



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO, DE CIÊNCIAS EXATAS E EDUCAÇÃO
DEPARTAMENTO DE ENG. DE CONTROLE, AUTOMAÇÃO E COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Fernanda de Oliveira Ferreira

Proposta metodológica para solução ótima do problema de carregamento de paletes em um plano bidimensional (2D-PCP)

Blumenau
2024

Fernanda de Oliveira Ferreira

Proposta metodológica para solução ótima do problema de carregamento de paletes em um plano bidimensional (2D-PCP)

Trabalho de Conclusão de Curso de Graduação em Engenharia de Controle e Automação do Centro Tecnológico, de Ciências Exatas e Educação da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Leonardo Rincon Mejia

Blumenau

2024

Ficha catalográfica gerada por meio de sistema automatizado gerenciado pela BU/UFSC.
Dados inseridos pelo próprio autor.

Ferreira, Fernanda de Oliveira

Proposta metodológica para solução ótima do problema de carregamento de paletes em um plano bidimensional (2D-PCP) / Fernanda de Oliveira Ferreira ; orientador, Leonardo Mejia Rincon, 2024.

84 p.

Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Campus Blumenau, Graduação em Engenharia de Controle e Automação, Blumenau, 2024.

Inclui referências.

1. Engenharia de Controle e Automação. 2. Problema de Carregamento de Paletes. 3. Otimização. 4. Tensor computacional. 5. Algoritmo de computação evolutiva. I. Rincon, Leonardo Mejia . II. Universidade Federal de Santa Catarina. Graduação em Engenharia de Controle e Automação. III. Título.

Fernanda de Oliveira Ferreira

Proposta metodológica para solução ótima do problema de carregamento de paletes em um plano bidimensional (2D-PCP)

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Engenheiro de Controle e Automação” e aprovado em sua forma final pelo Curso de Graduação em Engenharia de Controle e Automação.

Blumenau, 03 de Julho de 2024.

Banca Examinadora:

Prof. Dr. Leonardo Rincon Mejia
Universidade Federal de Santa Catarina

Prof. Dr. Daniel Alejandro Ponce Saldias
Universidade Federal de Santa Catarina

Prof. Dr. Julio Cesar Frantz
Centro Universitário de Brusque

Dedico este trabalho a Deus, por sua infinita sabedoria e bênçãos; à minha família, pelo amor, suporte incondicional e incentivo contínuo; ao meu companheiro de vida, pelo amor, pela paciência, encorajamento e por estar sempre ao meu lado; e a todos os professores que me guiaram com dedicação, conhecimento e inspiração. Meu sincero agradecimento a todos que contribuíram para a realização deste sonho e tornaram esta jornada possível.

AGRADECIMENTOS

Agradeço primeiramente a Deus, cuja infinita sabedoria e graça me sustentaram durante toda a jornada deste trabalho. Sem sua orientação e apoio constante, nada disso seria possível.

À minha amada família, em especial aos meus pais Teresinha e José, pela inabalável fé, amor incondicional e apoio inesgotável. Vocês são minha inspiração e minha fortaleza.

Ao meu companheiro de vida Eduardo, por seu apoio emocional, compreensão e por caminhar ao meu lado em cada desafio e conquista ao longo deste caminho.

Aos professores, tanto os que me guiaram diretamente quanto os que compartilharam seu conhecimento de forma indireta, minha gratidão. Em especial, aos mestres da Universidade Federal de Santa Catarina, pela dedicação e pelo incentivo ao conhecimento.

Ao meu orientador Prof. Dr. Leonardo Mejia Rincon, pela generosidade, pela paciência, orientação precisa e pelos *insights* valiosos que moldaram este trabalho.

À Universidade Federal de Santa Catarina, pela infraestrutura oferecida, pelo ambiente de aprendizado e por todas as oportunidades proporcionadas ao longo da minha formação.

Aos meus colegas de curso, pela colaboração, troca de experiências e amizades construídas, que tornaram esta jornada acadêmica ainda mais enriquecedora.

Por fim, a todos que, de alguma forma, contribuíram para este trabalho, meu sincero agradecimento.

Quando você é curioso, encontra muito o que fazer. Mas, é preciso ter coragem para realizar qualquer coisa. - Walt Disney

RESUMO

O PCP (Problema de Carregamento de Paletes) é um desafio significativo no campo da engenharia, com amplas aplicações em setores como paletização automatizada e robótica. Para resolver o PCP, uma variedade de algoritmos e técnicas têm sido utilizadas, incluindo heurísticas, meta-heurísticas, programação matemática e inteligência artificial. A paletização, em termos gerais, refere-se ao carregamento e organização eficiente de produtos em paletes para fins de transporte ou armazenamento. No entanto, isto representa um desafio multifacetado, dada a vasta gama de fatores que devem ser meticulosamente avaliados para alcançar uma solução ótima. A seleção de parâmetros contribui para inúmeras variações do PCP, sendo o 2D-PCP (Problema de Carregamento de Paletes Bidimensional) uma delas. Para resolver este problema, este estudo propõe uma nova metodologia desenvolvida para otimizar a distribuição de itens em um palete bidimensional. A estratégia proposta é dividida em duas etapas principais. A primeira consiste na minimização de uma função objetivo baseada na definição de um tensor computacional através da aplicação de um algoritmo de computação evolutiva, e a segunda consiste na eliminação de sobreposições através de uma nova abordagem algorítmica proposta. Cada camada tensora do modelo corresponde a um item destinado à colocação no palete. Esta metodologia apresenta benefícios notáveis, incluindo a fácil identificação de itens sobrepostos no palete. As descobertas sugerem que a solução computacional otimiza a utilização do espaço do palete e fornece informações precisas de posicionamento que poderiam ser usadas a posteriori por um sistema robotizado para executar o processo de peletização.

Palavras-chave: Problema de Carregamento de Paletes; Otimização; Tensor computacional; Algoritmo de computação evolutiva; Sobreposição

ABSTRACT

The PLP (Pallet Loading Problem) constitutes a significant focus in the field of engineering, with wide applications in various sectors, such as automated palletizing and robotics. To solve PLP, a variety of algorithms and techniques have been used, including heuristics, meta-heuristics, mathematical programming and artificial intelligence. Palletizing, in general terms, refers to the efficient loading and arranging of products on pallets for transportation or storage purposes. However, this represents a multifaceted challenge, given the wide range of factors that must be meticulously evaluated to achieve an optimal solution. Parameter selection contributes to numerous variations of PLP, being one of them the 2D-PLP (Two Dimensional Pallet Loading Problem). To solve this problem, this study proposes a new methodology developed to optimize the distribution of items on a two-dimensional pallet. The proposed strategy is divided into two main steps. The first consists of minimizing an objective function based on the definition of a computational tensor through the application of an evolutionary computing algorithm, and the second consists of eliminating overlaps through a new proposed algorithmic approach. Each tensor layer of the model corresponds to an item intended for placement on the pallet. This methodology has notable benefits, including easy identification of overlapped items on the pallet. The findings suggest that the computational solution optimizes pallet space utilization and provides accurate positioning information that could be used posteriorly by a robotic system to perform the palletizing process.

Keywords: Pallet Loading Problem; Optimization; Computational Tensor; Evolutionary Computing Algorithm; Overlap

LISTA DE FIGURAS

Figura 1 – (a) Enchendo uma caixa (b) Dimensões dos potes e caixas (GRANEMANN, 2020).	18
Figura 2 – (a) Palete produzido (b) Soltura da peça na paletização (PEGORETTI, 2021).	18
Figura 3 – (a) Peças paletizadas (b) Ensaio de paletização com peça prototipada (IKENOHUCHI, 2022).	19
Figura 4 – Cortes bidimensionais em chapas (MOURA <i>et al.</i> , 2019).	23
Figura 5 – Colocação do segundo molde e identificação de áreas sobrepostas (ALVES; AYMONE, 2014).	24
Figura 6 – Etapas do corte de um objeto de acordo com um padrão de corte 3-estágios (RANGEL; FIGUEIREDO, 2008).	24
Figura 7 – Espaços residuais em um contêiner carregado (SHENG <i>et al.</i> , 2016).	25
Figura 8 – Quatro arranjos de paletes em um contêiner (SHENG <i>et al.</i> , 2016).	25
Figura 9 – Padrões: (a) guilhotinado, (b) não-guilhotinado de 1 ^a ordem, e (c) não-guilhotinado de ordem superior (MORALES; MORABITO, 1997).	28
Figura 10 – Representação das peças retangulares em uma placa retangular (VIANNA; POLDI, 2005).	29
Figura 11 – Divisão do palete em cinco blocos, com orientação das caixas em cada bloco (MORALES; MORABITO, 1997).	29
Figura 12 – Heurísticas de bloco não guilhotina de segunda ordem (ARUN PRASAD; KRISHNAKUMAR, 2021).	30
Figura 13 – Classes de heurísticas BL: (a) Rotina de BL, (b) Rotina de BLF (HOPPER; TURTON, 2001).	33
Figura 14 – Melhores layouts para GA + BLF (esquerda) e GA + BLF (direita) (HOPPER; TURTON, 2001).	34
Figura 15 – Classificação de alguns métodos de Otimização, adaptado de (OLIVEIRA, 2006).	35
Figura 16 – Representação de mínimos local e global e máximos local e global (GOMES <i>et al.</i> , 2017).	36
Figura 17 – Fluxograma da evolução diferencial adaptado de (RINCON, 2012).	38
Figura 18 – Operação de mutação (PACHECO, 2015).	41
Figura 19 – Operação de cruzamento (PACHECO, 2015).	43
Figura 20 – Tensor de terceira ordem aplicado em uma situação real (KROONENBERG, 2008).	45
Figura 21 – Tensor aplicado ao contexto de RGB (ESTEVEES, 2020).	46
Figura 22 – Representação das camadas de índice 2, 3 e 4 de um tensor: (a) 3D. (b) 2D.	46

Figura 23 – Representação das camadas de índice 1 e 2 de um tensor: (a) 3D. (b) 2D.	47
Figura 24 – Aplicativo principal para execução da Fase 1.	49
Figura 25 – Fluxo de determinação do valor ótimo.	49
Figura 26 – Tela interativa que permite a escolha do conjunto a ser otimizado pelo usuário.	52
Figura 27 – (a) Um tensor 3D vazio construído para agrupar as informações sobre as caixas, (b) Uma caixa alocada em uma das camadas de um tensor.	54
Figura 28 – Posicionamento da caixa virtual.	55
Figura 29 – Restrições para para o posicionamento da caixa virtual: (a) Condições atendidas. (b) Condições não atendidas.	56
Figura 30 – (a) Representação bidimensional da disposição das caixas com sobreposição. (b) Áreas de sobreposição em evidência.	57
Figura 31 – (a) Visualização da Distribuição de Objetos no Tensor. (b) Tensor na sua representação bidimensional, destacando as caixas redimensionadas e suas coordenadas na disposição ideal.	58
Figura 32 – Aplicativo principal para execução da Fase 2 executado pelo terminal evidenciando a escolha do arquivo de entrada.	59
Figura 33 – Visualização da Distribuição de Objetos no Tensor Virtual Expandido: (a) Tensor virtual. (b) Décima primeira camada do tensor.	60
Figura 34 – Visualização do processo de movimentação do ciclo geral 1, após 10 ciclos: (a) Para baixo.. (b) Para a direita. (c) Para cima. (d) Para a esquerda.	61
Figura 35 – Visualização do processo de movimentação do ciclo geral 2, que consiste em 10 ciclos: (a) Para baixo. (b) Para a direita. (c) Para cima. (d) Para a esquerda.	61
Figura 36 – Entradas: (a) Um tensor 3D contendo informações sobre as caixas, (b) As coordenadas da melhor disposição das caixas.	62
Figura 37 – Círculo que contém o “X” representa o local da “Distância inválida”.	63
Figura 38 – Representação das posições das projeções.	64
Figura 39 – Projeções: (a) Realizada, sem nova sobreposição (b) Não realizada, causando sobreposição.	64
Figura 40 – Um ciclo completo de movimentação para baixo.	65
Figura 41 – Estudo de caso 1: 16 caixas do tipo 1.	67
Figura 42 – Estudo de caso 2: 12 caixas - 8 caixas do tipo 1 e 4 caixas do tipo 2 em orientação horizontal.	68
Figura 43 – Estudo de caso 3: 12 caixas - 8 caixas do tipo 1 e 4 caixas do tipo 2 em orientação vertical.	68
Figura 44 – Estudo de caso 4: 12 caixas - 8 do tipo 1, 2 do tipo 2 horizontais e 2 do tipo 2 verticais.	69

Figura 45 – Estudo de caso 5: 11 caixas - 8 do tipo 1, 2 do tipo 2 horizontais e 1 do tipo 3.	69
Figura 46 – Ambiente de Simulação de Robô Industrial.	70

LISTA DE TABELAS

Tabela 1 – Características Escolhidas pelo Usuário	53
Tabela 2 – Matriz de Características de objetos virtuais (MCV) na condição inicial	53
Tabela 3 – Matriz de Características de objetos virtuais (MCV) em função da coluna “Orientação”	55
Tabela 4 – Resumo de Tempo de Resultados.	70

LISTA DE ABREVIATURAS E SIGLAS

2D-PCP	Problema de Carregamento de Paletes Bidimensional
2D-PDC	Problema de Corte Bidimensional
2D-PDE	Problema de Empacotamento Bidimensional
2D-PLP	Two Dimensional Pallet Loading Problem
COBEM	Congresso Internacional de Engenharia Mecânica
DE	Algoritmo de Evolução Diferencial de Storn & Price
MCU	Matriz de Características Escolhidas pelo Usuário
MCV	Matriz de Características dos Objetos Virtuais
PCP	Problema de Carregamento de Paletes
PDC	Problema de Corte
PLP	Pallet Loading Problem
QI	Quociente de Inteligência
RNCs	Redes Neurais Convolucionais
TCC	Trabalho de Conclusão de Curso

SUMÁRIO

1	INTRODUÇÃO	16
1.1	MOTIVAÇÃO E JUSTIFICATIVA	17
1.2	OBJETIVOS	19
1.2.1	Objetivo Geral	19
1.2.2	Objetivos Específicos	19
1.3	ESTRUTURA DO DOCUMENTO	20
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	O PROBLEMA DE CARREGAMENTO DE PALETES (PCP)	21
2.2	PROBLEMAS SEMELHANTES AO 2D-PCP	22
2.3	APLICAÇÕES DOS PROBLEMAS	23
2.3.1	Aplicações para PDC	23
2.3.2	Aplicações para PCP	24
2.4	TRABALHOS SIMILARES AO PCP	26
2.5	PRINCIPAIS TÉCNICAS USADAS RELACIONADAS AOS TRABALHOS SIMILARES	27
2.5.1	Heurística de bloco	27
2.5.1.1	<i>Heurística de bloco aplicada no problema de corte</i>	<i>28</i>
2.5.1.2	<i>Heurística de bloco aplicada no 2D-PCP</i>	<i>30</i>
2.5.2	Meta-heurística	31
2.5.2.1	<i>Classe de heurísticas BL</i>	<i>33</i>
2.6	CLASSIFICAÇÃO DOS PRINCIPAIS MÉTODOS DE OTIMIZAÇÃO	33
2.7	ALGORITMO DE <i>STORN & PRICE</i>	35
2.7.1	Fluxograma da evolução diferencial: conceitos básicos	37
2.7.1.1	<i>População de indivíduos</i>	<i>38</i>
2.7.1.2	<i>Operações do DE: Mutação</i>	<i>40</i>
2.7.1.3	<i>Operações do DE: Cruzamento</i>	<i>41</i>
2.7.1.4	<i>Operações do DE: Seleção</i>	<i>42</i>
2.7.2	Parâmetros computacionais	44
2.8	TENSORES	44
3	METODOLOGIA	48
3.1	FASE 1: POSICIONAMENTO ÓTIMO DAS CAIXAS	48
3.1.1	Função Objetivo Teórica	50
3.1.2	Função objetivo implementada computacionalmente	51
3.1.2.1	<i>Parâmetros de entrada</i>	<i>52</i>
3.1.2.2	<i>Construção do tensor e alocação dos objetos virtuais em cada camada</i>	<i>54</i>
3.1.2.3	<i>Mensuração do Grau de Sobreposição</i>	<i>56</i>
3.1.3	Critério de parada atendido	58

3.2	FASE 2: CORREÇÃO DE SOBREPOSIÇÃO	59
4	RESULTADOS E DISCUSSÕES	67
4.1	RESULTADOS	67
4.1.1	Sistema Robotizado	70
4.2	DISCUSSÕES	70
5	CONCLUSÃO	74
5.0.1	Trabalhos Futuros	77
5.0.2	Publicação realizada	77
	REFERÊNCIAS	78
	ANEXO A – Publicação Realizada	84

1 INTRODUÇÃO

A paletização é um processo fundamental em diversas indústrias, essencial para a otimização da logística e da gestão da cadeia de suprimentos. Desde a sua introdução, a paletização revolucionou a maneira como os produtos são armazenados e transportados, permitindo uma maior eficiência na utilização do espaço e na movimentação de mercadorias. A evolução das técnicas de paletização, especialmente com o advento de novas tecnologias e algoritmos, tem proporcionado avanços significativos na forma como os itens são organizados e empacotados, refletindo diretamente na redução de custos operacionais e no aumento da produtividade.

No contexto industrial contemporâneo, o Problema de carregamento de Paletes (PCP) surge como um desafio crítico, com impacto direto na eficiência operacional e na sustentabilidade das operações logísticas. A capacidade de maximizar a utilização do espaço disponível em um palete não só otimiza os processos de armazenamento e transporte, mas também contribui para uma cadeia de suprimentos mais ágil e responsiva. Diante dessa importância, o desenvolvimento de novas metodologias para resolver o PCP são cruciais para atender às demandas crescentes e complexas do mercado global.

O PCP permanece como uma área crítica de pesquisa em engenharia, com sua influência permeando uma variedade de indústrias (ARUN PRASAD; KRISHNAKUMAR, 2021). Notavelmente, o seu impacto estende-se aos domínios da logística e da gestão da cadeia de abastecimento, onde o sucesso operacional depende da utilização eficiente do espaço e dos recursos (AHN; PARK; YOON, 2015).

Uma simplificação comum do PCP é sua variante bidimensional, que gira em torno do empacotamento de itens retangulares idênticos (caixas) em um palete retangular maior. A rotação de itens é permitida em passos de 90 graus, com a ressalva de que somente pacotes ortogonais são permitidos. O objetivo geral desta simplificação é maximizar o número de itens embalados no palete, evitando qualquer sobreposição (LU; CHA, 2014).

Para resolver este problema, este trabalho propõe uma nova metodologia que visa uma distribuição ótima de itens em um palete bidimensional. Essa metodologia busca minimizar uma função objetivo baseada em uma estrutura tensorial onde cada camada do tensor representa um objeto na interface. A dimensão de cada camada tensora é igual à dimensão do palete, e cada camada corresponde a um item que será colocado no palete. Assume-se que as caixas possuem altura uniforme (esta dimensão não interfere nas definições de posicionamento ótimo) e que as dimensões de comprimento e largura são conhecidas. Do ponto de vista técnico, esta pesquisa busca responder à seguinte pergunta:

“Dadas as dimensões de um palete e um conjunto de caixas retangulares, quais são os posicionamentos precisos para as caixas no palete de forma a evitar sobreposições e minimizar o espaço vago no palete?”

Apesar de o problema ser inicialmente percebido como singular, o método proposto o fragmenta em duas perguntas inter-relacionadas:

1^a, “onde as caixas devem ser posicionadas no palete para otimizar a utilização do espaço?”, e 2^a, “como podemos evitar a sobreposição potencial das caixas colocadas?”

Para minimizar a função objetivo proposta na presente pesquisa e a ser otimizada, foi utilizado o Algoritmo de Evolução Diferencial (DE) de Storn & Price. A abordagem proposta oferece o benefício significativo de permitir a verificação direta da sobreposição de objetos localizados nos paletes, permitindo a rejeição imediata dessas configurações em que sobreposições entre objetos sejam detectadas. Após o desenvolvimento completo da metodologia, acredita-se que a solução computacional melhorará significativamente a utilização do espaço do palete e fornecerá informações precisas de posicionamento que um sistema robotizado poderá usar a posteriori.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

Os desafios relacionados ao carregamento de itens em paletes, que abrangem desde pequenas a grandes caixas, até itens avulsos com formas e tamanhos variados, estão cada vez mais associados à robotização, uma realidade presente nas grandes empresas contemporâneas, conforme demonstram pesquisas tecnológicas recentes. A automatização dos procedimentos de carregamento trouxe uma série de vantagens, como o aumento da eficiência e a redução de erros humanos. No entanto, as soluções implementadas apresentam particularidades que frequentemente são abordadas de maneira específica, embora possam ser generalizadas para diversas aplicações.

Pesquisas recentes desenvolvidas no campus Blumenau da Universidade Federal de Santa Catarina, destacadas em trabalhos como os de Granemann (2020), Pegoretti (2021) e Ikenohuchi (2022), evidenciam que a integração de automação com flexibilidade e adaptabilidade continua a ser um desafio significativo. As empresas buscam constantemente aprimorar seus processos de paletização e carregamento robotizado e autônomo para atender de maneira eficiente à crescente diversidade de itens e às exigências do mercado.

A análise realizada por Granemann (2020) oferece uma exposição detalhada dos principais resultados alcançados e dos procedimentos executados no âmbito do projeto e implementação de uma célula de trabalho robotizada para o encaixotamento de produtos lácteos, conforme uma receita de produção previamente estabelecida. Este estudo exemplifica uma demanda específica atendida pela expertise da Torfresma Industrial, empresa que se destaca nos setores de agroindústria, robótica e automação industrial. A Torfresma projeta e fabrica uma variedade de equipamentos, incluindo sistemas para o processamento de carne, embalagem e selagem, encaixotamento, logística de caixas e paletização. Para uma representação visual do projeto, veja a Figura 1.

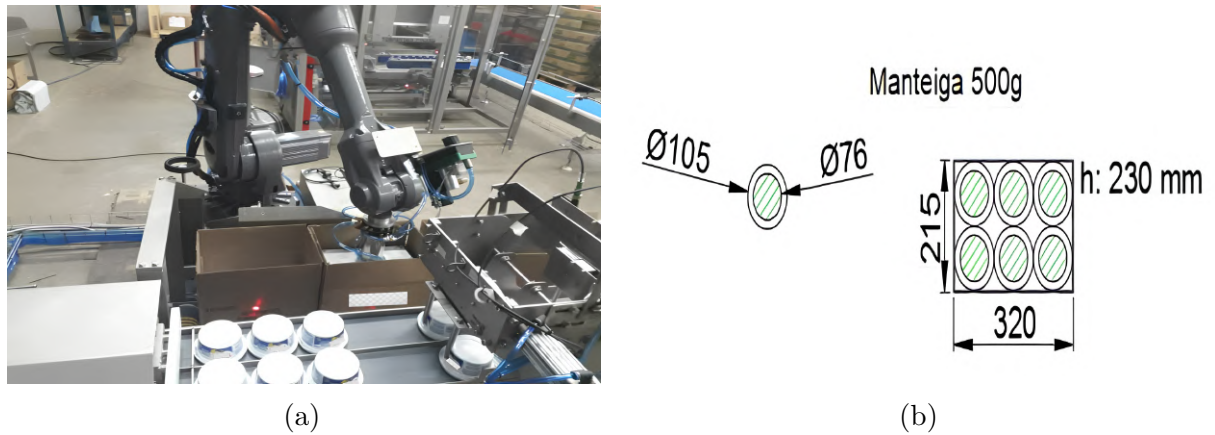


Figura 1 – (a) Enchendo uma caixa (b) Dimensões dos potes e caixas (GRANEMANN, 2020).

Outra iniciativa de pesquisa tecnológica relevante foi conduzida por Pegoretti (2021), abordando questões semelhantes às do projeto anterior, mas originada de uma demanda interna da empresa brasileira WEG S.A. O foco foi o desenvolvimento de uma célula robotizada para a manipulação e peletização de carcaças de motores elétricos, um esforço inovador realizado pela seção de Robotização de Processos Industriais da empresa. A Figura 2 oferece uma representação visual do processo, ilustrando o impacto e a aplicabilidade do projeto conforme descrito por Pegoretti (2021).



Figura 2 – (a) Palete produzido (b) Soltura da peça na paletização (PEGORETTI, 2021).

Uma abordagem alternativa para programar e simular um robô antropomórfico destinado à paletização de rotores estampados fabricados pela WEG S/A foi proposta por Ikenohuchi (2022). Esta iniciativa surgiu em resposta à crescente necessidade de otimização de processos logísticos, visando a economia de recursos em um ambiente de intensa competição no mercado e rápida evolução tecnológica. O trabalho proposto por Ikenohuchi aborda a concepção e simulação de um robô antropomórfico para a paletização de rotores auto-empacotados da WEG S/A. Ele inclui o desenvolvimento de um algoritmo para planejar o posicionamento e orientação de cada pacote, garantindo dinamismo e evitando colisões. O algoritmo foi testado no software KUKA.Sim e validado com peças

protótipo, demonstrando conformidade com os requisitos estabelecidos e oferecendo uma base para futuras aplicações. A Figura 3 ilustra os resultados obtidos pelo autor.



Figura 3 – (a) Peças paletizadas (b) Ensaio de paletização com peça prototipada (IKENOHUCHI, 2022).

Nos estudos mencionados anteriormente, destacam-se os benefícios da implementação de sistemas robóticos e os desafios enfrentados na adaptação de suas soluções às restrições físicas, como a prevenção de colisões. O principal objetivo em todos os casos é automatizar o processo de paletização, substituindo tarefas manuais e repetitivas pela utilização de robôs industriais. Além disso, a necessidade de desenvolver algoritmos para decidir como e onde alocar os itens é uma característica comum.

Esses estudos propuseram sistemas robóticos de paletização para diferentes tipos de produtos, como laticínios, rotores e carcaças de motores, destacando que a disposição dos itens em um palete é uma questão crucial para empresas de grande porte. Contudo, há uma lacuna na existência de uma abordagem programática genérica e flexível para resolver esse desafio, o que motivou o desenvolvimento da presente pesquisa.

1.2 OBJETIVOS

Nesta seção são descritos os objetivos geral e específicos estruturados para a execução do presente Trabalho de Conclusão de Curso.

1.2.1 Objetivo Geral

Desenvolver uma metodologia para a resolução ótima do problema de carregamento de paletes em um plano bidimensional 2D-PCP.

1.2.2 Objetivos Específicos

- Caracterizar um sistema 2D-PCP através do uso de variáveis fundamentais para a descrição do sistema de paletes.

- Determinar uma função objetivo para a maximização do uso do espaço do palete dado um conjunto de caixas conhecidas e caracterizadas.
- Resolver a função objetivo através do uso de um algoritmo computacional de evolução diferencial.
- Integrar os algoritmos de evolução diferencial com algoritmos adicionais desenvolvidos para evitar superposição.
- Validar o método proposto apresentando gráficos e simulações de alguns casos de estudo mostrando o processo de resolução do 2D-PCP.

1.3 ESTRUTURA DO DOCUMENTO

Este documento está organizado em seis capítulos que apresentam a base teórica, os materiais e métodos utilizados, bem como os principais resultados e conclusões obtidos.

O Capítulo 1 oferece uma breve introdução e caracterização do problema, motivação, justificativa, além dos objetivos geral e específicos do projeto.

O Capítulo 2 realiza uma revisão teórica das principais abordagens encontradas na literatura para a resolução de problemas do PCP. Este capítulo serve como uma fonte valiosa de insights para o desenvolvimento do algoritmo proposto e apresenta as principais ferramentas conceituais e de análise essenciais para o entendimento dos capítulos subsequentes.

No Capítulo 3, a metodologia proposta para a solução ótima do problema 2D-PCP é detalhada pela autora.

O Capítulo 4 apresenta os principais resultados obtidos, discussões, descrição dos casos de estudo abordados no projeto, bem como as dificuldades encontradas e os resultados alcançados.

Por fim, o Capítulo 5 apresenta as principais conclusões e perspectivas para trabalhos futuros.

Antecipadamente, conclui-se que os resultados foram satisfatórios, indicando que a solução computacional otimiza a utilização do espaço do palete e fornece informações precisas de posicionamento, as quais poderiam ser utilizadas por um sistema robotizado para executar processos de peletização.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo estabelece os fundamentos teóricos essenciais que sustentam este trabalho, proporcionando uma base conceitual sólida para a compreensão do tema em estudo. Para alcançar esse objetivo, é realizada uma revisão abrangente da literatura, oferecendo uma análise detalhada e elucidativa das principais abordagens, teorias e metodologias relevantes. Este capítulo não apenas contextualiza o problema dentro do estado da arte, mas também identifica lacunas e oportunidades para contribuições inovadoras, que serão exploradas nos capítulos subsequentes.

2.1 O PROBLEMA DE CARREGAMENTO DE PALETES (PCP)

O problema central abordado neste estudo pode ser concisamente descrito utilizando o termo PCP (Problema de Carregamento de Paletes), também conhecido em inglês como PLP (Pallet Loading Problem). Ao longo dos anos, esse desafio tem sido objeto de estudo por uma variedade de pesquisadores, cada um oferecendo suas próprias definições e métodos para entender e resolver esses problemas complexos. No contexto do **PCP**, diversas perspectivas importantes emergiram, cada uma destacando diferentes aspectos e abordagens para otimizar o carregamento de paletes. Algumas das principais variações do PCP são apresentadas na sequência:

- **Encontrar Arranjos Ótimos:** O desafio bidimensional (2D-PCP) de carregamento de paletes envolve descobrir os melhores arranjos para caixas retangulares idênticas em um palete retangular. Este aspecto tem sido um ponto focal de interesse para profissionais de pesquisa operacional há muitos anos, ressaltando a complexidade e relevância prática do problema (DOWSLAND, 1987).
- **Alocação de Itens em um Objeto Único:** Esta abordagem do 2D-PCP foca na alocação de um conjunto predeterminado de itens dentro de um único recipiente, enfatizando a importância de estratégias eficientes de disposição (HOPPER; TURTON, 2001).
- **Simplificação Bidimensional Comum:** Comumente, o problema é reduzido a uma questão bidimensional, onde caixas retangulares menores devem ser acomodadas em um palete retangular maior. A complexidade aumenta com a permissão para rotação dos itens em 90 graus, mas limitando-se a arranjos ortogonais (LU; CHA, 2014).

- **Extensão Tridimensional e Problema de Empacotamento:** Expandindo a problemática, considera-se também o empacotamento tridimensional eficiente em paletes, o que adiciona uma dimensão extra ao desafio. Esta abordagem é reconhecida como uma variação do “problema de empacotamento em um contêiner tridimensional”, elevando as questões de espaço e organização a um nível ainda mais complexo (GZARA; ELHEDHLI; YILDIZ, 2020).
- **Objetivo de Minimização de Desperdício:** O principal objetivo do 2D-PCP é otimizar a disposição dos tipos de caixas fornecidas para maximizar o uso do espaço disponível no palete e, conseqüentemente, minimizar o desperdício de área. Esse foco em eficiência reflete a demanda contínua por soluções sustentáveis e economicamente viáveis (ARUN PRASAD; KRISHNAKUMAR, 2021).

O cerne do problema reside na organização de itens, que podem ser similares ou distintos, em um palete. Geralmente, tanto os objetos quanto o próprio palete possuem forma retangular. Uma simplificação comum é adotar caixas de dimensões idênticas, todas retangulares, e considerar apenas duas dimensões. Além disso, os objetos podem ser rotacionados em incrementos de 90 graus. O objetivo fundamental é minimizar a área de desperdício no palete, tornando-o mais eficiente em termos de espaço utilizado.

2.2 PROBLEMAS SEMELHANTES AO 2D-PCP

O problema central deste trabalho, o 2D-PCP, também conhecido em algumas literaturas como 2D-PDE (Problema de Empacotamento Bidimensional), historicamente transcende o simples carregamento de paletes, ramificando-se em uma variedade de situações que permitem sua adaptação para resolver problemas semelhantes.

O PCP e o PDC (Problema de Corte) compartilham fundamentos baseados na analogia entre empacotar unidades menores sobre unidades maiores, como caixas sobre paletes, e cortar unidades maiores para produzir unidades menores, como chapas grandes em chapas menores. Essa similaridade, inicialmente subestimada na literatura, começou a ser reconhecida no final da década de 80, quando poucos autores haviam destacado e explorado essa conexão (MORALES; MORABITO, 1997).

Apesar dessas semelhanças fundamentais, existem distinções importantes entre os problemas de corte e empacotamento. Por exemplo, o problema de corte abrange considerações adicionais como a densidade do material, a direção do corte e a possibilidade de deformação do material, aspectos que não são primariamente preocupações no problema de paletização.

A partir do influente trabalho de Dyckhoff (1990), os problemas de corte e empacotamento passaram a ser vistos como pertencentes à mesma classe de desafios, com

uma estrutura lógica semelhante a outros problemas estudados em pesquisa operacional e gestão da produção. Essa abordagem unificada facilitou a aplicação de técnicas e metodologias desenvolvidas para um tipo de problema à resolução do outro, gerando avanços significativos na compreensão e solução de ambos os desafios (MORALES; MORABITO, 1997).

Por esse motivo, ao estudar o PCP, foram consultadas diversas referências que abordam soluções propostas para problemas de corte dimensional e outros problemas semelhantes, enriquecendo assim o entendimento sobre ambas as áreas.

2.3 APLICAÇÕES DOS PROBLEMAS

O PDC e O 2D-PCP são encontrados em muitas indústrias, sendo que diferentes setores incorporam diferentes restrições e objetivos. As indústrias de madeira, vidro e papel estão principalmente envolvidas no corte de figuras regulares, enquanto na construção naval, têxtil e indústria de couro, itens de formas irregulares e arbitrárias precisam ser empacotados (HOPPER; TURTON, 2001).

2.3.1 Aplicações para PDC

O 2D-PDC surge como um problema industrial, onde vários itens retangulares precisam ser cortados de um rolo de material. Como a ordem (conjunto de itens) é pequena em comparação com o material de estoque, a altura do objeto pode ser considerada como infinita. (HOPPER; TURTON, 2001). Dentre os casos de uso, pode-se citar:

Corte a Laser e CNC: Neste caso de uso o um algoritmo eficaz e eficiente auxilia na otimização do uso de material, minimizando o desperdício e maximizando a eficiência do corte. Veja a Figura 4 que ilustra cortes bidimensionais em chapas (MOURA *et al.*, 2019).

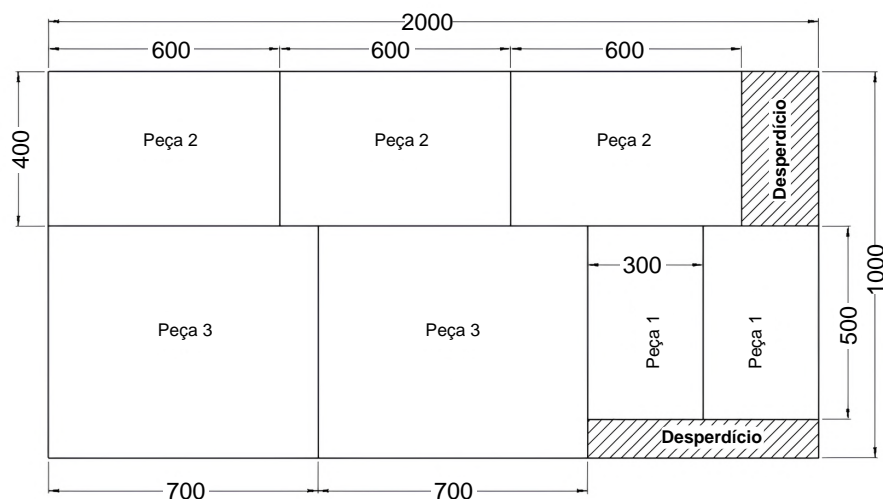


Figura 4 – Cortes bidimensionais em chapas (MOURA *et al.*, 2019).

Corte de Tecidos para Confecção: Na indústria têxtil, o algoritmo pode ser empregado para otimizar o corte de tecidos, minimizando os resíduos e melhorando a eficiência da produção de roupas e outros itens têxteis (ALVES; AYMONE, 2014). Em impressão de materiais como tecidos, papel e embalagens, o algoritmo pode ser empregado para otimizar o layout de impressão, garantindo o uso eficiente do material e minimizando o desperdício. Na Figura 5, pode-se observar que o problema de corte pode envolver formas irregulares e também exibe a questão de sobreposição como fator de tradução do sistema real para um simulado (ALVES; AYMONE, 2014).

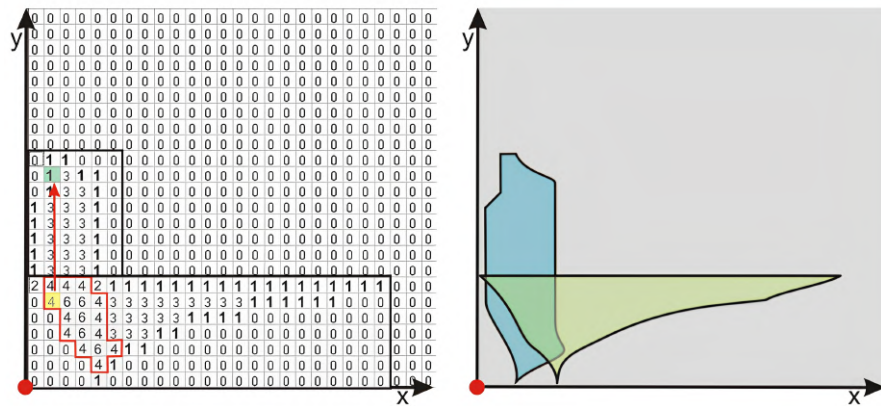


Figura 5 – Colocação do segundo molde e identificação de áreas sobrepostas (ALVES; AYMONE, 2014).

Fabricação de Móveis: Na indústria moveleira, o algoritmo pode ser aplicado para otimizar o corte de peças de madeira, reduzindo o desperdício de material e agilizando o processo de fabricação. A geração de padrões de corte para cortar painéis retangulares de madeira em itens retangulares menores é uma tarefa rotineira em indústrias de móveis. Além do objetivo usual de minimizar perdas, as indústrias procuram gerar padrões de corte que facilitem o processo de corte (RANGEL; FIGUEIREDO, 2008).

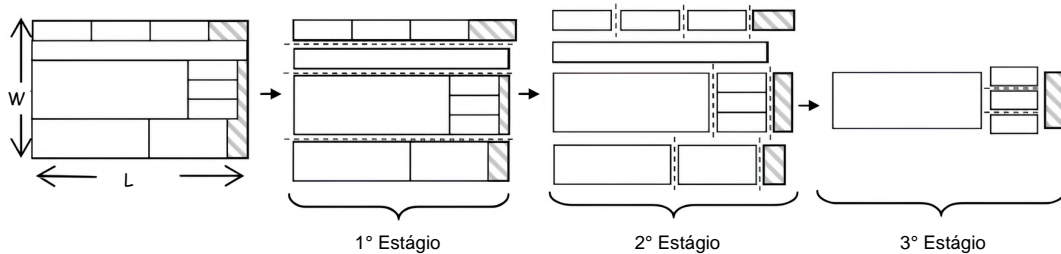


Figura 6 – Etapas do corte de um objeto de acordo com um padrão de corte 3-estágios (RANGEL; FIGUEIREDO, 2008).

2.3.2 Aplicações para PCP

Empacotamento de Contêineres: Fundamental para empresas de logística e transporte, essa aplicação do PCP busca maximizar a carga transportada por viagem,

reduzindo custos de transporte. As Figuras 7 e 8 ilustram, respectivamente, espaços residuais em um contêiner carregado e quatro arranjos de paletes dentro de um contêiner (SHENG *et al.*, 2016).

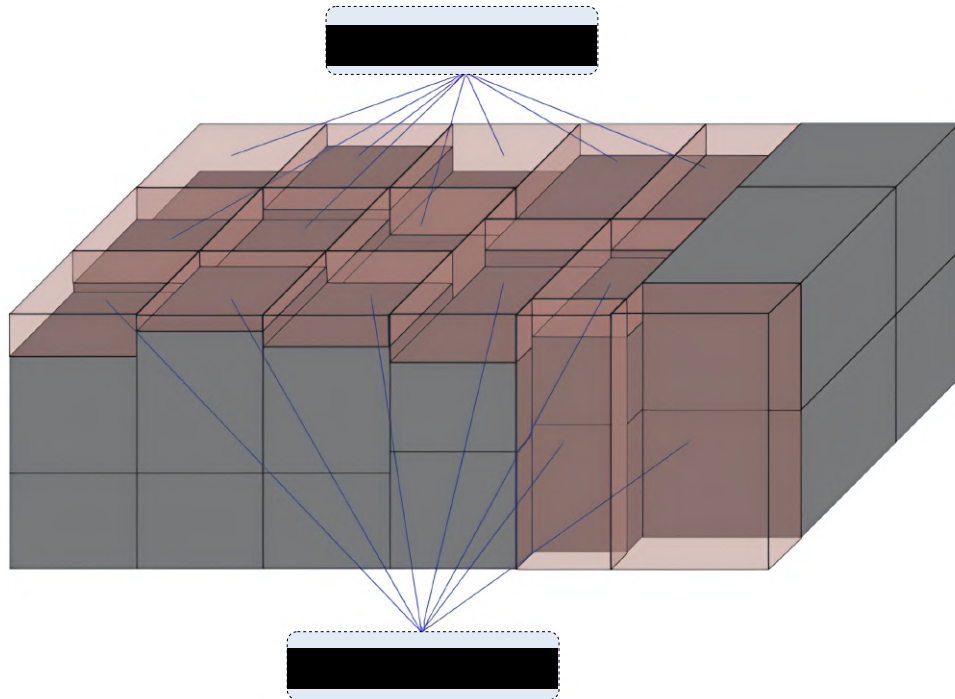


Figura 7 – Espaços residuais em um contêiner carregado (SHENG *et al.*, 2016).

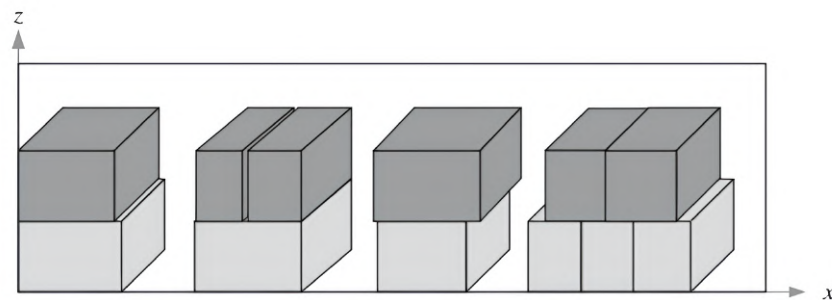


Figura 8 – Quatro arranjos de paletes em um contêiner (SHENG *et al.*, 2016).

O algoritmo de carregamento de paletes 2D (2D-PCP) tem aplicações surpreendentes além das tradicionais, como na Paletização Robótica e no corte de chapas metálicas, provando sua versatilidade em otimizar processos industriais e de fabricação. Este problema é vital na Engenharia Industrial, onde o objetivo comum é maximizar a utilização de material e minimizar a área desperdiçada. Este foco é crucial para indústrias de produção em massa, onde melhorias incrementais podem resultar em significativas economias de material e redução de custos (HOPPER; TURTON, 2001).

2.4 TRABALHOS SIMILARES AO PCP

Historicamente, o PCP tem sido o ponto focal de numerosos estudos, produzindo uma série de métodos concebidos para enfrentar este desafio. Vários procedimentos foram desenvolvidos para obter a disposição mais eficaz das caixas sobre um palete. O objetivo da presente seção é fazer menção às principais técnicas utilizadas na literatura para a solução do problema do PCP.

Tais estratégias abrangem métodos como heurísticas de bloco não guilhotina de ordem superior (ARUN PRASAD; KRISHNAKUMAR, 2021), abordagens meta-heurísticas (HOPPER; TURTON, 2001) e até mesmo algoritmos de aprendizagem por reforço híbrido (FANG *et al.*, 2023). Ao utilizar essas técnicas, numerosos estudos abordaram problemas de carregamento e embalagem de paletes, com foco na utilização e otimização eficiente do espaço. Eles oferecem informações valiosas aplicáveis ao 2D-PCP.

Por exemplo, Subir Bhattacharya, Roy e Sumita Bhattacharya (1998) exploraram o carregamento tridimensional de paletes com restrições práticas, usando técnicas de agrupamento e classificação. Ribeiro e Lorena (2007) aplicaram relaxação Lagrangeana e geração de colunas para resolver um problema semelhante. Os autores Subir Bhattacharya, Roy e Sumita Bhattacharya (1998) investigaram o PCP usando um algoritmo de busca em profundidade exata, introduziram novos conceitos de melhoria de eficiência, enquanto Nelien (1995) propôs um método de limite superior baseado em restrições estruturais. A otimização do carregamento de paletes e da utilização do espaço dentro da estrutura de múltiplos contêineres e atendimento de pedidos do cliente foi abordada por Gimenez-Palacios *et al.* (2023). Morabito, Morales e Widmer (2000) focaram na maximização da eficiência de distribuição no carregamento de caminhões. O carregamento de itens irregulares e não homogêneos em paletes tem sido uma área desafiadora tratada por meio de abordagens heurísticas e algoritmos exatos (BISCHOFF; JANETZ; RATCLIFF, 1995), (MASCARENHAS, 2005).

Além disso, vários estudos têm como objetivo otimizar os padrões de carregamento de paletes para maximizar os itens embalados e, ao mesmo tempo, minimizar as operações de transferência, aplicando uma série de técnicas, incluindo heurísticas e modelos baseados em teoria de grafos (DOWSLAND, 1990). Dyckhoff (1990) forneceu uma tipologia abrangente para problemas de corte e empacotamento. Outros estudos mostraram a relevância do empacotamento retangular e dos problemas de posicionamento irregular no carregamento de paletes. Estes empregaram métodos heurísticos e de rasterização para minimizar a sobreposição e garantir a contenção eficiente de itens irregulares, conforme o realizado por Scheithauer e Sommerweib (1998), Terno *et al.* (2000) e Martinez-Sykora *et al.* (2017). Por último, Os autores Alonso *et al.* (2017) e Sheng *et al.* (2016) exploram o carregamento de vários contêineres e o atendimento aos pedidos dos clientes com uso mínimo de caminhões e o enchimento eficiente de contêineres com caixas de produtos.

2.5 PRINCIPAIS TÉCNICAS USADAS RELACIONADAS AOS TRABALHOS SIMILARES

2.5.1 Heurística de bloco

A palavra heurística é derivada da palavra grega *heuriskein*, que significa descobrir. Atualmente esse termo é usado para descrever “um método que, baseado na experiência ou julgamento” parece conduzir a uma boa solução de um problema, mas que não garante produzir uma solução ótima. Também pode ser considerado como sendo um conhecimento circunstancial, não verificável, nem matematicamente verificável (FOULDS, 1984). Em termos mais simples, pode ser entendida como um método de busca por uma solução eficaz através de abordagens diretas, como força bruta.

Antes da exposição dos estudos que ilustram a aplicação da técnica da heurística de bloco e suas derivações, é apresentada uma definição dos termos encontrados na literatura. Inicialmente, essa definição é introduzida para o problema de corte, contudo sua aplicabilidade é ampliada para várias classes de problemas pertencentes ao domínio do PCP: Um corte é denominado “guilhotinado” quando, ao ser aplicado em um retângulo, resulta em dois novos retângulos, sendo o corte realizado exclusivamente na horizontal ou na vertical. Um padrão de corte é considerado guilhotinado quando é obtido através de cortes guilhotinados sucessivos (VIANNA; POLDI, 2005). As definições utilizadas para classificar os padrões produzidos nos problemas de corte são as seguintes:

- **Padrão guilhotinado:** Um corte é considerado guilhotinado se, ao ser realizado em um retângulo, produzir exatamente dois novos retângulos. Um padrão formado por cortes sucessivos guilhotinados é chamado de padrão guilhotinado (Figura 9a) (MORALES; MORABITO, 1997).
- **Padrão não-guilhotinado de 1^a ordem:** Um corte é chamado de não-guilhotinado de 1^a ordem se, ao ser efetuado em um retângulo, gerar cinco novos retângulos dispostos de forma a não formar um padrão guilhotinado. Um padrão formado por cortes sucessivos guilhotinados e/ou cortes não-guilhotinados de 1^a ordem é denominado padrão não-guilhotinado de 1^a ordem (Figura 9b). É importante observar que todo padrão guilhotinado também é um padrão não-guilhotinado de 1^a ordem, mas o contrário não é verdadeiro (MORALES; MORABITO, 1997).
- **Padrão não-guilhotinado de ordem superior:** Trata-se de um padrão que não pode ser obtido por meio de cortes sucessivos guilhotinados e/ou cortes não-guilhotinados de 1^a ordem. Veja a Figura 9c (MORALES; MORABITO,

1997).

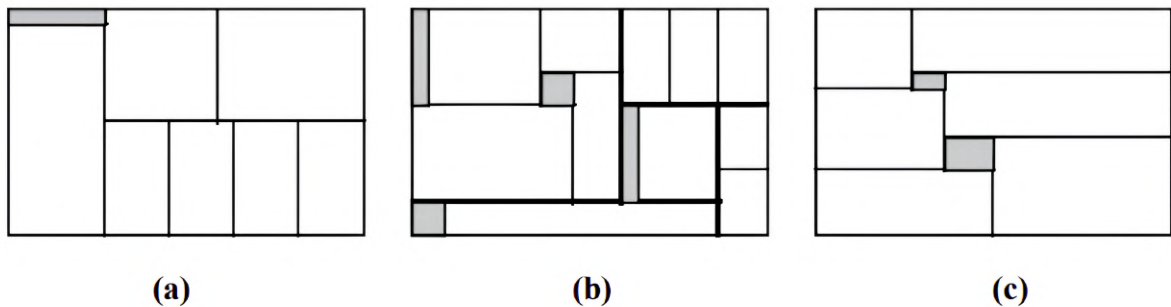


Figura 9 – Padrões: (a) guilhotinado, (b) não-guilhotinado de 1^a ordem, e (c) não-guilhotinado de ordem superior (MORALES; MORABITO, 1997).

A heurística de bloco é uma abordagem comumente utilizada para resolver problemas de otimização relacionados ao corte e ao carregamento de paletes. No contexto do problema de corte, a heurística de bloco é aplicada para otimizar o uso de material cortável, como chapas de madeira, metal, vidro, entre outros. O objetivo é cortar esses materiais em pedaços menores de maneira eficiente, minimizando o desperdício.

No problema de carregamento de paletes, a heurística de bloco é empregada para organizar eficientemente os itens em paletes, maximizando a utilização do espaço disponível e minimizando o número de paletes necessários para transportar os itens.

Em ambos os casos, a heurística de bloco funciona tentando agrupar os itens de uma maneira que maximize a eficiência do espaço, seguindo uma série de regras ou critérios para determinar como os itens devem ser organizados ou cortados. Essa abordagem é especialmente útil em situações onde encontrar a solução ótima é computacionalmente custoso ou impraticável, permitindo a obtenção de soluções razoáveis em um tempo aceitável.

2.5.1.1 Heurística de bloco aplicada no problema de corte

Considere agora que há disponíveis K tipos de placas com dimensões (L_k, W_k) , onde L_k representa o comprimento e W_k a largura da placa k ($k = 1, \dots, K$), em quantidades e_k , $k = 1, \dots, K$. Além disso, há um conjunto de pedidos com demandas conhecidas d_i , $i = 1, \dots, m$, para itens com dimensões (l_i, w_i) , onde l_i é o comprimento e w_i a largura da peça i , $i = 1, \dots, m$ (ver Figura 10).

O desafio consiste em satisfazer os pedidos cortando as placas em estoque, de forma a atender à demanda e minimizar as perdas de material. Este problema é conhecido na literatura como o Problema de Corte de Estoque Bidimensional (VIANNA; POLDI, 2005).

Um exemplo de uma heurística de bloco é o método proposto por Bischoff, Janetz e Ratcliff (1995). O palete (L, W) é inicialmente dividido em até 5 blocos, denominados

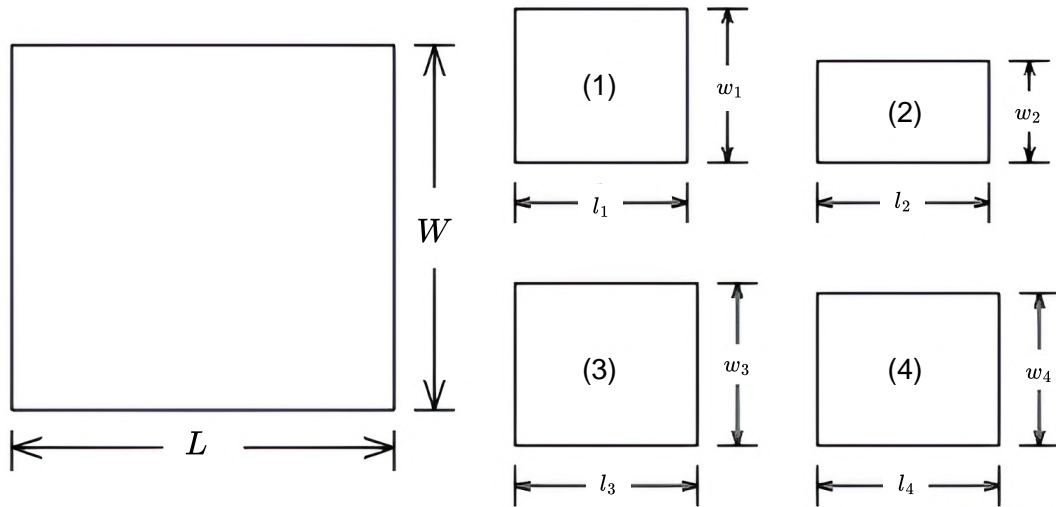


Figura 10 – Representação das peças retangulares em uma placa retangular (VIANNA; POLDI, 2005).

blocos 1, 2, 3, 4 e 5, respectivamente, com dimensões (L_i, W_i) , $i = 1, \dots, 5$, conforme a Figura 11. Uma orientação é prefixada para arranjar as caixas dentro de cada bloco (veja indicação na Figura), com exceção do bloco 3 (central), cuja orientação é pós-fixada (MORALES; MORABITO, 1997).

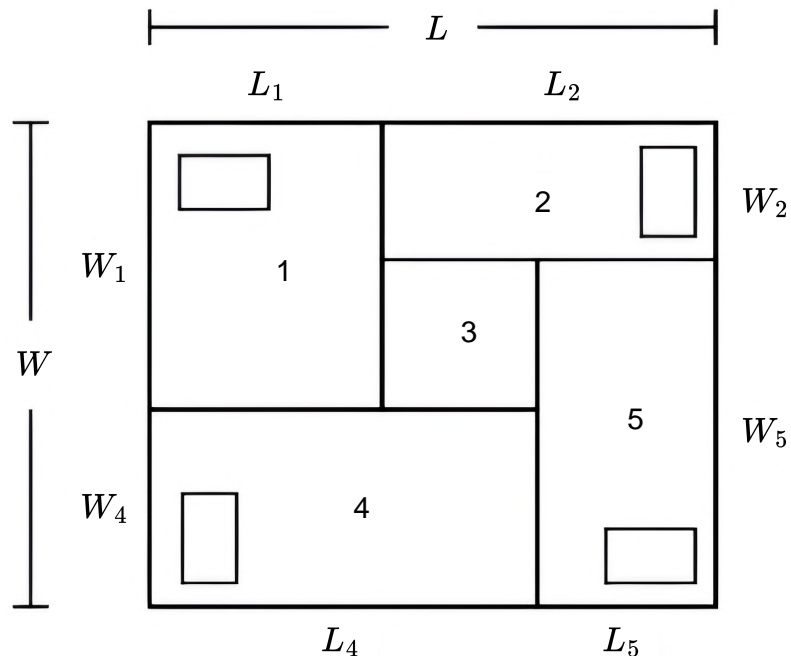


Figura 11 – Divisão do palete em cinco blocos, com orientação das caixas em cada bloco (MORALES; MORABITO, 1997).

Basicamente, o método examina todas as possíveis modificações nos blocos externos 1, 2, 4 e 5, variando suas dimensões (L_i, W_i) (inclusive a possibilidade das dimensões serem

nulas), e retorna o padrão que resultar no maior número de caixas no palete (MORALES; MORABITO, 1997).

2.5.1.2 Heurística de bloco aplicada no 2D-PCP

O método fundamental nas Heurísticas de Bloco consiste em dividir a área do palete em regiões menores denominadas Blocos e organizar as caixas de entrada fornecidas dentro desses blocos de modo a minimizar a área desperdiçada (ARUN PRASAD; KRISHNAKUMAR, 2021).

Há exemplos de estudos que utilizaram heurísticas de bloco não guilhotina de ordem superior. Veja na Figura 12 a característica física da modelagem de heurísticas de bloco de segunda ordem não guilhotinados. Para a divisão de blocos de segunda ordem não guilhotinados, obtém-se cinco blocos ao longo das bordas e um no centro, formando um polígono de seis lados. Sem repetir soluções simétricas ou duplicadas, pode-se dividir os blocos em seis cortes diferentes. Os blocos são rotulados como A, B, C, D e E, e suas dimensões são representadas por a_1 , a_2 , b_1 , b_2 , c_1 , c_2 , d_1 , d_2 , e_1 e e_2 (ARUN PRASAD; KRISHNAKUMAR, 2021).

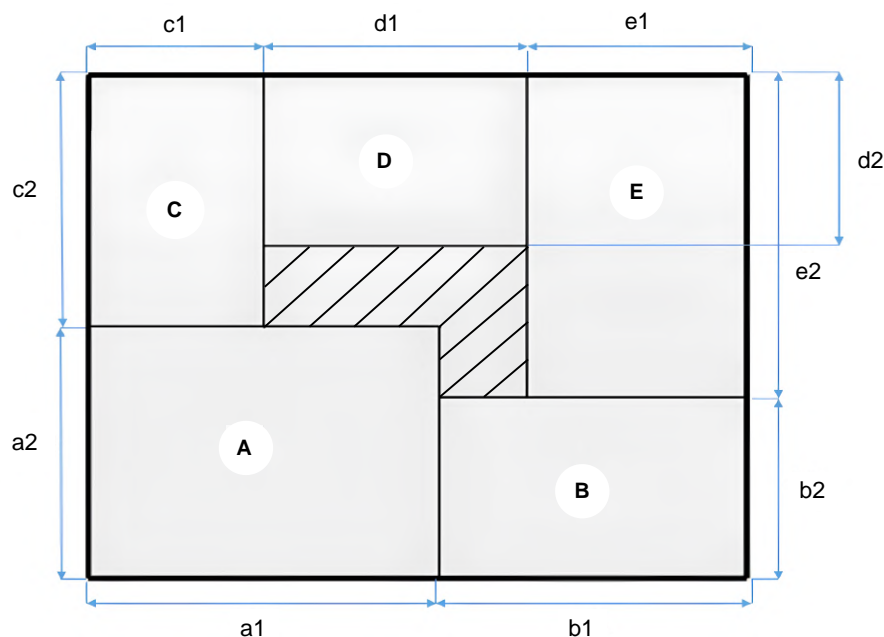


Figura 12 – Heurísticas de bloco não guilhotina de segunda ordem (ARUN PRASAD; KRISHNAKUMAR, 2021).

Fisicamente, o que se altera para o caso de cortes de primeira ordem é que o polígono central possui quatro lados, enquanto os polígonos ao redor também são quatro, conforme a heurística de primeira ordem.

No estudo conduzido por Arun Prasad e Krishnakumar (2021), foram testadas, comparadas e evidenciadas as vantagens das heurísticas de bloco de primeira e segunda

ordem. Em síntese, algumas considerações foram alcançadas:

- Usar caixas de diferentes tipos dentro de um bloco único pode resultar em arranjos de padrões com mais de quatro tipos de caixas diferentes.
- Não foram encontradas novas soluções de segunda ordem em comparação com a primeira ordem ao lidar com entradas de duas caixas, mas houve um aumento no número de arranjos na mesma solução.
- As heurísticas de blocos de primeira ordem são adequadas quando há apenas dois tipos diferentes de caixas, mas para mais de dois tipos, as heurísticas de segunda ordem são preferíveis.
- As heurísticas de segunda ordem são mais eficazes para obter arranjos de caixas de palete com menor desperdício de área, especialmente quando o número de tipos de caixas é maior.
- É recomendável começar com um limite baixo de área de desperdício ao usar heurísticas de segunda ordem e aumentar apenas conforme necessário para encontrar soluções em um tempo de computação mais curto.
- As heurísticas de segunda ordem oferecem uma variedade maior de arranjos de palete para escolher, dependendo das unidades de caixa disponíveis.

2.5.2 Meta-heurística

O termo “meta-heurística” deriva da composição de duas palavras gregas: “heurística”, já explicada no parágrafo anterior, e o prefixo “meta”, que significa “após”, indicando um nível superior de descoberta (BECCENERI *et al.*, 2008).

Uma meta-heurística pode ser vista como uma ferramenta algorítmica geral que pode ser aplicada a diferentes problemas de otimização, com modificações relativamente pequenas para torná-la adaptável a um problema específico (BECCENERI *et al.*, 2008).

O estudo conduzido por Hopper e Turton (2001) integrou de maneira brilhante e inteligente a heurística e a meta-heurística na resolução do PLP. Suas descobertas não apenas oferecem resultados valiosos, mas também contextualizam o trabalho em relação às pesquisas anteriores na área. Ele introduz um novo problema que, até o atual ponto deste estudo, não foi mencionado. Este problema é o problema de empacotamento retangular bidimensional (2D), onde um conjunto fixo de itens precisa ser alocado em um único objeto.

Notavelmente, este problema guarda semelhanças com o PCP, pois suas características são:

- Um conjunto de itens, que podem conter itens idênticos;
- Um único objeto com comprimento e largura fixos e altura infinita;
- Todas as peças são de formato retangular;
- Os itens podem ser rotacionados em 90° ;
- Não-guilhotinável.

Devido às características anteriormente mencionadas, percebe-se que o problema descrito apresenta semelhanças com o desafio abordado no presente estudo, o qual visa o desenvolvimento de uma solução eficaz para o problema em questão. Portanto, é essencial analisar as soluções que foram propostas e implementadas para problemas similares, pois elas podem oferecer *insights* valiosos para a nossa pesquisa.

Na literatura, é comum encontrar pesquisas que adotam abordagens híbridas, onde diferentes algoritmos são combinados para resolver um problema específico. Uma prática popular é a combinação de Algoritmos Genéticos (GA) com rotinas de colocação heurística. Nessa estratégia de dois estágios, o GA desempenha o papel de determinar a sequência na qual os itens devem ser empacotados, enquanto um segundo algoritmo é utilizado para descrever como essa sequência é alocada no objeto. Esta abordagem demonstra a eficácia da combinação de técnicas computacionais para lidar com problemas complexos de otimização, como evidenciado na pesquisa de Hopper e Turton (2001).

O estudo realizado por Hopper e Turton (2001) ressalta pelo menos mais duas abordagens relacionadas ao uso de Algoritmos Genéticos (GAs) na resolução do problema investigado. Em uma dessas categorias de abordagens, o objetivo é incorporar parte das informações de layout na estrutura de dados do GA. No entanto, mesmo com essa incorporação, são necessárias algumas regras adicionais para fixar a posição no layout.

Uma terceira categoria destacada engloba um estudo adicional que propôs uma técnica de codificação 2D para abordar um problema de carregamento de paletes com retângulos idênticos. Nessa técnica, o layout é representado por uma matriz 2D que identifica as posições disponíveis tanto na vertical quanto na horizontal, sendo que a colocação horizontal é priorizada. Embora essa abordagem funcione bem para problemas de pequena escala, para otimizar resultados em problemas de médio porte, foram introduzidos operadores adicionais de reparo e aprimoramento (HOPPER; TURTON, 2001).

2.5.2.1 Classe de heurísticas BL

Existe a classe de heurísticas BL (Bottom-Left), trata-se de um procedimento de empacotamento que preserva a estabilidade BL do layout. O item é alocado em uma posição BL estável se não puder ser movido mais para a esquerda ou para baixo. Uma posição válida é encontrada quando o retângulo colide com o layout parcial em seus lados inferior e esquerdo. Na Figura 13a é mostrado o posicionamento de uma sequência de retângulos, que é descrita pela permutação (2, 6, 4, 3, 0, 1, 5) (HOPPER; TURTON, 2001).

A principal limitação deste método reside na formação de espaços vazios no layout, quando itens maiores bloqueiam o movimento de itens subsequentes. Em contrapartida, sua complexidade temporal é apenas $O(N^2)$, onde N representa o número de itens a serem empacotados. Devido a essa baixa complexidade, essa heurística se mostra vantajosa em uma abordagem híbrida com uma meta-heurística. Isso ocorre porque a rotina de decodificação precisa ser executada sempre que a qualidade de uma solução é avaliada, contribuindo assim para a otimização global (HOPPER; TURTON, 2001). Na Figura 13b está demonstrada a política de posicionamento usando o mesmo exemplo de permutação da Figura 13a.

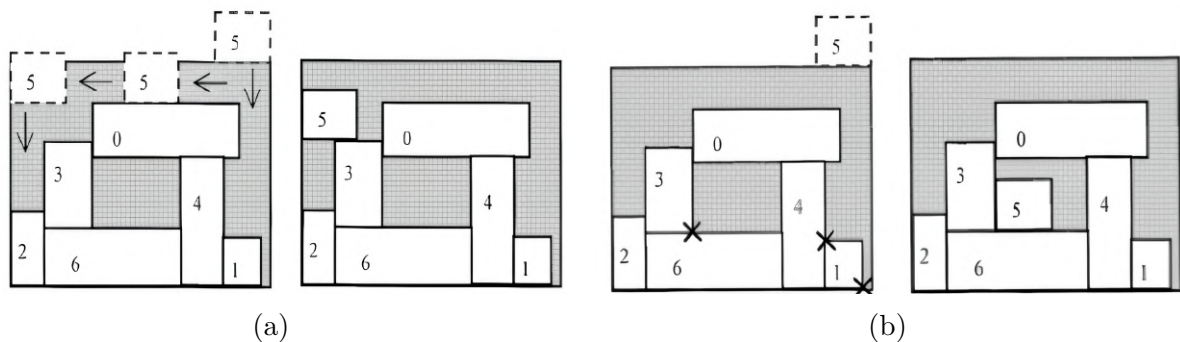


Figura 13 – Classes de heurísticas BL: (a) Rotina de BL, (b) Rotina de BLF (HOPPER; TURTON, 2001).

A diferença é evidente na Figura 14. No recorte à esquerda, é mostrado o resultado final da combinação de uma meta-heurística com o BL, enquanto à direita está o resultado da combinação com o BLF. Isso ilustra a capacidade do BLF em reduzir a presença de lacunas vazias.

2.6 CLASSIFICAÇÃO DOS PRINCIPAIS MÉTODOS DE OTIMIZAÇÃO

Comumente, os métodos de otimização subdividem-se em dois grandes grupos: os Métodos Determinísticos e os Métodos Naturais. Os métodos Determinísticos dividem-se ainda em Diretos e Indiretos. No grupo dos Diretos fazem parte os Métodos Gradientes Conjugados, Simplex e Nelder-Mead, enquanto o grupo dos Indiretos é composto pelos

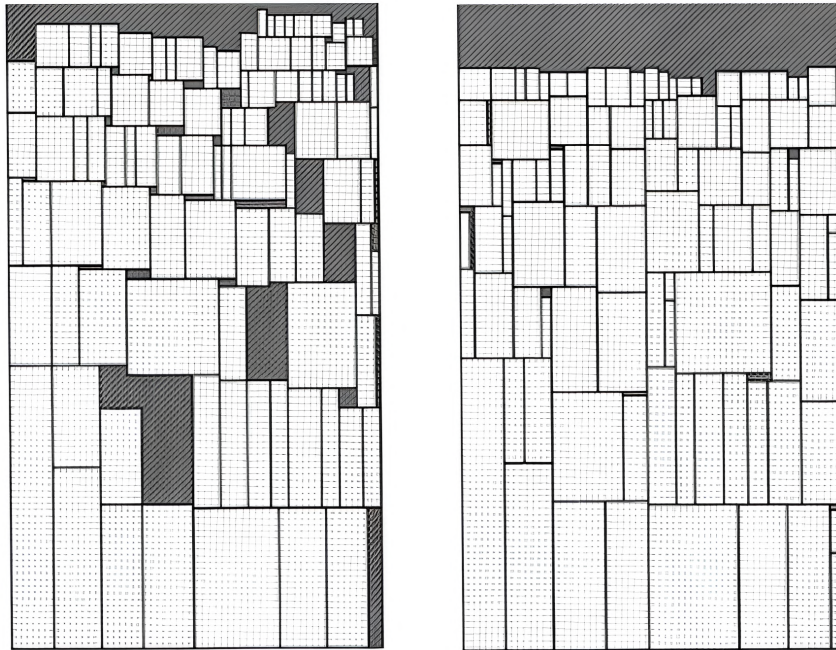


Figura 14 – Melhores layouts para GA + BLF (esquerda) e GA + BLF (direita) (HOPPER; TURTON, 2001).

métodos de Powell, da Máxima Descida, Direções Conjugadas, da Variável Métrica e de Newton. Do grande grupo dos Naturais derivam três grupos principais: os Baseados em população, Busca Tabu e Recozimento Simulado. Existem três derivações principais a partir do subgrupo Baseados em população: Otimização por Exame de Partículas, Sistema de Colônia de Formigas e, por fim, Algoritmos Evolucionários. Dos Algoritmos Evolucionários derivam as classificações em Estratégias de Evolução, Algoritmos Genéticos, Programação Evolutiva, Programação Genética e Evolução Diferencial. Tal classificação está apresentada na Figura 15.

A maioria dos métodos determinísticos se fundamenta no cálculo de derivadas ou em suas estimativas, exigindo dados do vetor gradiente. Isso implica em buscar o ponto onde o gradiente se anula ou em utilizar sua direção como referência (OLIVEIRA, 2006).

Os métodos naturais, incluindo os algoritmos evolucionários (AEs), destacam-se por sua abordagem na busca da solução ideal através do uso de regras de probabilidade, operando de forma “aleatória direcionada”. Esses métodos confiam unicamente nas informações fornecidas pela função de otimização, dispensando a necessidade de dados sobre derivadas ou possíveis descontinuidades (OLIVEIRA, 2006).

Os algoritmos genéticos são abordagens de busca semi-aleatória fundamentadas na teoria da evolução de Charles Darwin. Operando com conjuntos de candidatos, denominados populações, esses métodos são constantemente modificados com base em dois princípios essenciais da evolução natural: seleção e variação. Esses princípios espelham a competição na qual os indivíduos mais adaptados têm maior probabilidade de reprodução e transmissão de seus genes às gerações seguintes, enquanto os menos adaptados tendem

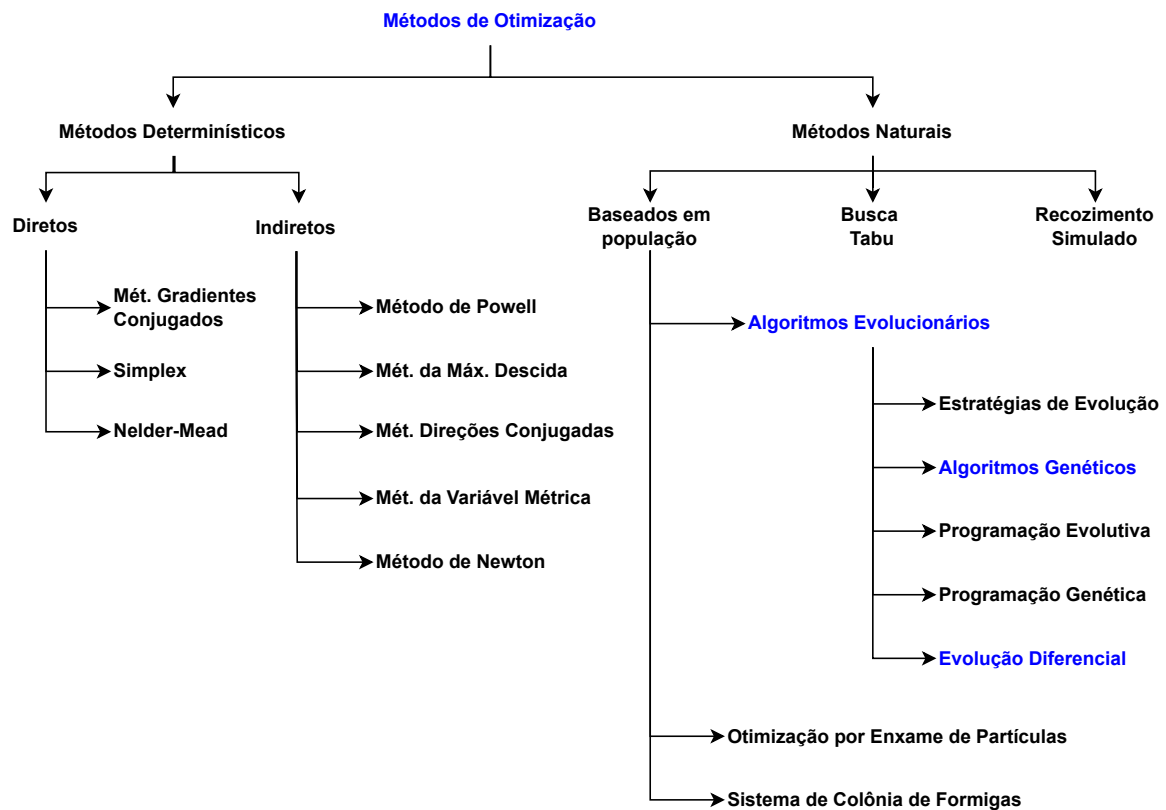


Figura 15 – Classificação de alguns métodos de Otimização, adaptado de (OLIVEIRA, 2006).

a desaparecer da população (DEB, 2001).

A Evolução Diferencial (ED) é um método de otimização que se baseia na geração de novos indivíduos, chamados vetores doadores, a partir da combinação de diferenças entre dois indivíduos aleatórios da população e um terceiro indivíduo. Essa abordagem permite uma rápida convergência para soluções ótimas, além de ser fácil de implementar e robusta. Existem diferentes estratégias dentro da evolução diferencial, que variam na forma como o processo de mutação é realizado. Essas estratégias podem ser ajustadas para lidar com diferentes tipos de problemas de otimização (OLIVEIRA, 2006).

Diferenciação evolutiva é um algoritmo específico dos Algoritmos Evolucionários que foi desenvolvido em 1995 por Rainer Storn e Kenneth Price (*Storn & Price*).

2.7 ALGORITMO DE STORN & PRICE

A compreensão dos métodos heurísticos pode ser ilustrada por meio de uma analogia com o desafio de identificar, empiricamente, o ponto mais alto da Terra. Resolver este problema de forma heurística envolveria iniciar a busca a partir de um ponto na superfície da Terra e explorar montanhas em busca de picos mais altos. Durante esse processo, várias montanhas seriam escaladas e suas alturas comparadas. O ponto mais alto seria progressivamente ajustado conforme novas descobertas surgissem, representando

os chamados “ótimos locais”. Eventualmente, a busca seria interrompida por motivos como a ausência de descobertas significativas por um período prolongado, preocupações com segurança ou restrições de tempo e recursos financeiros. O ponto mais alto seria então determinado, ainda que sem uma validação científica absoluta, reconhecendo que outro pico mais alto pode permanecer oculto (COLIN, 2007). Na Figura 16 está ilustrada a representação de mínimos local e global e máximos local e global.

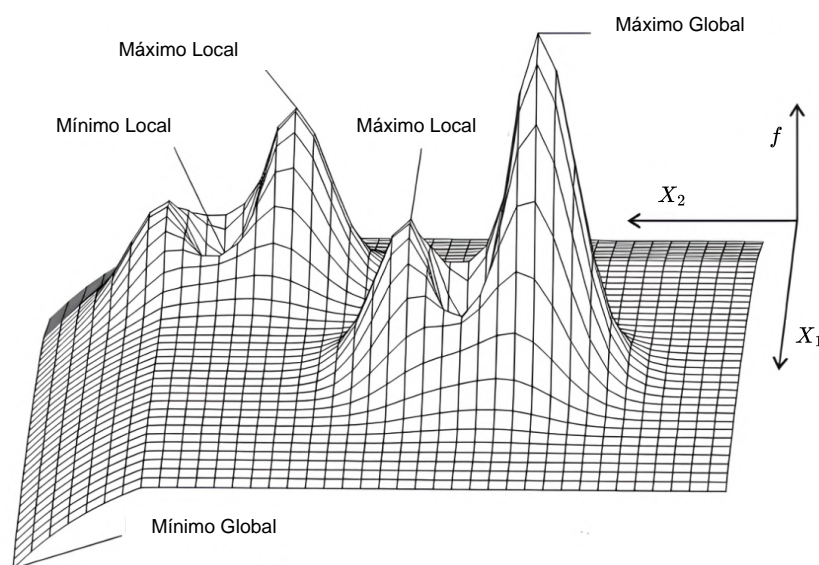


Figura 16 – Representação de mínimos local e global e máximos local e global (GOMES *et al.*, 2017).

Em termos simples, um método eurístico é uma abordagem prática para resolver um problema que pode ou não garantir uma solução perfeita, mas é boa o suficiente para ser útil. Em resumo, métodos heurísticos são como atalhos inteligentes para resolver problemas complexos, priorizando a eficiência prática sobre a garantia de soluções perfeitas.

O algoritmo de Evolução Diferencial (DE), conhecido em inglês como *Differential Evolution*, foi introduzido por Storn & Price em 1997. Este método heurístico é projetado para resolver problemas de otimização contínua sem depender de derivadas. Em vez disso, ele emprega estratégias de busca baseadas na evolução e na diferença entre vetores de solução. Essa abordagem torna o DE particularmente eficaz em uma variedade de problemas de otimização onde a forma da função objetivo (chamada também de função custo ou ainda função de adaptação - *fitness*) pode ser desconhecida ou difícil de modelar matematicamente (STORN, R.; PRICE, 1997).

Desde de sua criação, o DE se apresenta como um simples, mas poderoso algoritmo de otimização numérica para busca da solução ótima global, sendo aplicado com sucesso na solução de vários problemas de otimização difícil. Num problema típico de minimização, encontra-se um ótimo local quando qualquer movimento a ser feito piore o valor atual da função objetivo. Um ótimo global corresponde ao menor valor da função objetivo, entre todos os ótimos locais existentes no espaço de busca (STORN, R.; PRICE, 1997).

O DE possui as seguintes características (CHENG; HWANG, 2001):

- É um algoritmo de busca estocástica, originado dos mecanismos de seleção natural. Um processo estocástico pode ser definido como uma sequência de variáveis aleatórias indexadas ao tempo e também a eventos. É uma variável que se desenvolve no tempo de maneira parcialmente aleatória e imprevisível. Os princípios da natureza nos quais os GAs se inspiram são simples. De acordo com a teoria de C. Darwin, o princípio de seleção privilegia os indivíduos mais aptos com maior longevidade e, portanto, com maior probabilidade de reprodução. Indivíduos com mais descendentes têm mais chance de perpetuarem seus códigos genéticos nas próximas gerações. Tais códigos genéticos constituem a identidade de cada indivíduo e estão representados nos cromossomas.
- O algoritmo é simples e de fácil entendimento, com poucos parâmetros de controle para conduzir à otimização.
- É eficaz para solucionar problemas de otimização com função objetivo descontínua, pois não necessita de informações sobre derivadas da mesma.
- Manipula uma população de soluções que utiliza diferentes regiões no espaço de busca, tornando o algoritmo robusto a mínimos locais.
- É eficaz mesmo trabalhando com uma população pequena;
- Permite as variáveis serem otimizadas como números reais, sem processamento extra.

Na Figura 17, está representado o fluxograma da evolução diferencial.

2.7.1 Fluxograma da evolução diferencial: conceitos básicos

A Seção 2.7.1 explora os fundamentos essenciais desse método de otimização. Inicialmente, uma população de indivíduos é criada aleatoriamente para iniciar o processo de evolução. Cada indivíduo é representado como um vetor de parâmetros que cobre todo o espaço de busca do problema. Ao longo das iterações do algoritmo, esses vetores são submetidos a operadores evolutivos como mutação, cruzamento e seleção, que visam manter a diversidade da população, evitar convergências prematuras e buscar a melhor solução possível. Através desses processos iterativos, a população evolui até que um critério de

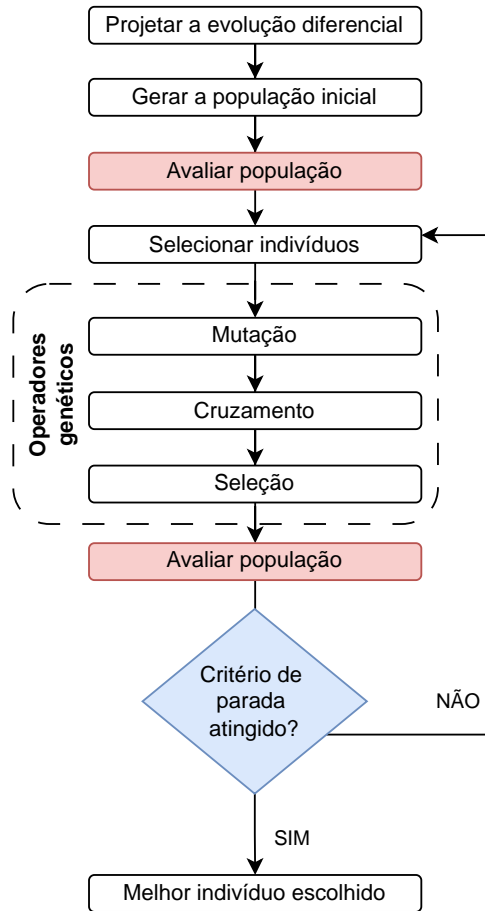


Figura 17 – Fluxograma da evolução diferencial adaptado de (RINCON, 2012).

parada seja alcançado, proporcionando uma solução ótima para o problema de otimização em questão.

2.7.1.1 População de indivíduos

A inicialização da população determina o processo de criação dos indivíduos para o primeiro ciclo do algoritmo. Tipicamente, a população inicial é formada a partir de indivíduos aleatoriamente criados. Populações iniciais aleatórias podem ser semeadas com bons cromossomas para uma evolução mais rápida, quando se conhece, a priori, o valor de boas “sementes” (STORN, K. P. R., 2007).

A população inicial é determinada aleatoriamente e composta com N_p indivíduos, são vetores, que cobrem todo o espaço de busca. Para um problema com n variáveis de projeto, cada vetor possui n componentes.

A população inicial (S) do algoritmo DE é determinada aleatoriamente e composta de N_p indivíduos (vetores), que cobrem todo o espaço de busca Ω . Onde cada indivíduo possui n componentes (parâmetros). Um indivíduo é descrito então como:

$$x_{i,k}, \quad i = 1, 2, \dots, N_p \quad (1)$$

onde k representa o número de interações. No geral, a população é criada através de uma distribuição de probabilidade uniforme. Após criada a população, a mesma é submetida a ação de operadores evolutivos, repetidos em sequência até que a condição de parada seja atendida. Esse critério pode ser convergência da população, um erro mínimo atingido ou um valor pré-definido de iterações. São três tipos de operadores: mutação, cruzamento e seleção.

Para melhor entendimento, suponha que estamos usando o algoritmo DE para otimizar um problema com ($N = 3$) vetores de parâmetros, cada um de dimensão ($n = 2$).

Na iteração inicial ($k = 0$), a população inicial S é distribuída uniformemente na região de busca. Por exemplo, teria-se:

$$S = \{x_{1,0}, x_{2,0}, x_{3,0}\}$$

Digamos que esses vetores sejam:

$$x_{1,0} = \begin{bmatrix} 1 & 2 \end{bmatrix}$$

$$x_{2,0} = \begin{bmatrix} 3 & 4 \end{bmatrix}$$

$$x_{3,0} = \begin{bmatrix} 5 & 6 \end{bmatrix}$$

Agora, na iteração seguinte ($k = 1$), cada vetor em S será alvo de substituição. Isso envolverá as etapas de mutação, cruzamento e aceitação, mas para simplificar, vamos apenas considerar a fase de mutação.

Por exemplo, suponha que, após a fase de mutação, obtivemos os seguintes vetores mutantes:

$$\hat{x}_{1,1} = \begin{bmatrix} 1.2 & 2.3 \end{bmatrix}$$

$$\hat{x}_{2,1} = \begin{bmatrix} 3.1 & 3.9 \end{bmatrix}$$

$$\hat{x}_{3,1} = \begin{bmatrix} 5.2 & 6.1 \end{bmatrix}$$

Esses vetores mutantes agora serão utilizados nas etapas de cruzamento e aceitação para formar a nova população para a próxima iteração do algoritmo. Esse processo continua até que um critério de parada seja atingido ou um número máximo de iterações seja alcançado.

Os operadores do DE se baseiam no princípio da evolução natural cujos os principais objetivos são:

- Manter a diversidade da população;
- Evitar convergências prematuras;

- Obter a melhor solução para o problema;

2.7.1.2 Operações do DE: Mutação

Na fase de mutação, para cada vetor alvo $x_{i,k}$, onde $i = 1, 2, \dots, Np$, um vetor mutante $\hat{x}_{i,k}$ é obtido através de um processo de mutação.

$$\hat{x}_{i,k} = x_{\alpha,k} + F \cdot (x_{\beta,k} - x_{\gamma,k}) \quad (2)$$

A operação mutação (2) é representada por M_d , onde os termos indicam:

- $\alpha, \beta, \gamma \in (1, 2, \dots, Np)$ e $\alpha \neq \beta \neq \gamma$.
- $\hat{x}_{i,k}$ é o vetor mutante para o vetor alvo $x_{i,k}$ na iteração k .
- $x_{\alpha,k}$, $x_{\beta,k}$ e $x_{\gamma,k}$ são três vetores diferentes selecionados aleatoriamente da população atual S para a operação de mutação.
- F é um fator de escala (maior do que zero) que controla a magnitude da mutação.

Para facilitar a compreensão, em um exemplo numérico ilustrativo, considera-se a seguinte população S na iteração k :

$$S = \begin{bmatrix} 4 & 5 & 6 \\ 1 & 2 & 3 \\ 7 & 8 & 9 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 9 & 6 & 1 \end{bmatrix}$$

Para calcular o vetor mutante $\hat{x}_{i,k}$ para o primeiro vetor alvo $x_{1,k}$, são selecionados aleatoriamente os índices α , β e γ , sendo respectivamente 1, 4 e 7. Supõe-se também que o fator de escala F seja 0.5.

Substituindo na expressão, obtém-se:

$$\hat{x}_{1,k} = x_{\alpha,k} + F \cdot (x_{\beta,k} - x_{\gamma,k})$$

$$\hat{x}_{1,k} = x_{2,k} + 0.5 \cdot (x_{1,k} - x_{3,k})$$

Substituindo os valores correspondentes da população S , chega-se a:

$$\begin{aligned}\hat{x}_{1,k} &= [4 \ 5 \ 6] + 0.5 \cdot ([1 \ 2 \ 3] - [7 \ 8 \ 9]) \\ \hat{x}_{1,k} &= [4 \ 5 \ 6] + 0.5 \cdot [-6 \ -6 \ -6] \\ \hat{x}_{1,k} &= [4 \ 5 \ 6] + [-3 \ -3 \ -3] \\ \hat{x}_{1,k} &= [1 \ 2 \ 3]\end{aligned}$$

Portanto, o vetor mutante $\hat{x}_{1,k}$ para o primeiro vetor alvo $x_{1,k}$ é $[1 \ 2 \ 3]$. Na Figura 18 está a ilustração da operação de mutação. Neste exemplo, a população possui 8 indivíduos e foram escolhidos aleatoriamente os indivíduos 1, 4 e 7 para realizar a operação.

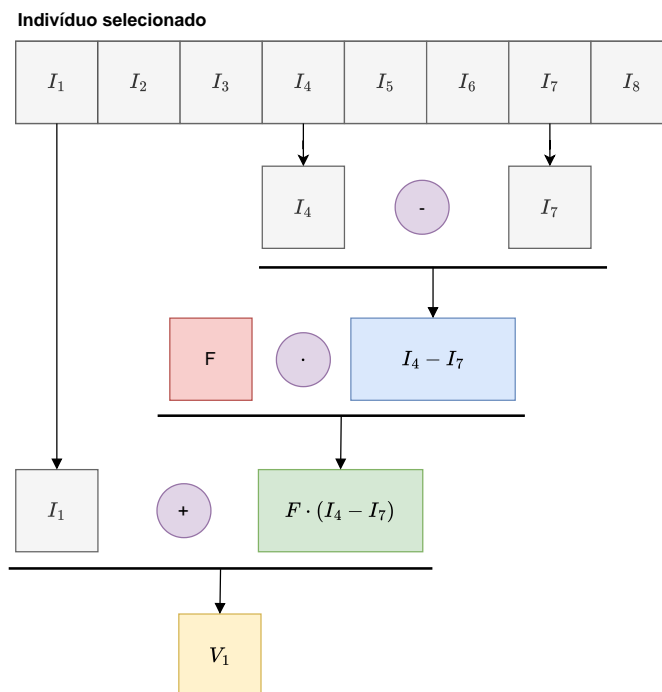


Figura 18 – Operação de mutação (PACHECO, 2015).

2.7.1.3 Operações do DE: Cruzamento

Esta operação foi proposta por *Storn & Price* para aumentar a diversidade da população. Consiste em cruzar indivíduos da população original ($x_{i,k}$) e da população mutada ($\hat{x}_{i,k}$) e assim obter o vetor de teste ($y_{i,k}$).

Ao fim desta operação, todos os indivíduos cruzados formarão uma nova população, a população de cruzamento (S_c) de mesmo tamanho e dimensão das populações obtidas anteriormente.

O cruzamento é definido por,

$$y_{i,k}^j = \begin{cases} \hat{x}_{i,k}^j & \text{se } R_j \leq CR \text{ ou } j = I_i \\ x_{i,k}^j & \text{se } R_j > CR \text{ e } j \neq I_i \end{cases} \quad (3)$$

onde,

- I_i é um inteiro escolhido aleatoriamente no conjunto I , ou seja, $I_i \in I = \{1, 2, \dots, n\}$;
- O sobrescrito j representa o j -ésimo componente dos vetores respectivos.
- $R_j \in (0, 1)$, desenhado aleatoriamente para cada j .

Para melhor compreender o conceito existe uma explicação didática de simples entendimento: Chame o primeiro indivíduo da S_c de $y_{i,k}^j$, resultado do cruzamento de um indivíduo da população original S , $x_{i,k}^j$ com um indivíduo da população mutada S_m o $\hat{x}_{i,k}^j$ para ficar bem didático e fácil de compreender, imagine o primeiro indivíduo da população, ambos selecionados de maneira aleatória. Esses indivíduos possuem dimensão n (lembre-se da matriz que discutimos no início das apresentações na qual cada linha é um novo indivíduo, ou seja, um vetor $[1,n]$) e vamos chamar cada uma dessas n posições de genes. Por exemplo, se o indivíduo é um vetor de dimensão 5, $v = [1 \ 2 \ 3 \ 4 \ 5]$, ele possui 5 genes, um pra cada posição de v . A Figura 19 ilustra a representação do processo de cruzamento.

O objetivo é cruzar os genes dos Indivíduos $x_{i,k}^j$ e $\hat{x}_{i,k}^j$ para gerar o indivíduo $y_{i,k}^j$. Suponha $y_{i,k}^j = [4,4,4,4]$ e $\hat{x}_{i,k}^j = [5,5,5,5]$, na qual os genes 1 e 3 são de $y_{i,k}^j$ e 2 e 4 são de $\hat{x}_{i,k}^j$. Para determinar qual gene é transmitido existe uma taxa de cruzamento C_r , definida no intervalo $[0,1]$. Basicamente, com base neste intervalo um número aleatório é escolhido R_j , se ele for menor que o C_r , o gene mutante é transmitido, caso contrário o gene da população corrente é passado adiante. Então, no exemplo, tem-se $y_{i,k}^j = [4,4,4,4]$ e $\hat{x}_{i,k}^j = [5,5,5,5]$, o número aleatório é sorteado 4 (neste caso $j = 1,2,3,4$ vezes), logo são 4 comparações para decidir qual gene deve ser transmitido para $y_{i,k}^j$. Todo este processo apresentado é para a geração de apenas um indivíduo do S_c , repetido para o N_p indivíduos da população. A expressão a seguir ilustra o cruzamento apresentada em (3).

2.7.1.4 Operações do DE: Seleção

Para entender o conceito de seleção, é importante primeiro compreender o conceito de função objetivo. Uma função objetivo pode ser maximizada ou minimizada e pode ser unidimensional, ou seja, envolve apenas uma variável de projeto, ou multidimensional, o

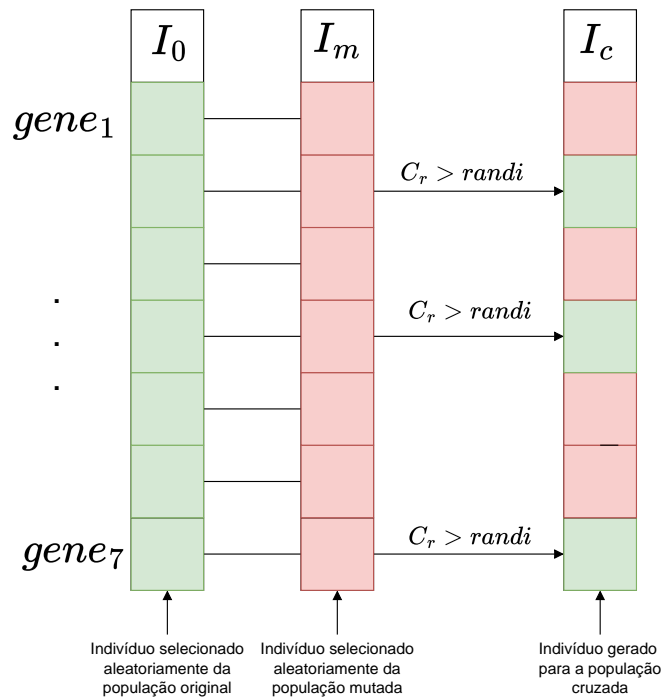


Figura 19 – Operação de cruzamento (PACHECO, 2015).

que significa que envolve mais de uma variável de projeto. As variáveis de projeto, ou variáveis de decisão, são aquelas que podem ser ajustadas durante o processo de otimização, ou seja, são valores que impactam no resultado da função objetivo.

Por exemplo, uma função objetivo pode ser $f(x) = \text{sen}(x)$, esta será a função de avaliação do exemplo, da qual o valor de erro pode ser mensurado. É conhecido que o valor mínimo da função seno é -1, com isso a otimização caminhará para esse valor ao longo das interações, e toda população (neste caso unidimensional, pois só há um indivíduo em cada interação) vai ser avaliada pela função seno. Quanto mais longe de -1, menos apto é aquele indivíduo.

O processo de seleção (mutação e cruzamento) continua até que todos os membros de S sejam considerados. Depois que todos os vetores de teste $y_{i,k}$ de N foram gerados, a aceitação é aplicada. Na fase de aceitação, o valor da função no vetor de teste, $f(y_{i,k})$, é comparado com $f(x_{i,k})$. O resultado da seleção é a população *best* (S_B).

$$x_{i,k+1} = \begin{cases} y_{i,k}, & \text{se } f(y_{i,k}) \leq f(x_{i,k}), \\ x_{i,k}, & \text{caso contrário.} \end{cases} \quad (4)$$

Sendo f a função objetivo representada. Desta forma, os indivíduos mais aptos são passados para a próxima geração, formando a população dos melhores indivíduos. No caso do seno, os indivíduos mais aptos são aqueles com valores próximos de -1. Com isso, é finalizada uma iteração do algoritmos. Na próxima interação, a S será igualada a S_B e todo o processo é realizado novamente até que um critério de parada seja atingido. Ao final de todo o processo, basta escolher o indivíduo da S_B que possua o melhor valor na

avaliação da função objetivo.

No final o objetivo é determinar uma solução ótima. A solução ótima de um problema de programação linear é a solução admissível que apresenta melhor valor para a função objetivo. Quando o objetivo é minimizar a função objetivo, a solução ótima é a solução admissível que permite obter o menor valor para a função objetivo. Para o caso onde o objetivo é a maximização da função objetivo, a solução ótima é a solução admissível que permite obter o maior valor para a função objetivo (BERNARDO, 2016).

2.7.2 Parâmetros computacionais

A compreensão dos parâmetros do algoritmo Differential Evolution (DE) é essencial para sua aplicação eficaz em problemas de otimização. Parâmetros como o tamanho da população (**NP**), o número máximo de iterações (**itermax**), os limites do espaço de busca (**XVmin** e **XVmax**), o fator de escala na mutação (**F**) e a taxa de cruzamento (**CR**) desempenham papéis cruciais na dinâmica e no desempenho do algoritmo. Cada um desses elementos influencia diretamente a capacidade do DE de convergir para uma solução ótima, garantindo a integridade e a validade das soluções exploradas durante o processo de otimização.

Os parâmetros utilizados no DE determinam a eficiência e a qualidade da solução encontrada. O tamanho da população (**NP**) afeta a diversidade inicial, permitindo uma exploração mais ampla do espaço de soluções. O número máximo de iterações (**itermax**) define o tempo de execução do algoritmo, impactando a convergência para uma solução aceitável. Os limites do espaço de busca (**XVmin** e **XVmax**) asseguram que as soluções candidatas permaneçam dentro de um intervalo válido e relevante para o problema, enquanto o fator de escala na mutação (**F**) e a taxa de cruzamento (**CR**) controlam a magnitude das mudanças aplicadas e a probabilidade de aceitação das soluções modificadas.

A escolha da estratégia de evolução também é fundamental, pois determina como as soluções candidatas são combinadas e ajustadas ao longo das gerações. Finalmente, a frequência de saída de resultados intermediários (**refresh**) permite monitorar o progresso do algoritmo e ajustar os parâmetros conforme necessário, contribuindo para um ajuste fino do processo de otimização.

2.8 TENSORES

Tensores são objetos matemáticos úteis para descrever propriedades física, como vetores e matrizes. A classificação (também chamada de ordem) de um tensor é definida com base no número de direções (dimensão da matriz) necessárias para descrevê-lo (UNIVERSIDADE DE CAMBRIDGE, 2024).

Um tensor é um operador multilinear que transforma um espaço linear em outro. Na Figura 20 está ilustrado um tensor de terceira ordem aplicado em uma situação real.

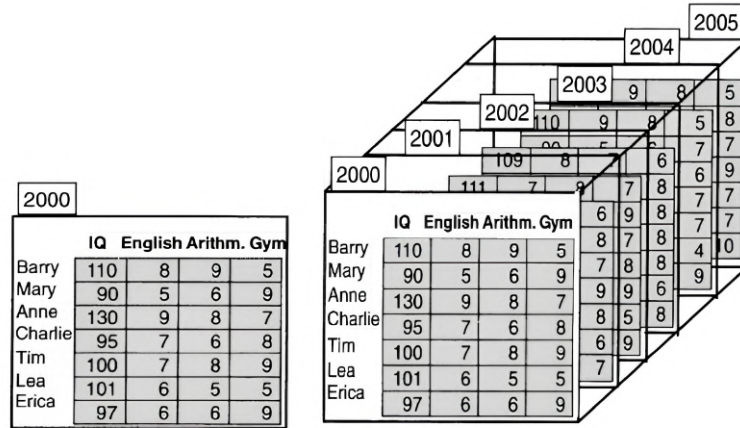


Figura 20 – Tensor de terceira ordem aplicado em uma situação real (KROONENBERG, 2008).

Note que ele é útil para processar uma informação multilinear, em cada camada temos a associação de um ano, indo de 2000 até 2005. Para cada ano temos informações úteis de notas para cada aluno e a informação do QI (Quociente de Inteligência) deles, desta representação pode-se retirar diversas informações úteis, como somar e dividir para obter média de notas, retirar a maior média geral, menor, etc.

Outro exemplo prático consiste no uso de tensores RNCs (Redes Neurais Convolucionais) que trabalham diretamente com Tensores, que são entidades geométricas introduzidas na matemática e na física para generalizar a noção de escalares, vetores e matrizes (ESTEVEZ, 2020). As entradas das RNCs são imagens, que devem ser transformadas em tensores normalmente tridimensionais, devido a caracterização de uma imagem digital em três camadas RGB que é a abreviatura de um sistema de cores aditivas em que o Vermelho (Red), o Verde (Green) e o Azul (Blue) são combinados de várias formas de modo a reproduzir um largo espectro cromático. Veja a Figura 21.

Neste projeto, opta-se pela utilização de sinais multidimensionais através da aplicação de tensores, aproveitando-se da flexibilidade da álgebra tensorial. Destaca-se especialmente a capacidade de matricular os tensores. Essa escolha decorre da necessidade de lidar com um sistema multidimensional no projeto. Os métodos tensoriais desempenham um papel crucial, pois são capazes de processar simultaneamente diversas variabilidades presentes nos sinais, oferecendo uma abordagem eficiente para lidar com a complexidade dos dados multidimensionais envolvidos.

Na presente pesquisa optou-se por utilizar tensores em que cada camada representa a disposição de uma caixa dentro de um palete, considerando as dimensões de comprimento (C) e largura (L). Cada camada é equivalente às dimensões do palete. Na Figura 22a, é apresentada uma representação dessa camada do tensor, em que a área amarela indica a

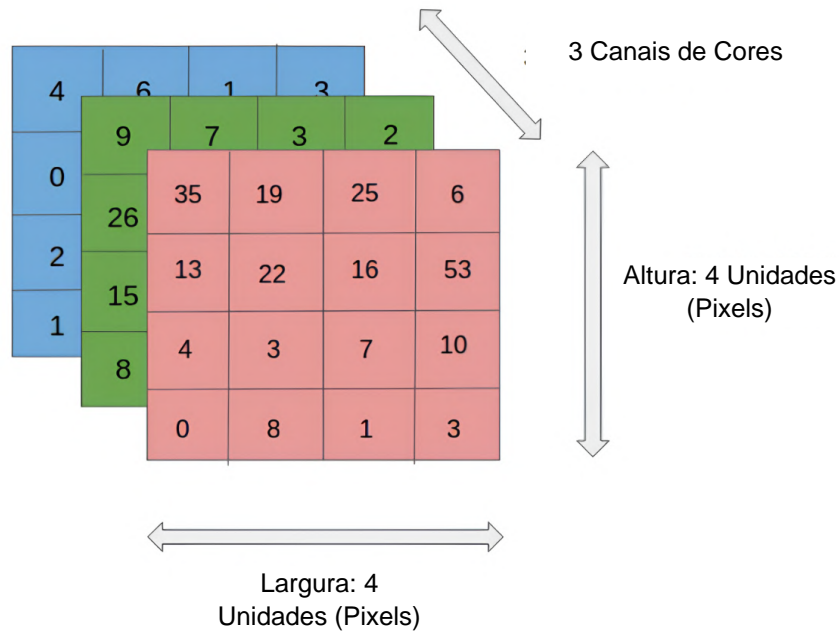


Figura 21 – Tensor aplicado ao contexto de RGB (ESTEVEES, 2020).

localização da caixa dentro do palete. Essa abordagem se destaca pela sua simplicidade e eficácia na verificação de sobreposições, realizada através de uma simples soma de camadas, conforme ilustrado na Figura 22b. É perceptível que áreas com uma tonalidade amarela mais clara indicam sobreposição entre duas camadas, enquanto no centro dessas sobreposições destaca-se um ponto amarelo mais intenso, indicando a interseção das três camadas. A representação é baseada em matrizes binárias, e as cores foram aplicadas apenas para facilitar a visualização.

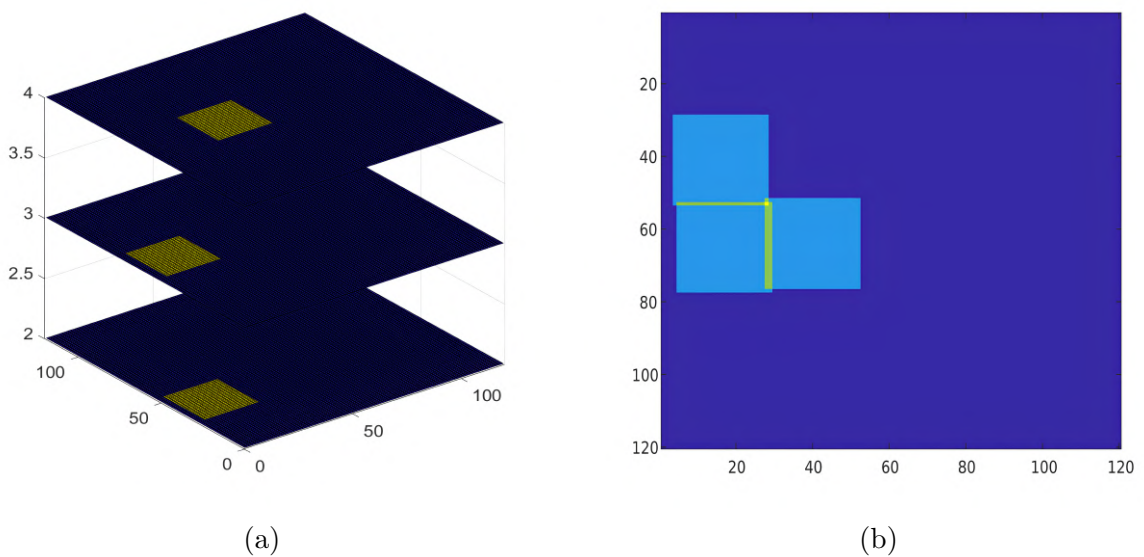


Figura 22 – Representação das camadas de índice 2, 3 e 4 de um tensor: (a) 3D. (b) 2D.

A detecção da ausência de sobreposição também é possível. Na Figura 23a, é

apresentada a ilustração de duas camadas. Na Figura 23b, é mostrada a representação bidimensional dessas duas camadas, destacando a ausência de sobreposição.

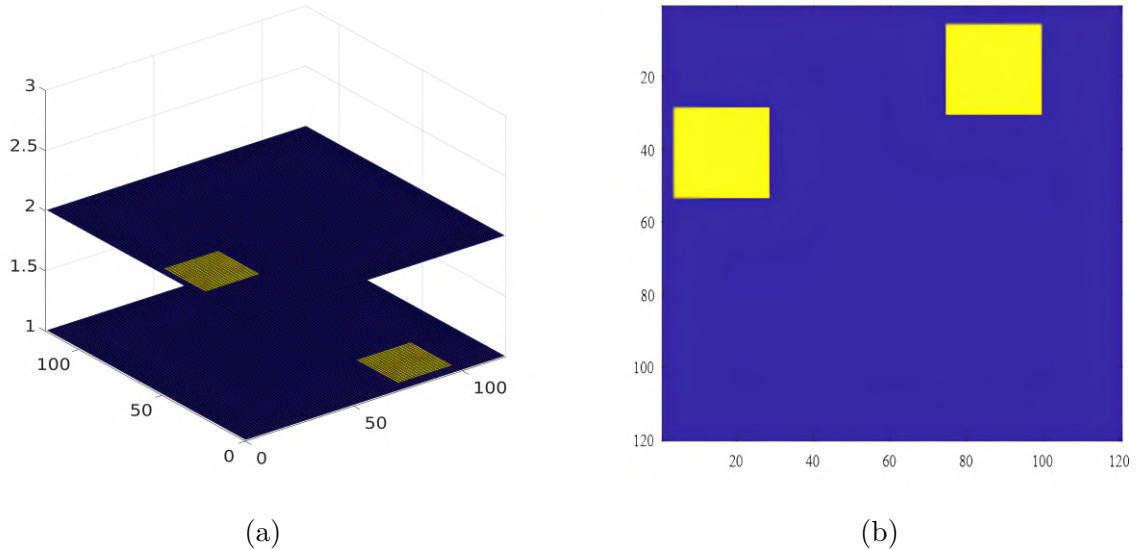


Figura 23 – Representação das camadas de índice 1 e 2 de um tensor: (a) 3D. (b) 2D.

Portanto, optou-se por utilizar tensores devido às características intrínsecas desses elementos, que proporcionam uma capacidade única de tradução de sistemas físicos para virtuais. Essa escolha se justifica pela vantagem de não apenas detectar sobreposições, mas também tornar o sistema sensível à eliminação delas. A representação tensorial permitiu lidar com casos de sobreposição, ao mesmo tempo que possibilitou a eliminação dessas sobreposições e prevenção de novas colisões. Essa abordagem oferece uma solução abrangente e eficaz para a gestão de interações complexas entre elementos físicos e virtuais.

3 METODOLOGIA

A presente seção detalha a abordagem metodológica adotada para a execução da pesquisa, fornecendo uma estrutura sólida para a análise e interpretação dos dados. Objetiva-se garantir a confiabilidade e validade do estudo e assegurar a replicabilidade dos resultados, contribuindo assim para a construção do conhecimento científico na área de estudo.

Dada a natureza dual da pergunta de pesquisa, a metodologia proposta é dividida em duas fases. A fase inicial visa descobrir a posição e orientação ótimas de um conjunto de caixas cuja as dimensões são reduzidas de propósito dentro de um percentual definido. A fase subsequente busca eliminar possíveis sobreposições quando as caixas são redimensionadas para seu tamanho original. Ambas as etapas dependem do conceito de tensores, em que cada camada reflete as dimensões do palete e representa uma caixa distinta a ser acomodada no palete. Nas subseções a seguir, são apresentadas e discutidas as etapas que compõem o método proposto.

3.1 FASE 1: POSICIONAMENTO ÓTIMO DAS CAIXAS

Para esta fase, projetou-se uma interface gráfica interativa que permite definir os valores das variáveis necessárias para a otimização, como o tamanho do palete em X e Y, o tamanho da população (NP), o número máximo de iterações (intermax), o passo máximo do DE (F), a taxa de cruzamento (CR), a taxa de atualização da tela (refresh) e a estratégia de evolução. Após pressionar o botão “Executar”, a otimização da função objetivo é iniciada. A Figura 24 representa o aplicativo principal utilizado para a execução desta fase.

A função objetivo desempenha um papel crucial na otimização, permitindo a avaliação da população. Recorda-se que no fluxograma apresentado na Figura 17, a execução dessa função é parte da etapa “Avaliar população” da evolução diferencial. Ela fornece o valor necessário para verificar se o critério de parada foi atingido.

O fluxo para determinar esse valor está detalhado na Figura 25. Primeiramente, verifica-se se há pontos fora da caixa; se sim, o critério de parada não é atendido. Caso contrário, verifica-se se há alguma caixa fora do palete. Se houver caixas que não podem ser alocadas dentro do palete devido a suas dimensões, o critério também não é atendido. Em seguida, verifica-se se há sobreposição entre as caixas alocadas no palete. Se houver, o critério não é atendido, e o valor ótimo deve ser proporcional ao grau de sobreposição. Se não houver sobreposição, o critério é atendido, e o melhor valor ótimo é encontrado.

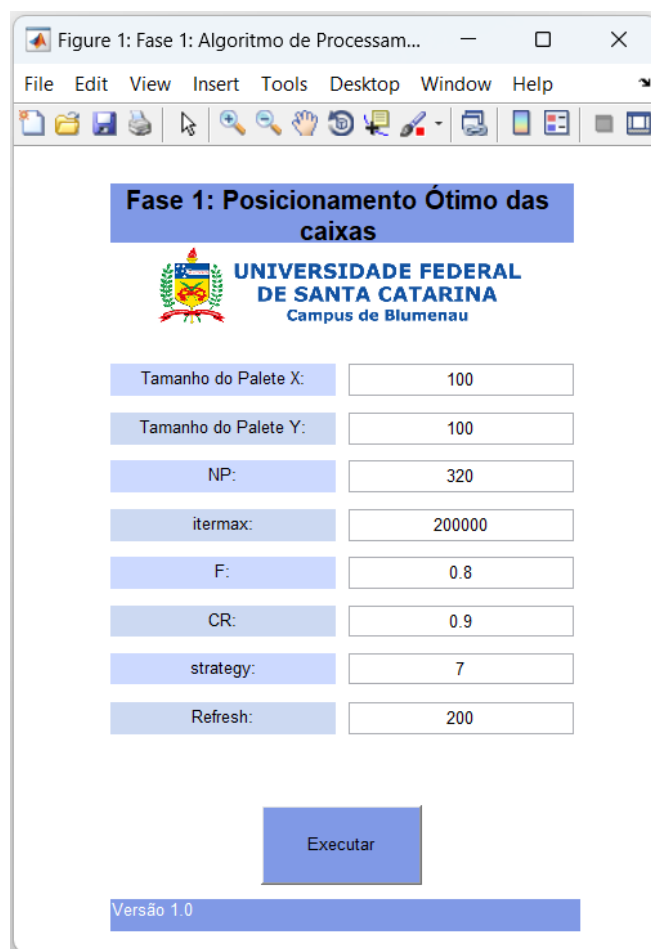


Figura 24 – Aplicativo principal para execução da Fase 1.

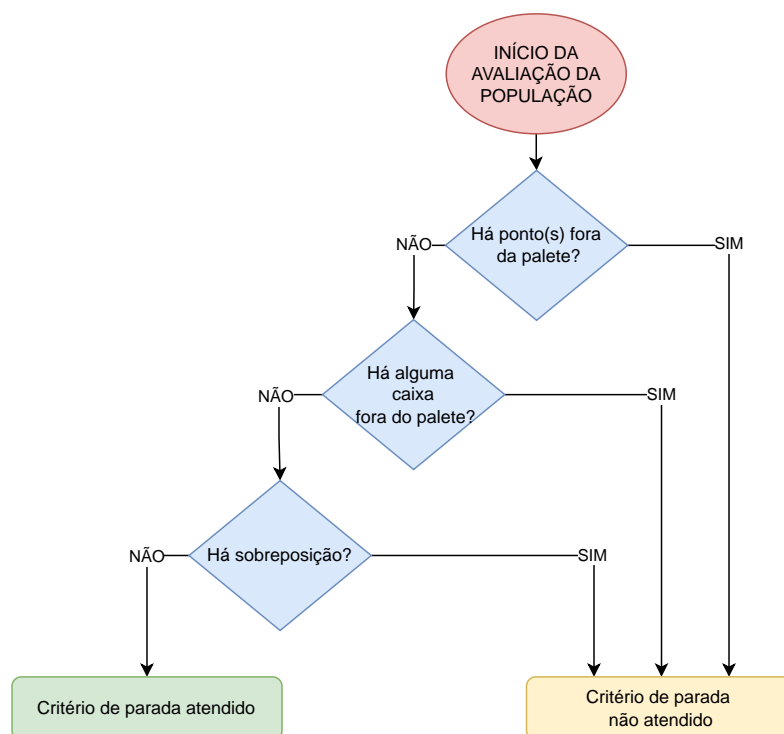


Figura 25 – Fluxo de determinação do valor ótimo.

3.1.1 Função Objetivo Teórica

No âmbito da otimização, todo problema pode ser expresso matematicamente pela definição de sua função objetivo. Essa função desempenha um papel crucial ao quantificar e expressar o objetivo específico que deve ser alcançado no problema em questão. Ao avaliar e aprimorar essa função, é possível mensurar de forma eficaz o quão bem uma determinada solução ou conjunto de decisões atende aos requisitos e critérios do problema. A função objetivo orienta a busca pela solução ótima que melhor se alinha ao resultado desejado ou objetivo da otimização, garantindo a obtenção dos resultados mais favoráveis.

Considera-se importante destacar que a entrada para o problema PCP estudado é dada como um conjunto de variáveis (L, W, x_i, y_i, h_i) . Onde, L : é o comprimento do palete; W : é a largura do palete; (x_i, y_i) : é o par ordenado que representa a posição coordenada da caixa; (a_i, b_i) : representam as dimensões de cada caixa, e h_i : é a orientação da caixa. $h_i = 1$ significa orientação vertical e $h_i = 0$ significa orientação horizontal.

Nesta fase, o foco é na minimização da área não ocupada em um palete com dimensões $L \times W$, onde um conjunto de N caixas de dimensões $a_i \times b_i \quad \forall i = 1 \dots N$ está disposto. O problema de otimização é formulado por meio do uso de uma entidade matemática conhecida como tensor, que organiza os dados em uma estrutura multidimensional. Nesse contexto, emprega-se um tensor (T) com N camadas, onde cada camada representa uma matriz binária indicando a posição e orientação de um objeto individual redimensionado.

Antes de iniciar o processo de otimização o redimensionamento das caixas é proposto para reduzir o custo computacional durante o processo de otimização, sugerindo-se o uso de valores entre 80% e 100% do tamanho original para cada caixa. Valores inferiores podem levar a mínimos locais na função objetivo proposta. Ao combinar as camadas do tensor, obtém-se um mapa completo que mostra a distribuição de todas as caixas colocadas no palete, representando a área total ocupada.

A otimização proposta busca minimizar a área total não ocupada que é calculada subtraindo a área total ocupada $\left(\sum_{i=1}^N T_i(a_i, b_i, x_i, y_i)\right)$ da área total do palete $(L \cdot W)$, conforme indicado na Equação (5)a da Função Objetivo. A bordagem proposta proporciona uma quantificação precisa do espaço disponível dentro do palete, facilitando a determinação de uma disposição otimizada das caixas.

minimize:

$$L \cdot W - \sum_{i=1}^N T_i(a_i, b_i, x_i, y_i) \quad (5a)$$

sujeito a:

$$x_i + a_i \leq L \quad (5b)$$

$$y_i + b_i \leq W \quad (5c)$$

$$x_i \geq 1 \quad (5d)$$

$$y_i \geq 1 \quad (5e)$$

$$\max \left(\sum_{i=1}^N T_i(a_i, b_i, x_i, y_i) \right) \leq 1 \quad (5f)$$

Além disso, o problema de otimização do presente estudo incorpora dois conjuntos distintos de restrições para garantir a inexistência de sobreposição entre as caixas e para proibir o posicionamento de qualquer objeto além dos limites do palete. As equações (5)b, (5)c, (5)d e (5)e introduzem termos de penalidade que consideram casos em que as caixas ultrapassam as fronteiras do palete, enquanto a Equação (5)f define rigorosamente a restrição relacionada à sobreposição. Essas restrições protegem efetivamente a integridade do arranjo das caixas, garantindo que cada caixa ocupe um espaço único e não sobreposto, permanecendo dentro dos limites prescritos do palete. A utilização dessas equações facilita o estabelecimento de um *framework* de otimização abrangente e eficiente que não apenas otimiza o espaço vazio, mas também respeita as fronteiras espaciais e as restrições de arranjo impostas pelo problema de empacotamento.

3.1.2 Função objetivo implementada computacionalmente

A presente seção visa descrever a metodologia computacional para otimização da disposição de caixas em um palete. Inicialmente, são definidos os parâmetros de entrada através da construção da Matriz de Características dos Objetos Virtuais (MCV). Esta matriz é composta por duas partes: Estática e Dinâmica, onde a parte Estática (MCU) é configurada pelo usuário através de uma interface interativa. Essa interface permite a definição de características como dimensões, orientação e presença das caixas a serem otimizadas.

Após a configuração pelo usuário, os dados são processados para ajustar as dimensões e remover caixas não presentes, resultando na MCV final. Em seguida, um tensor é construído para alocar as caixas virtualmente dentro do palete, respeitando-se diversas restrições físicas como posição inicial, limites do palete e condições de sobreposição entre caixas.

Finalmente, é realizado o cálculo do grau de sobreposição entre as caixas alocadas para determinar a eficiência da disposição. Este processo de avaliação visa minimizar a área vazia no palete, garantindo a máxima utilização do espaço disponível.

3.1.2.1 Parâmetros de entrada

Nesta fase é construída a Matriz de Características dos Objetos Virtuais (MCV), que incorpora as características das caixas a serem paletizadas. Ela é dividida em duas partes distintas: Estática e Dinâmica. A parte Estática é uma Matriz de Características Escolhidas pelo Usuário (MCU), enquanto a parte Dinâmica é um vetor que contém as coordenadas otimizadas utilizadas para posicionar as caixas dentro do palete.

Para começar a compor a MCU, o usuário precisa primeiro definir algumas características do conjunto que deseja otimizar. Para facilitar esse processo, uma interface interativa foi desenvolvida, onde o usuário pode inserir esses dados. A representação visual dessa interface está disponível na Figura 26.

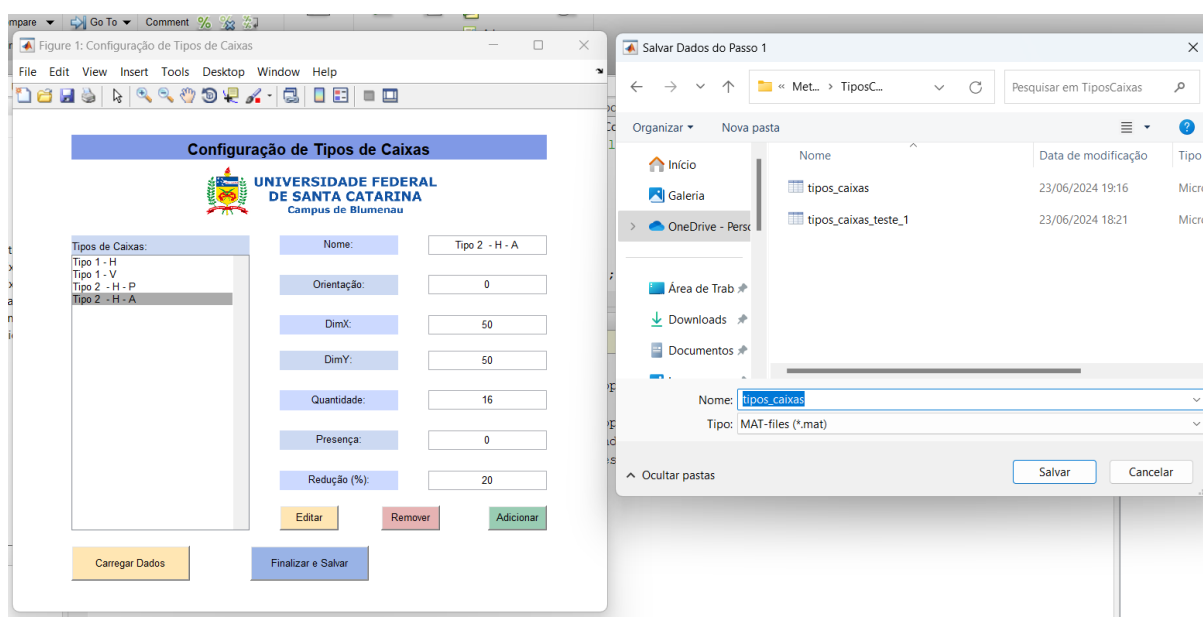


Figura 26 – Tela interativa que permite a escolha do conjunto a ser otimizado pelo usuário.

A partir da interface, o usuário pode carregar conjuntos de dados chamados “Tipos de Caixas”, que representam os diferentes tipos de caixas cujos posicionamentos ele deseja otimizar. Da mesma forma, ele pode finalizar e salvar o conjunto de caixas configurado após a definição.

Cada conjunto pode incluir um ou mais tipos de caixas, que são categorias definidas pelo usuário para descrever melhor características específicas que ele deseja representar na configuração. Para cada tipo de caixa configurado, o usuário deve especificar obrigatoriamente os seguintes parâmetros: dimensões de largura e comprimento, quantidade, orientação (horizontal ou vertical), presença (presente ou não), e o percentual de redução das dimensões das caixas.

A título de exemplo, o resultado da alimentação de dados através da interface ilustrada na Figura 26, podem ser sintetizados na Tabela 1. Neste exemplo o usuário escolheu configurar o conjunto com quatro tipos de caixas, denominados como “Tipo 1 - H”, “Tipo 1 - V”, “Tipo 2 - H - P” e “Tipo 2 - V - A”. Para cada tipo, definiu que desejava duas unidades de cada. Portanto, a matriz resultante contém oito linhas. Os tipos que começam com “Tipo 1 -” indicam se as caixas devem ser posicionadas horizontalmente (“H”) ou verticalmente (“V”). Já os tipos que começam com “Tipo 2 - V -”, contém informações a mais, que indicam se o conjunto está presente (“P”) ou ausente (“A”) na otimização que ele deseja realizar.

Tabela 1 – Características Escolhidas pelo Usuário

Nome	Largura	Comprimento	Orientação	Presença	Redução (%)
Tipo 1 - H	25,2	25,2	0	1	10%
Tipo 1 - H	25,2	25,2	0	1	10%
Tipo 1 - V	25,2	25,2	1	1	15%
Tipo 1 - V	25,2	25,2	1	1	15%
Tipo 2 - V - P	50	25	1	1	20%
Tipo 2 - V - P	50	25	1	1	20%
Tipo 2 - V - A	50	25	1	0	20%
Tipo 2 - V - A	50	25	1	0	20%

Na metodologia proposta, os dados organizados na Tabela 1 são processados antes de serem enviados para a função objetivo, visando aumentar a eficiência do algoritmo de otimização. Por exemplo, cada tipo de caixa possui um percentual de redução aplicado às suas dimensões. Portanto, as dimensões na Tabela 1 são ajustadas e arredondadas para números inteiros, conforme mostrado na Tabela 2. Além disso, as caixas que não estão presentes no arranjo devem ser eliminadas. No exemplo em discussão, das 8 caixas adicionadas pelo usuário, apenas 6 estão efetivamente presentes no palete. Adicionalmente, a porção Dinâmica também é arredondada para números inteiros.

Tabela 2 – Matriz de Características de objetos virtuais (MCV) na condição inicial

Estática				Dinâmica	
Largura	Comprimento	Orientação	Presença	Pos. X	Pos. Y
23	23	0	1	3	21
23	23	0	1	40	26
21	21	1	1	38	11
21	21	1	1	36	4
40	20	1	1	15	32
40	20	1	1	49	39

Após os ajustes das partes Dinâmica e Estática, a MCV em sua condição inicial é obtida, conforme ilustrado na Tabela 2. É importante destacar que, através da análise

dos valores mínimo e máximo da parte Dinâmica, é possível responder ao questionamento: “Há ponto(s) fora do palete?”. Por exemplo, se as dimensões do palete são iguais a 100 [u.m] de largura e 100 [u.m] de comprimento posição resulta em 105 ou -1, então há pontos fora das dimensões do palete.

3.1.2.2 Construção do tensor e alocação dos objetos virtuais em cada camada

Nesta etapa, um tensor é construído para agrupar as informações do conjunto de caixas que estão sendo alocadas no palete. Inicialmente, cada camada do tensor possui as dimensões bidimensionais do palete, e o tensor possui um número de camadas igual ao número de caixas a serem otimizadas. A Figura 27a ilustra o tensor vazio; as bordas amarelas são apenas para facilitar a visualização das camadas. Já a Figura 27b ilustra uma caixa alocada em uma das camadas de um tensor.

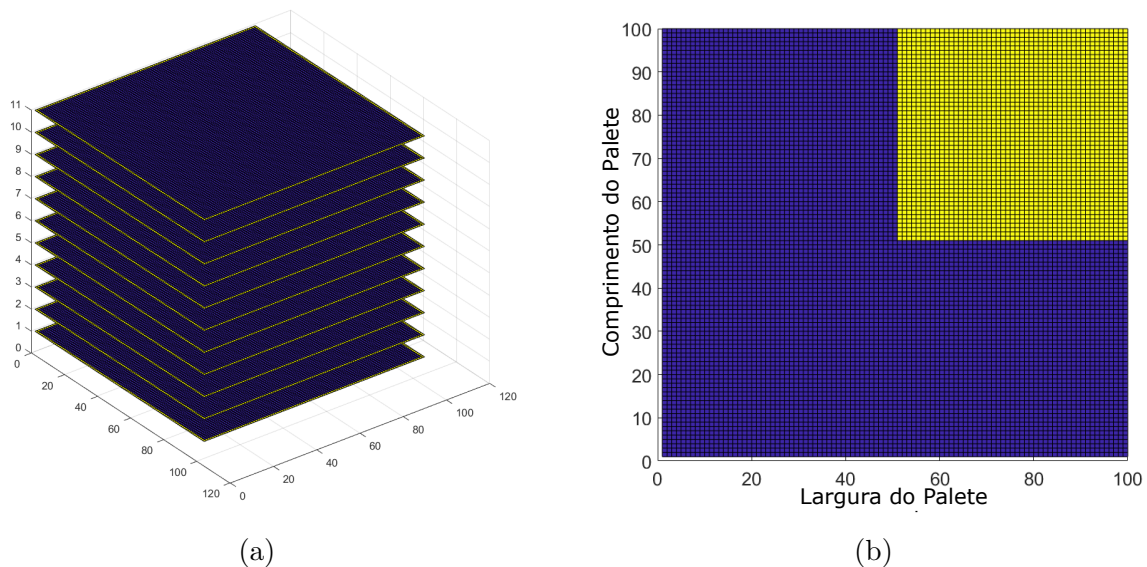


Figura 27 – (a) Um tensor 3D vazio construído para agrupar as informações sobre as caixas, (b) Uma caixa alocada em uma das camadas de um tensor.

Primeiramente, atualizam-se as informações de largura e comprimento das caixas que devem estar presentes no palete, considerando a orientação. Na Tabela 2, são apresentadas as condições iniciais dessas informações. Na Tabela 3, os dados são rearranjados com base na validação da coluna “Orientação”. Nesta nova tabela, os elementos destacados em azul e vermelho mudam de coluna. Os elementos destacados em azul representam caixas quadradas, e essa mudança de coluna não tem efeito prático sobre suas dimensões. Por outro lado, os elementos destacados em vermelho efetivamente mudam sua orientação ao trocar as dimensões de largura e comprimento. Quando nos referimos às dimensões do objeto virtualizado, estamos considerando as dimensões já ajustadas conforme a orientação.

Após a criação da Tabela 3, determinam-se os vértices da caixa virtual para compará-los com as extremidades do palete, verificando se a caixa fica completamente dentro dos limites ou se os ultrapassa.

Tabela 3 – Matriz de Características de objetos virtuais (MCV) em função da coluna “Orientação”

Estática				Dinâmica	
Largura	Comprimento	Orientação	Presença	Pos. X	Pos. Y
23	23	0	1	3	21
23	23	0	1	40	26
21	21	1	1	38	11
21	21	1	1	36	4
20	40	1	1	15	32
20	40	1	1	49	39

Na Figura 28, observa-se que a caixa virtual é definida a partir do ponto inicial (L_i, C_i) , e utilizando-se as dimensões de largura e comprimento, resultando nos seguintes vértices:

- **Origem (inferior esquerdo):** (L_i, C_i)
- **Inferior direito:** $(L_i + \text{Largura}, C_i) = (L_j, C_i)$
- **Superior esquerdo:** $(L_i, C_i + \text{Comprimento}) = (L_i, C_j)$
- **Superior direito:** $(L_i + \text{Largura}, C_i + \text{Comprimento}) = (L_j, C_j)$

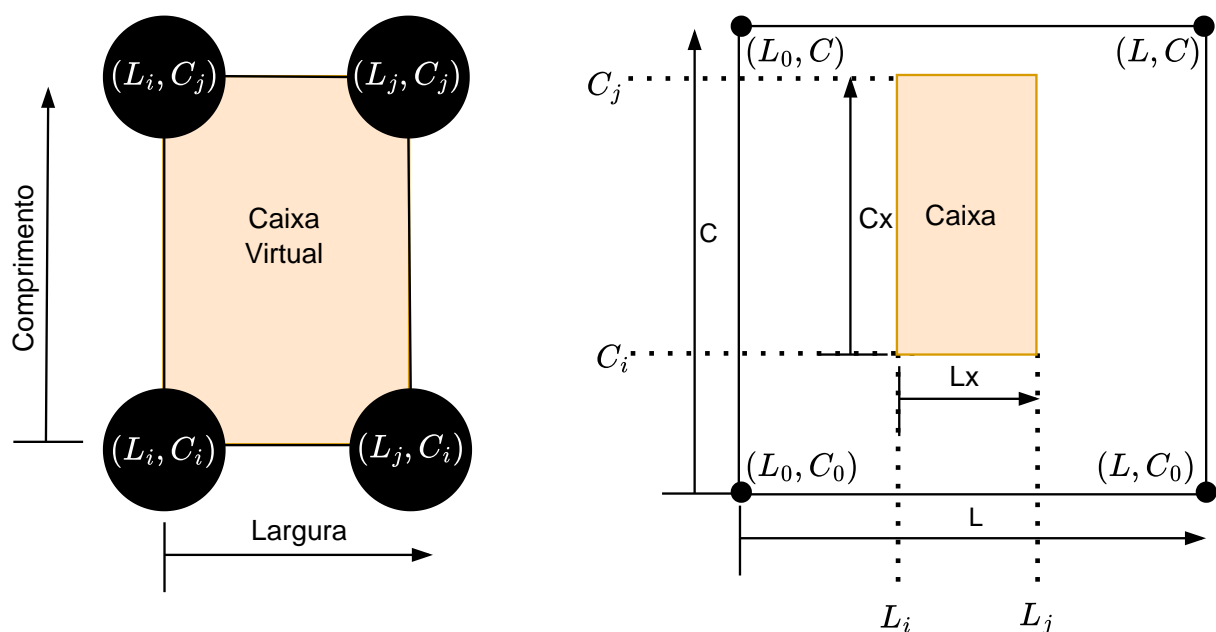


Figura 28 – Posicionamento da caixa virtual.

Para alocar um objeto virtual em uma camada de tensor, ele é representado preenchendo os valores correspondentes com 1, indicando sua presença nesse espaço. Isso requer que as quatro restrições teóricas definidas nas equações de (5)b a (5)e sejam fisicamente válidas e respeitadas da seguinte forma:

$$L_i \geq L_0 \tag{6)a}$$

$$C_i \geq C_0 \tag{6)b}$$

$$L_i + \text{Largura} \leq L \tag{6)c}$$

$$C_i + \text{Comprimento} \leq C \tag{6)d}$$

A Figura 29a ilustra a caixa virtual já posicionada dentro do palete pois as expressões presente em (6) foram atendidas. Na Figura 29b é apresentado quatro casos nos quais as condições para as expressões (6)a, (6)b, (6)c e (6)d não são satisfeitas, correspondendo às caixas marcadas como B, C, D e E, respectivamente.

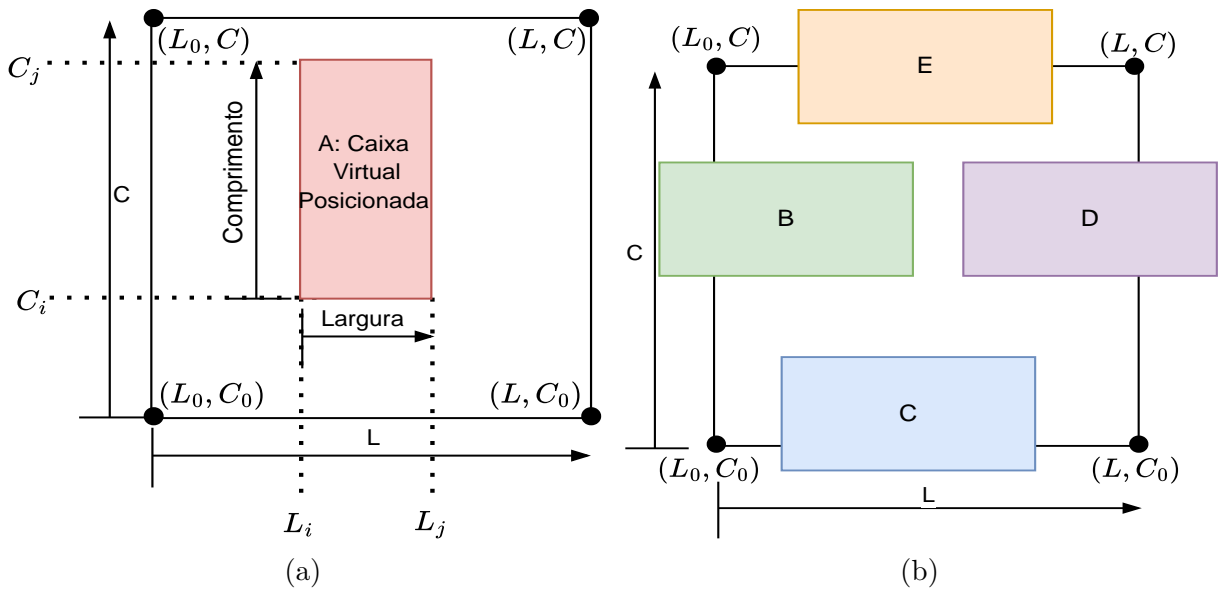


Figura 29 – Restrições para para o posicionamento da caixa virtual: (a) Condições atendidas. (b) Condições não atendidas.

Se a condição descrita na Expressão (6) não for atendida, o objeto virtual não é alocado e o critério não é cumprido. Este cenário é semelhante ao ilustrado na Figura 25, que corresponde à resposta negativa à pergunta “Há alguma caixa fora do palete?”. Se a resposta for positiva, o processo de busca pelo valor ótimo é interrompido. Caso contrário, o processo avança para a etapa de mensuração do grau de sobreposição.

3.1.2.3 Mensuração do Grau de Sobreposição

Esta é a última verificação proposta na metodologia para a etapa de “Avaliação da população”. Esta verificação responde ao questionamento “Há sobreposição?” apresentado

na Figura 25 e relacionado à Equação (5)f. Idealmente, se todas as caixas no arranjo fossem alocadas no palete sem sobreposição e ocupando todo o espaço, o valor máximo na representação bidimensional seria igual a um, como mostrado na Figura 23. No entanto, geralmente há sobreposições, conforme está ilustrado na Figura 22. Por isso, foi desenvolvida uma estratégia para medir o grau de sobreposição.

Para medir o grau de sobreposição, todas as camadas do tensor são somadas para obter uma representação bidimensional, como apresentada na Figura 31b. A partir dessa representação, são realizadas operações para diferenciar a porção ocupada efetivamente da porção sobreposta da camada.

Na representação bidimensional, posições com valores iguais a um ou zero indicam que não há sobreposição naquela área. Para identificar regiões com sobreposição, podemos subtrair uma unidade de cada posição na matriz somada. Em uma situação sem sobreposição, os valores máximos seriam zero e as regiões anteriormente vazias teriam um valor de -1. Ao substituir -1 por 0, obtemos uma matriz totalmente zerada, indicando ausência de sobreposição.

No entanto, em casos com sobreposição, as posições que originalmente tinham valores 2, 3, etc., passarão a ter valores 1, 2, etc., após a subtração, indicando ainda a presença de sobreposição. Assim, ao realizar essa subtração e ajustar os valores negativos para zero, podemos destacar as áreas com sobreposição. A matriz resultante mostrará apenas os valores relacionados à sobreposição, facilitando a identificação dessas regiões.

Na Figura 30a está ilustrada uma representação bidimensional da disposição das caixas com sobreposição. Já na Figura 30b é mostrada a matriz após o tratamento, evidenciando apenas as regiões com sobreposição após a subtração de uma unidade e ajuste dos valores negativos para zero.

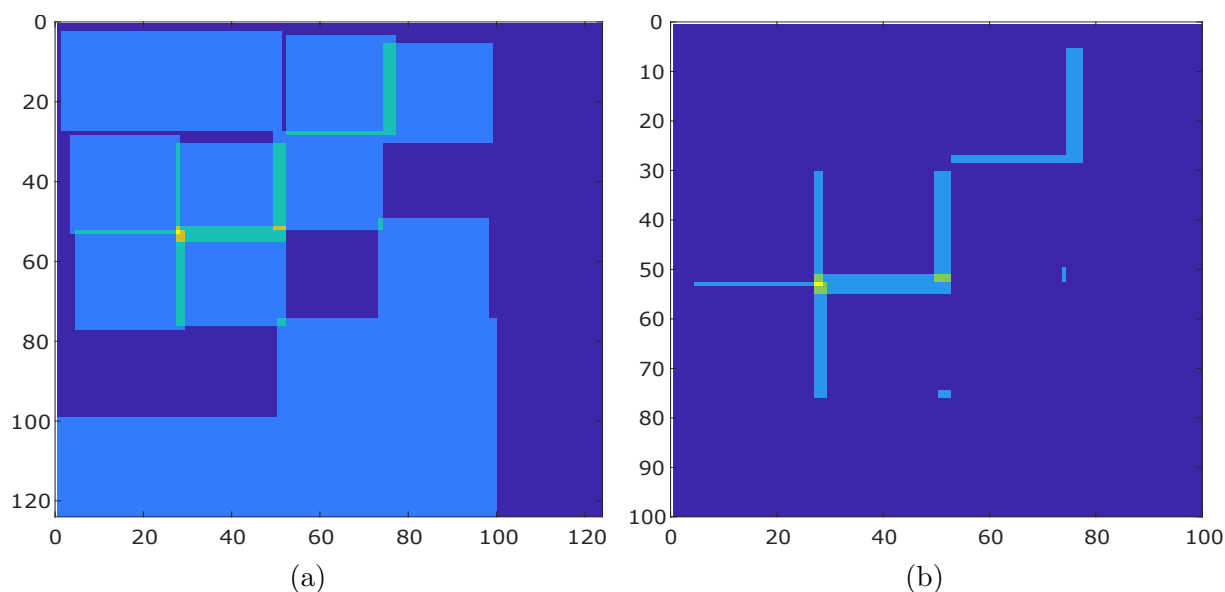


Figura 30 – (a) Representação bidimensional da disposição das caixas com sobreposição. (b) Áreas de sobreposição em evidência.

Especificamente, essa etapa mede o grau de sobreposição que a alocação dos objetos pode gerar e soma essa medida com o cálculo da área vazia para obter o valor ótimo. Esta estratégia demonstrou ser eficiente, pois o objetivo é minimizar a área vazia, de modo que, quanto menor o grau de sobreposição, melhor será atendido o critério de parada.

3.1.3 Critério de parada atendido

O critério de parada é alcançado quando as condições definidas são satisfeitas, conforme mostrado na Figura 25. Isso marca o término da Fase 1, resultando na geração de dados que alimentam a Fase 2.

O processo de otimização discutido resulta na formação de um tensor. Dentro desse tensor, cada camada é uma matriz que representa um objeto distinto e redimensionado. As áreas ocupadas pelo objeto são marcadas com 1, enquanto as áreas vazias são representadas por 0. Isso é exemplificado na Figura 31a, onde as áreas amarelas indicam as regiões ocupadas pelas caixas, enquanto as áreas azuis indicam os espaços vazios.

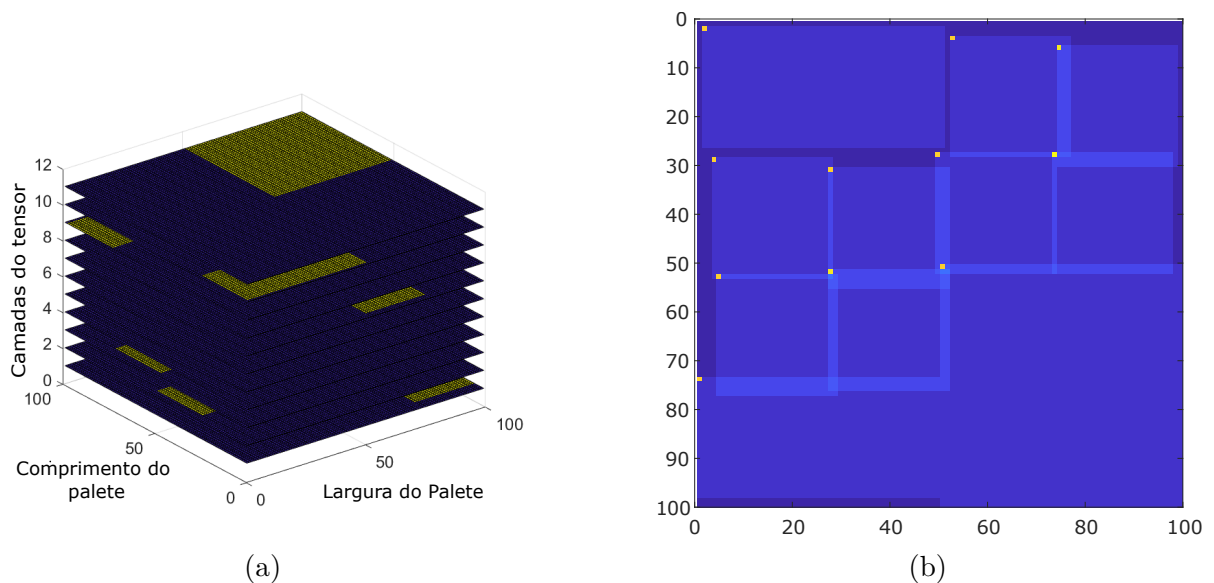


Figura 31 – (a) Visualização da Distribuição de Objetos no Tensor. (b) Tensor na sua representação bidimensional, destacando as caixas redimensionadas e suas coordenadas na disposição ideal.

Essa representação visual é apresentada em um cenário com um palete de dimensões 100 [u.m] por 100 [u.m], contendo um total de 11 camadas (cada uma representando uma caixa). A partir desse tensor otimizado, é possível extrair as coordenadas finais (x , y) para qualquer caixa no sistema. Esse exemplo destaca a utilidade do tensor para obter as coordenadas finais como uma característica crucial na descrição da posição dos objetos. Essas informações serão, por fim, utilizadas para reorganizar os objetos redimensionados nos paletes.

3.2 FASE 2: CORREÇÃO DE SOBREPOSIÇÃO

Esta seção detalha o procedimento de correção de sobreposições. Esta etapa ocorre após o redimensionamento das caixas para seus tamanhos originais, o que potencialmente gera sobreposições. Para entender melhor essa etapa, pode-se fazer uma analogia: é como se o palete fosse inclinado sequencialmente em direção a cada um de seus quatro lados principais, fazendo com que as caixas se separassem mecanicamente por gravidade.

O procedimento de correção de sobreposições abrange diversas etapas, desde a inicialização até a exibição final do resultado. Isso inclui a escolha dos parâmetros do usuário e o carregamento de dados, a criação de um tensor expandido para facilitar o processo de movimentação, e ciclos de movimentações nas quatro direções principais (para cima, para baixo, para a direita e para a esquerda). Como resultado, são geradas representações visuais das caixas e suas coordenadas de posicionamento, minimizando sobreposições.

Na primeira etapa, o usuário escolhe os parâmetros que guiam o algoritmo de correção de sobreposição e realiza o carregamento de dados. Similar aos processos de definição e uso de dados da Fase 1, foi desenvolvida uma tela interativa para a Fase 2. Essa interface permite que o usuário interaja diretamente com o algoritmo de correção de sobreposição de dados. Na Figura 32 está ilustrado o aplicativo principal utilizado para executar a Fase 2 através do terminal, destacando a seleção do arquivo de entrada.

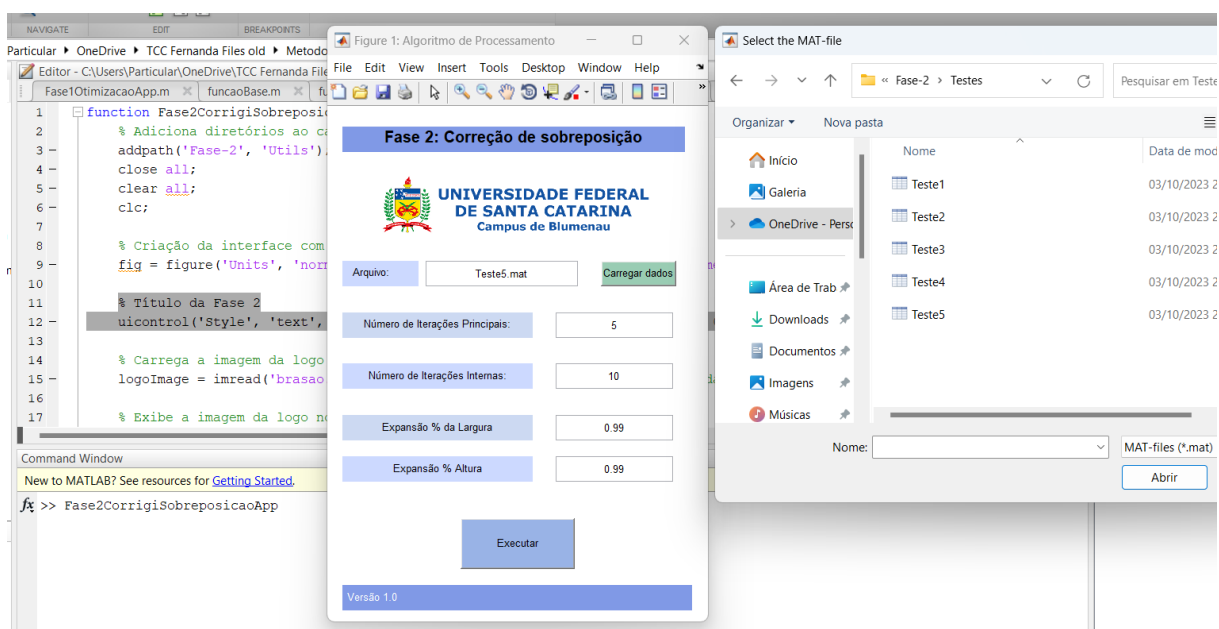


Figura 32 – Aplicativo principal para execução da Fase 2 executado pelo terminal evidenciando a escolha do arquivo de entrada.

Através da tela interativa, o usuário pode carregar arquivos contendo o conjunto de caixas que deseja corrigir quanto à sobreposição. Além disso, é possível ele definir o número de ciclos de movimentações principais desejados, assim como o número de movimentações

internas e as porcentagens que determinam o quanto o palete será expandido para facilitar a movimentação das caixas.

Dentre as etapas iniciais do algoritmo de correção de sobreposição das caixas, está a criação de um tensor virtual expandido para aumentar a flexibilidade na movimentação das caixas. Para realizar essa expansão, seleciona-se o menor valor de largura e o menor valor de comprimento das dimensões das caixas, e aplica-se individualmente o percentual escolhido.

Esses valores, após serem convertidos em números inteiros, são acrescidos às dimensões do palete. Essa estratégia evita mudanças muito drásticas na disposição das caixas em comparação com o resultado da sobreposição. Dessa forma, a expansão do palete é realizada de maneira controlada. Ilustrações desse processo de ampliação são apresentadas nas Figuras 33a e 33b, onde a região expandida é destacada em verde.

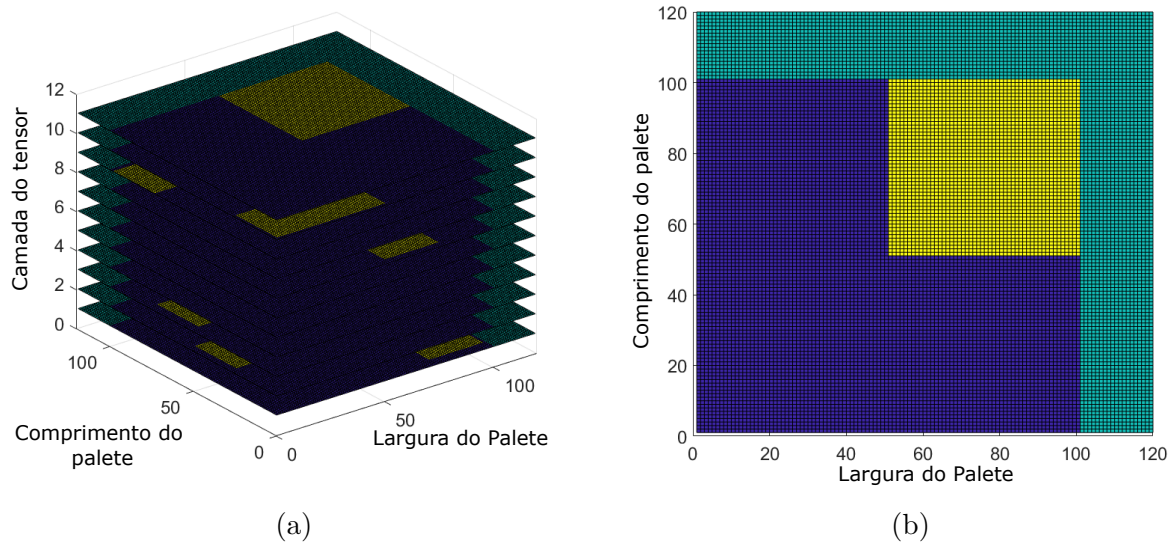


Figura 33 – Visualização da Distribuição de Objetos no Tensor Virtual Expandido: (a) Tensor virtual. (b) Décima primeira camada do tensor.

Posteriormente, o tensor virtualmente expandido alimenta uma função de correção de sobreposição, juntamente com o vetor de coordenadas de posição da caixa e a direção desejada de movimento. Estes parâmetros desempenham um papel crucial no processo de correção de sobreposição e nos movimentos das caixas.

Para realizar a movimentação das caixas virtuais no sistema, definiu-se quatro tipos de movimentos inspirados nas principais direções: **mB** para mover a caixa para baixo, **mD** para mover a caixa para a direita, **mC** para mover a caixa para cima e **mE** para mover a caixa para a esquerda. Esses comandos oferecem uma forma intuitiva e eficiente de manipular as caixas dentro do ambiente virtual.

A estratégia para corrigir sobreposições envolve ciclos de movimentos sequenciais, que seguem esta ordem: para baixo, para a direita, para cima e para a esquerda, e depois recomeçam. Esses ciclos se repetem continuamente, começando novamente com o

movimento para baixo, e assim por diante.

O usuário pode ajustar o “Número de interações principais” para determinar quantas vezes esse ciclo é executado. Além disso, há o “Número de interações internas”, que permite ao usuário especificar quantas vezes cada movimento dentro do ciclo deve ser repetido antes de avançar para o próximo movimento. Por exemplo, é possível configurar duas repetições internas para cada movimento: duas vezes para baixo, duas vezes para a direita, duas vezes para cima e duas vezes para esquerda.

Na Figuras 34a, 34b, 34c e 34d, são mostradas as disposições bidimensionais das caixas no palete após 10 ciclos de movimentação para baixo, para a direita, para cima e para a esquerda, respectivamente. Essas movimentações marcam o término do primeiro ciclo geral de movimentação, conforme ilustrado na Figura 34.

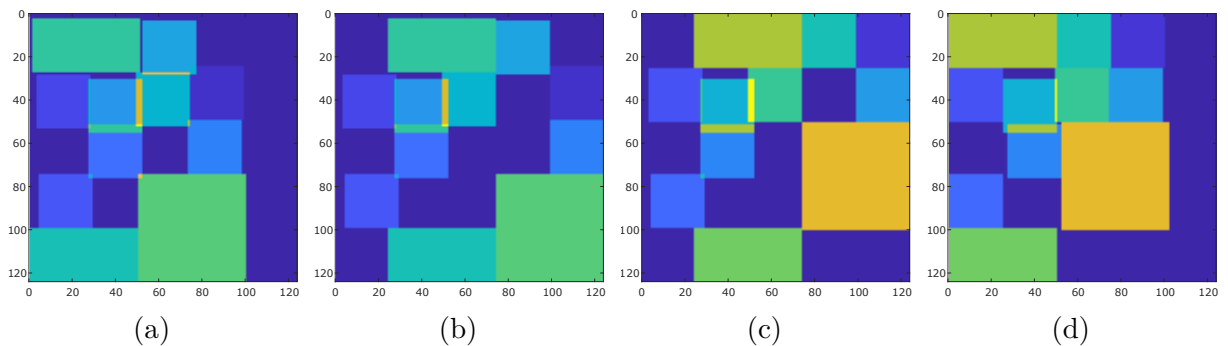


Figura 34 – Visualização do processo de movimentação do ciclo geral 1, após 10 ciclos: (a) Para baixo.. (b) Para a direita. (c) Para cima. (d) Para a esquerda.

Realizar a repetição do Ciclo 1 significa que o usuário escolheu executar “2 interações principais”. Na Figura 35, esta representa essa repetição do ciclo 1, ou seja, 10 ciclos de movimentações das caixas para baixo, seguidos por 10 ciclos de movimentações para a direita, mais 10 ciclos de movimentações para cima e, finalmente, 10 ciclos de movimentações para a esquerda

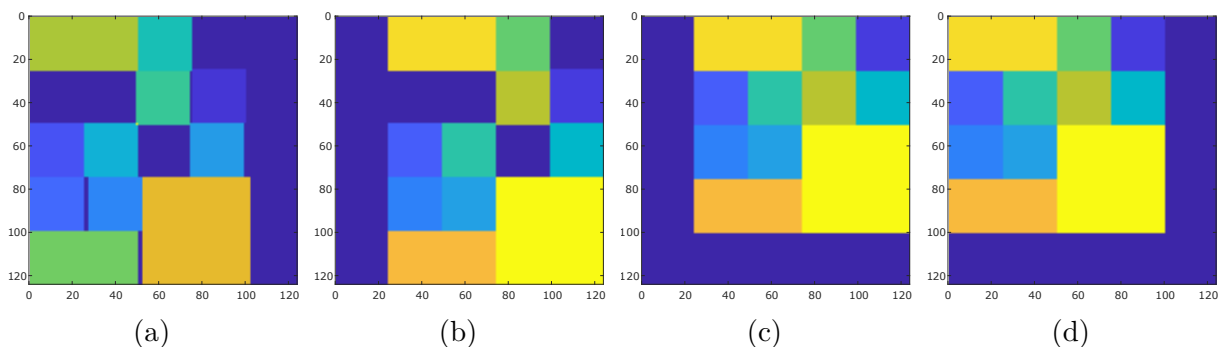


Figura 35 – Visualização do processo de movimentação do ciclo geral 2, que consiste em 10 ciclos: (a) Para baixo. (b) Para a direita. (c) Para cima. (d) Para a esquerda.

A ordem dos movimentos influencia diretamente a distribuição das caixas no palete. Por exemplo, a predominância de movimentos ascendentes (como mC) seguidos pelo movimento para a esquerda (como mE) resulta na concentração das caixas no canto superior

esquerdo do palete. Isso ocorre porque as caixas são inicialmente deslocadas para cima e depois para a esquerda, levando à acumulação no canto superior do palete conforme os movimentos são realizados sequencialmente.

Para realizar cada movimento individual, o algoritmo requer um tensor 3D com todas as informações sobre as caixas, um vetor com as coordenadas da disposição ótima das caixas, e a especificação do tipo de movimento (para cima, para baixo, para a esquerda ou para a direita). Ao final de cada movimento, o algoritmo gera duas saídas: uma representação visual das caixas no palete virtual expandido, conforme ilustrado na Figura 36a, e um vetor contendo as coordenadas dos vértices superiores esquerdos de cada caixa, representados pelos pontos amarelos na Figura 36b. Essas saídas são usadas iterativamente para realimentar parte da entrada do processo.

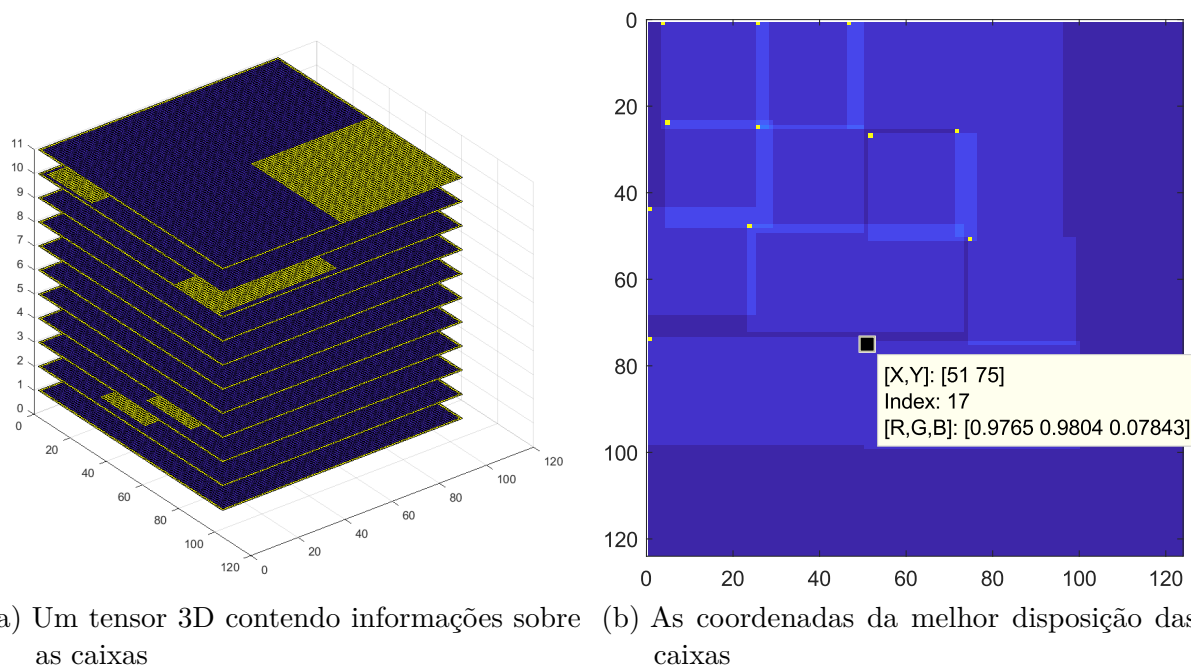


Figura 36 – Entradas: (a) Um tensor 3D contendo informações sobre as caixas, (b) As coordenadas da melhor disposição das caixas.

No processo de um movimento único (todas as caixas para a direita, por exemplo), cada caixa é movida de forma independente, uma de cada vez. Para determinar a sequência de qual caixa mover, é utilizado um critério que seleciona a caixa mais próxima da origem. Essa estratégia tem se mostrado eficiente. No entanto, para garantir que a mesma caixa não seja escolhida novamente na próxima iteração, ao final do processo de movimentação, a informação sobre a distância da caixa em relação à origem é atualizada para um valor muito grande, referenciado na metodologia proposta como “distância inválida”.

Uma sugestão para a criação desta “distância inválida” está na Equação (7). Nessa equação, L e C são as dimensões do palete na camada bidimensional, e ∞ é um número expressivamente grande, dentro do contexto do algoritmo.

$$\text{distância inválida} = \sqrt{(L + \infty)^2 + (C + \infty)^2} \quad (7)$$

A distância é considerada inválida, pois, na prática, está fora das dimensões do palete. Para ilustrar essa situação, observe a Figura 37. Nela, o “X” circulado indica o local de um ponto que gera a distância inválida. Os círculos numerados representam os pontos utilizados para calcular as distâncias válidas de cada caixa em relação à origem. O círculo que contém o número 10 é o que está praticamente sobre a origem.

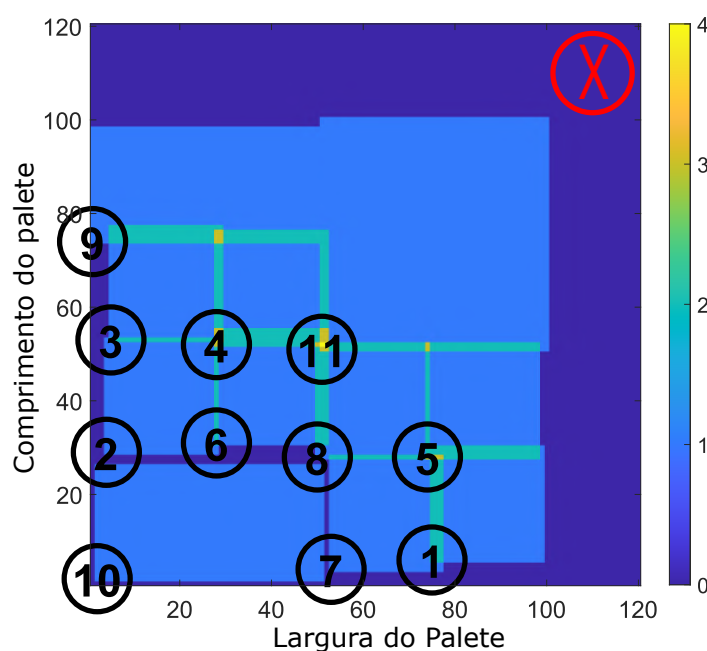


Figura 37 – Círculo que contém o “X” representa o local da “Distância inválida”.

A caixa selecionada é movida até os limites físicos do paleta ou até que a porção que se pretende movimentar não origina uma nova sobreposição, o que ocorrer antes. Para fazer as validações destes critérios constrói-se projeções do movimento.

Essas projeções são baseadas nas coordenadas especificadas do objeto virtual que representa a caixa, e no tipo de movimento desejado.

Na Figura 38, é possível observar as projeções para cada exemplo de movimento: as representações verdes indicam as matrizes onde a operação retira elementos da representação bidimensional da distribuição das caixas, enquanto as vermelhas indicam onde a operação coloca elementos.

O processo de movimentação compara a projeção com a representação bidimensional de todas as caixas para verificar se a caixa pode ser ou não movida. É importante destacar ao leitor que apesar das projeções de cada movimento estarem agrupadas em pares na Figuras 38, na prática apenas um tipo e para uma caixa realiza-se a construção projeção por vez.

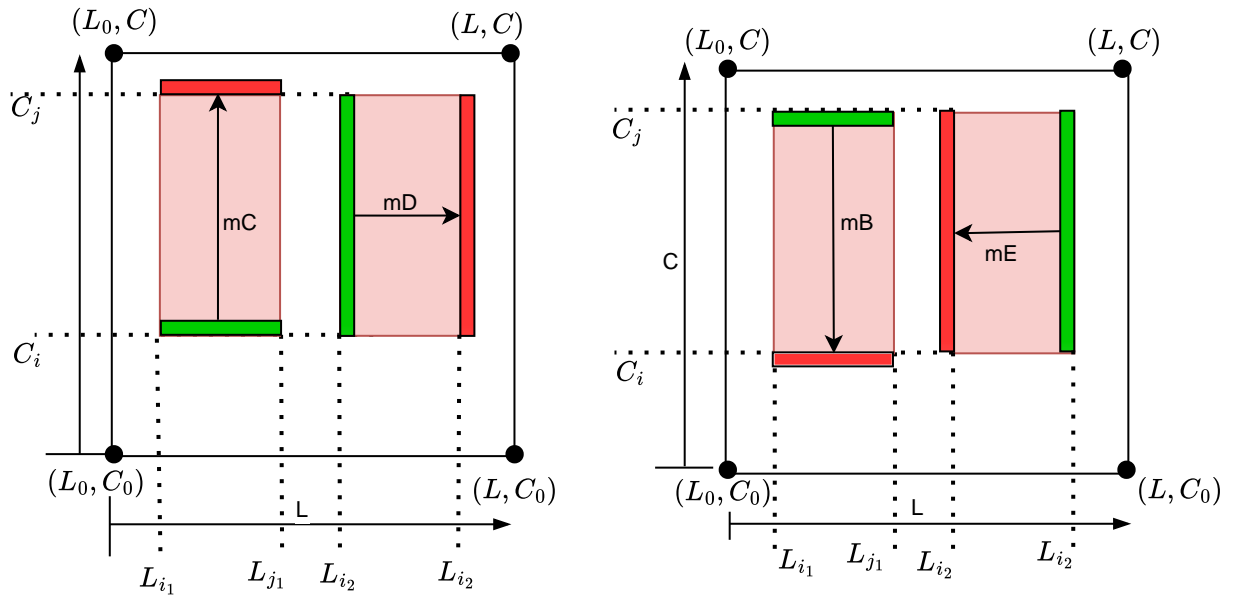


Figura 38 – Representação das posições das projeções.

Na Figura 39a, na qual a seta sinaliza que o movimento é para baixo, observa-se que, caso a movimentação não cause uma nova sobreposição, a caixa poderá ser movida. Enquanto na Figura 39b há a ilustração de um caso onde a movimentação não é possível de ser realizada, a região circulada indica onde a nova sobreposição iria ocorrer caso o movimento fosse efetivado.

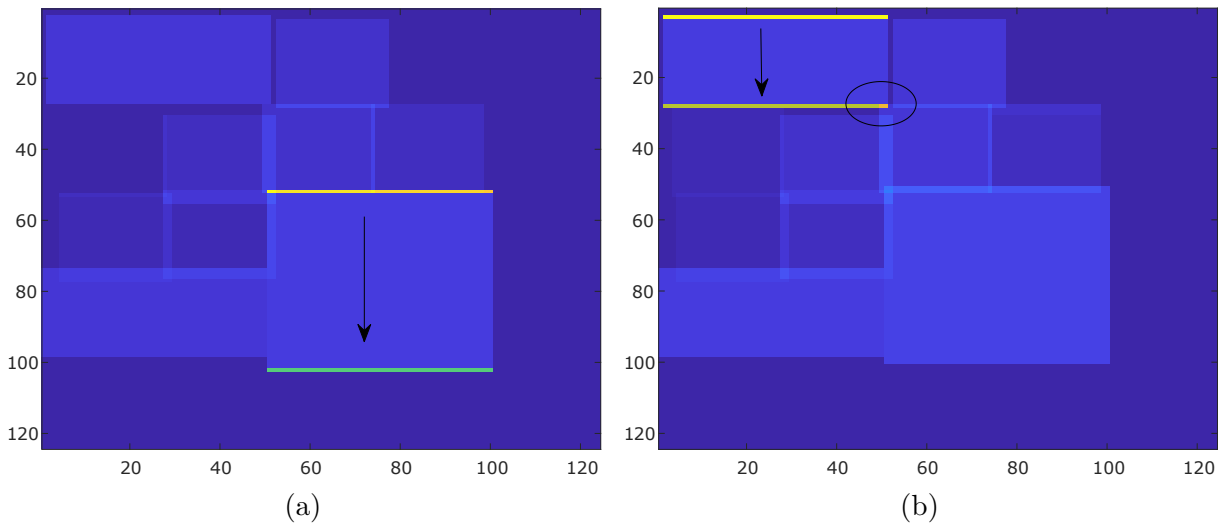


Figura 39 – Projeções: (a) Realizada, sem nova sobreposição (b) Não realizada, causando sobreposição.

Na Figura 40, observa-se um ciclo de movimentação das caixas para baixo, com alguns saltos para facilitar a representação. Os pequenos retângulos amarelos representam a projeção de onde os elementos da caixa que está sendo movimentada deixariam de estar presentes. Os pequenos retângulos verdes, por sua vez, indicam as projeções que seriam preenchidas caso a movimentação da caixa fosse realizada. Todos os quadros apresentados na Figura 40 mostram os momentos em que as caixas não podem mais ser movimentadas

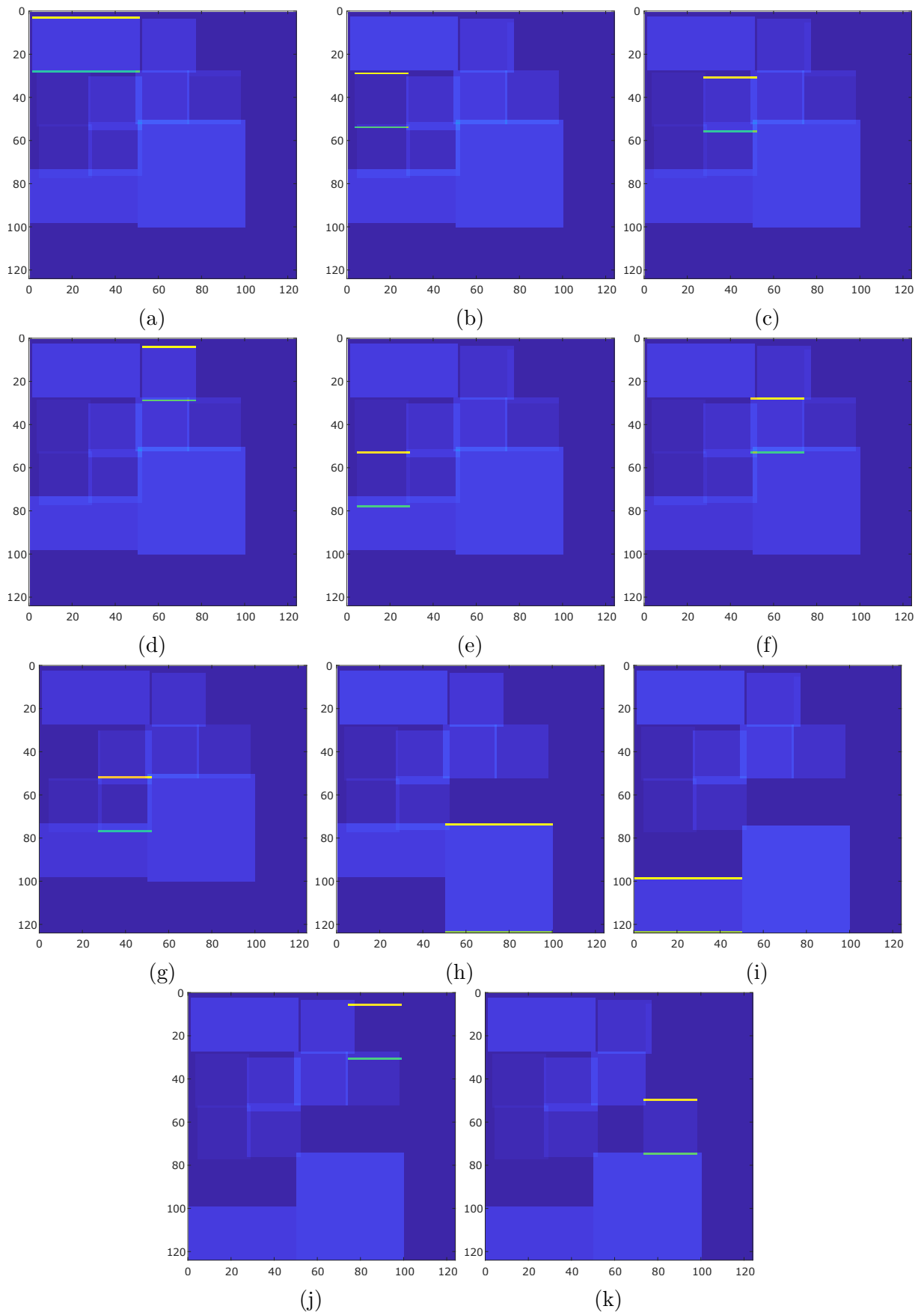


Figura 40 – Um ciclo completo de movimentação para baixo.

dentro do ciclo, seja porque a projeção, ao ser comparada com as caixas no palete, indica que a região já está ocupada por outra caixa, seja porque excede o limite físico do palete.

Por exemplo, observe que o primeiro objeto a ser movido, mostrado na Figura 40a, se desloca apenas pequenas unidades, pois ao tentar mover-se para baixo, gera uma nova sobreposição. Na Figura 40b, em comparação com a Figura 40a, é possível perceber que mover o objeto na direção desejada não é viável, pois a porção projetada resulta em sobreposição. O mesmo problema ocorre nas Figuras 40c até 40g.

Na Figura 40h é mostrado o momento em que a caixa em destaque não pode mais ser movida, pois a projeção indica que o limite do palete foi atingido. Comparando com a Figura 40g, é possível ver a caixa no instante anterior ao início de seu movimento para baixo. Esse mesmo processo pode ser verificado para a caixa, cujo instante inicial está na Figura 40h e o final na Figura 40i.

Em conclusão, o algoritmo de movimentação de caixas virtualizadas apresentado nesta seção demonstra um processo sistemático e iterativo para corrigir sobreposições e otimizar a distribuição das caixas no palete. Utilizando quatro tipos principais de movimentos – para baixo, para a direita, para cima e para a esquerda – e controlando a quantidade de interações principais e internas, o sistema pode ajustar a posição das caixas de maneira eficiente. As representações visuais e projeções detalhadas garantem a precisão dos movimentos, evitando novas sobreposições e respeitando os limites físicos do palete. A estratégia de atualizar a “distância inválida” impede que a mesma caixa seja selecionada repetidamente, otimizando o desempenho do algoritmo. Dessa forma, o processo descrito aprimora a organização das caixas no espaço virtual.

4 RESULTADOS E DISCUSSÕES

Este capítulo apresenta os resultados obtidos a partir de cinco estudos de caso detalhados, que exploram a otimização do posicionamento de caixas em um espaço bidimensional. Cada estudo de caso é discutido em termos de suas configurações específicas de caixas, incluindo dimensões e orientações, bem como os métodos utilizados para resolver desafios como sobreposição e eficiência de espaço. Os resultados são analisados criticamente, destacando tanto a eficácia da metodologia proposta quanto as lições aprendidas durante o processo de experimentação. Essa análise proporciona *insights* valiosos para a aplicação prática da técnica em sistemas robóticos de paletização, demonstrando sua capacidade de lidar com uma variedade de cenários logísticos e industriais complexos.

4.1 RESULTADOS

Nesta seção, são apresentados os resultados de cinco estudos de caso. As dimensões do palete foram definidas em 100 cm por 100 cm e mantidas constantes para todos os casos. Foram utilizados três tipos diferentes de caixas: **Caixa 1**, com dimensões de 25 cm por 25 cm; **Caixa 2**, com dimensões de 25 cm por 50 cm; e **Caixa 3**, com dimensões de 50 cm por 50 cm. Todas as caixas têm a mesma altura.

O primeiro estudo de caso é o mais simples, onde as posições de 16 caixas do tipo 1 são otimizadas. Inicialmente, as dimensões dessas caixas são reduzidas em 80%, tornando-se 20 cm por 20 cm. Em seguida, o algoritmo de computação evolutiva é utilizado para encontrar as posições ideais para esses 16 elementos sem sobreposição. Com esse resultado preliminar (veja a Figura 41a), as caixas são redimensionadas de 80% para 100%, o que gera sobreposição, como pode ser visto na Figura 41b. Para corrigir essa sobreposição, utiliza-se o algoritmo de correção de sobreposição, obtendo o resultado final, conforme mostrado na Figura 41c.

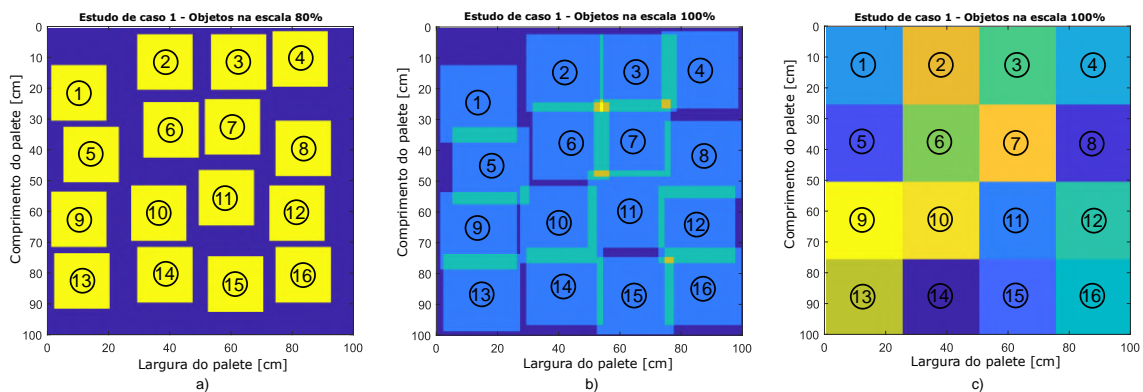


Figura 41 – Estudo de caso 1: 16 caixas do tipo 1.

No segundo estudo de caso, foram adicionadas caixas do tipo 2, as quais foram obrigatoriamente orientadas na vertical. Neste caso, 8 caixas do tipo 1 foram utilizadas

juntamente com 4 caixas do tipo 2 na orientação vertical. Para este caso específico, apenas as caixas do tipo 1 passaram por um redimensionamento de 80%, enquanto as caixas do tipo 2 mantiveram 100% de suas dimensões originais. Os resultados de posicionamento otimizado, alcançados sem sobreposições, podem ser observados na Figura 42a. Após o processo de otimização, o processo de redimensionamento é executado, inevitavelmente causando sobreposição, como mostra a Figura 42b. Por fim, o processo de correção de sobreposição é aplicado e a sobreposição é resolvida. A disposição final é mostrada na Figura 42c.

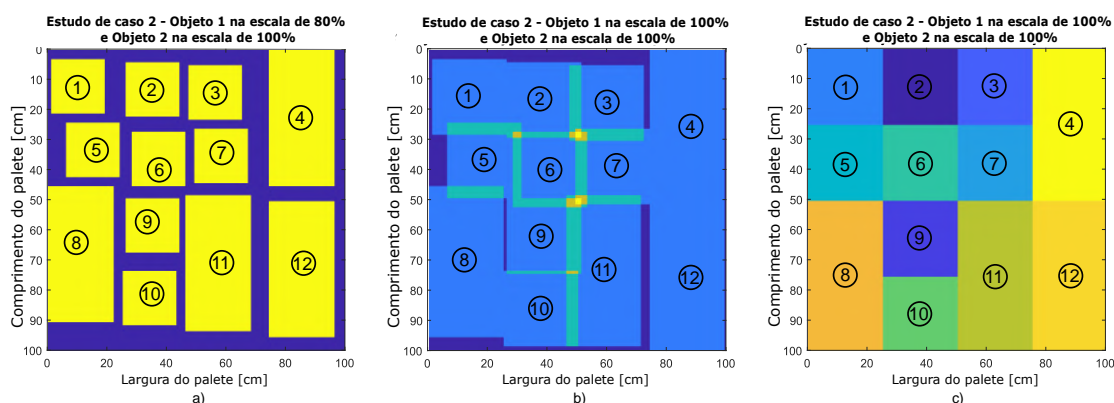


Figura 42 – Estudo de caso 2: 12 caixas - 8 caixas do tipo 1 e 4 caixas do tipo 2 em orientação horizontal.

O terceiro estudo de caso é uma variação do segundo, onde o mesmo número de caixas é considerado, mas agora as 4 caixas do tipo 2 são forçadas a serem orientadas horizontalmente. O resultado da otimização pode ser observado na Figura 43a. Em seguida, o resultado apresentado na Figura 43a passa pelo processo de redimensionamento, gerando as sobreposições apresentadas na Figura 43b, que são corrigidas pelo algoritmo de correção de sobreposição, resultando no resultado final apresentado na Figura 43c.

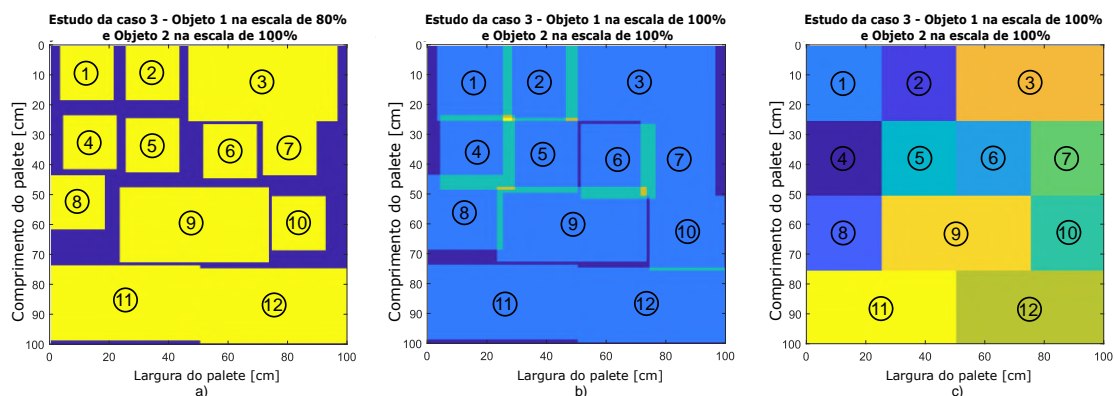


Figura 43 – Estudo de caso 3: 12 caixas - 8 caixas do tipo 1 e 4 caixas do tipo 2 em orientação vertical.

No quarto estudo de caso, é considerada uma variação combinando as condições do segundo caso com o terceiro caso. Nesse caso, foram utilizadas 8 caixas do tipo 1 juntamente

com 4 caixas do tipo 2, mas desta vez o algoritmo escolhe livremente a orientação ótima para as caixas do tipo 2. Aqui, as caixas do tipo 1 passam por um redimensionamento de 80%, enquanto as caixas do tipo 2 passam por um redimensionamento de 90%. A otimização inicial é apresentada na Figura 44a. Após a aplicação do processo de redimensionamento, ocorre sobreposição, como pode ser observado na Figura 44b. A sobreposição é facilmente corrigida pelo algoritmo de correção de sobreposição, como apresentado na Figura 44c.

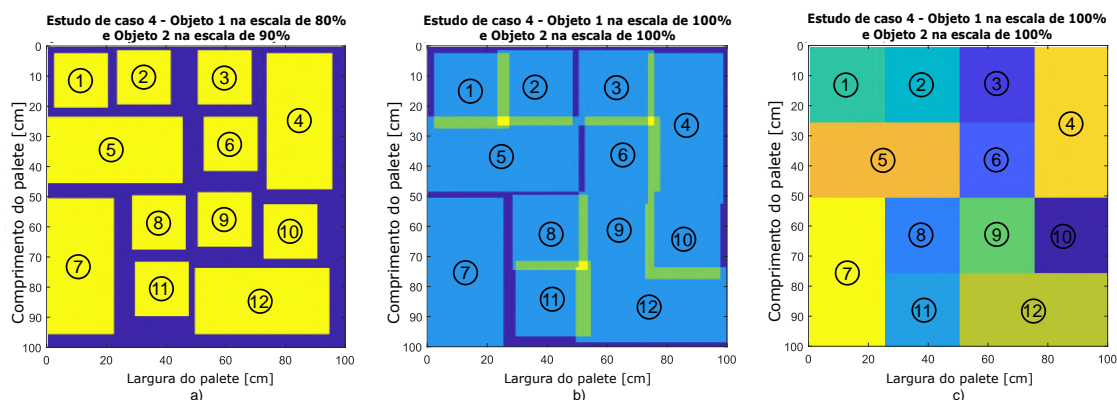


Figura 44 – Estudo de caso 4: 12 caixas - 8 do tipo 1, 2 do tipo 2 horizontais e 2 do tipo 2 verticais.

O último estudo de caso utiliza 8 caixas do tipo 1, 2 caixas do tipo 2 e 1 caixa do tipo 3. Aqui, as caixas do tipo 1 passam por um redimensionamento de 80% e as caixas do tipo 2 e 3 passam por um redimensionamento de 90% antes do processo de otimização. Na Figura 45a, observamos a solução ideal para este caso. A sobreposição ocorre quando o redimensionamento é aplicado e pode ser vista na Figura 45b. Finalmente, os resultados obtidos a partir da correção de sobreposição são apresentados na Figura 45c.

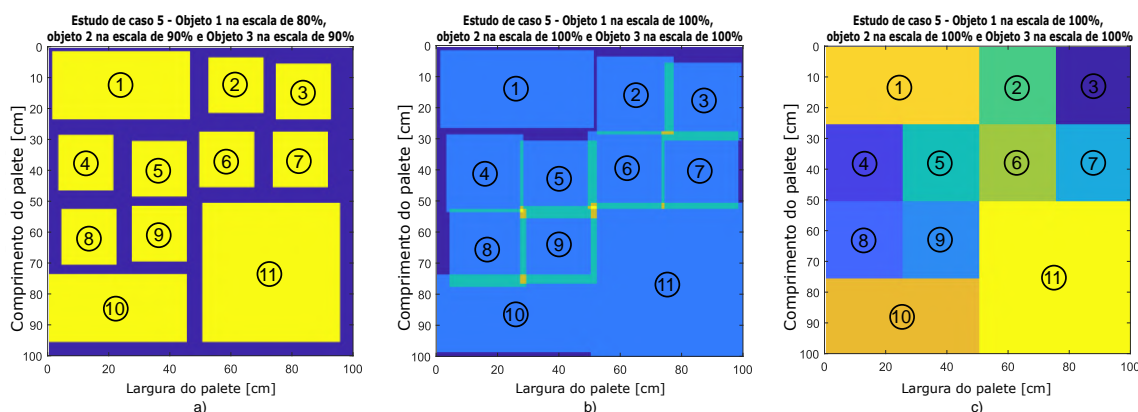


Figura 45 – Estudo de caso 5: 11 caixas - 8 do tipo 1, 2 do tipo 2 horizontais e 1 do tipo 3.

Para finalizar esta seção, o tempo gasto em cada simulação é apresentado na Tabela 4. Esta tabela apresenta individualmente o tempo necessário para a aplicação do algoritmo de otimização (Estágio 1), para o algoritmo de correção de sobreposição (Estágio 2) e, finalmente, apresenta o tempo total gasto (Estágio 1 + Estágio 2).

Tabela 4 – Resumo de Tempo de Resultados.

Estudos de de caso	Tempo de simulação para a Etapa 1 (s)	Tempo de simulação para a Etapa 2 (s)	Tempo total total (s)
Estudo de caso 1	14,20	3,36	17,57
Estudo de caso 2	19,78	3,31	23,09
Estudo de caso 3	21,78	3,14	24,92
Estudo de caso 4	23,50	3,12	26,62
Estudo de caso 5	24,21	2,87	27,09

4.1.1 Sistema Robotizado

Os resultados obtidos com a metodologia proposta podem servir como parâmetros de entrada para o modelo de cinemática de posição de um manipulador robótico, possibilitando a colocação precisa das caixas sobre um palete. Essa abordagem garante que as posições finais determinadas sejam utilizadas de forma eficiente, otimizando o processo de paletização.

Uma maneira simples e eficaz de validar esses resultados é através do uso de simuladores como CoppeliaSim, ROS e outros. Esses simuladores são amplamente reconhecidos pela comunidade científica e permitem testar os algoritmos em ambientes virtuais que refletem cenários do mundo real, sem a necessidade de protótipos físicos. Isso não só facilita a validação, mas também reduz custos e acelera o desenvolvimento.

Para fins representativos, na Figura 46 é apresentada uma imagem no CoppeliaSim que exemplifica essa situação. No entanto, é importante destacar que esta imagem não representa uma simulação de um caso real, mas sim uma ilustração da potencial aplicação dos resultados obtidos com a metodologia proposta.

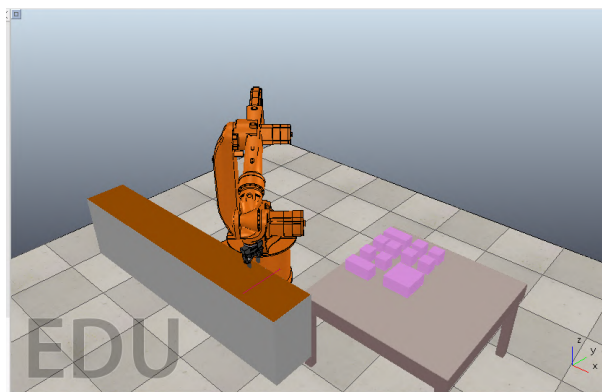


Figura 46 – Ambiente de Simulação de Robô Industrial.

4.2 DISCUSSÕES

Para introduzir informações sobre métricas de validação em uma metodologia que resolve o problema de carregamento de paletes, é importante destacar que a análise realizada foi temporária, baseada em um conjunto conhecido de caixas. No contexto de

programação linear e otimização, uma “instância” refere-se a uma versão específica do problema com seus próprios dados e parâmetros. A validação de desempenho de algoritmos frequentemente utiliza dois critérios principais: medição da área desperdiçada e tempo de processamento. Por exemplo, Arun Prasad e Krishnakumar (2021) utilizaram a medição por área desperdiçada em seu estudo. Quanto ao tempo de processamento, estudos como os de Morales e Morabito (1997), Subir Bhattacharya, Roy e Sumita Bhattacharya (1998), Mascarenhas (2005) e Ribeiro e Lorena (2007) são notáveis. Em particular, Morales e Morabito (1997) forneceu uma base comparativa indicando que o tempo médio para o método de *LagClus* (*Lagrangian Clustering*), uma técnica baseada em métodos Lagrangianos utilizada na resolução de problemas de otimização com restrições, foi de 403,4 segundos em média, para 20 casos de estudo realizados.

Nos experimentos realizados, a variedade de caixas simuladas variou de 11 a 16. Todos os casos estudados foram resolvidos dentro de um intervalo de tempo aceitável, atingindo com sucesso o objetivo de determinar com precisão as posições de um conjunto de caixas para preencher de forma otimizada o espaço do palete sem sobreposição. Esses resultados demonstram a eficácia da metodologia proposta, que mostrou ser capaz de lidar com diferentes configurações e cenários de carga.

Os experimentos evidenciam a robustez da solução diante da diversidade dos elementos. A metodologia manteve sua eficiência mesmo quando as caixas apresentavam disparidades dimensionais significativas, indicando que o algoritmo é altamente adaptável. Isso é crucial para aplicações práticas, onde a variedade de dimensões e formas das caixas é comum, e a capacidade de adaptação do algoritmo garante que a solução seja eficaz em uma ampla gama de situações.

No experimento com o maior número de caixas, todas as unidades possuíam dimensões idênticas, ocupando completamente o espaço disponível no palete. Notavelmente, essa configuração resultou no menor tempo total de processamento. Isso destaca a eficácia da metodologia em situações onde as caixas são uniformes em tamanho e forma, permitindo um preenchimento mais direto e eficiente do espaço disponível. Este cenário ilustra como a uniformidade dos elementos pode ser explorada para otimizar ainda mais o tempo de processamento.

Por outro lado, o cenário com menos elementos, onde uma caixa abrangia um quarto da área total, apresentou um tempo total maior. Isso ocorreu devido à ausência de redução de tamanho para essa caixa durante a fase inicial. A fase inicial não conseguiu otimizar o espaço eficientemente devido ao grande tamanho da caixa em relação ao espaço disponível. No entanto, essa configuração registrou a menor duração na segunda fase, destacando a importância desta fase na redução do tempo total e na flexibilidade da solução inicial. Este resultado sublinha a importância da segunda fase para ajustar a solução inicial e melhorar a eficiência geral do algoritmo.

Os casos 3 e 4, que envolviam combinações mistas de caixas de diferentes tamanhos,

registraram durações semelhantes. Isso demonstra que a metodologia pode manter uma performance consistente mesmo quando lida com variações significativas nas dimensões dos itens. A capacidade de o algoritmo lidar com tais variações sem perda substancial de eficiência é uma prova de sua robustez e versatilidade. Essa consistência é vital para aplicações em ambientes reais, onde a variabilidade nas dimensões das caixas é uma constante.

A divisão da metodologia proposta em duas fases mostrou-se particularmente eficaz. A primeira fase facilita a inclusão de variáveis como orientação e posição, permitindo uma solução mais abrangente que pode incluir novas variáveis no futuro, como peso do objeto e centro de massa. O objetivo desta fase é aumentar a precisão do ajuste final e garantir a resolubilidade do problema. Ao lidar com essas variáveis adicionais, a metodologia pode ser adaptada para aplicações ainda mais complexas, como sistemas robóticos que precisam considerar a estabilidade da carga. Essa flexibilidade para incorporar novas variáveis de decisão aumenta a aplicabilidade do algoritmo em diversos cenários industriais.

A segunda fase constitui um processo simples e eficiente que contribui significativamente para a redução do tempo necessário para alcançar o objetivo final. Esta fase aproveita os resultados da primeira fase para ajustar finamente a posição das caixas, garantindo que o espaço do palete seja utilizado da maneira mais eficiente possível. Essa abordagem em duas fases não só melhora a precisão do posicionamento das caixas, mas também reduz o tempo total de processamento, combinando os melhores aspectos de ambas as etapas. A eficiência temporal é um aspecto crítico, especialmente em operações logísticas onde o tempo de processamento pode impactar diretamente na produtividade e nos custos operacionais.

Além disso, a integração de algoritmos de evolução diferencial com algoritmos adicionais desenvolvidos para evitar a superposição provou ser uma estratégia eficiente. A adaptação dessas técnicas assegurou uma solução robusta e eficiente para o problema de carregamento de paletes em um plano bidimensional (2D-PCP). Essa abordagem demonstrou ser flexível, permitindo a inclusão de novas variáveis de decisão e mantendo a eficiência do algoritmo, mesmo quando confrontado com a variabilidade nas dimensões das caixas. A flexibilidade do algoritmo é um ponto crucial, pois permite que ele seja adaptado a diferentes tipos de problemas de carregamento, potencialmente ampliando seu uso para outras aplicações logísticas e industriais.

Os resultados alcançados mostram que a metodologia proposta não apenas atinge seus objetivos iniciais, mas também abre caminho para futuras melhorias e adaptações. Para aprimorar ainda mais o processo de posicionamento de caixas, recomenda-se a introdução de novas variáveis de decisão, como o centro de massa e o peso das caixas. Além disso, um aumento na diversidade dos tipos e tamanhos de caixas considerados, incorporando cenários com dimensões mais realistas, pode ampliar a aplicabilidade do algoritmo e fornecer *insights* adicionais. Esses refinamentos são essenciais para desenvolver

uma solução que não apenas funcione em ambientes controlados, mas também em situações do mundo real, onde as variáveis são mais complexas e imprevisíveis.

Em suma, a metodologia proposta cumpriu seu objetivo de determinar o posicionamento preciso de caixas dentro de um espaço bidimensional sem sobreposição para aplicação em um sistema robótico. A pesquisa destacou a flexibilidade e adaptabilidade do algoritmo, juntamente com as eficiências de tempo derivadas da divisão do processo. Embora existam possíveis melhorias para a metodologia proposta, o estudo atual estabelece uma base sólida para investigações subsequentes, facilitando a aplicação do algoritmo em cenários mais complexos e realistas. A metodologia mostrou-se robusta e eficiente, pronta para ser expandida e aprimorada conforme novas necessidades surgem no campo da logística e da robótica industrial.

5 CONCLUSÃO

Com base nos estudos e desenvolvimentos apresentados neste trabalho, evidencia-se que a metodologia proposta para resolver o problema de carregamento de paletes em um plano bidimensional (2D-PCP) demonstra características notavelmente inovadoras. Ela integra de maneira pioneira a técnica avançada de otimização conhecida como algoritmo Storn and Price com métodos heurísticos, aproveitando características que podem ser agrupadas em matrizes e traduzindo o sistema físico em um sistema computacional descrito por tensores. Essa abordagem não só busca maximizar o uso do espaço do palete, mas também visa evitar sobreposições, estabelecendo um novo padrão para eficiência e precisão na logística industrial.

Além disso, o foco na progressão da pesquisa visa preparar a metodologia para futuras aplicações práticas, promovendo avanços significativos na automação e eficiência dos processos industriais relacionados ao carregamento de paletes.

Com base nos estudos e desenvolvimentos apresentados neste trabalho, conclui-se que a metodologia proposta para resolver o problema de carregamento de paletes em um plano bidimensional (2D-PCP) atingiu seus objetivos de forma eficaz e robusta. Através de uma abordagem cuidadosamente planejada que envolveu a caracterização detalhada do sistema, a formulação de uma função objetivo precisa e a implementação de algoritmos de computação evolutiva diferencial integrados a métodos para evitar sobreposição, foi possível não apenas maximizar o uso do espaço do palete, mas também evitar sobreposições. Os resultados obtidos nas simulações e experimentos validaram a eficiência da metodologia, proporcionando uma base sólida para contribuições significativas na área de logística e automação industrial.

Notavelmente, o objetivo de desenvolver uma metodologia para a resolução ótima do problema de carregamento de paletes em um plano bidimensional (2D-PCP) foi atendido através de uma série de etapas estratégicas e bem planejadas que garantiram a precisão e eficiência necessárias para a validação da metodologia proposta.

Primeiramente, o objetivo de caracterizar um sistema 2D-PCP através do uso de variáveis fundamentais para a descrição do sistema de paletes foi atendido de maneira meticulosa e detalhada. Esta caracterização envolveu a identificação e definição de variáveis essenciais que descrevem com precisão o sistema de paletes e permitiram o desenvolvimento de uma metodologia robusta e eficiente para o problema de carregamento de paletes em um plano bidimensional.

Especificamente, as variáveis fundamentais como orientação, posição das caixas, dimensões das caixas e o espaço disponível no palete foram cuidadosamente definidas. A orientação refere-se aos indicativos de horizontal e vertical, enquanto a posição se refere às coordenadas específicas de cada caixa dentro do plano bidimensional do palete. As dimensões das caixas, incluindo largura e comprimento, foram essenciais para calcular o

espaço ocupado e para garantir que as caixas pudessem ser acomodadas sem sobreposição. O espaço disponível no palete foi mapeado para entender a área total que poderia ser utilizada para o carregamento.

Além disso, foi considerada a forma e a disposição geométrica do palete sendo retangular, garantindo que a metodologia pudesse lidar com diferentes configurações de paletes e tamanhos variados. Esta caracterização detalhada permitiu a criação de um modelo preciso do problema 2D-PCP, facilitando o desenvolvimento de algoritmos que pudessem explorar eficientemente o espaço disponível e encontrar soluções ótimas para o arranjo das caixas.

Seguindo as etapas necessárias para alcançar o objetivo final, a definição da função objetivo foi um passo fundamental no processo de otimização do espaço do palete. O objetivo de determinar uma função objetivo para maximizar o uso do espaço disponível no palete, considerando um conjunto conhecido e caracterizado de caixas, foi completamente alcançado através de um rigoroso processo de formulação e refinamento. Esta função foi essencial para guiar o posicionamento das caixas, cujas dimensões foram reduzidas no plano bidimensional do palete, visando otimizar a utilização do espaço sem gerar sobreposições.

A função objetivo foi elaborada com o claro propósito de maximizar o uso do espaço, levando em consideração as dimensões das caixas, a área total disponível no palete e as restrições de sobreposição entre as unidades. Essa formulação não só facilitou a comparação entre diferentes configurações de arranjo das caixas, mas também assegurou que cada solução final encontrada pelo algoritmo de otimização fosse eficiente e praticamente aplicável.

Inicialmente, a função objetivo foi concebida levando em consideração variáveis cruciais como as dimensões das caixas, a área total do palete e as restrições de posicionamento. As dimensões das caixas foram meticulosamente integradas à função para calcular a área ocupada por cada caixa e para determinar como elas poderiam ser acomodadas de forma mais eficiente possível. Isso permitiu maximizar não apenas a quantidade de caixas que poderiam ser carregadas, mas também a forma como eram dispostas para otimizar o espaço.

Durante o desenvolvimento da função objetivo, foram realizados ajustes iterativos com base em testes experimentais e simulações computacionais. Esses testes verificaram a eficácia da função em diferentes cenários e condições. A validação experimental permitiu ajustes finos na função objetivo para melhorar sua precisão e adaptabilidade a diferentes configurações de caixas e paletes.

O objetivo de resolver a função objetivo através do uso de um algoritmo de computação evolutiva diferencial foi alcançado com sucesso, proporcionando uma abordagem eficaz e robusta para encontrar soluções ótimas no problema de carregamento de paletes em um plano bidimensional (2D-PCP).

A escolha de um algoritmo de computação evolutiva diferencial foi motivada pela

sua capacidade de lidar com problemas complexos de otimização que envolvem múltiplas variáveis e restrições, como é o caso do 2D-PCP. Este tipo de algoritmo baseia-se em conceitos inspirados na evolução natural, onde uma população de soluções candidatas é iterativamente refinada ao longo de várias gerações para convergir para uma solução ótima ou próxima dela.

No contexto do carregamento de paletes, o algoritmo de computação evolutiva diferencial foi aplicado para explorar uma ampla gama de configurações possíveis das caixas no palete, considerando variáveis como posição, orientação e tamanho das caixas. Durante cada iteração do algoritmo, as soluções candidatas foram avaliadas com base na função objetivo previamente definida, que visava maximizar o uso do espaço do palete sem gerar sobreposições entre as caixas.

O objetivo de integrar os algoritmos de evolução diferencial com algoritmos adicionais desenvolvidos para evitar superposição foi alcançado de maneira eficiente e eficaz, fortalecendo a metodologia para resolver o problema de carregamento de paletes em um plano bidimensional (2D-PCP).

A integração dos algoritmos de evolução diferencial com os algoritmos adicionais desenvolvidos para evitar superposições foi um marco significativo no desenvolvimento da metodologia. Durante o processo de otimização, cada configuração de caixas gerada pelo algoritmo foi submetida a verificações rigorosas para detectar possíveis interseções ou sobreposições entre as unidades. Caso alguma sobreposição fosse identificada, os algoritmos adicionais aplicavam ajustes automáticos nas posições das caixas, garantindo que todas as soluções finais fossem livres de conflitos.

A validação do método proposto, através da apresentação de gráficos e simulações de casos de estudo do problema de carregamento de paletes em um plano bidimensional (2D-PCP), demonstrou de forma conclusiva a eficácia e robustez da metodologia desenvolvida. Os resultados obtidos nas simulações confirmaram que o algoritmo de evolução diferencial, integrado aos algoritmos adicionais para evitar superposição, foi capaz de encontrar soluções ótimas que maximizam o uso do espaço do palete enquanto evitam a sobreposição.

No geral, a metodologia demonstrou flexibilidade e adaptabilidade, permitindo a introdução de novas variáveis de decisão e gerando resultados preliminares de forma eficiente. A divisão em duas etapas contribuiu para uma redução significativa de tempo, mantendo a precisão e a otimização desejadas.

A validação experimental da metodologia foi conduzida através de testes extensivos utilizando uma ampla gama de cenários de carregamento de paletes. Esses testes não apenas confirmaram a eficácia da metodologia na prática, mas também proporcionaram insights valiosos para ajustes finos na implementação dos algoritmos. Os resultados obtidos demonstraram consistentemente que a metodologia era robusta e eficiente, capaz de resolver com sucesso o desafiador problema do 2D-PCP em condições diversas.

Em resumo, a metodologia desenvolvida não apenas alcançou, mas superou os objetivos inicialmente estabelecidos. A combinação de uma caracterização detalhada do sistema 2D-PCP, uma função objetivo precisa, o uso de algoritmos de evolução diferencial e a integração de métodos para evitar sobreposição resultou em uma solução robusta e eficiente para o problema de carregamento de paletes. A validação experimental confirmou não apenas a aplicabilidade prática da metodologia, mas também seu potencial para otimizar processos industriais complexos, promovendo ganhos significativos em eficiência operacional e redução de custos. A continuidade desta pesquisa e a implementação de melhorias sugeridas poderão abrir caminho para avanços ainda mais significativos na área de logística e automação industrial.

5.0.1 Trabalhos Futuros

Apesar dos resultados bem-sucedidos, a expansão potencial do estudo merece consideração. Para aprimorar ainda mais o processo de posicionamento de caixas, propõe-se a introdução de novas variáveis de decisão, como o centro de massa e o peso das caixas. Além disso, sugere-se um aumento na diversidade dos tipos e tamanhos de caixas considerados, incorporando cenários com dimensões realistas. Esses refinamentos podem ampliar a aplicabilidade do algoritmo e fornecer *insights* adicionais.

5.0.2 Publicação realizada

Dentro do escopo deste Trabalho de Conclusão de Curso (TCC), destaca-se a relevância de um artigo científico anteriormente desenvolvido e publicado. O artigo intitulado "Proposta de uma metodologia para otimização do problema 2-DPLP" discute a melhoria na distribuição de itens em paletes bidimensionais, utilizando algoritmos de computação evolutiva e uma abordagem algorítmica para resolver problemas de sobreposição (FERREIRA *et al.*, 2023). Este estudo foi apresentado durante o 27^o Congresso Internacional de Engenharia Mecânica (COBEM 2023), organizado pela Sociedade Brasileira de Ciências Mecânicas e Engenharia em Florianópolis. No ANEXO A - Publicação Realizada, encontra-se a primeira página do referido artigo.

REFERÊNCIAS

- AHN, S.; PARK, C.; YOON, K. Um algoritmo de ramificação e limitação de melhoramento para o problema de carregamento de paletes utilizando uma estrutura em escada. **Expert Systems with Applications**, v. 42, n. 21, p. 7676–7683, 2015. ISSN 0957-4174.
- ALONSO, M.T.; ALVAREZ-VALDES, R.; IORI, M.; PARREÑO, F.; TAMARIT, J.M. Modelos matemáticos para problemas de carregamento de múltiplos contêineres. **Omega**, v. 66, p. 106–117, 2017. ISSN 0305-0483.
- ALVES, Andressa Schneider; AYMONE, José Luís Farinatti. Algoritmo Heurístico para o Problema do Corte Bidimensional de Peças Irregulares na Indústria do Vestuário. **Podes Revista Eletrônica Pesquisa Operacional Para O Desenvolvimento**, Rio de Janeiro, v. 6, n. 3, p. 434–450, set. 2014.
- ARUN PRASAD, S.; KRISHNAKUMAR, P. Heurísticas de blocos de ordem superior para problemas de carregamento de paletes 2D com múltiplas entradas de caixas. **Materials Today: Proceedings**, v. 46, p. 4625–4633, 2021. Conferência Internacional sobre Avanços em Materiais e Aplicações de Manufatura. ISSN 2214-7853.
- BECCENERI, José; RAMOS, Fernando; CAMPOS VELHO, Haroldo; SILVA, José; LORENA, Luiz; VIJAYKUMAR, Nandamudi; SANTOS, Rafael; ROSA, Reinaldo; TRAVELHO, Jerônimo. Meta-Heurísticas e Otimização Combinatória: Aplicações em Problemas Ambientais, jan. 2008.
- BERNARDO, Maria do Rosário Matos. **Programação Linear (2ª parte)**. 2016. Disponível em: <https://repositorioaberto.uab.pt/bitstream/10400.2/5865/1/Programa%C3%A7%C3%A3o%20linear%20-2parte.pdf>.
- BHATTACHARYA, Subir; ROY, Rahul; BHATTACHARYA, Sumita. Um algoritmo exato de busca em profundidade para o problema de carregamento de paletes. **European Journal of Operational Research**, v. 110, n. 3, p. 610–625, 1998. ISSN 0377-2217.
- BISCHOFF, E.E.; JANETZ, F.; RATCLIFF, M.S.W. Carregamento de paletes com itens não idênticos. **European Journal of Operational Research**, v. 84, n. 3, p. 681–692, 1995. Corte e Embalagem. ISSN 0377-2217.

CHENG, S. L.; HWANG, C. Aproximação ótima de sistemas lineares por um algoritmo de evolução diferencial. **IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans**, v. 31, n. 6, p. 698–707, 2001.

COLIN, Emerson Carlos. **Pesquisa Operacional: 170 aplicações em estratégia, finanças, logística, produção, marketing e vendas**. Rio de Janeiro: LTC, 2007.

DEB, Kalyanmoy. **Otimização Multiobjetivo utilizando Algoritmos Evolucionários**. [S.l.]: John Wiley & Sons, 2001.

DOWSLAND, Kathryn A. Carregamento de paletes automatizado eficiente. **European Journal of Operational Research**, v. 44, n. 2, p. 232–238, 1990. Corte e Embalagem. ISSN 0377-2217.

DOWSLAND, Kathryn A. Um algoritmo exato para o problema de carregamento de paletes. **European Journal of Operational Research**, v. 31, n. 1, p. 78–84, 1987. ISSN 0377-2217.

DYCKHOFF, Harald. Uma tipologia de problemas de corte e embalagem. **European Journal of Operational Research**, v. 44, n. 2, p. 145–159, 1990. Corte e Embalagem. ISSN 0377-2217.

ESTEVEVES, Toni. **Agrupando conceitos e classificando imagens com Deep Learning: Uma introdução aos conceitos de CNN com Python, Scikit Learning e Keras**. 2020. Disponível em: <https://estevestoni.medium.com/agrupando-conceitos-e-classificando-imagens-com-deep-learning-5b2674f99539>. Acesso em: 19 mar. 2024.

FANG, Jie; RAO, Yunqing; ZHAO, Xusheng; DU, Bing. Um Algoritmo Híbrido de Aprendizado por Reforço para Problemas de Empacotamento Irregular 2D. **Matemática**, v. 11, n. 2, p. 327, 2023. ISSN 2227-7390.

FERREIRA, Fernanda de Oliveira; ZANONI, João Victor; TEODORO, Aline; RINCON, Leonardo Mejia. A methodological proposal for the optimal solution of the 2-DPLP problem. *In: 27º Congresso Internacional de Engenharia Mecânica (COBEM 2023)*. Florianópolis, SC, Brasil: Sociedade Brasileira de Ciências Mecânicas e Engenharia, 4–8 dez. 2023.

FOULDS, L. R. **Otimização Combinatória para Graduados**. Nova York: Springer-Verlag, 1984. P. 114.

GIMENEZ-PALACIOS, I.; ALONSO, M.T.; ALVAREZ-VALDES, R.; PARREÑO, F. Problemas de carregamento multi-contêiner com condições de entrega múltipla e dividida. **Computers & Industrial Engineering**, v. 175, p. 108844, 2023. ISSN 0360-8352.

GOMES, Fabrício Maciel; PEREIRA, Félix Monteiro; MARINS, Fernando Augusto Silva; SILVA, Messias Borges. Estudo comparativo entre os métodos gradiente reduzido generalizado e algoritmo genético em otimização com múltiplas respostas. **Revista Produção Online**, Associação Brasileira de Engenharia de Produção - ABEPRO, v. 17, n. 2, p. 592–619, 14 jun. 2017.

GRANEMANN, André Luiz. **Desenvolvimento de um sistema robótico para embalagem de potes para uma empresa de laticínios**. 2020. Trabalho de Conclusão de Curso – Universidade Federal de Santa Catarina, Centro de Blumenau. Apresentado à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

GZARA, Fatma; ELHEDHLI, Samir; YILDIZ, Burak C. O Problema de Carregamento de paletes: Empacotamento de recipientes tridimensionais com restrições práticas. **European Journal of Operational Research**, v. 287, n. 3, p. 1062–1074, 2020. ISSN 0377-2217.

HOPPER, E.; TURTON, B.C.H. Uma investigação empírica de algoritmos meta-heurísticos e heurísticos para um problema de empacotamento 2D. **European Journal of Operational Research**, v. 128, n. 1, p. 34–57, 2001. ISSN 0377-2217.

IKENOHUCHI, Jardson Yudi Barroso. **Programação e simulação de um robô antropomórfico para paletização de rotores estampados**. 2022. Trabalho de Conclusão de Curso – Universidade Federal de Santa Catarina, Centro de Blumenau. Apresentado à Universidade Federal de Santa Catarina como parte dos requisitos para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

KROONENBERG, Pieter M. **Análise de Dados Multiway Aplicada**. [S.l.]: John Wiley & Sons, 2008.

LU, Yiping; CHA, Jianzhong. Um algoritmo rápido para identificar instâncias de tamanho mínimo das classes de equivalência do Problema de Carregamento de paletes.

European Journal of Operational Research, v. 237, n. 3, p. 794–801, 2014. ISSN 0377-2217.

MARTINEZ-SYKORA, A.; ALVAREZ-VALDES, R.; BENNELL, J.A.; RUIZ, R.; TAMARIT, J.M. Matheurísticas para o problema de empacotamento irregular de recipientes com rotações livres. **European Journal of Operational Research**, v. 258, n. 2, p. 440–455, 2017. ISSN 0377-2217.

MASCARENHAS, Walter F. Dois aspectos do problema de carregamento de paletes. **Electronic Notes in Discrete Mathematics**, v. 19, p. 381–387, 2005. Proceedings of GRACO2005. ISSN 1571-0653.

MORABITO, Reinaldo; MORALES, Silvia Regina; WIDMER, João Alexandre. Otimização do carregamento de produtos paletizados em caminhões. **Transportation Research Part E: Logistics and Transportation Review**, v. 36, n. 4, p. 285–296, 2000. ISSN 1366-5545.

MORALES, Silvia Regina; MORABITO, Reinaldo. Uma heurística simples e eficaz para resolver o problema do carregamento de paletes do produtor. **Revista Brasileira de Pesquisa Operacional**, v. 4, n. 1, p. 52–75, abr. 1997.

MOURA, Cassiano Rodrigues; KRSYSCYK, José Vitor; ULBRICHT, Gerson; CHAGAS, Gil Magno Portal. Otimização de cortes bidimensionais em chapas através da aplicação de modelos matemáticos. **Revista Gestão Industrial**, Ponta Grossa, v. 15, n. 3, p. 59–75, jul. 2019.

NELIEN, Josef. Como usar restrições estruturais para calcular um limite superior para o problema de carregamento de paletes. **European Journal of Operational Research**, v. 84, n. 3, p. 662–680, 1995. Corte e Empacotamento. ISSN 0377-2217.

OLIVEIRA, Giovana Trindade da Silva. **Estudo e Aplicações da Evolução Diferencial**. 2006. Dissertação – Universidade Federal de Uberlândia, Uberlândia, MG. Dissertação apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Uberlândia, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Mecânica. Área de concentração: Mecânica dos Sólidos e Vibrações. Orientadora: Prof^a. Dr^a. Sezimária F. Pereira Saramago.

PACHECO, André. **O algoritmo Evolução Diferencial - DE**. 2015. Disponível em: <https://computacaointeligente.com.br/algoritmos/o-algoritmo-evolucao-diferencial/>. Acesso em: 14 mar. 2024.

PEGORETTI, Felipe Raizer. **Programação e simulação de uma célula robótica para manipulação de carcaças de motores elétricos**. 2021. Trabalho de Conclusão de Curso – Universidade Federal de Santa Catarina, Centro de Blumenau. Departamento de Engenharia de Controle e Automação e Computação, Centro Tecnológico, Ciências Exatas e Educação.

RANGEL, Socorro; FIGUEIREDO, Altamir G. de. O Problema de Corte de Estoque em Indústrias de Móveis de Pequeno e Médio Portes. **Pesquisa Operacional**, [S.L.], v. 28, n. 3, p. 451–472, dez. 2008.

RIBEIRO, Glaydston Mattos; LORENA, Luiz Antonio Nogueira. Relaxação Lagrangeana com agrupamentos e geração de colunas para o problema de carregamento de paletes do fabricante. **Computers & Operations Research**, v. 34, n. 9, p. 2695–2708, 2007. ISSN 0305-0548.

RINCON, Leonardo Mejia. **Otimização da capacidade de carga de um manipulador paralelo 3RRR simétrico em trajetórias com contato**. 2012. Dissertação de Mestrado – Universidade Federal de Santa Catarina, Florianópolis.

SCHEITHAUER, Gubtram; SOMMERWEIB, Uta. Heurística 4-Block para o problema de empacotamento de retângulos. **European Journal of Operational Research**, v. 108, n. 3, p. 509–526, 1998. ISSN 0377-2217.

SHENG, Liu; HONGXIA, Zhao; XISONG, Dong; CHANGJIAN, Cheng. Um algoritmo heurístico para carregamento de contêineres com paletes e caixas de preenchimento. **European Journal of Operational Research**, v. 252, n. 3, p. 728–736, 2016. ISSN 0377-2217.

STORN, K. P. R. **Evolução Diferencial (DE) para Otimização de Funções Contínuas**. Berkeley: International Computer Science Institute, 2007.

STORN, R.; PRICE, K. Evolução diferencial - Um heurística simples e eficiente para otimização global em espaços contínuos. **J. Global Optimiz**, v. 11, p. 341–359, 1997.

TERNO, Johannes; SCHEITHAUER, Guntram; SOMMERWEISS, Uta; RIEHME, Jan. Uma abordagem eficiente para o problema de carregamento de múltiplos paletes. **European Journal of Operational Research**, v. 123, n. 2, p. 372–381, 2000. ISSN 0377-2217.

UNIVERSIDADE DE CAMBRIDGE. **O que é um tensor?** Universidade de Cambridge. 2024. Disponível em: https://www.doitpoms.ac.uk/tlplib/tensors/what_is_tensor.php.

VIANNA, Andréa Carla Gonçalves; POLDI, Kelly Cristina. O Problema de Corte de Estoque Bidimensional Aplicado a uma Indústria de Esquadrias Metálicas. *In*: SOCIEDADE BRASILEIRA DE PESQUISA OPERACIONAL. SBPO XXXVII Simpósio Brasileiro de Pesquisa Operacional. Gramado, RS: [s.n.], Setembro 2005.

ANEXO A – Publicação Realizada



COBEM-2023-0816

A methodological proposal for the optimal solution of the 2-DPLP problem

Fernanda de Oliveira Ferreira*

fernanda_o_ferreira@hotmail.com

João Victor Zanoni*

jvzanoni111@gmail.com

Aline Souza Teodoro**

line_tsa@hotmail.com

Leonardo Mejia Rincon**

leonardo.mejia.rincon@ufsc.br

*Undergraduate program in Automation, Control, and Computation Engineering (CAC) Federal University of Santa Catarina - Campus Blumenau. Santa Catarina, Brazil.

**Graduate Program in Textile Engineering (PGETEX) Federal University of Santa Catarina - Campus Blumenau. Blumenau, Santa Catarina, Brazil

Abstract. The Pallet Loading Problem (PLP) constitutes a significant focus in the field of industrial engineering, with wide-ranging applications in multiple industries such as automated palletization and robotics. To solve the PLP, a variety of algorithms and techniques have been used, including heuristics, meta-heuristics, mathematical programming, and artificial intelligence. Palletization, in broad terms, pertains to efficiently loading and arranging products onto pallets for either transportation or storage purposes. Nevertheless, this represents a multifaceted challenge, given the wide array of factors that must be meticulously assessed to achieve an optimal solution. Parameter selection contributes to numerous PLP variations, with the two-dimensional Distributor's Pallet Loading Problem (2-DPLP) being one of them. To solve this problem, this paper proposes a novel methodology devised to optimize the distribution of items on a two-dimensional pallet. Our proposed strategy is separated into two main stages. The first consists of minimizing an objective function based on the definition of a computational tensor through the application of an evolutionary computation algorithm, and the second one consists of overlapping elimination through a new proposed algorithmic approach. Each tensor layer in our model corresponds to an item destined for placement on the pallet. This methodology has notable benefits, including the easy identification of overlapping items on the pallet. Our findings suggest that the computational solution optimizes pallet space utilization and provides precise placement information that can be used by a robotic system in order to execute the palletizing process.

Keywords: Two-dimensional pallet loading problem, Evolutionary computation, Tensor, Methodological proposal, Robotic System.

1. INTRODUCTION

The Pallet Loading Problem (PLP) stands as a critical research area in industrial engineering, with its influence pervading a variety of industries (Arun Prasad and Krishnakumar, 2021). Notably, its impact extends to logistics and supply chain management domains, where operational success hinges upon the efficient spatial and resource utilization (Ahn *et al.*, 2015). The role of logistics in the textile industry cannot be understated, spanning the entire supply chain from sourcing manufacturing inputs to the distribution of the final product.

A common simplification of the PLP is its two-dimensional variant, which revolves around packing identical rectangular items (boxes) onto a larger, rectangular pallet. Items rotation is permitted in steps of 90 degrees, with the caveat that only orthogonal packages are allowed. The overarching goal for this simplification is to maximize the number of packed items onto the pallet while precluding any overlap (Lu and Cha, 2014).

To solve this problem, this paper proposes a new methodology aiming at an optimal distribution of items in a two-dimensional pallet. This methodology aims to minimize an objective function based on a tensorial structure where each layer of the tensor represents an object in the interface. The dimension of each tensor's layer is equal to the dimension of the pallet and, each tensor layer corresponds to an item that will be placed on the pallet. To minimize the proposed objective function, the Evolution Differential algorithm of Storn and Price was used. Our approach provides the significant benefit of allowing straightforward verification of pallet overlap, along with immediate rejection of overlapping cases. Upon the methodology's full development, we anticipate our computational solution to significantly enhance pallet space utilization and provide precise placement information to a robotic system.

Historically, the Distributor's Pallet Loading Problem (DPLP) has been the focal point of numerous studies, yielding an array of methods designed to address this challenge. Various procedures have been developed to obtain the most