 (0,23)	18,97 (19,63)	34,36 (33,40)

Fonte: Elaborado pela autora.

Analisando os resultados obtidos para o Set 2, na Tabela 10, pode-se observar que o PILS, em todos os níveis de iterações testados, apresenta bom desempenho quanto à inferioridade, sendo melhor que todos os outros métodos testados. Para 10 iterações, o desempenho em percentil é menos competitivo, em contrapartida, para 20, o terceiro melhor desempenho, e para 30, o melhor dentre todos as abordagens. O PILS também se mostra muito forte em relação ao tempo de CPU requerido, para todos os níveis testados.

Com o teste destas duas instâncias, pode-se verificar que o PILS apresenta um bom desempenho na aplicação ao sequenciamento de MMAL, obtendo boas soluções, com tempo computacional muito competitivo, quando comparado a outros métodos. Mesmo com o menor nível de iterações no seu processo de busca, ele foi capaz de superar as meta-heurísticas ACO,

GA, SA e TS, nas condições em que estas abordagens foram modeladas e parametrizadas por McMullen (2001b), para os conjuntos de instância testados.

Sendo assim, é possível responder a primeira questão desta pesquisa. Sim, o PILS pode ser considerado eficiente no sequenciamento de MMAL.

4.2 APLICAÇÃO EM UM CASO REAL

Após testar o PILS em instâncias da literatura e compará-los a outro método de resolução, verificou-se que a escolha deste método para aplicação ao problema de sequenciamento de MMAL foi justificada. A seguir, será feita uma aplicação a um caso real, e o sequenciamento resultante será comparado ao sequenciamento realizado atualmente.

O objeto de estudo do caso real trata-se de uma linha de montagem de uma empresa multinacional do ramo metalúrgico, do segmento de autopeças. Nesta linha de montagem, seis modelos diferentes de chassis são montados. Estes modelos possuem componentes comuns, mas também componentes distintos. Atualmente, o sequenciamento de cada turno é feito de modo a agrupar a produção de cada modelo (ex.: 1-1-1-2-2-2-2-3...).

A linha de montagem foi projetada de forma que o tempo de setup entre os modelos pode ser considerado zero. Portanto, para manter a diferença entre o consumo real e ideal de componentes ao longo do turno baixa, o programador da linha pode optar por ter um número maior de setups. Por outro lado, um ponto que pode ser visto como negativo em relação a fazer um alto número de setups está em eventuais erros operacionais durante a montagem.

Ao entregar ao decisor um *trade-off* entre o número de setups e a variação no consumo de componentes, ele fica apto a escolher uma sequência que melhore a variação, sem prejudicar a qualidade da montagem.

A demanda utilizada para o teste foi baseada em uma média de produção por turno de três meses, e ela foi reduzida utilizando o máximo divisor comum entre a demanda de cada modelo. A demanda considerada está na Tabela 11. Quanto aos componentes, os que são consumidos em quantidades iguais por todos os modelos foram excluídos. O consumo por produto de cada um dos 36 componentes considerados está na Tabela 12.

Tabela 11 - Matriz de Demandas (Q_i) do Caso Real por Produto (A_i)

Produto (A_i)	A_0	A_1	A_2	A_3	A_4	A_5	$Q = \sum Q_i$
Demanda	1	8	3	2	3	1	18

Fonte: Elaborado pela autora.

Tabela 12 - Matriz de componentes (a_j) do Caso Real

Produto (A_i) Componente (a_j)	A_0	A_1	A_2	A_3	A_4	A_5
a_0	2	2	0	2	2	2
a_1	0	0	2	0	0	0
a_2	0	0	0	1	0	1
a_3	1	1	0	0	1	0
a_4	0	0	1	0	0	0
a_5	1	1	0	1	1	1
a_6	0	0	1	0	0	0
a_7	0	0	0	0	2	2
a_8	2	2	2	2	0	0
a_9	0	2	0	2	2	2
a_{10}	2	0	0	0	0	0
a_{11}	0	0	2	0	0	0
a_{12}	6	6	0	6	6	6
a_{13}	0	0	4	0	0	0
a_{14}	2	2	2	2	0	0
a_{15}	0	0	0	0	2	2
a_{16}	2	2	2	0	0	0
a_{17}	0	0	0	2	0	0
a_{18}	0	0	0	0	3	0
a_{19}	0	0	0	0	0	3
a_{20}	4	0	0	0	0	0
a_{21}	0	2	0	2	2	2
a_{22}	4	4	0	4	4	4
a_{23}	1	1	0	1	1	1
a_{24}	0	0	1	0	0	0
a_{25}	3	3	0	3	3	3
a_{26}	0	0	3	0	0	0
a_{27}	2	0	0	0	0	0
a_{28}	0	2	0	2	2	2
a_{29}	0	0	0	1	0	0
a_{30}	1	1	1	0	0	0
a_{31}	0	0	3	0	0	0
a_{32}	0	0	2	0	0	0
a_{33}	0	0	1	0	0	0
a_{34}	1	1	0	1	1	1
a_{35}	0	0	2	0	0	0

Fonte: Elaborado pela autora.

O PILS foi testado com três níveis diferentes de iterações do algoritmo: 18 (PILS-18), 36 (PILS-36) e 54 (PILS-54) iterações. Cada teste foi executado 10 vezes. A comparação dos resultados é mostrada na Tabela 13.

Desta vez, como seria inviável a obtenção da solução ótima por enumeração completa, dado que o número de soluções possíveis é 2.205.403.200, as métricas para avaliar as soluções são diferentes do que as utilizadas na subseção anterior.

Para cada teste, a coluna Solução Média representa a média das soluções das 10 execuções para cada setup associado e, entre parênteses, o *desvio* calculado por (11). A coluna do *Gap*, calculada por (12), representa a diferença entre a Solução Média e a melhor solução conhecida, entre todos os testes. As últimas linhas da tabela mostram a média das métricas para cada teste e o tempo de execução médio (em segundos).

$$Desvio = \frac{\text{Solução média} - \text{melhor solução do teste}}{\text{melhor solução do teste}} \cdot 100\% \quad (11)$$

$$Gap = \frac{\text{Solução média} - \text{melhor solução conhecida}}{\text{melhor solução conhecida}} \cdot 100\% \quad (12)$$

Tabela 13 - Comparação entre os níveis de iterações para o Caso Real

Setups	PILS-18		PILS-36		PILS-54	
	Solução Média	GAP	Solução Média	GAP	Solução Média	GAP
6	185,41 (0,0%)	0,0%	185,41 (0,0%)	0,0%	185,41 (0,0%)	0,0%
7	161,16 (12,3%)	12,3%	157,56 (9,8%)	9,8%	155,61 (8,4%)	8,4%
8	122,14 (2,1%)	2,1%	121,51 (1,6%)	1,6%	121,89 (1,9%)	1,9%
9	109,62 (2,4%)	2,8%	107,96 (1,2%)	1,2%	107,91 (0,8%)	1,2%
10	91,52 (1,3%)	1,3%	93,22 (3,2%)	3,2%	91,00 (0,8%)	0,8%
11	84,91 (0,7%)	0,7%	85,97 (2%)	2,0%	84,89 (0,7%)	0,7%
12	81,25 (2,3%)	2,3%	80,93 (1,9%)	1,9%	79,60 (0,3%)	0,3%
13	76,55 (0,0%)	0,0%	76,55 (0,0%)	0,0%	76,55 (0,0%)	0,0%
14	73,70 (0,0%)	0,0%	73,70 (0,0%)	0,0%	73,70 (0,0%)	0,0%
15	73,62 (0,0%)	0,0%	73,62 (0,0%)	0,0%	73,62 (0,0%)	0,0%
16	73,53 (0,0%)	0,0%	73,53 (0,0%)	0,0%	73,53 (0,0%)	0,0%
17	73,49 (0,0%)	0,0%	73,49 (0,0%)	0,0%	73,49 (0,0%)	0,0%
18	74,03 (0,0%)	0,0%	74,03 (0,0%)	0,0%	74,03 (0,0%)	0,0%
Média	98,53 (1,6%)	1,7%	98,27 (1,5%)	1,5%	97,79 (1,0%)	1,0%
Tempo (s)	52,904		106,152		149,963	

Fonte: Elaborado pela autora.

A sequência executada atualmente na linha de montagem tem 6 setups e resulta em um Consumo de 225,45. Nos resultados, é possível notar que mesmo que o decisor opte por fazer

apenas 6 setups, existe um modo melhor de realizar o sequenciamento, que foi obtido por todos os testes.

Analisando a média das soluções obtidas, o PILS-18 apresentou a pior solução e o maior desvio e *gap* entre os três testes, porém o melhor tempo computacional. O PILS-36 forneceu solução, desvio e *gap* um pouco melhores, em mais do dobro do tempo do PILS-18. Finalmente, o PILS-54 apresentou a melhor solução, com os menores desvios, em quase o triplo do tempo do PILS-18.

O desempenho dos testes variou para as sequências com setups menores ou iguais a 12. Para maior do que 12, todos os testes obtiveram os mesmos resultados. Portanto, se o decisor entender que vale a pena fazer mais de 12 setups, para obter um melhor nivelamento do consumo de componentes, ele pode escolher qualquer um dos números de iterações do algoritmo, que ele tende a fornecer o mesmo resultado, ou resultados muito próximos. Caso contrário, ele deve fazer o *trade-off* entre o quão perto da solução ótima ele quer estar e o quanto ele está disposto a esperar pelo resultado.

Por outro lado, como o processo de sequenciamento ocorre, geralmente, uma vez por turno, mesmo o PILS-54 demorando cerca de três vezes mais que o PILS-18, os menos de três minutos que ele demora podem ser considerados insignificantes se comparados as oito horas de jornada que o sequenciamento irá atender.

A Tabela 14 mostra a melhor sequência obtida, ou seja, com a menor variação do consumo de componentes, para cada nível de setup.

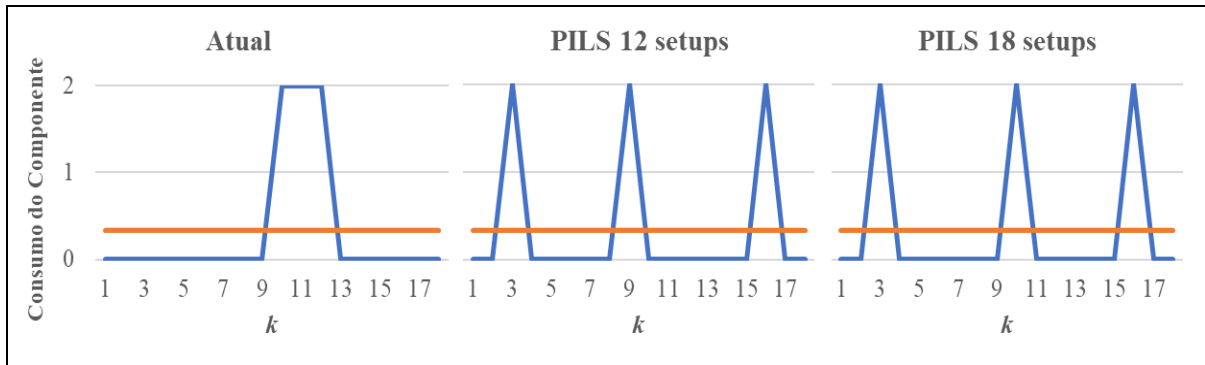
Tabela 14 - Melhores sequências obtidas pelo PILS

Setup	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Consumo
6	3	3	0	4	4	4	2	2	2	1	1	1	1	1	1	1	1	5	185,41
7	3	3	0	4	4	4	2	2	1	1	1	1	1	1	1	1	2	5	143,51
8	1	1	1	1	1	2	2	4	4	4	0	3	3	5	2	1	1	1	119,61
9	1	1	1	2	4	4	4	0	2	3	3	1	1	1	1	1	2	5	106,66
10	1	1	1	2	4	4	0	3	3	2	5	1	1	1	1	1	2	4	90,31
11	1	1	1	2	4	4	3	0	2	5	1	1	1	1	1	2	4	3	84,29
12	3	4	2	1	1	1	1	4	2	0	5	1	1	1	1	2	4	3	79,40
13	1	1	4	2	3	1	1	4	2	0	5	1	1	1	1	2	4	3	76,55
14	1	1	4	2	3	1	1	4	2	0	5	1	1	3	2	4	1	1	73,70
15	1	4	2	1	3	1	1	4	2	0	5	1	1	3	2	4	1	1	73,62
16	1	4	2	1	3	1	1	4	2	0	5	1	1	3	1	2	4	1	73,53
17	1	4	2	1	3	1	1	4	2	1	0	5	1	3	1	2	4	1	73,49
18	1	4	2	1	3	1	0	4	1	2	1	5	1	3	1	2	4	1	74,03

Fonte: Elaborado pela autora.

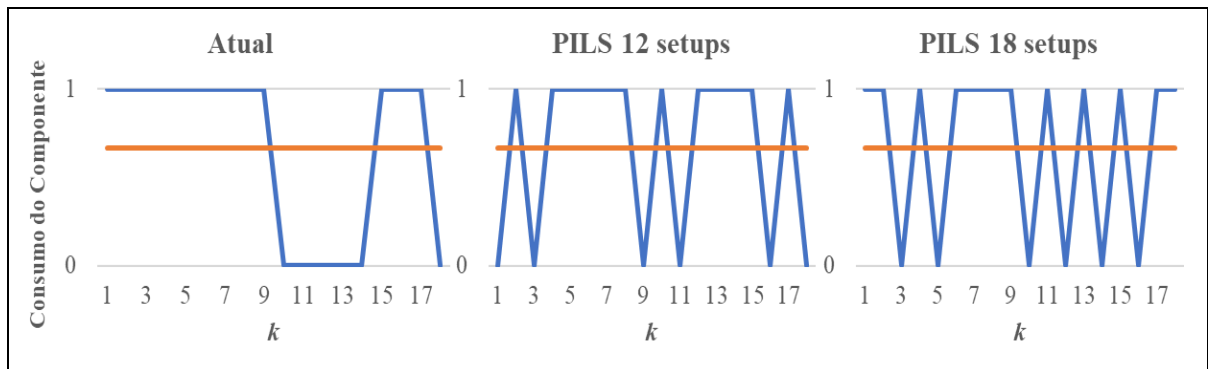
Para auxiliar a visualização das diferenças entre a sequência realizada atualmente e sequências resultantes do PILS, nas Figuras 29, 30, 31 e 32 são apresentados gráficos do consumo dos componentes a_1 , a_3 , a_{12} e a_{16} , respectivamente, ao longo dos instantes.

Figura 29 - Consumo do componente a_1 atualmente, PILS com 12 e 18 setups



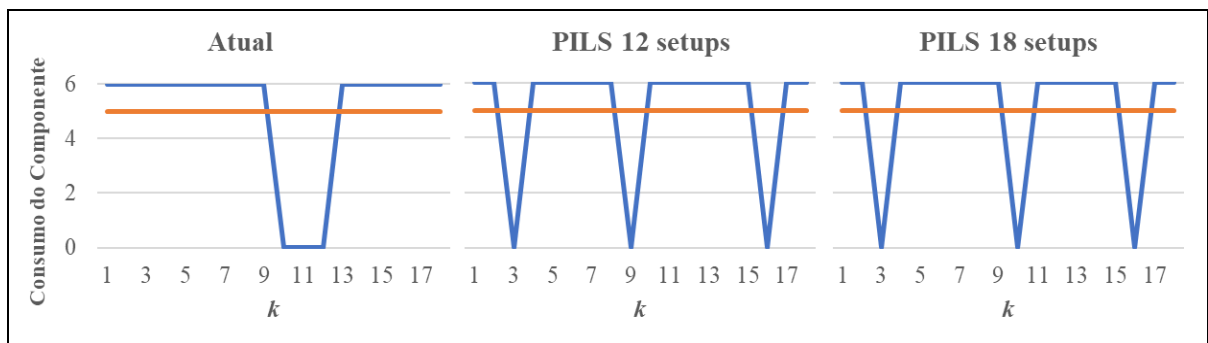
Fonte: Elaborado pela autora.

Figura 30 - Consumo do componente a_3 atualmente, PILS com 12 e 18 setups



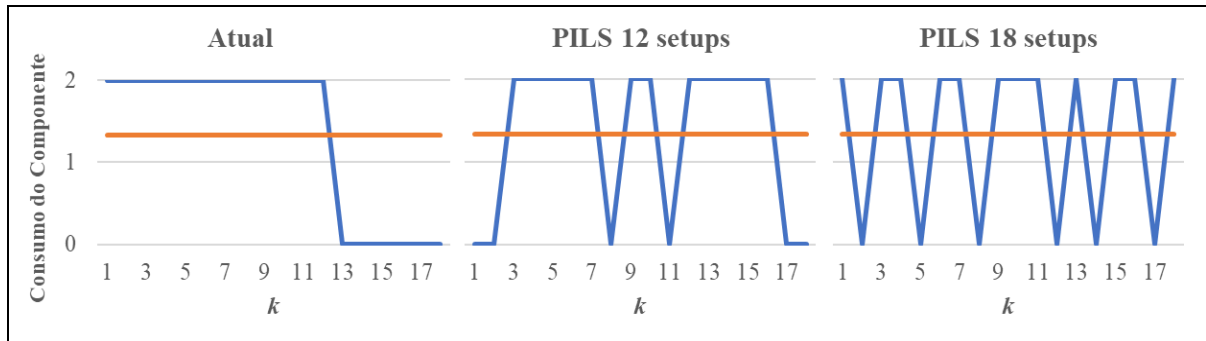
Fonte: Elaborado pela autora.

Figura 31 - Consumo do componente a_{12} atualmente, PILS com 12 e 18 setups



Fonte: Elaborado pela autora.

Figura 32 - Consumo do componente a_{16} atualmente, PILS com 12 e 18 setups



Fonte: Elaborado pela autora.

Os consumos exibidos nas figuras são referentes as sequências: atual (0-1-1-1-1-1-1-1-2-2-2-3-3-4-4-4-5), com 12 setups da Tabela 14, com 18 setups da Tabela 14 e, em laranja, o que seria o consumo ideal em cada instante.

Pode-se observar que o consumo de componentes no sequenciamento atual acontece de forma desnivelada, com picos e vales concentrados. Por outro lado, à medida que o número de setups aumenta, os picos e vales se distribuem ao longo dos instantes, para do consumo se aproximar do consumo ideal.

O consumo do Componente a_1 , por exemplo, é concentrado nos instantes 10, 11 e 12, e é zero em todo o restante, sendo que seu consumo médio ideal por instante é aproximadamente 0,333. Na sequência de 12 setups do PILS, o consumo acontece nos instantes 3, 9 e 16; e com 18 setups, 3, 10 e 17, ou seja, distribuído entre os instantes.

Mantendo o consumo dos componentes próximo do seu consumo médio é possível:

- Dimensionar adequadamente o *kanban*, enquanto na sequência aleatória, o *kanban* pode ser superdimensionado para absorver as variações na demanda;
- Evitar estoques intermediários excessivos;
- Evitar as paradas de linha;
- Evitar o desbalanceamento de carga do processo fornecedor e etc.

Estes resultados podem responder a segunda questão deste trabalho. Ou seja, considerando os objetivos escolhidos para a otimização do sequenciamento neste trabalho, os resultados obtidos melhoram o desempenho da linha de montagem, em relação ao que é praticado, pois as sequencias propostas reduzem a variabilidade no consumo de componentes.

5 CONCLUSÃO

5.1 RESPOSTAS ÀS QUESTÕES DE PESQUISA E OBJETIVOS

Este trabalho teve seu foco no Sequenciamento de Linhas de Montagem de Modelos Mistos Multiobjetivo. O objetivo almejado do sequenciamento é a minimização da distância entre o consumo real e ideal de componentes ao longo dos instantes, e do número de setups, simultaneamente.

Quanto à primeira questão desta pesquisa, de verificar se o PILS é eficiente no sequenciamento de MMAL, foi positivamente respondida. Testando o PILS nas duas instâncias da literatura, pode-se verificar que ele apresenta um bom desempenho na aplicação ao sequenciamento de MMAL, obtendo boas soluções, com tempo computacional muito competitivo, quando comparado a outros métodos. Mesmo com o menor nível de iterações no seu processo de busca, ele foi capaz de superar as meta-heurísticas ACO, GA, SA e TS, nas condições em que estas abordagens foram modeladas e parametrizadas por McMullen (2001b), para os conjuntos de instância testados. Então sim, o PILS pode ser considerado eficiente no sequenciamento de MMAL.

Os resultados obtidos no caso real podem responder a segunda questão deste trabalho, ou seja, considerando os objetivos escolhidos para a otimização do sequenciamento neste trabalho, os resultados obtidos melhoram o desempenho da linha de montagem em relação ao que é praticado.

Na aplicação ao caso real, o PILS é capaz de oferecer um conjunto de boas soluções para o decisor optar. Essas soluções foram obtidas em um tempo irrelevante, quando comparado ao horizonte ao qual o sequenciamento é destinado, e todas apresentam melhora do nivelamento do consumo de componentes ao longo dos períodos, o que consagra o atingimento do objetivo geral deste trabalho.

Desta forma, conclui-se que todas as questões de pesquisa definidas na seção 1.3 que motivaram este trabalho foram respondidas, bem como os objetivos geral e específicos estruturados em seguida.

Como conclusão da contribuição prática, pode-se afirmar que o uso da abordagem proposta se mostra útil na obtenção de boas soluções para suportar o processo de sequenciamento da linha de produção. A solução apresentada é capaz de compor um sistema de manufatura inteligente, e auxiliar na trajetória rumo à Indústria 4.0.

Do ponto de vista acadêmico, fica evidente a aplicação inédita do PILS ao problema do sequenciamento, a consideração dos objetivos de minimizar simultaneamente a variação da taxa de componentes e o número de setups e a aplicação a um caso real contribuem para minimizar as lacunas identificadas.

5.2 SUGESTÕES PARA TRABALHOS FUTUROS

Para futuras pesquisas, algumas sugestões e recomendações podem ser seguidas:

- incluir no modelo mais detalhes práticos encontrados em um ambiente real de produção, criando metas e, portanto, uma nova função objetivo a ser otimizada;
- generalizar o modelo, aplicando-o a casos reais de maior dimensão e complexidade;
- adaptar o modelo a casos em que existam múltiplas linhas de montagem de mesmos modelos;
- desenhar um modelo conectado a informações dos sistemas e sensores da linha, para que a sequência possa considerar inputs em tempo real, como previsão de demanda, mão-de-obra disponível, níveis de estoque dos componentes, disponibilidade de equipamentos.

REFERÊNCIAS

- AKGÜNDÜZ, O. S.; TUNALI, S. An adaptive genetic algorithm approach for the mixed-model assembly line sequencing problem. **International Journal of Production Research**, London, v. 48, n. 17, p. 5157-5179, 2010. Disponível em: <https://doi.org/10.1080/00207540903117857>. Acesso em: 21 set. 2014.
- AKGÜNDÜZ, O. S.; TUNALI, S. A review of the current applications of genetic algorithms in mixed-model assembly line sequencing. **International Journal of Production Research**, London, v. 49, n. 15, p. 4483-4503, 2011. Disponível em: <https://doi.org/10.1080/00207543.2010.495085>. Acesso em: 10 maio 2014.
- AL-AOMAR, R. Handling multi-lean measures with simulation and simulated annealing. **Journal of the Franklin Institute**, Philadelphia, v. 348, n. 7, p. 1506-1522, 2011. Disponível em: <https://doi.org/10.1016/j.jfranklin.2010.05.002>. Acesso em: 07 jun. 2014.
- AL-E-HASHEM, S. M. J. M.; ARYANEZHAD, M. B.; JABBARZADEH, A. A new approach to solve a mixed-model assembly line with a bypass subline sequencing problem. **The International Journal of Advanced Manufacturing Technology**, Guildford, v. 52, n. 9-12, p. 1053-1066, 2011. Disponível em: <https://doi.org/10.1007/s00170-010-2797-x>. Acesso em: 19 maio 2016.
- ALPAY, S. GRASP with path relinking for a multiple objective sequencing problem for a mixed-model assembly line. **International Journal of Production Research**, London, v. 47, n. 21, p. 6001-6017, 2009. Disponível em: <https://doi.org/10.1080/00207540802158291>. Acesso em: 10 maio 2014.
- AROUI, K.; ALPAN, G.; FREIN, Y. Minimising work overload in mixed-model assembly lines with different types of operators: a case study from the truck industry. **International Journal of Production Research**, London, v. 55, n. 21, p. 6305-6326, 2017. Disponível em: <https://doi.org/10.1080/00207543.2017.1346313>. Acesso em: 08 jul. 2019.
- ARROYO, J. E. C. **Heurísticas e metaheurísticas para otimização combinatória multiobjetivo**. 2002. Tese (Doutorado em Engenharia Elétrica) – Universidade Estadual de Campinas, Campinas, 2002. Disponível em: http://repositorio.unicamp.br/bitstream/REPOSIP/260313/1/Arroyo_JoseEliasClaudio_D.pdf. Acesso em: 09 set. 2015.
- ASSIS, L. P. *et al.* Multiobjective vehicle routing problem with fixed delivery and optional collections. **Optimization Letters**, Heidelberg, v. 7, n. 7, p. 1419-1431, 2013. Disponível em: <https://doi.org/10.1007/s11590-012-0551-z>. Acesso em: 22 jun. 2019.
- AZIZ, B. Modeling and analyzing an industry 4.0 communication protocol. **IEEE Internet Of Things Journal**, Dartmouth, v. 7, n. 10, p. 10120-10127, 2020. Disponível em: <https://doi.org/10.1109/JIOT.2020.2999325>. Acesso em: 12 jan. 2021.
- BALLER, R. *et al.* The assembly line feeding problem: an extend formulation with multiple line feeding policies and a case study. **International Journal of Production Economics**, Amsterdam, v. 222, p. 1-19, 2020. Disponível em: <https://doi.org/10.1016/j.ijpe.2019.09.010>. Acesso em: 12 jan. 2021.

BAUTISTA, J.; CANO, J. Minimizing work overload in mixed model assembly lines. **International Journal of Production Economics**, Amsterdam, v. 112, n. 1, p. 177-191, 2008. Disponível em: <https://doi.org/10.1016/j.ijpe.2006.08.019>. Acesso em: 22 maio 2016.

BAUTISTA, J.; COMPANYS, R.; COROMINAS, A. Heuristics and exact algorithms for solving the Monden problem. **European Journal of Operational Research**, Amsterdam, v. 88, n. 1, p. 101-113, 1996. Disponível em: [https://doi.org/10.1016/0377-2217\(94\)00165-0](https://doi.org/10.1016/0377-2217(94)00165-0). Acesso em: 12 dez. 2011.

BELFIORE, P.; FÁVERO, L. P. **Pesquisa operacional**: para cursos de engenharia. Rio de Janeiro: Elsevier, 2013. 277 p.

BERTRAND, J. W. M.; FRANSOO, J. C. Operations management research methodologies using quantitative modeling. **International Journal of Operations & Production Management**, Baltimore, v. 22, n. 2, p. 241-264, 2002. Disponível em: <https://doi.org/10.1108/01443570210414338>. Acesso em: 19 jun. 2019.

BENOTSMANE, R.; DUDÁS, L. Robotic production oriented engine design and manufacturing. In: VEHICLE AND AUTOMOTIVE ENGINEERING, 3., 2020, Miskolc. **Proceedings** [...]. Miskolc: Springer, 2021. Disponível em: https://doi.org/10.1007/978-981-15-9529-5_34. Acesso em: 12 jan. 2021.

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: overview and conceptual comparison. **ACM Computing Surveys**, New York, v. 35, n. 3, p. 268-308, 2003. Disponível em: <https://doi.org/10.1145/937503.937505>. Acesso em: 20 jun. 2019.

BOLAT, A.; SAVSAR, M.; AL-FAWZAN, M. A. Algorithms for real-time scheduling of jobs on mixed model assembly lines. **Computers & Operations Research**, Oxford, v. 21, n. 5, p. 487-498, 1994. Disponível em: [https://doi.org/10.1016/0305-0548\(94\)90098-1](https://doi.org/10.1016/0305-0548(94)90098-1). Acesso em: 17 maio 2016.

BORTOLINI, M. *et al.* Assembly system design in the Industry 4.0 era: a general framework. 2017. **IFAC-Papers Online**, Oxford, v. 50, n. 1, p. 5700-5706, 2017. Disponível em: <https://doi.org/10.1016/j.ifacol.2017.08.1121>. Acesso: 12 jan. 2021.

BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information Sciences**, New York, v. 237, p. 82-117, 2013. Disponível em: <https://doi.org/10.1016/j.ins.2013.02.041>. Acesso em: 20 jun. 2019.

BOYSEN, N.; EMDE, S. Scheduling the part supply of mixed-model assembly lines in line-integrated supermarkets. **European Journal of Operational Research**, Amsterdam v. 239, n. 3, p. 820-829, 2014. Disponível em: <https://doi.org/10.1016/J.EJOR.2014.05.029>. Acesso em: 08 jun. 2014.

BOYSEN, N.; FLIEDNER, M.; SCHOLL, A. Sequencing mixed-model assembly lines: survey, classification and model critique. **European Journal of Operational Research**, Amsterdam, v. 192, n. 2, p. 349-373, 2009. Disponível em: <https://doi.org/10.1016/J.EJOR.2007.09.013>. Acesso em: 08 jun. 2014.

BOYSEN, N.; FLIEDNER, M.; SCHOLL. Production planning of mixed-model assembly lines: overview and extensions. **Production Planning & Control: the management of operations**, Abingdon, v. 20, n. 5, p. 455-471, 2009. Disponível em: <https://doi.org/10.1080/09537280903011626>. Acesso em: 04 set. 2015.

BUERGIN, J. *et al.* Local order scheduling for mixed-model assembly lines in the aircraft manufacturing industry. **Production Engineering**, Berlin, v. 12, n. 6, p. 759-767, 2018. Disponível em: <https://doi.org/10.1007/s11740-018-0852-x>. Acesso em: 27 dez. 2018.

BUKCHIN, J.; DAR-EL, E. M.; RUBINOVITZ, J. Mixed model assembly line design in a make-to-order environment. **Computers & Industrial Engineering**, Oxford, v. 41, n. 4, p. 405-421, 2002. Disponível em: [https://doi.org/10.1016/S0360-8352\(01\)00065-1](https://doi.org/10.1016/S0360-8352(01)00065-1). Acesso em: 17 maio 2016.

BUKCHIN, Y.; RABINOWITCH, I. A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs. **European Journal of Operational Research**, Amsterdam, v. 174, n. 1, p. 492-508, 2006. Disponível em: <https://doi.org/10.1016/j.ejor.2005.01.055>. Acesso em: 17 maio 2016.

CAMUSSI, N. B.; CERDÁ, J.; CAFARO, D. C. Mathematical formulations for the optimal sequencing and lot sizing in multiproduct synchronous assembly lines. **Computers & Industrial Engineering**, Oxford, v. 152, 2021. Disponível em: <https://doi.org/10.1016/j.cie.2020.107006>. Acesso em: 12 jan. 2021.

CELANO, G.; COSTA, A.; FICHERA, S. A comparative analysis of sequencing heuristics for solving the Toyota Goal Chasing problem. **Robotics and Computer Integrated Manufacturing**, Oxford, v. 20, n. 6, p. 573-581, 2004. Disponível em: <https://doi.org/10.1016/j.rcim.2004.07.009>. Acesso em: 12 jan. 2021.

CHAKRAVARTY, A. K.; SHTUB, A. The effect of learning on the operation of mixed-model assembly lines. **Production and Operations Management**, Baltimore, v. 1, n. 2, p. 198-211, 1992. Disponível em: <https://doi.org/10.1111/j.1937-5956.1992.tb00352.x>. Acesso em: 12 dez. 2011.

CHINCHULUUN, A.; PARDALOS, P. M. A survey of recent developments in multiobjective optimization. **Annals in Operations Research**, Dordrecht, v. 154, p. 29-50, 2007. Disponível em: <https://doi.org/10.1007/s10479-007-0186-0>. Acesso em: 20 jan. 2021.

CHUTIMA, P.; JITMETTA, K. Adaptive biogeography-based optimisation for two-sided mixed-model assembly line sequencing problems. **International Journal of Operational Research**, London, v. 16, n. 4, p. 390-420, 2013. Disponível em: <https://doi.org/10.1504/IJOR.2013.052712>. Acesso em: 07 jul. 2019.

CHUTIMA, P.; PINKOOMPEE, P. Multi-objective sequencing problems of mixed-model assembly systems using memetic algorithms. **ScienceAsia**, Bangkok, v. 35, n. 3, p. 295-305, 2009. Disponível em: <https://doi.org/10.2306/scienceasia1513-1874.2009.35.295>. Acesso em: 22 maio 2016.

COELLO-COELLO, C. A.; LAMONT, G. B.; VAN VELDHUIZEN, D. A. **Evolutionary algorithms for solving multi-objective problems**. 2nd. ed. New York: Springer Science,

2007.

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L. **Introduction to algorithms**. 2. ed. Rio de Janeiro: Campus, 2002.

CHAVES, A. A. **Uma meta-heurística híbrida com busca por agrupamentos aplicada a problemas de otimização combinatória**. 2009. Tese (Doutorado em Computação Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2009. Disponível em: <http://www.lac.inpe.br/~lorena/antonio/tese-antonio.pdf>. Acesso em: 05 dez. 2011.

DAR-EL, E. M. Mixed-model assembly line sequencing problems. **Omega: The International Journal of Management Science**, Oxford, v. 6, n. 4, p. 313-323, 1978. Disponível em: [https://doi.org/10.1016/0305-0483\(78\)90004-X](https://doi.org/10.1016/0305-0483(78)90004-X). Acesso em: 20 ago. 2014.

DAR-EL, E. M.; CUCUY, S. Optimal mixed-model sequencing for balanced lines. **Omega: The International Journal of Management Science**, Oxford, v. 5, n. 4, p. 333-342, 1977. Disponível em: [https://doi.org/10.1016/0305-0483\(77\)90115-3](https://doi.org/10.1016/0305-0483(77)90115-3). Acesso em: 04 set. 2015.

DUPLAGA, E. A.; HAHN, C. K.; HUR, D. Mixed-model assembly line sequencing at Hyundai Motor Company. **Production and Inventory Management Journal**, Baltimore, v. 37, n. 3, p. 20-27, 1996. Disponível em: https://www.researchgate.net/publication/280083175_Mixed-Model_Assembly_Line_Sequencing_at_Hyundai_Motor_Company. Acesso em: 04 set. 2010.

FATHI, M., GHOBAKHLOO, M. Enabling mass customization and manufacturing sustainability in industry 4.0 Context: a novel heuristic algorithm for in-plant material supply optimization. **Sustainability**, Basel, v. 12, n. 16, p. 1-15, 2020. Disponível em: <https://doi.org/10.3390/su12166669>. Acesso em: 01 nov. 2014.

FATTAHI, P. *et al.* Sequencing mixed-model assembly lines by considering feeding lines. **The International Journal of Advanced Manufacturing Technology**, Guildford, v. 61, n. 5-8, p. 677-690, 2012. Disponível em: <https://doi.org/10.1007/s00170-011-3726-3>. Acesso em: 08 jun. 2014.

GALINDRES-GUANCHÁ, L. F.; TORO-OCAMPO, E. M.; GALLEGU-RENDÓN, R. A. Multi-objective MDVRP solution considering route balance and cost using the ILS metaheuristic. **International Journal of Industrial Engineering Computations**, Waterloo, v. 9, n. 1, p. 33-46, 2018. Disponível em: <https://doi.org/10.5267/j.ijiec.2017.5.002>. Acesso em: 22 jun. 2019.

GEIGER, M. J. Foundations of the pareto iterated local search metaheuristic. In: INTERNATIONAL CONFERENCE ON MULTIPLE CRITERIA DECISION MAKING, 18., 2006, Chania. **Proceedings** [...], Chania, 2006. Disponível em: <https://arxiv.org/abs/0809.0406>. Acesso em: 04 jan. 2019.

GEIGER, M. J. Decision support for multi-objective flow shop scheduling by the pareto iterated local search methodology. **Computers & Industrial Engineering**, Oxford, v. 61, n. 3, p. 805-812, 2011. Disponível em: <https://doi.org/10.1016/j.cie.2011.05.013>. Acesso em: 04 set. 2015.

GHINATO, P. Elementos fundamentais do sistema toyota de produção. In: ALMEIDA, A. T.; SOUZA, F. D. C. (ed.). **Produção e competitividade: aplicações e inovações**. Recife: Editora da UFPE, 2000, p. 169-189. Disponível em: <http://www.scielo.br/pdf/prod/v5n2/v5n2a04>. Acesso em: 7 jun. 2014.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers & Operations Research**, Oxford, v. 13, n. 5, p. 533-549, 1986. Disponível em: [https://doi.org/10.1016/0305-0548\(86\)90048-1](https://doi.org/10.1016/0305-0548(86)90048-1). Acesso em: 04 nov. 2014.

GOMES, H. C.; NEVES, F. A.; SOUZA, M. J. F. Multi-objective metaheuristic algorithms for the resource-constrained project scheduling problem with precedence relations. **Computers & Operations Research**, Oxford, v. 44, p. 92-104, 2014. Disponível em: <https://doi.org/10.1016/j.cor.2013.11.002>. Acesso em: 22 jun. 2019.

GÜDEN, H.; MERAL, S. An adaptive simulated annealing algorithm-based approach for assembly line balancing and a real-life case study. **The International Journal of Advanced Manufacturing Technology**, Guildford, v. 84, v. 5-8, p. 1539-1559, 2016. Disponível em: <https://doi.org/10.1007/s00170-015-7802-y>. Acesso em: 27 dez. 2018.

HAIMES, Y. Y.; LASDON, L. S.; WISMER, D. A. On a bicriterion formulation of the problems of integrated system identification and system optimization. **IEEE Transactions on Systems, Man and Cybernetics**, New York, v. 1, p. 296-297, 1971. Disponível em: <https://doi.org/10.1109/TSMC.1971.4308298>. Acesso em: 22 jun. 2019.

HAMZADAYI, A.; YILDIZ, G. A simulated annealing algorithm based approach for balancing and sequencing of mixed-model U-lines. **Computers & Industrial Engineering**, Oxford, v. 66, n. 4, p. 1070-1084, 2013. Disponível em: <https://doi.org/10.1016/j.cie.2013.08.008>. Acesso em: 01 nov. 2014.

HYUN, C. J.; KIM, Y.; KIM, Y. K. A genetic algorithm for multiple objective sequencing problems in mixed model assembly lines. **Computers & Operations Research**, Oxford, v. 25, n. 7-8, p. 675-690, 1998. Disponível em: [https://doi.org/10.1016/S0305-0548\(98\)00026-4](https://doi.org/10.1016/S0305-0548(98)00026-4). Acesso em: 10 maio 2014.

INMAN, R. R.; BULFIN, R. L. Sequencing JIT mixed-model assembly lines. **Management Science**, Hanover, v. 37, n. 7, p. 901-904, 1991. Disponível em: <https://doi.org/10.1287/mnsc.37.7.901>. Acesso em: 12 dez. 2011.

OLIVEIRA JUNIOR, P. L.; ARROYO, J. E. C.; SOUZA, V. A. A. Heurística GRASP e ILS para o problema no-wait flowshop scheduling multiobjetivo. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL (SBPO), 42., 2010, Bento Gonçalves. **Anais [...]** Bento Gonçalves: UFSM, 2010. Disponível em: <http://www.din.uem.br/sbpo/sbpo2010/pdf/72558.pdf>. Acesso em: 09 set. 2015.

KAGERMANN, H. *et al.* **Recommendations for implementing the strategic initiative industrie 4.0**. Munich: Acatech – National Academy of Science and Engineering, 2013. Disponível em: <https://www.din.de/blob/76902/e8cac883f42bf28536e7e8165993f1fd/recommendations-for-implementing-industry-4-0-data.pdf>. Acesso em: 16 jan. 2021.

- KIM, Y. K.; HYUN, C. J.; KIM, Y. Sequencing in mixed model assembly lines: a genetic algorithm approach. **Computers & Operations Research**, Oxford, v. 23, n. 12, p. 1131-1145, 1996. Disponível em: [https://doi.org/10.1016/S0305-0548\(96\)00033-0](https://doi.org/10.1016/S0305-0548(96)00033-0). Acesso em: 12 dez. 2012.
- KIM, Y. K.; KIM, J. Y.; KIM, Y. A coevolutionary algorithm for balancing and sequencing in mixed model assembly lines. **Applied Intelligence**, Dordrecht, v. 13, n. 3, p. 247-258, 2000. Disponível em: <https://doi.org/10.1023/A:1026568011013>. Acesso em: 17 maio 2016.
- KORKMAZEL, T.; MERAL, S. Bicriteria sequencing methods for the mixed-model assembly line in just-in-time production systems. **European Journal of Operational Research**, Amsterdam, v. 131, n. 1, p. 188-207, 2001. Disponível em: [https://doi.org/10.1016/S0377-2217\(00\)00061-8](https://doi.org/10.1016/S0377-2217(00)00061-8). Acesso em: 10 maio 2014.
- KOTANI, S.; ITO, T.; OHNO, K. Sequencing problem for a mixed-model assembly line in the Toyota production system. **International Journal of Production Research**, London, v. 42, n. 23, p. 4955-4974, 2004. Disponível em: <https://doi.org/10.1080/0020754042000270377>. Acesso em: 04 set. 2015.
- KRENCZYK, D.; DZIKI, K. A simulated annealing based method for sequencing problem in mixed model assembly lines. In: INTERNATIONAL CONFERENCE ON SOFT COMPUTING MODELS IN INDUSTRIAL AND ENVIRONMENTAL APPLICATIONS, 15., 2020, Burgos. **Proceedings** [...]. Burgos: Springer, 2021, p. 331-341. Disponível em: https://doi.org/10.1007/978-3-030-57802-2_32. Acesso em: 12 jan. 2021.
- KUBIAK, W. Minimizing variation of production rates in just-in-time systems: a survey. **European Journal of Operational Research**, Amsterdam, v. 66, n. 3, p. 259-271, 1993. Disponível em: [https://doi.org/10.1016/0377-2217\(93\)90215-9](https://doi.org/10.1016/0377-2217(93)90215-9). Acesso em: 23 jun. 2014.
- KUBIAK, W.; SETHI, S. A note on "level schedules for mixed-model assembly lines in just-in-time production systems". **Management Science**, Oxford, v. 37, n. 1, p. 121-122, 1991. Disponível em: <https://doi.org/10.1287/mnsc.37.1.121>. Acesso em: 07 jun. 2014.
- KUCUKKOC, I.; ZHANG, D. Z. Simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. **International Journal of Production Research**, London, v. 52, n. 12, p. 3665-3687, 2014. Disponível em: <https://doi.org/10.1080/00207543.2013.879618>. Acesso em: 24 set. 2014.
- LEU, Y.; HUANG, P. Y.; RUSSEL, R. S. Using beam search techniques for sequencing mixed-model assembly lines. **Annals of Operations Research**, Dordrecht, v. 70, n. 0, p. 379-397, 1997. Disponível em: <https://doi.org/10.1023/A:1018938608304>. Acesso em: 16 jan. 2012.
- LEU, Y.; MATHESON, L. A.; REES, L. P. Sequencing mixed-model assembly lines with Genetic Algorithms. **Computers & Industrial Engineering**, Oxford, v. 30, n. 4, p. 1027-1036, 1996. Disponível em: [https://doi.org/10.1016/0360-8352\(96\)00050-2](https://doi.org/10.1016/0360-8352(96)00050-2). Acesso em: 04 set. 2015.
- LI, J.; GAO, J.; SUN, L. Sequencing minimum product sets on mixed-model U-lines to minimise work overload. **International Journal of Production Research**, London, v. 50, n.

18, p. 4977-4993, 2012. Disponível em: <https://doi.org/10.1080/00207543.2011.624559>. Acesso em: 01 nov. 2014.

LIAN, K. *et al.* A modified colonial competitive algorithm for the mixed-model U-line balancing and sequencing problem. **International Journal of Production Research**, London, v. 50, n. 18, p. 5117-5131, 2012. Disponível em: <https://doi.org/10.1080/00207543.2011.653453>. Acesso em: 01 nov. 2014.

LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. **Operations Research**, Berlin, v. 21, n. 2, p. 498-516, 1973. Disponível em: <https://doi.org/10.1287/opre.21.2.498>. Acesso em: 04 nov. 2014.

LIU, Q. *et al.* Advanced scatter search approach and its application in a sequencing problem of mixed-model assembly lines in a case company. **Engineering Optimization**, Abingdon, v. 46, n. 11, p. 1485-1500, 2014. Disponível em: <https://doi.org/10.1080/0305215X.2013.846334>. Acesso em: 01 abr. 2015.

LOURENÇO, H. R. Logistics management: an opportunity for metaheuristics. In: REGO, C.; ALIDAEE, B. (ed.). **Metaheuristics optimization via memory and evolution**. Boston: Kluwer Academic Publishers, 2005, p. 329-356. Disponível em: https://doi.org/10.1007/0-387-23667-8_15. Acesso em: 21 jun. 2019.

LOURENÇO, H.; MARTIN, O.; STUTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. A. (ed.). **Handbook of metaheuristics**. New York: Springer, 2003, p. 320-353. Disponível em: https://doi.org/10.1007/0-306-48056-5_11. Acesso em: 07 jun. 2014.

LYONS, A. C.; UM, J.; SHARIFI, H. Product variety, customisation and business process performance: a mixed methods approach to understanding their relationships. **International Journal of Production Economics**, Amsterdam, v. 221, 2020. Disponível em: <https://doi.org/10.1016/j.ijpe.2019.08.004>. Acesso em: 16 jan. 2021.

MAKAROUNI, I.; PSARRAS, J.; SISKOS, E. Interactive bicriterion decision support for a large scale industrial scheduling system. **Annals of Operations Research**, Dordrecht, v. 227, n. 1, p. 45-61, 2015. Disponível em: <https://doi.org/10.1007/s10479-013-1406-4>. Acesso em: 03 set. 2015.

MANSOURI, S. A. A multi-objective genetic algorithm for mixed-model sequencing on JIT assembly lines. **European Journal of Operational Research**, Amsterdam, v. 167, n.3, p. 696-716, 2005. Disponível em: <https://doi.org/10.1016/j.ejor.2004.07.016>. Acesso em: 10 maio 2014.

MARTÍ, R.; REINELT, G. Heuristic methods. In: MARTÍ, R.; REINELT, G. **The linear ordering problem, exact and heuristic methods in combinatorial optimization**. Heidelberg: Springer Berlin Heidelberg, 2011. p. 17-40. Disponível em: https://link.springer.com/chapter/10.1007/978-3-642-16729-4_2. Acesso em: 04 nov. 2014.

McMULLEN, P. R. A kohonen self-organizing map approach to addressing a multiple objective, mixed-model JIT sequencing problem. **International Journal of Production Economics**, Amsterdam, v. 72, n. 1, p. 59-71, 2001. Disponível em: [https://doi.org/10.1016/S0925-5273\(00\)00091-8](https://doi.org/10.1016/S0925-5273(00)00091-8). Acesso em: 12 dez. 2011.

McMULLEN, P. R. An ant colony optimization approach to addressing a just in time sequencing problem with multiple objectives. **Artificial Intelligence in Engineering**, Oxford, v. 15, n. 3, p. 309-317, 2001. Disponível em: [https://doi.org/10.1016/S0954-1810\(01\)00004-8](https://doi.org/10.1016/S0954-1810(01)00004-8). Acesso em: 09 nov. 2015.

McMULLEN, P. R. An efficient frontier approach to addressing JIT sequencing problems with setup via search heuristics. **Computers & Industrial Engineering**, Oxford, v. 41, n. 3, p. 335-353, 2001. Disponível em: [https://doi.org/10.1016/S0360-8352\(01\)00059-6](https://doi.org/10.1016/S0360-8352(01)00059-6). Acesso em: 08 jan. 2020.

McMULLEN, P. R.; FRAZIER, G. V. A simulated annealing approach to mixed-model sequencing with multiple objectives on a just-in-time line. **IIE Transactions**, Philadelphia, v. 32, n. 8, p. 679-686, 2000. Disponível em: <https://doi.org/10.1023/A:1007611509671>. Acesso em: 12 dez. 2011.

MILTENBURG, J. Level schedules for mixed-model assembly lines in just-in-time production systems. **Management Science**, Oxford, v. 35, n. 2, p. 192-207, 1989. Disponível em: <https://doi.org/10.1287/mnsc.35.2.192>. Acesso em: 23 jun. 2014.

MILTENBURG, J.; SINNAMON, G. Scheduling mixed-model multi-level just-in-time production systems. **International Journal of Production Research**, London, v. 27, n. 9, p. 1487-1509, 1989. Disponível em: <https://doi.org/10.1080/00207548908942636>. Acesso em: 24 jul. 2019.

MIRANDA, D. M. BRANKE, J.; CONCEIÇÃO, S. V. Algorithms for the multi-objective vehicle routing problem with hard time windows and stochastic travel time and service time. **Applied Soft Computing**, Amsterdam, v. 70, p. 66-79, 2018. Disponível em: <https://doi.org/10.1016/j.asoc.2018.05.026>. Acesso em: 22 jun. 2019.

MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. **Computers & Operations Research**, Oxford, v. 24, n. 11, p. 1097-1100, 1997. Disponível em: [https://doi.org/10.1016/S0305-0548\(97\)00031-2](https://doi.org/10.1016/S0305-0548(97)00031-2). Acesso em: 07 jun. 2014.

MONDEN, Y. **Toyota production system: practical approach to production management**. Norcross: Industrial Engineering and Management Press; Institute of Industrial Engineers, 1983. p. 181-192.

MORADI, H.; ZANDIEH, M.; MAHDAVI, I. Non-dominated ranked genetic algorithm for a multi-objective mixed-model assembly line sequencing problem. **International Journal of Production Research**, London, v. 49, n. 12, p. 3479-3499, 2011. Disponível em: <https://doi.org/10.1080/00207540903433882>. Acesso em: 27 maio 2016.

NADERI, B. *et al.* Multi-objective no-wait flowshop scheduling problems: models and algorithms. **International Journal of Production Research**, London, v. 50, n. 10, p. 2592-2608, 2012. Disponível em: <https://doi.org/10.1080/00207543.2010.543937>. Acesso em: 22 jun. 2019.

OLIVEIRA, A. C. **Algoritmos evolutivos híbridos com detecção de regiões promissoras em espaços de busca contínuos e discretos**. 2005. Tese (Doutorado em Computação

Aplicada) – Instituto Nacional de Pesquisas Espaciais, São José dos Campos, 2005.
Disponível em: <http://www.lac.inpe.br/~lorena/alexandre/tese-alexandre.pdf>. Acesso em: 24 nov. 2013.

PONNAMBALAM, S. G.; ARAVINDAN, P.; RAO, M. S. Genetic algorithms for sequencing problems in mixed model assembly lines. **Computers & Industrial Engineering**, Oxford, v. 45, n. 4, p. 669-690, 2003. Disponível em: <https://doi.org/10.1016/j.cie.2003.09.001>. Acesso em: 10 maio 2014.

RABBANI, M.; RAHIMI-VAHED; A. TORABI, S. A. Real options approach for a mixed-model assembly line sequencing problem. **The International Journal of Advanced Manufacturing Technology**, Guildford, v. 37, n. 11-12, p. 1209-1219, 2008. Disponível em: <https://doi.org/10.1007/s00170-007-1058-0>. Acesso em: 10 maio 2014.

RAHIMI-VAHED, A. R.; MIRZAEI, A. H. A hybrid multi-objective shuffled frog-leaping algorithm for a mixed-model assembly line sequencing problem. **Computers & Industrial Engineering**, Oxford, v. 53, n. 4, p. 642-666, 2007. Disponível em: <https://doi.org/10.1016/j.cie.2007.06.007>. Acesso em: 04 set. 2015.

RAHIMI-VAHED, A. R.; MIRGHORBANI, S. M.; RABBANI, M. A hybrid multi-objective particle swarm algorithm for a mixed-model assembly line sequencing problem. **Engineering Optimization**, Abingdon, v. 39, n. 8, p. 877-898, 2007. Disponível em: <https://doi.org/10.1080/03052150701512042>. Acesso em: 04 set. 2015.

RAHIMI-VAHED, A. R. *et al.* A multi-objective scatter search for a mixed model assembly line sequencing problem. **Advanced Engineering Informatics**, Oxford, v. 21, n. 1, p. 85-99, 2007. Disponível em: <https://doi.org/10.1016/j.aei.2006.09.007>. Acesso em: 12 dez. 2011.

SAIF, U. *et al.* Multi-objective artificial bee colony algorithm for order oriented simultaneous sequencing and balancing of multi-mixed model assembly line. **Journal of Intelligent Manufacturing**, Dordrecht, v. 39, n. 3, p. 1195-1220, 2019. Disponível em: <https://doi.org/10.1007/s10845-017-1316-4>. Acesso em: 08 jul. 2019.

SALEHI, M. *et al.* Multi-criteria sequencing problem in mixed-model synchronous assembly lines. **The International Journal of Advanced Manufacturing Technology**, Guildford, v. 67, n. 1-4, p. 983-993, 2013. Disponível em: <https://doi.org/10.1007/s00170-012-4542-0>. Acesso em: 24 set. 2014.

SANCHES, A. L.; MARINS, F. A. S.; MONTEVECHI, J. A. B. Proposta para modelagem do sequenciamento de linhas de montagem múltiplas em ambiente de produção enxuta utilizando simulação. In: SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL (SBPO), 42., 2010, Bento Gonçalves. **Anais [...]**. Bento Gonçalves: UFSM, 2010, p. 1153-1164. Disponível em: <http://www.din.uem.br/sbpo/sbpo2010/pdf/72327.pdf>. Acesso em: 03 ago. 2019.

SCHOLL, A. **Balancing and sequencing of assembly lines**. 2nd ed. Heidelberg: Physica-Verlag, 1999. 318 p. Disponível em: <https://www.springer.com/gp/book/9783790811803>. Acesso em: 27 maio 2016.

SCHULZ, S.; NEUFELD, J. S.; BUSCHER, U. A multi-objective iterated local search

algorithm for comprehensive energy-aware hybrid flow shop scheduling. **Journal of Cleaner Production**, Oxford, v. 224, n. 1, p. 421-434, 2019. Disponível em: <https://doi.org/10.1016/j.jclepro.2019.03.155>. Acesso em: 22 jun. 2019.

SHINGO, S. **O sistema Toyota de produção do ponto de vista da engenharia de produção**. 2. ed. Porto Alegre: Artes Médicas, 1996. 291 p.

SCOPUS. **Base de dados bibliográficos**. Disponível em: <https://www.scopus.com>. Acesso em: 17 jan 2021.

SILVA, G. G. M. P.; TUBINO, D. F.; SEIBEL, S. Linhas de montagem: revisão da literatura e oportunidades para pesquisas futuras. **Production**, São Paulo, v. 25, n. 1, p. 170-182, 2015. Disponível em: <https://doi.org/10.1590/S0103-65132014005000001>. Acesso em: 09 abr. 2015.

SIMARIA, A. S.; VILARINHO, P. M. 2-ANTBAL: an ant colony optimisation algorithm for balancing two-sided assembly lines. **Computers & Industrial Engineering**, Oxford, v. 56, n. 2, p. 489-506, 2009. Disponível em: <https://doi.org/10.1016/j.cie.2007.10.007>. Acesso em: 24 set. 2014.

SOARES, P. B. *et al.* O. Análise bibliométrica da produção científica brasileira sobre Tecnologia de Construção e Edificações na base de dados Web of Science. **Ambiente Construído**, Porto Alegre, v. 16, n. 1, p. 175-185, 2016. Disponível em: <https://doi.org/10.1590/s1678-86212016000100067>. Acesso em: 01 jul. 2019.

SOTOUDEH-ANVARI, A.; HAFEZALKOTOB, A. A bibliography of metaheuristics-review from 2009 to 2015. **International Journal of Knowledge-based and Intelligent Engineering Systems**, Amsterdam, v. 22, n.1, p. 83-95, 2018. Disponível em: <https://doi.org/10.3233/KES-180376>. Acesso em: 11 jun. 2019.

STONE, K. B. Four decades of lean: a systematic literature review. **International Journal of Lean Six Sigma**, Bingley, v. 3, n. 2, p. 112-132, 2012. Disponível em: <https://doi.org/10.1108/20401461211243702>. Acesso em: 02 nov. 2014.

TAMURA, T.; LONG, H.; OHNO, K. A sequencing problem to level part usage rates and work loads for a mixed-model assembly line with a bypass subline. **International Journal of Production Economics**, Amsterdam, v. 60-61, p. 557-564, 1999. Disponível em: [https://doi.org/10.1016/S0925-5273\(98\)00169-8](https://doi.org/10.1016/S0925-5273(98)00169-8). Acesso em: 24 maio 2016.

TAVAKKOLI-MOGHADDAM, R.; RAHIMI-VAHED, A. R. Multi-criteria sequencing problem for a mixed-model assembly line in a JIT production system. **Applied Mathematics and Computation**, New York, v. 181, n. 2, p. 1471-1481, 2006. Disponível em: <https://doi.org/10.1016/j.amc.2006.02.033>. Acesso em: 10 maio 2014.

THOMOPOULOS, N. Balancing-sequencing for mixed-model assembly. **Management Science**, Oxford, v. 14, n. 2, p. B59-B75, 1967. Disponível em: <https://doi.org/10.1287/mnsc.14.2.B59>. Acesso em: 23 jun. 2014.

TRANFIELD, D.; DENYER, D.; SMART, P. Towards a methodology for developing evidence-informed management knowledge by means of systematic review. **British Journal**

of Management, Oxford, v. 14, n. 3, p. 207-222, 2003. Disponível em: <https://doi.org/10.1111/1467-8551.00375>. Acesso em: 28 maio 2015.

WOMACK, J. P.; JONES, D. T. **A mentalidade enxuta nas empresas lean thinking: elimine o desperdício e crie riqueza**. Rio de Janeiro: Elsevier, 2004. 408 p.

WÖRNER, D. Sequenzbildung in der automobilmontage anhand von materialorientierten Nivellierungsstrategien. **Audi Dissertationsreihe**, Göttingen, v.106, p. 174, 2015. Disponível em: <https://cuvillier.de/en/shop/publications/6985-sequenzbildung-in-der-automobilmontage-anhand-von-materialorientierten-nivellierungsstrategien>. Acesso em: 12 jan. 2021.

XIAOBO, Z.; OHNO, K. Algorithms for sequencing mixed models on an assembly line in a JIT production system. **Computers & Industrial Engineering**, Oxford, v. 32, n. 1, p. 47-56, 1997. Disponível em: [https://doi.org/10.1016/S0360-8352\(96\)00193-3](https://doi.org/10.1016/S0360-8352(96)00193-3). Acesso em: 04 set. 2015.

XIAOBO, Z.; OHNO, K. A sequencing problem for a mixed-model assembly line in a JIT production system. **Computers & Industrial Engineering**, Oxford, v. 27, n. 1-4, p. 71-74, 1994. Disponível em: [https://doi.org/10.1016/0360-8352\(94\)90240-2](https://doi.org/10.1016/0360-8352(94)90240-2). Acesso em: 04 set. 2015.

XIAOBO, Z.; OHNO, K. Properties of a sequencing problem for a mixed model assembly line with conveyor stoppages. **European Journal of Operational Research**, Amsterdam, v. 124, n. 3, p. 560-570, 2000. Disponível em: [https://doi.org/10.1016/S0377-2217\(99\)00198-8](https://doi.org/10.1016/S0377-2217(99)00198-8). Acesso em: 04 set. 2015.

XIAOBO, Z.; ZHOU, Z. Algorithms for Toyota's goal of sequencing mixed models on an assembly line with multiple workstations. **Journal of the Operational Research Society**, Basingstonke, v. 50, n. 7, p. 704-710, 1999. Disponível em: <https://doi.org/10.1057/palgrave.jors.2600750>. Acesso em: 24 set. 2014.

XIAOBO, Z. *et al.* Sequence-to-customer goal with stochastic demands for a mixed model assembly line. **International Journal of Production Research**, London, v. 44, n. 24, p. 5279-5305, 2006. Disponível em: <https://doi.org/10.1080/00207540600597195>. Acesso em: 23 maio 2016.

XIONG, N. *et al.* A walk into metaheuristics for engineering optimization: principles, methods and recent trends. **International Journal of Computational Intelligence Systems**, Paris, v. 8, n. 4, p. 606-636, 2015. Disponível em: <https://doi.org/10.1080/18756891.2015.1046324>. Acesso em: 04 set. 2015.

XU, L. D.; XU, E. L.; LI, L. Industry 4.0: state of the art and future trends. **International Journal of Production Research**, London, v. 56, n. 8, p. 2941-2962, 2018. Disponível em: <https://doi.org/10.1080/00207543.2018.1444806>. Acesso em: 12 jan. 2021.

XU, J. *et al.* An iterated local search for the multi-objective permutation flowshop scheduling problem with sequence-dependent setup times. **Applied Soft Computing**, Amsterdam, v. 52, p. 39-47, 2017. Disponível em: <https://doi.org/10.1016/j.asoc.2016.11.031>. Acesso em: 22 jun. 2019.

YANO, C. A.; RACHAMADUGU, R. Sequencing to minimize work overload in assembly lines with product options. **Management Science**, Oxford, v. 35, n. 5, p. 572-586, 1991. Disponível em: <https://doi.org/10.1287/mnsc.37.5.572>. Acesso em: 03 set. 2015.

YAVUZ, M.; AKÇALI, E. Production smoothing in just-in-time manufacturing systems: a review of the models and solution approaches. **International Journal of Production Research**, London, v. 45, n. 16, p. 3579-3597, 2007. Disponível em: <https://doi.org/10.1080/00207540701223410>. Acesso em: 10 maio 2014.

ZADEH, L. Optimality and non-scalar-valued performance criteria. **IEEE Transactions on Automatic Control**, Piscataway, v. 8, p. 59–60, 1963. Disponível em: <https://doi.org/10.1109/TAC.1963.1105511>. Acesso em: 16 jan. 2021.

ZANDIEH, M.; MORADI, H. An imperialist competitive algorithm in mixed model assembly line sequencing problem to minimise unfinished works. **International Journal of Systems Science: Operations & Logistics**, Abingdon, v. 6, n. 2, p. 179-192, 2017. Disponível em: <https://doi.org/10.1080/23302674.2017.1401152>. Acesso em: 08 jul. 2019.

ZHANG, B.; XU, L.; ZHANG, J. Balancing and sequencing problem of mixed-model U-shaped robotic assembly line: Mathematical model and dragonfly algorithm based approach. **Applied Soft Computing**, Amsterdam, v. 98, 2021. Disponível em: <https://doi.org/10.1016/j.asoc.2020.106739>. Acesso em: 12 jan. 2021.

ZHONG, R. Y. *et al.* Intelligent manufacturing in the context of industry 4.0: a review. **Engineering**, Beijing, 2017, v. 3, n.5, p. 616-630, 2017. Disponível em: <https://doi.org/10.1016/J.ENG.2017.05.015>. Acesso em: 16 jan. 2021.

ZHOU, A. *et al.* Multiobjective evolutionary algorithms: a survey of the state of the art. **Swarm and Evolutionary Optimization**, Amsterdam, v. 1, n. 1, p. 32-49, 2011. Disponível em: <https://doi.org/10.1016/j.swevo.2011.03.001>. Acesso em: 26 abr. 2015.

ZHU, Q.; ZHANG, J. Ant colony optimisation with elitist ant for sequencing problem in a mixed model assembly line. **International Journal of Production Research**, London, v. 49, n. 15, p. 4605-4626, 2011. Disponível em: <https://doi.org/10.1080/00207543.2010.493534>. Acesso em: 04 out. 2014.

ZHUQI, X.; SHUSAKU, H. A study on sequencing method for the mixed-model assembly line in just-in-time production systems. **Computers & Industrial Engineering**, Oxford, v. 27, n. 1-4, p. 225-228, 1994. Disponível em: [https://doi.org/10.1016/0360-8352\(94\)90276-3](https://doi.org/10.1016/0360-8352(94)90276-3). Acesso em: 04 set. 2015.

APÊNDICE A - CÓDIGO EM C++

```

/*****
** PROGRAMA EM C PARA RESOLVER O PROBLEMA DE SEQUENCIAMENTO **
** DE LINHAS DE MONTAGEM APLICANDO O PILS **
**
** AUTOR: MARIANA MONTEIRO USHIZIMA (FEG/UNESP) **
*****/

#include <algorithm>
#include <cstdio>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>
#include <math.h>
#include <stdbool.h>

#pragma argsused
#pragma hdrstop

#define MAX(x,y) ((x)<(y) ? (y) : (x))

//demanda de produtos, quantidade necessária e quantidade total de cada componente
char nomearq1[50] = "Set2_J.txt";

/**** Dados da linha de montagem ****/
#define p 5 //número de produtos
#define c 5 //número de componentes
#define Q 12 //quantidade de demanda diária de todos os produtos

#define k1 10 //número de soluções geradas

/**** Variáveis Globais ****/
int A[p], //matriz com a demanda de cada produto
    B[p][c], //matriz com as quantidade necessária de componentes para produzir uma
    unidade de um produto
    N[c]; //vetor com a quantidade total necessária dos componentes
int IterMax = 20;

/**** Estrutura: TSol - Representar uma solução ****/
struct TSol
{
    int seq[Q];
    double fu;
    int fs;
};

/**** Vetor Fronteira Pareto ****/
TSol FP[Q]; //vetor que armazena soluções Papprox
float FPmedia[Q];
int Xinv[Q]; // vetor que armazena informação se solução foi investigada

/**** Declaração de Funções ****/
void LerArquivos();
void ImprimirSolucao(TSol s);
TSol GerarSolucaoInicial();
double CalcularFU(TSol s);
int CalcularFS(TSol s);

```

```

TSol PrimeiroMovimento(TSol s, int pos1, int pos2);
TSol SegundoMovimento(TSol s, int pos1, int pos2);
TSol TerceiroMovimento(TSol s, int pos1, int pos2);
void GerarVizinhanca(TSol s);
TSol Perturbar(TSol s);
void PILS();
double randomico(double min, double max);
int irandomico(int min, int max);

FILE*tempos;

clock_t CPUinicio,
      CPUatual,
      CPUfim;

float tempoMelhor = 0.0,
      tempoTotal = 0.0,
      tempoMedio = 0.0;

FILE*resultadoFinal;

/*****
*** Método: Main()
*** Função: Programa principal
*****/
int main()
{
    for(int w=0; w<1; w++)
    {
        for(int j=0; j<Q; j++)
            FPmedia[j] = 0;

        resultadoFinal = fopen("Resultado_PILS.txt", "a");

        fprintf(resultadoFinal, "\n*****\n\n");
        fclose(resultadoFinal);

        for(int i=0; i<k1; i++)
        {
            tempoMelhor = 0.0,
            tempoTotal = 0.0;

            printf("\n***** Iteracao %d *****\n", i+1);

            srand(time(NULL));
            LerArquivos();

            for(int j=0; j<Q; j++)
                FP[j].fu = 99999999;

            CPUinicio = CPUfim = clock();

            resultadoFinal = fopen("Sequencias_PILS.txt", "a");
            fprintf(resultadoFinal, "\n\nSOLUCAO ENCONTRADA %d", i+1);
            fprintf(resultadoFinal, "\n*****");
            fprintf(resultadoFinal, "\n\n Sequenciamento | FS | FU | Tempo Atual \n");
            fclose(resultadoFinal);

            PILS();

```

```

        resultadoFinal = fopen("Resultado_PILS.txt","a");
        fprintf(resultadoFinal, "\n\nSOLUCAO ENCONTRADA %d", i+1);
        fprintf(resultadoFinal, "\nSetups | Usage \n");

        for(int j=0; j<Q; j++)
            fprintf(resultadoFinal, " %d |%.3lf\n", j+1, FP[j].fu);

        fprintf(resultadoFinal, "\n\nTempo total: %.3f", (float)(CPUfim - CPUinicio)/CLOCKS_PER_SEC);
        fclose(resultadoFinal);

        for(int j=0; j<Q; j++)
            FPmedia[j] += FP[j].fu;

    tempoMedio += tempoTotal;
}

        resultadoFinal = fopen("Resumo_PILS.txt","a");

        fprintf(resultadoFinal, "\n*****\n\n");

        fprintf(resultadoFinal, "\nSetups | Usage \n");
        for(int j=0; j<Q; j++)
            fprintf(resultadoFinal, " %d |%.3lf\n", j+1, FPmedia[j]/k1);
        fprintf(resultadoFinal, "\n\nTempo total: %.3f", tempoMedio/k1);
        fclose(resultadoFinal);
    }
    return 0;
}

/*****
*** Método: LerArquivos() ***
*** Função: Ler os arquivos de produtos e componentes ***
*****/
void LerArquivos()
{
    FILE *arquivo;

    //Abrir arquivo
    arquivo = fopen(nomearq1, "r");
    if(!arquivo)
    {
        printf("O Arquivo %s nao pode ser aberto.\n", nomearq1);
        getchar();
        exit(1);
    }

    // Ler as demandas dos produtos
    for(int j=0; j<p; j++)
        fscanf(arquivo, "%d", &A[j]);

    //Ler a quantidade necessaria de cada componente
    for(int i=0; i<p; i++)
        for(int j=0; j<c; j++)
            fscanf(arquivo, "%d", &B[i][j]);

    //Ler a quantidade total necessaria de cada componente
    for (int i=0; i<c; i++)

```

```

        fscanf(arquivo, "%d", &N[i]);

        fclose(arquivo);
    }

/*****
*** Método: ImprimirSolucao() ***
*** Função: Imprimir a melhor solucao encontrada no arquivo de resultados ***
*****/
void ImprimirSolucao(TSol s)
{
    resultadoFinal = fopen("Sequencias_PILS.txt", "a");

    for (int i=0; i<Q; i++)
        fprintf(resultadoFinal, "%d ", s.seq[i]);

    fprintf(resultadoFinal, "| %d ", s.fs);
    fprintf(resultadoFinal, "| %.3f ", s.fu);
    fprintf(resultadoFinal, "| %.3f \n", (float)(CPUatual - CPUinicio)/CLOCKS_PER_SEC);

    fclose(resultadoFinal);
}

/*****
*** Método: GerarSolucaoInicial() ***
*** Função: Gerar Solucao Inicial Aleatorimante ***
*****/
TSol GerarSolucaoInicial()
{
    TSol s;

    //número de produtos sequenciados
    int prod[p];

    //inicializar o número de produtos sequenciados
    for (int k=0; k<p; k++)
        prod[k] = 0;

    //alocar os Q produtos para na linha
    for (int j=0; j<Q; j++)
    {
        int x = irandomico(0,p-1);

        //encontrar um produto que nao tenha sido totalmente sequenciado
        while (prod[x] >= A[x])
            x = irandomico(0,p-1);

        s.seq[j] = x;
        prod[x]++;
    }

    s.fu = CalcularFU(s);
    s.fs = CalcularFS(s);

    //imprimir solucao
    ImprimirSolucao(s);

    return s;
}

```

```

/*****
*** Método: CalcularFU(TSol s)
*** Função: Calcula o valor da função usage da solução s
*****/
double CalcularFU(TSol s)
{
    double DTotal = 0.0, Dk, d;
    int X[c], K, aux1;

    //inicializar X
    for(int i=0; i<c; i++)
        X[i] = 0;

    //calcular cada valor de Dk
    for(int i=0; i<Q; i++)
    {
        K = i+1;

        //calcular a diferenca entre o ideal e o real
        Dk = 0;
        for (int j=0; j<c; j++)
        {
            aux1 = 0 ;

            aux1 += B[s.seq[i]][j];

            double aux = (double)K*N[j];
            d = (aux/Q) - X[j] - aux1;
            Dk = Dk + d*d;

            //ajustar o consumo X[j]
            X[j] = X[j] + aux1;
        }

        //Dk = sqrt(Dk);

        DTotal += Dk;
    }

    CPUatual = clock();

    return DTotal;
}

/*****
*** Método: CalcularFS(TSol s)
*** Função: Calcula o valor da função setup da solução s
*****/
int CalcularFS(TSol s)
{
    int STotal = 1;

    //calcular cada valor de sk
    for(int i=0; i<Q-1; i++)
    {
        if(s.seq[i+1] - s.seq[i] != 0)
            STotal++;
    }

    return STotal;
}

```

```

}

/*****
*** Método: PrimeiroMovimento(TSol s, int a, int b) ***
*** Função: Realizar troca entre dois produtos ***
*****/
TSol PrimeiroMovimento(TSol s, int pos1, int pos2)
{
    TSol sViz = s;

    sViz.seq[pos1] = s.seq[pos2];
    sViz.seq[pos2] = s.seq[pos1];

    return sViz;
}

/*****
*** Método: SegundoMovimento(TSol s,int a, int b) ***
*** Função: Shift para frente ***
*****/
TSol SegundoMovimento(TSol s, int pos1, int pos2)
{
    TSol sViz = s;

    sViz.seq[pos2] = s.seq[pos1];
    for(int a=pos1; a<pos2; a++)
        sViz.seq[a] = s.seq[a+1];

    return sViz;
}

/*****
*** Método: TerceiroMovimento(TSol s,int a, int b) ***
*** Função: Shift para trás ***
*****/
TSol TerceiroMovimento(TSol s, int pos1, int pos2)
{
    TSol sViz = s;

    sViz.seq[pos2] = s.seq[pos1];
    for(int a=pos2+1; a<=pos1; a++)
        sViz.seq[a] = s.seq[a-1];

    return sViz;
}

/*****
*** Método: GerarVizinhanca() ***
*** Função: Gera todos os vizinhos da solução inicial ***
*****/
void GerarVizinhanca(TSol sInicial)
{
    TSol sViz, sMelhorViz;
    int m, inViz[3];

    sMelhorViz = sInicial;

    for(int i=0; i<3; i++)
        inViz[i] = 0;

```

```

while((inViz[0] == 0) || (inViz[1] == 0) || (inViz[2] == 0))
{
    m = irandomico(1,3);

    switch(m)
    {
        case 1:    // fazer todos os Swaps
            {
                if(inViz[0] == 0)
                {
                    inViz[0] = 1;

                    for(int j=0; j<Q; j++)
                    {
                        for(int k=j+1; k<Q; k++)
                        {
                            sViz = PrimeiroMovimento(sInicial, j, k);
                            sViz.fu = CalcularFU(sViz);
                            sViz.fs = CalcularFS(sViz);
                            ImprimirSolucao(sViz);

                            if(FP[sViz.fs-1].fu <= sViz.fu)
                                continue;
                            else
                            {
                                FP[sViz.fs-1] = sViz;
                                Xiniv[sViz.fs-1] = 0;
                            }

                            if(sMelhorViz.fu <= sViz.fu)
                            {
                                {
                                    else if(sViz.fs == sInicial.fs)
                                    {
                                        sMelhorViz = sViz;

                                        for(int i=0; i<3; i++)
                                            inViz[i] = 0;
                                    }
                                }
                            }
                        }
                    }
                }

                sInicial = sMelhorViz;
                break;
            }
            else
                break;

        case 2:    // fazer todos os Shifts para frente
            {
                if(inViz[1] == 0)
                {
                    inViz[1] = 1;

                    for(int j=0; j<Q; j++)
                    {
                        for(int k=j+1; k<Q; k++)
                        {
                            sViz = SegundoMovimento(sInicial, j, k);

```

```

sViz.fu = CalcularFU(sViz);
sViz.fs = CalcularFS(sViz);
ImprimirSolucao(sViz);

if(FP[sViz.fs-1].fu <= sViz.fu)
    continue;
else
{
    FP[sViz.fs-1] = sViz;
    Xinv[sViz.fs-1] = 0;
}

if(sMelhorViz.fu <= sViz.fu)
{
}
else if(sViz.fs == sInicial.fs)
{
    sMelhorViz = sViz;

    for(int i=0; i<3; i++)
        inViz[i] = 0;
}
}
}

sInicial = sMelhorViz;
break;
}
else
break;
}

case 3: // fazer todos os Shifts para trás
{
    if(inViz[2] == 0)
    {
        inViz[2] = 1;

        for(int j=1; j<Q; j++)
        {
            for(int k=0; k<j; k++)
            {
                sViz = TerceiroMovimento(sInicial, j, k);
                sViz.fu = CalcularFU(sViz);
                sViz.fs = CalcularFS(sViz);
                ImprimirSolucao(sViz);

                if(FP[sViz.fs-1].fu <= sViz.fu)
                {
                }
                else
                {
                    FP[sViz.fs-1] = sViz;
                    Xinv[sViz.fs-1] = 0;
                }

                if(sMelhorViz.fu <= sViz.fu)
                {
                }
                else if(sViz.fs == sInicial.fs)

```

```

        {
            sMelhorViz = sViz;

            for(int i=0; i<3; i++)
                inViz[i] = 0;
        }
    }
}

sInicial = sMelhorViz;
break;
    }
else
break;
}
}
}

Xinv[sInicial.fs-1] = 1;
}

/*****
*** Método: Perturbar(TSol s)
*** Função: Fazer perturbação aleatoria
*****/
TSol Perturbar(TSol s)
{
    TSol sViz;

    int j = irandomico(0, Q-4);          //escolher uma posição

    sViz = PrimeiroMovimento(s, j, j+3);
    sViz = PrimeiroMovimento(sViz, j+1, j+2);

    sViz.fs = CalcularFS(sViz);
    sViz.fu = CalcularFU(sViz);

    ImprimirSolucao(sViz);
    return sViz;
}

/*****
*** Método: PILS()
*** Função: Aplica a metaheurística PILS no MMAL
*****/
void PILS()
{
    int Iter = 0;
    int investigar = 1;

    TSol s, sPert;

    // Criar a solucao inicial do ILS
    s = GerarSolucaoInicial();

    // Alocar a solução gerada no conjunto Papprox
    FP[s.fs-1] = s;

    // Gerar Vizinhos
    GerarVizinhanca(s);

```

```

// Inicio do Laço Externo
while(Iter < IterMax)
{

    Iter++;
    investigar = 0;

    // Buscar Vizinhanças não Investigadas
    for(int i=0; i<Q; i++)
    {
        if((Xinv[i] == 0) && (FP[i].fu < 999999))
            investigar = 1;
    }
    if(investigar == 1)
    {
        int escolha = irandomico(0,Q-1);

        while((Xinv[escolha] == 1) || (FP[escolha].fu >= 999999))
        {
            escolha = irandomico(0,Q-1);
        }

        GerarVizinhanca(FP[escolha]);
    }
    else
    {
        int escolha = irandomico(0,Q-1);

        while((FP[escolha].fu >= 999999))
        {
            escolha = irandomico(0,Q-1);
        }

        sPert = Perturbar(FP[escolha]);
        GerarVizinhanca(sPert);
    }
}

//Finalizar o tempo computacional do método
CPUfim = clock();
tempoTotal += (float)(CPUfim - CPUinicio)/CLOCKS_PER_SEC;
}

/*****
*** Método: randomico(double min, double max)                ***
*** Função: Gera um numero aleatorio entre min e max          ***
*****/
double randomico(double min, double max)
{
    return ((double)(rand()%10000)/10000.0)*(max-min)+min;
}

/*****
*** Método: irandomico(int min, int max)                      ***
*** Função: Gera um numero inteiro aleatorio entre min e max  ***
*****/
int irandomico(int min, int max)
{
    return (int)randomico(0,max-min+1.0) + min;
}

```