UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA METALÚRGICA E DE MATERIAIS

CURSO DE GRADUAÇÃO EM ENGENHARIA METALÚRGICA

GABRIEL SANTOS VASCONCELOS

SOFTWARE PARA OTIMIZAÇÃO DO CORTE DE MATERIAIS NA INDÚSTRIA EM GERAL

FORTALEZA

2026

GABRIEL SANTOS VASCONCELOS

SOFTWARE PARA OTIMIZAÇÃO DO CORTE DE MATERIAIS NA INDÚSTRIA EM GERAL

Trabalho de conclusão de curso apresentada ao Curso de Graduação em Engenharia Metalúrgica da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Metalúrgica.

Orientador: Prof. Dr. Francisco Marcondes

FORTALEZA

2026

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

V45s Vasconcelos, Gabriel Santos.
Software para otimização do corte de materiais na indústria em geral / Gabriel Santos Vasconcelos. – 2026.
52 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Metalúrgica, Fortaleza, 2026.
Orientação: Prof. Dr. Francisco Marcondes.

1. otimização de corte. 2. algoritmos heurísticos. 3. corte linear. 4. corte bidimensional. I. Título.

CDD 669

GABRIEL SANTOS VASCONCELOS

SOFTWARE PARA OTIMIZAÇÃO DO CORTE DE MATERIAIS NA INDÚSTRIA EM GERAL

Trabalho de conclusão de curso apresentada ao Curso de Graduação em Engenharia Metalúrgica da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Metalúrgica.

Aprovado em: 16 de janeiro de 2026

BANCA EXAMINADORA

_____

Prof. Dr. Francisco Marcondes (Orientador)

Universidade Federal do Ceará (UFC)

_____

Prof. Dr. Ricardo Emílio F. Quevedo Nogueira

Universidade Federal do Ceará (UFC)

_____

Eng. João Victor Barroso Xavier

Universidade Federal do Ceará (UFC)

Ao Prof. Dr. Francisco Marcondes, pela orientação.
À minha família, em especial aos meus pais, pelo apoio constante.
A todos que fizeram parte desta jornada.

AGRADECIMENTOS

Primeiramente, expresso minha sincera gratidão ao meu orientador, Prof. Dr. Francisco Marcondes, pela confiança depositada, pela orientação meticulosa e pelo apoio intelectual decisivo ao longo de toda a realização deste trabalho. Sua experiência e disponibilidade foram fundamentais para o rigor e a conclusão desta pesquisa.

Agradeço profundamente à minha família, especialmente minha mãe, Aurícelia Alves dos Santos, dedicada professora que sempre valorizou o saber, e ao meu pai, cujo trabalho incansável em conjunto com ela garantiu todas as oportunidades. Aos meus tios, tias e primos, que formam uma família onde a educação sempre foi prioridade, meu reconhecimento pelo constante incentivo e pelo exemplo que carrego.

Por fim, registro meu apreço a todos os professores, colegas e amigos que, de alguma forma, contribuíram e compartilharam desta jornada acadêmica.

RESUMO

Este trabalho apresentou o desenvolvimento de um sistema de otimização de corte de materiais, uma ferramenta computacional de corte de materiais industriais que possui algoritmos para corte linear (1D) e corte em área (2D) juntamente com uma interface gráfica com configuração flexível de restrições de fácil entendimento. O problema de otimização de corte de materiais é fundamental na indústria, impactando diretamente os custos de produção e a sustentabilidade ambiental através da geração de resíduos. O objetivo geral desse trabalho foi desenvolver um sistema completo de otimização de corte que suportasse ambos os tipos de corte. A metodologia adotada envolveu a implementação de algoritmos de busca exaustiva com heurísticas para contornar limitações computacionais, utilizando programação dinâmica e geração iterativa de combinações para corte linear (1D), e o algoritmo *bottom-left* modificado com suporte a rotação automática para corte bidimensional (2D). O sistema foi desenvolvido em Python com interface gráfica fazendo uso da biblioteca padrão do *python*, *Tkinter*, incorporando funcionalidades de configuração, visualização gráfica dos resultados obtidos e exportação de relatórios. A validação foi realizada através de seis cenários de teste que foram apresentados no decorrer desse trabalho, três para corte linear e três para corte bidimensional, utilizando dados simulados com diferentes configurações de materiais e peças, para mostrar o funcionamento da ferramenta em diferentes cenários. Os principais resultados demonstraram capacidade de encontrar soluções eficientes de corte com eficiências superiores a 93% em todos os cenários testados, processando problemas com múltiplos tipos de peças em tempos de processamento viáveis. O sistema demonstrou capacidade de reduzir desperdício de material comparado a métodos manuais, sendo adaptável a diferentes contextos industriais. A principal contribuição do trabalho está no desenvolvimento de uma ferramenta completa e acessível que integra várias funcionalidades em uma única plataforma, oferecendo suporte simultâneo para ambos os tipos de corte e mostrando viabilidade de abordagens heurísticas para contornar limitações computacionais de problemas de alta complexidade computacional.

Palavras-chave: otimização de corte, problema de corte de estoque, algoritmos heurísticos, corte linear, corte bidimensional.

ABSTRACT

This work presented the development of a material cutting optimization system, a computational tool for industrial material cutting that incorporates algorithms for linear cutting (1D) and area cutting (2D), along with a graphical user interface featuring flexible and easily understandable constraint configuration. The material cutting optimization problem is fundamental in industry, directly impacting production costs and environmental sustainability through waste generation. The general objective of this work was to develop a complete cutting optimization system capable of supporting both types of cutting. The adopted methodology involved the implementation of exhaustive search algorithms with heuristics to overcome computational limitations, using dynamic programming and iterative generation of combinations for linear cutting (1D), and a modified bottom-left algorithm with automatic rotation support for bidimensional cutting (2D). The system was developed in Python with a graphical interface built using Python's standard library, Tkinter, incorporating features for configuration, graphical visualization of the obtained results, and report exportation. Validation was carried out through six test scenarios presented throughout this work, three for linear cutting and three for bidimensional cutting, using simulated data with different configurations of materials and pieces to demonstrate the tool's operation in different scenarios. The main results demonstrated the system's ability to find efficient cutting solutions with efficiencies above 93% in all tested scenarios, processing problems with multiple types of pieces within viable processing times. The system showed the ability to reduce material waste compared to manual methods, while being adaptable to different industrial contexts. The main contribution of this work lies in the development of a complete and accessible tool that integrates multiple functionalities into a single platform, offering simultaneous support for both types of cutting and demonstrating the feasibility of heuristic approaches to overcome the computational limitations of highly complex problems.

Keywords: cutting optimization, cutting stock problem, heuristic algorithms, linear cutting, two-dimensional cutting.

SUMÁRIO

1 INTRODUÇÃO 11
2 OBJETIVOS 14
2.1 Objetivo Geral 14
2.2 Objetivos Específicos 14
3 REVISÃO BIBLIOGRÁFICA 15
3.1 Problema de Corte de Materiais na Indústria 15
3.2 Problemas de Otimização Combinatória 16
3.3 Cutting Stock Problem 17
3.4 Algoritmos de Corte Linear (1D) 18
3.5 Algoritmos de Corte em Área (2D) 19
3.6 Heurísticas e Estratégias de Otimização 21
3.7 Aplicações Industriais da Otimização de Corte 22
4 METODOLOGIA 24
4.1 Tipo de Pesquisa 24
4.2 Arquitetura do Sistema 24
4.2.1 Estrutura Modular 24
4.2.2 Fluxo de Dados 25
4.2.3 Persistência de Dados 26
4.3 Algoritmos Utilizados 26
4.3.1 Algoritmo de Otimização para Corte Linear (1D) 26
4.3.2 Algoritmo de Otimização para Corte em Área (2D) 28
4.3.2.1 Algoritmo Bottom-Left 29
4.3.2.2 Adaptações implementadas 29
4.3.2.3 Geração de Combinações de Quantidades 29
4.3.2.4 Estratégia de busca inteligente 29
4.3.2.5 Detecção de colisões 29
4.3.2.6 Fase de Preenchimento de Espaços 29
4.3.3 Heurísticas e Otimizações 30
4.4 Tecnologias Empregadas 30
4.4.1 Linguagem de Programação 30
4.4.2 Interface Gráfica 30
4.4.3 Bibliotecas Auxiliares 30
4.4.4 Estrutura de Dados 30

4.5 Validação 30
4.5.1 Estratégia de Validação 31
4.5.2 Cenários de Teste 31
4.5.3 Métricas de Avaliação 31
4.5.4 Análise de Resultados 31
4.5.5 Limitações da Validação 31
5 RESULTADOS E DISCUSSÃO 32
5.1 Cenários de Teste 32
5.1.1 Cenário 1D-1: Corte de Bobina de Fibra de Carbono 32
5.1.2 Cenário 1D-2: Corte de Barras Metálicas 32
5.1.3 Cenário 1D-3: Corte com Múltiplas Restrições 32
5.1.4 Cenário 2D-1: Corte de Chapa de MDF 32
5.1.5 Cenário 2D-2: Corte com Rotação Permitida 33
5.1.6 Cenário 2D-3: Corte com Margem de Segurança 33
5.2 Resultados Obtidos 33
5.2.1 Resultados do Cenário 1D-1 33
5.2.2 Resultados do Cenário 1D-2 35
5.2.3 Resultados do Cenário 1D-3 36
5.2.4 Resultados do Cenário 2D-1 38
5.2.5 Resultados do Cenário 2D-2 40
5.2.6 Resultados do Cenário 2D-3 41
5.3 Análise e Discussão dos Resultados 43
5.3.1 Análise do Cenário 1D-1..... 43
5.3.2 Análise do Cenário 1D-2..... 44
5.3.3 Análise do Cenário 1D-3..... 44
5.3.4 Análise do Cenário 2D-1..... 44
5.3.5 Análise do Cenário 2D-2..... 45
5.3.6 Análise do Cenário 2D-3..... 45
5.3.7 Comportamento das restrições..... 45
5.3.8 Comparação entre Cenários Simples e Complexos..... 46
5.3.9 Comparação de cenários 2D..... 46
5.3.10 Comparação de métodos manuais..... 46
5.4 Comparação entre Corte 1D e Corte 2D..... 47
5.5 Limitações Identificadas..... 47
5.5.1 Limitações do algoritmo..... 47

5.5.2 Limitações do Ambiente de Teste e dos Dados..... 48
5.5.3 Impactos das limitações..... 48
6 CONCLUSÃO 49
7 REFERÊNCIAS 51

LISTA DE FIGURAS

Figura 1 Fluxo de dados do sistema de otimização de corte..... 25
Figura 2 Algoritmo de Otimização para Corte Linear (1D)..... 27
Figura 3 Interface do software durante a configuração do cenário de teste 1D-1... 33
Figura 4 Resultado final obtido no cenário 1D-1)..... 34
Figura 5 Visualização gráfica do layout de corte do teste 1D-1)..... 34
Figura 6 Interface do software durante a configuração do cenário de teste 1D-2.... 35
Figura 7 Resultado final obtido no cenário 1D-2..... 35
Figura 8 Visualização gráfica do layout de corte obtido no cenário 1D-2..... 36
Figura 9 Interface do software durante a configuração do teste 1D-3..... 36
Figura 10 Resultado final obtido no cenário 1D-3..... 37
Figura 11 Visualização gráfica do layout de corte obtido no cenário 1D-3..... 38
Figura 12 Interface do software durante a configuração do teste 2D-1..... 38
Figura 13 Resultado final obtido no cenário 2D-1..... 39
Figura 14 Estatísticas do resultado final obtido no cenário 2D-1..... 39
Figura 15 Interface do software durante a configuração do teste 2D-2..... 40
Figura 16 Resultado final obtido no cenário 2D-2..... 40
Figura 17 Estatísticas do resultado final obtido no cenário 2D-2..... 41
Figura 18 Interface do software durante a configuração do teste 2D-3..... 41
Figura 19 Interface do software durante a configuração da margem de segurança 2D-3..... 42
Figura 20 Resultado final obtido no cenário 2D-3..... 42
Figura 21 Estatísticas do resultado final obtido no cenário 2D-3..... 43

1 INTRODUÇÃO

A indústria, de modo geral, busca continuamente aumentar sua eficiência produtiva, reduzindo a geração de sobras e promovendo um melhor aproveitamento dos materiais utilizados. No contexto da manufatura, o corte de materiais, como barras, chapas, bobinas e tecidos, exerce influência direta sobre esses indicadores, uma vez que a forma como o material é seccionado impacta significativamente os níveis de desperdício e o desempenho global do processo produtivo. Dessa forma, a otimização do corte configura-se como um fator determinante para a eficiência industrial e para a redução de perdas de matéria-prima (Morabito, 1994; Arenales, 1999).

Ao aprofundar a discussão sobre a otimização do corte de materiais, destaca-se o *Cutting Stock Problem* (problema de corte de estoque), um problema clássico da otimização combinatória, amplamente estudado na literatura e intimamente relacionado ao conhecido problema da mochila, esse problema consiste em determinar a melhor forma de cortar peças de diferentes dimensões a partir de um material disponível, buscando minimizar o desperdício e maximizar o aproveitamento do insumo. Trata-se de uma situação recorrente no cotidiano industrial, na qual operadores frequentemente realizam essas combinações de forma manual, resultando em soluções limitadas, pouco eficientes e suscetíveis a erros de cálculo, especialmente diante do elevado número de combinações possíveis (Gilmore; Gomory, 1961; Garey; Johnson, 1979; Nemhauser; Wolsey, 1988).

Apesar do impacto significativo que o problema de corte exerce sobre a geração de sobras e o desperdício de materiais, muitas empresas ainda utilizam métodos manuais para calcular as combinações de corte. Essa prática contribui para perdas que poderiam ser evitadas por meio do uso de ferramentas computacionais adequadas. Soma-se a isso a carência de soluções que ofereçam ampla gama de aplicações, como suporte ao corte unidimensional (1D) e bidimensional (2D), controle de registros de cortes, análise de custos, geração de relatórios e cadastro de peças com diferentes dimensões para múltiplos produtos, limitando a eficiência do planejamento industrial (Cherri, 2006; Abuabara, 2006).

Diante desse cenário, este Trabalho de Conclusão de Curso (TCC) apresenta o desenvolvimento de um *software* voltado à otimização do corte de materiais na indústria, não obstante a ferramenta possua potencial de aplicação em diferentes

setores produtivos, o enfoque principal deste trabalho está direcionado ao setor metal-mecânico, em razão de sua elevada demanda por processos de corte eficientes e do alto custo associado à matéria-prima utilizada (Morabito, 1994; Abuabara, 2006).

O *software* desenvolvido contempla duas modalidades de corte, o corte linear (1D), aplicado a materiais como barras e bobinas, e o corte em área (2D), destinado a chapas e materiais planos. A proposta busca atender às necessidades de diferentes contextos industriais, com ênfase nas áreas de Engenharia Metalúrgica e metal-mecânica, oferecendo uma solução flexível e adaptável às características específicas de cada processo produtivo (Arenales, 1999; Bogue, 2020).

O problema central abordado por esta pesquisa refere-se à elevada geração de desperdício de materiais e à baixa eficiência nos processos de corte em diversos setores industriais. Diante disso, o trabalho propõe o desenvolvimento de uma ferramenta computacional de otimização de corte que seja adaptável a diferentes realidades industriais, com o objetivo de reduzir desperdícios e melhorar a eficiência no consumo de matéria-prima. A relevância do desenvolvimento fundamenta-se em sua importância científica, ao contribuir para o avanço dos métodos de otimização de corte; social, ao colaborar com a sustentabilidade e a redução do consumo de recursos e energia; e tecnológica, ao oferecer uma solução aplicável a múltiplos contextos industriais (Ribeiro; Morales, 2020; Wolsey, 1998).

Ao final deste trabalho, tem-se como objetivo geral apresentar um *software* de otimização de corte de materiais voltado à aplicação industrial, bem como demonstrar suas funcionalidades e possibilidades de uso em diferentes cenários produtivos. Como objetivos específicos, destacam-se o desenvolvimento de algoritmos de otimização para corte linear (1D) e em área (2D), a implementação de uma interface gráfica que permita a configuração dos parâmetros e a visualização dos resultados, a incorporação de análises de custos, geração de relatórios e histórico de combinações, a validação da ferramenta por meio de cenários simulados e a documentação da metodologia e dos resultados obtidos, evidenciando a necessidade e a aplicabilidade da solução proposta (Gilmore; Gomory, 1961; Nemhauser; Wolsey, 1988; Bogue, 2020).

A metodologia adotada combina pesquisa aplicada e desenvolvimento de software, utilizando técnicas de otimização combinatória, como algoritmos *bottom-left* para problemas bidimensionais e busca exaustiva com heurísticas para problemas unidimensionais, a implementação foi realizada em linguagem *Python*, com interface

gráfica desenvolvida em *Tkinter*, e a validação ocorreu por meio de cenários simulados baseados em situações reais da indústria metal-mecânica e de outros setores produtivos (Arenales, 1999; Ribeiro; Morales, 2020).

2 OBJETIVOS

2.1 Objetivo Geral

Desenvolver um sistema computacional de otimização de corte de materiais industriais que integre algoritmos para corte linear (1D) e corte em área (2D), incorporando restrições práticas e oferecendo uma interface gráfica do usuário (GUI) amigável para configuração e visualização dos resultados.

2.2 Objetivos Específicos

- Implementar algoritmos de otimização para corte linear utilizando estratégias de busca exaustiva com geração iterativa de combinações, permitindo processar problemas com múltiplos tipos de peças e restrições complexas.
- Implementar algoritmos de otimização para corte em área utilizando o algoritmo *bottom-left* modificado com suporte a rotação automática de peças e verificação de colisões.
- Desenvolver um sistema de restrições que suporte quantidade mínima e máxima por tipo de peça, prioridades de posicionamento, não repetição de peças e margens de segurança.

3 REVISÃO BIBLIOGRÁFICA

3.1 Problema de Corte de Materiais na Indústria

O corte de materiais constitui uma das operações fundamentais nos processos de manufatura, envolvendo a divisão de matérias-primas em peças menores, de acordo com dimensões e quantidades previamente especificadas. Essa operação está presente em diversos setores industriais, como o metal-mecânico, madeireiro, têxtil, vidraceiro e gráfico, entre outros. A eficiência do processo de corte exerce impacto direto sobre os custos de produção, o tempo de fabricação e a sustentabilidade ambiental, especialmente em função da geração de resíduos associada a essa etapa produtiva (Morabito, 1994; Arenales, 1999).

A relevância econômica do problema de corte de materiais é expressiva, sobretudo em contextos nos quais a matéria-prima representa uma parcela significativa do custo final do produto. Em setores como a construção civil e a manufatura industrial, que utilizam insumos de elevado valor, como aço, madeira e vidro, a otimização do aproveitamento do material pode resultar em reduções substanciais de custos. O desperdício de material não apenas configura uma perda financeira direta, mas também acarreta impactos ambientais relacionados à extração intensiva de recursos naturais e ao descarte de resíduos industriais (Cherri, 2006; Abuabara, 2006).

O impacto do desperdício de materiais nos processos produtivos extrapola a dimensão econômica, estendendo-se a aspectos relacionados à eficiência operacional e à competitividade industrial. Empresas que conseguem reduzir perdas por meio de um planejamento adequado do corte tendem a apresentar melhor desempenho financeiro e maior capacidade de competir em mercados caracterizados por margens reduzidas. Nesse sentido, estabelece-se uma relação inversa entre desperdício e eficiência produtiva, na qual melhorias na gestão do corte de materiais contribuem diretamente para o aumento da eficiência global do sistema produtivo (Bogue, 2020; Ribeiro; Morales, 2020).

A complexidade inerente ao problema de corte de materiais decorre da necessidade de conciliar múltiplas variáveis, tais como as dimensões das peças, as quantidades demandadas, as restrições de produção e as características físicas do material disponível. Essa complexidade justifica a adoção de métodos sistemáticos de otimização, baseados em programação linear, programação inteira e otimização

combinatória, capazes de auxiliar o processo de tomada de decisão, substituindo ou complementando abordagens empíricas tradicionalmente empregadas na indústria (Gilmore; Gomory, 1961; Nemhauser; Wolsey, 1988; Wolsey, 1998).

3.2 Problemas de Otimização Combinatória

A otimização combinatória representa uma área fundamental da ciência da computação e da pesquisa operacional, dedicada ao estudo de problemas de otimização nos quais o espaço de soluções é discreto e finito. Tais problemas caracterizam-se pela necessidade de selecionar a melhor solução entre um conjunto finito de alternativas possíveis, considerando uma ou mais funções objetivo e um conjunto de restrições que devem ser satisfeitas.

A natureza combinatória desses problemas decorre da necessidade de examinar diferentes combinações ou arranjos de elementos, em que cada configuração representa uma solução potencial. Como o número de soluções possíveis cresce exponencialmente com o tamanho da instância do problema, a exploração manual torna-se inviável mesmo para problemas de dimensões moderadas, conferindo-lhes elevada complexidade computacional.

A complexidade computacional dos problemas de otimização combinatória é frequentemente classificada por meio da teoria da complexidade computacional (Garey; Johnson, 1979). Muitos desses problemas pertencem à classe NP (Nondeterministic Polynomial Time), na qual uma solução candidata pode ser verificada em tempo computacionalmente eficiente (polinomial). Entretanto, até o momento, não se conhece um algoritmo capaz de resolver esses problemas de forma exata com baixo tempo de computação.

Problemas particularmente desafiadores são classificados como NP-completos, indicando que qualquer problema em NP pode ser reduzido a eles em tempo computacionalmente eficiente (polinomial), o que sugere que a obtenção de soluções exatas pode demandar elevado custo computacional à medida que o tamanho do problema aumenta (Garey; Johnson, 1979).

A relação entre problemas de otimização combinatória e problemas industriais é estreita, uma vez que muitos desafios práticos enfrentados nos processos de produção e logística podem ser modelados como problemas combinatórios (Ribeiro; Morales, 2020). Problemas de roteamento de veículos, agendamento de tarefas,

designação de recursos e corte de materiais são exemplos de aplicações industriais que apresentam características combinatórias. A capacidade de formular e resolver esses problemas de forma sistemática oferece vantagens competitivas significativas para organizações que conseguem implementar soluções adequadas.

Os métodos exatos para problemas de otimização combinatória incluem técnicas como programação inteira e algoritmos de *branch-and-bound*, que garantem a obtenção da solução ótima, embora possam apresentar elevado custo computacional quando aplicados a problemas de grande porte (Nemhauser; Wolsey, 1988; Wolsey, 1998).

3.3 *Cutting Stock Problem*

O *Cutting Stock Problem*, também conhecido como problema de corte de estoque, representa um dos problemas clássicos de otimização combinatória aplicados à indústria. Sua origem remonta ao desenvolvimento da pesquisa operacional na década de 1960, quando Gilmore e Gomory (1961) buscaram modelar matematicamente o desafio de cortar materiais em estoque em peças menores, de modo a minimizar o desperdício ou maximizar o aproveitamento do material disponível.

A formulação geral do *Cutting Stock Problem* envolve a determinação de como cortar um ou mais objetos de dimensões conhecidas, denominados materiais ou estoques, em peças menores de tamanhos e quantidades especificadas, de forma a otimizar uma função objetivo, tipicamente relacionada à minimização do desperdício, à minimização do número de materiais utilizados ou à maximização do aproveitamento do material (Gilmore; Gomory, 1961). O problema pode assumir diferentes variantes, dependendo das dimensões envolvidas: corte linear ou unidimensional, quando apenas uma dimensão é relevante; corte bidimensional, quando duas dimensões devem ser consideradas; e corte tridimensional, em casos mais complexos (Arenales, 1999).

A relação entre o *Cutting Stock Problem* e o problema da mochila, outro problema clássico de otimização combinatória, é estreita e amplamente reconhecida na literatura. O problema da mochila envolve a seleção de itens para serem colocados em uma mochila com capacidade limitada, maximizando o valor total dos itens selecionados sem exceder essa capacidade. No contexto do corte de materiais, cada

material disponível pode ser interpretado como uma mochila com capacidade equivalente à dimensão do material, enquanto as peças a serem cortadas representam itens com pesos e valores específicos. Essa analogia permite a aplicação de técnicas originalmente desenvolvidas para o problema da mochila na resolução de instâncias do Cutting Stock Problem.

Os principais desafios associados ao *Cutting Stock Problem* incluem a explosão combinatória do espaço de soluções à medida que o número de tipos de peças aumenta, a necessidade de considerar múltiplas restrições práticas, como quantidades mínimas e máximas, prioridades de produção e características específicas do material e a dificuldade de encontrar soluções ótimas em tempo computacional viável para problemas de dimensão realista. Essas características tornam o problema computacionalmente desafiador e justificam o desenvolvimento de métodos especializados para sua resolução.

A literatura apresenta diversas variantes do *Cutting Stock Problem*, cada uma adaptada a diferentes contextos industriais. Problemas envolvendo múltiplos materiais com dimensões distintas, custos variáveis associados a diferentes tipos de material, restrições de precedência ou compatibilidade entre peças e problemas multiobjetivo que consideram simultaneamente desperdício, custo e tempo de produção são exemplos de extensões do problema clássico com ampla aplicação prática (Morabito, 1994).

3.4 Algoritmos de Corte Linear (1D)

Os algoritmos de corte linear, também denominados algoritmos de corte unidimensional, tratam de problemas de otimização nos quais apenas uma dimensão do material é relevante para o processo de corte. Essa classe de problemas é comum em indústrias que trabalham com materiais no formato de barras, rolos, bobinas ou perfis, nos quais o comprimento é a dimensão crítica, enquanto largura e altura são fixas ou não restritivas.

Os métodos de otimização aplicados ao corte linear podem ser classificados em abordagens exatas e aproximadas. As abordagens exatas buscam encontrar a solução ótima por meio de técnicas como programação dinâmica, *branch-and-bound* e programação inteira linear (Nemhauser; Wolsey, 1988; Wolsey, 1998). A programação dinâmica, em particular, adapta-se de forma eficiente aos problemas de

corte linear, permitindo a decomposição do problema em subproblemas menores, resolvidos recursivamente, com armazenamento das soluções intermediárias para evitar cálculos redundantes.

As técnicas de *branch-and-bound* exploram o espaço de soluções de forma sistemática, utilizando limites superiores e inferiores para eliminar regiões do espaço de busca que não podem conter a solução ótima. Essa abordagem é especialmente eficaz quando combinada com relaxações lineares do problema, que fornecem limites capazes de orientar a exploração do espaço de soluções de maneira mais eficiente.

As abordagens aproximadas para o corte linear incluem métodos heurísticos que priorizam a velocidade de execução em detrimento da garantia de otimalidade. Heurísticas baseadas em ordenação, como a organização das peças por tamanho decrescente ou crescente antes do corte, constituem estratégias simples, porém frequentemente eficazes. Métodos gulosos que selecionam peças com base em critérios como maior tamanho, menor desperdício ou maior valor também oferecem soluções rápidas para problemas de dimensão moderada.

Os algoritmos de geração de padrões representam uma classe importante de métodos para o corte linear, nos quais padrões de corte são gerados dinamicamente conforme necessário. Cada padrão define uma forma específica de cortar um material em uma combinação de peças, e o problema de otimização consiste em determinar quais padrões utilizar e em quais quantidades para atender à demanda ao menor custo possível. Essa abordagem permite separar a geração de padrões viáveis da otimização de sua utilização, facilitando a resolução de problemas complexos.

As vantagens dos algoritmos de corte linear incluem relativa simplicidade computacional quando comparados a problemas multidimensionais, possibilidade de obtenção de soluções ótimas em tempo viável para muitos problemas práticos e facilidade de incorporação de restrições adicionais, como quantidades mínimas e máximas. Entre as limitações, destacam-se a dependência da qualidade da modelagem do problema, a possível explosão combinatória em situações com muitos tipos de peças e grandes quantidades, e a necessidade de adaptação às restrições específicas de cada contexto industrial.

3.5 Algoritmos de Corte em Área (2D)

Diante da elevada complexidade computacional associada aos problemas de corte de materiais, especialmente aqueles classificados como NP-difíceis, a utilização de métodos heurísticos e metaheurísticos tem se consolidado como uma alternativa viável para a obtenção de soluções satisfatórias em tempo computacional reduzido. Diferentemente dos métodos exatos, essas abordagens não garantem necessariamente a solução ótima global, porém são capazes de produzir soluções de boa qualidade, aceitáveis do ponto de vista prático e operacional, sobretudo em instâncias de grande porte (Silva; Ribeiro, 2004; Martello; Toth, 1990).

As heurísticas caracterizam-se por estratégias construtivas ou de melhoria que exploram regras específicas do problema, geralmente baseadas em critérios empíricos ou em propriedades estruturais do modelo. No contexto do problema de corte, destacam-se heurísticas clássicas como *First Fit, Best Fit* e suas variações, que buscam alocar peças de forma sequencial, priorizando o melhor aproveitamento possível do material disponível. Embora simples, tais métodos apresentam desempenho relevante em aplicações industriais, especialmente quando combinados com estratégias de refinamento local (Gilmore; Gomory, 1961; Dyckhoff, 1990).

Por sua vez, as metaheurísticas surgem como uma evolução das heurísticas tradicionais, oferecendo estruturas mais gerais e flexíveis para a exploração do espaço de soluções. Técnicas como algoritmos genéticos, busca tabu, *simulated annealing* e colônia de formigas têm sido amplamente aplicadas a problemas de corte e empacotamento, demonstrando capacidade de escapar de ótimos locais e de lidar com múltiplas restrições simultaneamente. Essas abordagens baseiam-se em princípios inspirados em fenômenos naturais ou processos cognitivos e apresentam bom desempenho em problemas de alta dimensionalidade (Glover; Laguna, 1997; Blum; Roli, 2003).

A aplicação de metaheurísticas ao problema de corte de materiais permite incorporar critérios adicionais relevantes ao ambiente industrial, como redução de sobras, balanceamento da produção, limitação de padrões de corte e adaptação a demandas dinâmicas. Além disso, essas técnicas possibilitam a hibridização com métodos exatos ou heurísticas construtivas, resultando em algoritmos híbridos que combinam robustez computacional e qualidade das soluções obtidas (Poldi; Arenales, 2009; Wäscher *et al*., 2007).

Do ponto de vista prático, a adoção de métodos heurísticos e metaheurísticos reflete uma tendência crescente na indústria, onde a necessidade de respostas

rápidas e soluções eficientes supera a exigência de otimalidade estrita. Dessa forma, essas abordagens tornam-se particularmente adequadas para sistemas produtivos reais, nos quais a variabilidade das demandas e as restrições operacionais impõem desafios adicionais aos modelos matemáticos tradicionais. Assim, o uso dessas técnicas contribui de maneira significativa para a melhoria da eficiência produtiva, redução de desperdícios e aumento da competitividade organizacional (Ribeiro; Morales, 2020; Arenales, 2015).

3.6 Heurísticas e Estratégias de Otimização

O uso de heurísticas em problemas de corte de materiais surge da necessidade de encontrar soluções de boa qualidade em tempo computacional viável, especialmente quando métodos exatos se tornam impraticáveis devido à complexidade do problema. Heurísticas são estratégias algorítmicas que utilizam conhecimento específico do problema ou regras práticas para guiar a busca por soluções, priorizando a eficiência computacional em detrimento de garantias formais de otimalidade.

As heurísticas em problemas de corte podem ser classificadas em construtivas, nas quais uma solução é construída por meio de decisões locais, e de melhoria, nas quais uma solução inicial é refinada por meio de modificações iterativas. Heurísticas construtivas são frequentemente utilizadas para gerar soluções iniciais rapidamente, enquanto heurísticas de melhoria podem ser aplicadas para refinar soluções obtidas por meio de outros métodos.

As vantagens do uso de heurísticas em termos de tempo de processamento são significativas. Enquanto métodos exatos podem requerer tempo exponencial no pior caso, heurísticas bem projetadas tipicamente executam em tempo polinomial, permitindo o processamento de instâncias de problemas de dimensão realista em tempo aceitável. Essa característica torna as heurísticas particularmente valiosas em contextos industriais, nos quais decisões de corte devem ser tomadas rapidamente e o custo computacional deve ser controlado.

A qualidade das soluções obtidas por meio de heurísticas varia de acordo com a estratégia utilizada e com as características específicas da instância do problema. Embora heurísticas não garantam a obtenção da solução ótima, frequentemente produzem soluções muito próximas da otimalidade, com diferenças de qualidade

muitas vezes compensadas pela vantagem em velocidade de execução. Em muitos contextos práticos, soluções de alta qualidade são preferíveis a soluções ótimas que demandam tempo computacional proibitivo.

Estratégias comuns de heurísticas em problemas de corte incluem a ordenação de peças por critérios como tamanho, prioridade ou valor antes do processamento, a limitação dinâmica do espaço de busca por meio de critérios de poda, a geração iterativa de combinações em vez de exploração exaustiva e a utilização de estimativas ou aproximações para guiar decisões quando o cálculo exato é custoso. A combinação de múltiplas estratégias heurísticas frequentemente produz resultados superiores à aplicação isolada de cada estratégia.

O desenvolvimento de heurísticas eficazes requer compreensão profunda das características do problema e dos fatores que influenciam a qualidade das soluções. Análises experimentais e refinamento iterativo são frequentemente necessários para ajustar parâmetros e estratégias heurísticas visando desempenho ótimo em classes específicas de problemas. A literatura apresenta diversas heurísticas desenvolvidas e testadas em diferentes contextos, oferecendo uma base sólida para adaptação a novos problemas específicos.

3.7 Aplicações Industriais da Otimização de Corte

As aplicações industriais da otimização de corte estendem-se por diversos setores, cada um apresentando características específicas que influenciam a modelagem e resolução do problema. O setor metal-mecânico representa uma das principais áreas de aplicação, onde otimização de corte de chapas metálicas, barras e perfis é fundamental para controlar custos de matéria-prima, frequentemente representando parcela significativa dos custos de produção (Abubara, 2006). A indústria automotiva, construção civil e fabricação de equipamentos industriais são exemplos de setores que se beneficiam intensivamente de técnicas de otimização de corte de materiais metálicos.

O setor madeireiro apresenta características particulares relacionadas à variabilidade natural do material, necessidade de considerar veios e nós da madeira, e diferentes classes de qualidade que afetam a utilização de cada porção do material (Morabito, 1994). A otimização de corte de painéis, chapas e tábuas de madeira é uma etapa essencial para maximizar aproveitamento de recursos naturais e reduzir

impacto ambiental da exploração florestal. A indústria moveleira e construção civil são os principais consumidores de produtos madeireiros que se beneficiam de otimização de corte.

O setor têxtil enfrenta desafios específicos relacionados a padrões de corte de tecidos, consideração de estampas e direção da trama, e necessidade de corte de múltiplas camadas simultaneamente. A otimização de corte têxtil busca minimizar desperdício considerando estas particularidades, sendo fundamental para competitividade em um setor caracterizado por margens reduzidas e necessidade de produção em larga escala. A indústria de confecção e decoração são os principais aplicadores de técnicas de otimização de corte têxtil.

Outros setores com aplicações significativas incluem indústria vidraceira, onde corte de vidro plano deve considerar fragilidade do material e padrões específicos de utilização, indústria gráfica, onde otimização de impressão em folhas de papel reduz custos de matéria-prima e tempo de produção, indústria de embalagens, onde corte de cartolina e papelão requer consideração de dobras e encaixes, e indústria aeroespacial, onde otimização de corte de materiais compósitos de alto valor é crítica para viabilidade econômica de projetos.

A literatura apresenta diversos estudos de casos e validações de técnicas de otimização de corte em contextos industriais reais, demonstrando potencial de redução de desperdício, desperdício esse que geralmente é inferior a 10% comparado a métodos empíricos ou manuais (Morabito, 1994; Cherri, 2006). Estes ganhos de eficiência traduzem-se diretamente em benefícios econômicos e ambientais, justificando investimento em desenvolvimento e implementação de sistemas de otimização de corte (Bogue, 2020).

4 METODOLOGIA

Nessa seção será apresentada a metodologia empregada no desenvolvimento da ferramenta computacional de otimização de corte de materiais que foi desenvolvido nesse presente trabalho. Será abordado as tecnologias utilizadas, algoritmos, pesquisa, arquitetura e validação.

4.1 Tipo de pesquisa

Nesse trabalho foi desenvolvido um software cujo objetivo principal foi criar uma solução fácil de usar para otimização de corte de material, tendo como principal aplicação o setor industrial. A pesquisa foi aplicada, explorando diferentes técnicas de otimização combinatória para desenvolver o software e descrevendo a implementação dela e os resultados obtidos. A pesquisa que foi feita visa resolver problemas reais do dia a dia presente nas indústrias e gerar um produto útil ao final desse trabalho (Gil, 2019). O desenvolvimento tecnológico envolve a criação da ferramenta (software), prototipação e validação da ferramenta (Salomon, 2011).

4.2 Arquitetura do Sistema

O software foi desenvolvido seguindo uma arquitetura modular, com o objetivo de facilitar a manutenção, a escalabilidade e a inclusão de novas funcionalidades ao longo do tempo. A separação em módulos permite que cada componente seja desenvolvido, testado e atualizado de forma independente, reduzindo a complexidade geral do sistema.

4.2.1 Estrutura Modular

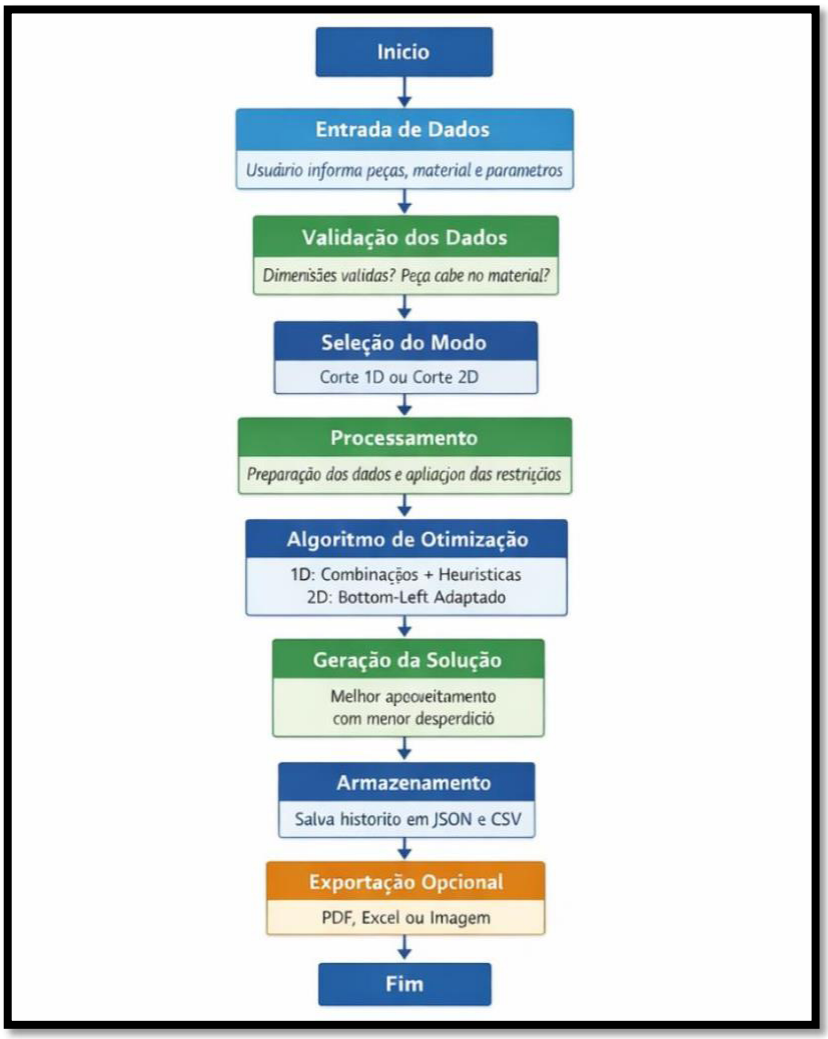
A arquitetura do sistema é composta por módulos bem definidos, cada um responsável por uma funcionalidade específica. O módulo de Interface Gráfica utiliza a biblioteca *Tkinter* para criação de janelas, formulários e elementos visuais, sendo responsável pela interação entre o usuário e a lógica de otimização. O módulo de Gerenciamento de Dados realiza o controle de peças, configurações do usuário e histórico de combinações, garantindo a persistência das informações por meio de arquivos JSON e CSV.

Os módulos de Otimização 1D e Otimização 2D implementam, respectivamente, os algoritmos responsáveis pelo corte linear e pelo corte em área. O módulo de Visualização é responsável pela geração de representações gráficas dos layouts de corte, enquanto o módulo de Exportação permite a geração de relatórios e resultados nos formatos PDF, Excel e imagens.

4.2.2 Fluxo de Dados

O funcionamento do *software* segue um fluxo de dados bem definido. Inicialmente, ocorre a entrada de dados, na qual o usuário informa as peças, dimensões do material a ser cortado e demais parâmetros de otimização. Em seguida, os dados passam por uma etapa de validação, garantindo que as informações estejam corretas antes do processamento. Abaixo, a figura 1 ilustra esse processo:

Figura 1- Fluxo de dados do sistema de otimização de corte



Fonte: autoria própria (2025).

Após a validação, os algoritmos realizam o processamento e a otimização do corte do material. Os resultados são então apresentados ao usuário por meio de visualizações gráficas e armazenados no histórico do sistema. Por fim, de forma opcional, os dados podem ser exportados em diferentes formatos para análise posterior.

4.2.3 Persistência de Dados

As configurações criadas pelo usuário, assim como o cadastro de peças, são armazenadas no arquivo *config.json*. O histórico de cálculos de corte linear e bidimensional é mantido em arquivos JSON e CSV (historico_1d.json, historico_2d.json, historico_1d.csv e historico_2d.csv). Além disso, modelos de configuração são armazenados na pasta *templates*/, permitindo reutilização e padronização de cenários.

4.3 Algoritmos Utilizados

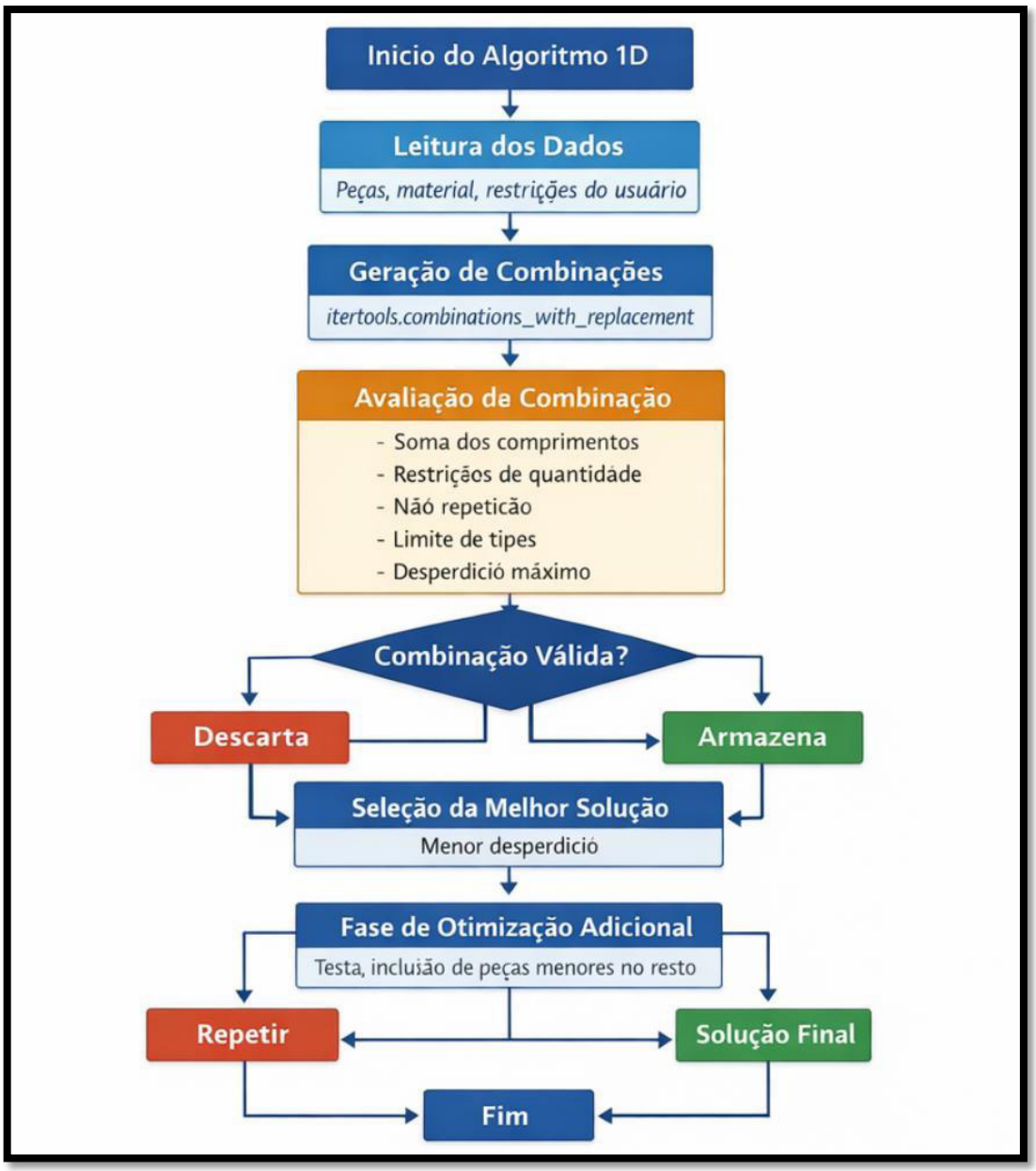
Durante o desenvolvimento da ferramenta, foram implementados algoritmos próprios para os processos de corte linear (1D) e corte em área (2D), incorporando heurísticas e restrições configuráveis pelo usuário. A definição desses algoritmos levou em consideração a necessidade de conciliar qualidade das soluções com viabilidade computacional, uma vez que problemas de corte apresentam crescimento combinatório do espaço de busca à medida que aumenta o número de peças e restrições envolvidas.

4.3.1 Algoritmo de Otimização para Corte Linear (1D)

O algoritmo desenvolvido para o corte linear (1D) utiliza uma estratégia de busca exaustiva combinada com heurísticas, com o objetivo de encontrar a melhor combinação possível de peças para minimizar o desperdício de material. Essa abordagem consiste em avaliar sistematicamente diferentes combinações de peças, respeitando as restrições impostas pelo usuário, como limites de quantidade, proibição de repetição, número máximo de tipos distintos e limites aceitáveis de desperdício.

A escolha por uma busca exaustiva controlada justifica-se pelo fato de que, em problemas de corte linear, a avaliação completa das combinações possíveis tende a produzir soluções de alta qualidade. No entanto, para evitar explosão combinatória e tempos de processamento inviáveis, o algoritmo incorpora filtros e critérios heurísticos que eliminam antecipadamente combinações inviáveis ou pouco promissoras. Dessa forma, o processo mantém um equilíbrio entre abrangência da busca e eficiência computacional. O funcionamento geral do algoritmo de corte linear está representado na figura 2.

Figura 2- Algoritmo de Otimização para Corte Linear (1D)



Fonte: autoria própria (2025).

4.3.1.1 Geração de Combinações

A geração de combinações é realizada utilizando a função *itertools.combinations_with_replacement,* respeitando limites de quantidade e restrições definidas pelo usuário.

4.3.1.2 Avaliação das Combinações

Cada combinação gerada é avaliada considerando se a soma dos comprimentos das peças não ultrapassa o comprimento do material disponível, se atende às quantidades mínimas e máximas por peça, se respeita a restrição de não repetição, se atende ao limite de tipos diferentes de peças, se o desperdício gerado está dentro do limite máximo permitido e se o restante atende ao mínimo útil configurado.

4.3.1.3 Seleção da Melhor Solução

A melhor solução inicial é selecionada com base na combinação que apresenta a maior soma de comprimentos utilizados, resultando no menor desperdício possível.

4.3.1.4 Fase de Otimização Adicional

Após a seleção da solução inicial, o algoritmo executa uma fase adicional de otimização, na qual o espaço restante é analisado e peças menores são adicionadas de forma iterativa, desde que todas as restrições do usuário sejam respeitadas.

4.3.1.5 Métodos de Seleção

O sistema oferece dois métodos de seleção de peças: o método Selecionar, que utiliza apenas as peças marcadas pelo usuário, e o método Excluir, que utiliza todas as peças, exceto aquelas marcadas para exclusão.

4.3.2 Algoritmo de Otimização para Corte em Área (2D)

O algoritmo de corte bidimensional utiliza uma variante adaptada do algoritmo *Bottom-Left,* modificada para suportar múltiplos tipos de peças e restrições específicas do problema.

4.3.2.1 Algoritmo Bottom-Left

O algoritmo *Bottom-Left* posiciona as peças iniciando pelo canto inferior esquerdo do material base, movendo-se progressivamente para a direita e depois para cima, buscando minimizar os espaços vazios.

4.3.2.2 Adaptações Implementadas

Foram implementadas adaptações que incluem suporte a múltiplos tipos de peças, rotação quando permitida, margens de segurança entre peças, respeito a quantidades mínimas e máximas e consideração de prioridades de posicionamento.

4.3.2.3 Geração de Combinações de Quantidades

O algoritmo calcula inicialmente a quantidade máxima teórica de cada peça com base na área do material. Em seguida, gera combinações dentro dos limites configurados pelo usuário, aplicando limites dinâmicos para evitar explosão combinatória, priorizando peças de maior área e maior prioridade.

4.3.2.4 Estratégia de Busca Inteligente

A estratégia de busca varia conforme a quantidade de tipos de peças, utilizando testes completos para poucos tipos e amostragem controlada quando o número de peças aumenta.

4.3.2.5 Detecção de Colisões

A detecção de colisões é realizada por meio da verificação de sobreposição entre peças representadas por coordenadas geométricas. O algoritmo testa diferentes posições seguindo a estratégia *Bottom-Left* e considera margens de segurança definidas pelo usuário.

4.3.2.6 Fase de Preenchimento de Espaços

Após a definição da melhor combinação inicial, o sistema identifica espaços vazios no layout e tenta preenchê-los com peças menores de forma iterativa, aumentando o aproveitamento do material.

4.3.3 Heurísticas e Otimizações

O desempenho e a qualidade das soluções são aprimorados por meio de heurísticas como ordenação por prioridade, ordenação por tamanho, limitação dinâmica do espaço de busca e validação prévia da viabilidade do corte.

4.4 Tecnologias Empregadas

O desenvolvimento do software utilizou tecnologias de código aberto e multiplataforma, visando portabilidade e facilidade de manutenção.

4.4.1 Linguagem de Programação

A linguagem *Python* 3.x foi escolhida devido à sua simplicidade, legibilidade, ampla disponibilidade de bibliotecas e compatibilidade com diferentes sistemas operacionais.

4.4.2 Interface Gráfica

A interface gráfica foi desenvolvida com *Tkinter* e organizada em um sistema de abas, contemplando um guia informativo, funcionalidades para corte 1D e funcionalidades para corte 2D.

4.4.3 Bibliotecas Auxiliares

Bibliotecas como *Pillow, ReportLab e OpenPyXL* foram utilizadas para exportação de imagens, geração de relatórios em PDF e criação de planilhas Excel. O funcionamento da otimização não depende dessas bibliotecas, apenas as funcionalidades de exportação.

4.4.4 Estrutura de Dados

Foram utilizadas estruturas nativas do *Python*, como listas, dicionários e tuplas, além de JSON para serialização e CSV para armazenamento do histórico de cálculos.

4.5 Validação

A validação da ferramenta foi realizada por meio de cenários simulados representativos do ambiente industrial. Os testes e validações foram executados em um notebook *Lenovo Gaming* 3i, com as seguintes especificações técnicas: Modelo: *Lenovo Gaming 3i*, Processador: Intel Core i5 10300H (4 núcleos, 8 threads, frequência base de 2,5 GHz, turbo até 4,5 GHz), memória RAM: 16 GB DDR4 operando a 2.933 MHz, sistema operacional: Windows 10 Home, armazenamento: SSD NVMe PCIe (256 GB), placa de vídeo: NVIDIA GTX 1650 de 4 GB (não utilizada diretamente pelos algoritmos, mas presente no sistema)

4.5.1 Estratégia de Validação

Devido à ausência de dados reais de empresas, a validação foi baseada em cenários simulados fundamentados na literatura e em situações típicas da prática industrial.

4.5.2 Cenários de Teste

Foram definidos cenários específicos para os modos de corte 1D e 2D, contemplando diferentes materiais, restrições e objetivos de otimização.

4.5.3 Métricas de Avaliação

As métricas utilizadas incluíram eficiência de uso, desperdício, tempo de processamento e número de peças utilizadas ou posicionadas.

4.5.4 Análise dos Resultados

Os resultados foram analisados considerando comparação com métodos manuais, atendimento às restrições configuradas e desempenho computacional.

4.5.5 Limitações da Validação

Reconhece-se que a validação baseada em cenários simulados apresenta limitações, como ausência de variabilidade real e integração com sistemas industriais, aspectos que podem ser abordados em trabalhos futuros.

5 RESULTADOS E DISCUSSÃO

Nesse capítulo será apresentado os resultados obtidos durante os testes de validação do software. Foram criados vários cenários de testes com diferentes cenários industriais e esses testes foram realizados usando a ferramenta desenvolvida nesse trabalho. Os resultados são analisados de forma crítica e fazendo um relacionamento com o conteúdo presente na revisão bibliográfica.

5.1 Cenários de Teste

Para validar o funcionamento do sistema desenvolvido, foram definidos seis cenários de teste, três para corte linear (1D) e três para corte em área (2D). Os cenários foram criados simulando diferentes contextos indústrias para testar as funcionalidades que foram desenvolvidas.

5.1.1 Cenário 1D-1: Corte de Bobina de Fibra de Carbono

Nesse primeiro cenário, ocorre uma simulação de otimização de corte de uma bobina de fibra de carbono (100 metros), com 16 peças diferentes (comprimentos entre 19,520 m e 63,550 m). Objetivo: minimizar desperdício em material de alto custo.

5.1.2 Cenário 1D-2: Corte de Barras Metálicas com Restrições de Quantidade

Simula o corte de barras metálicas (40 metros), com 5 tipos de peças diferentes. Testa também as restrições de quantidade mínima e máxima, utiliza valores decimais nos comprimentos para simular um cenário mais realistas.

5.1.3 Cenário 1D-3: Corte com Múltiplas Restrições

Nesse cenário ocorre uma simulação com um material de 75 metros com 8 tipos de peças. Esse teste valida o funcionamento simultâneo de múltiplas restrições: não repetição, sistema de prioridades e limites de quantidade máxima.

5.1.4 Cenário 2D-1: Corte de Chapa de MDF

Chapa de MDF com dimensões de 2,0 m x 1,5 m (3,0 m²), com 8 tipos de peças (portas, janelas, painéis e detalhes). Algumas peças possuem rotação permitida. Testa o algoritmo *bottom-left* modificado.

5.1.5 Cenário 2D-2: Corte com Rotação Permitida

Chapa de 3,0 m x 2,0 m (6,0 m²), com 7 tipos de peças, todas com rotação permitida. Valida o funcionamento do algoritmo de rotação automática.

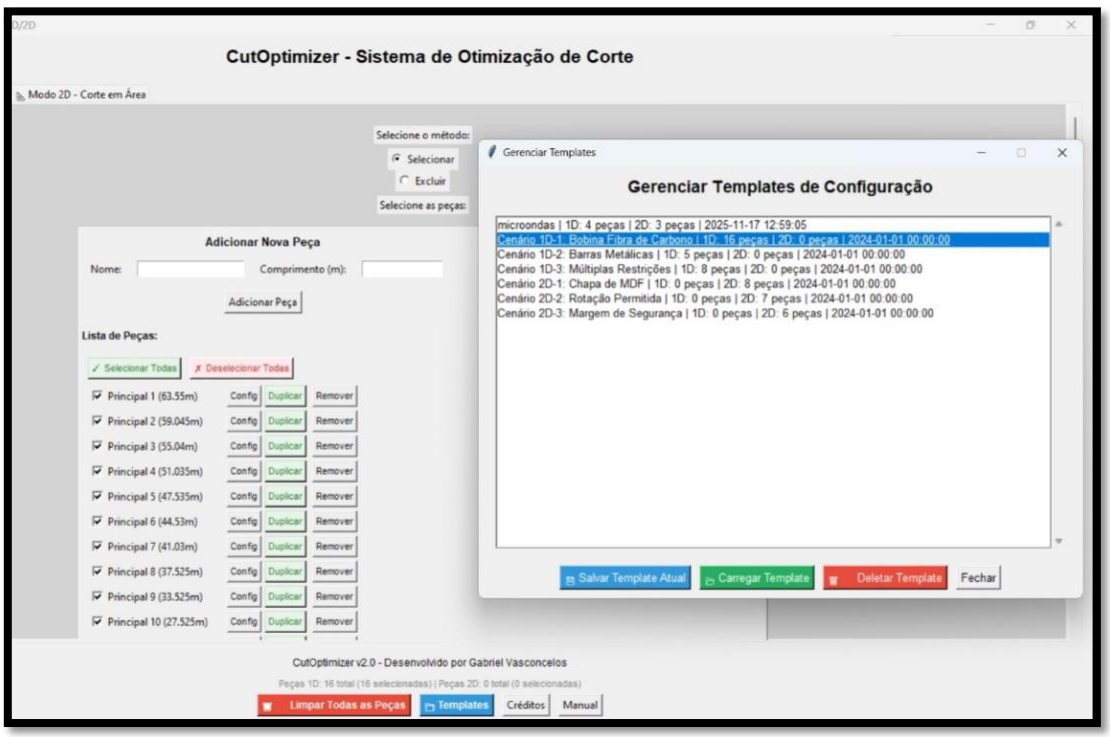
5.1.6 Cenário 2D-3: Corte com Margem de Segurança

Chapa de 2,5 m x 1,8 m (4,5 m²), com 6 tipos de peças, todas com rotação permitida. Testa a aplicação de margem de segurança de 0,02 m entre peças.

5.2 Resultados Obtidos

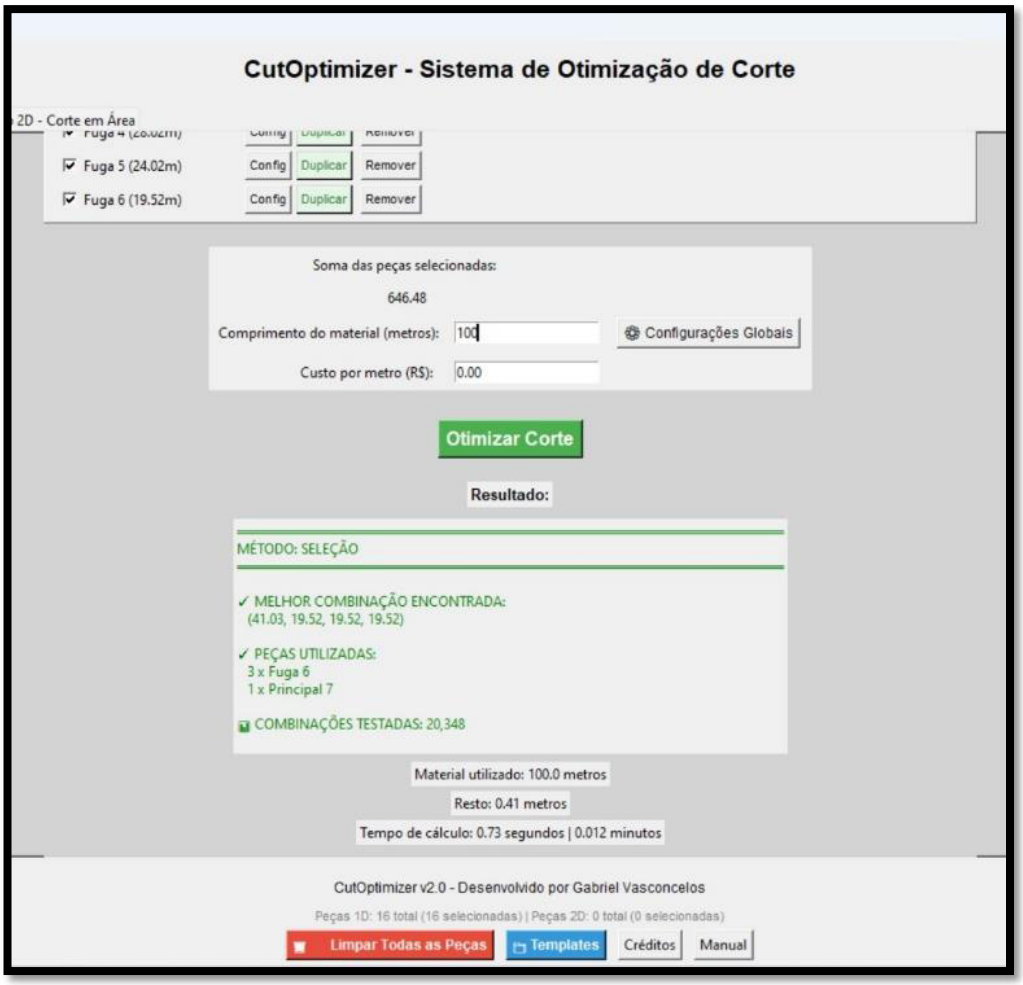
5.2.1 Resultados do Cenário 1D-1: Corte de Bobina de Fibra de Carbono

Figura 3- Interface do software durante a configuração do cenário de teste 1D



Fonte: autoria própria (2025).

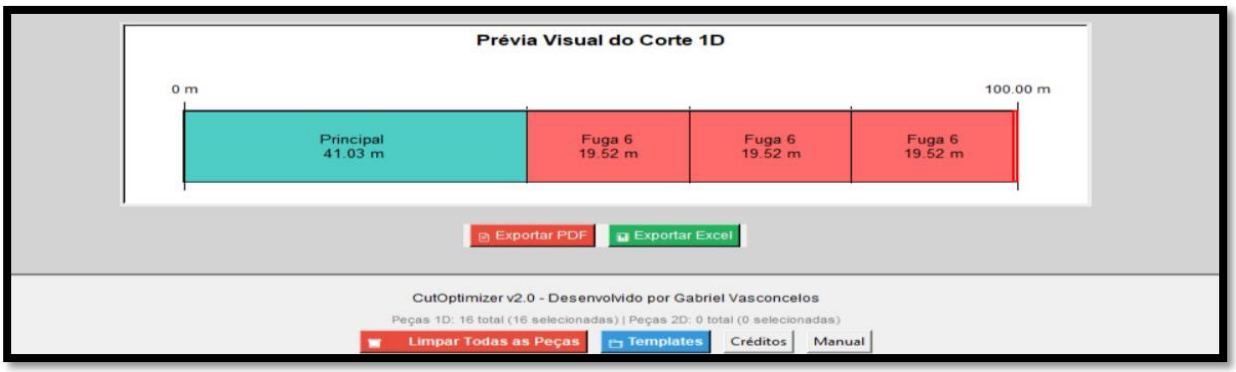
Figura 4- Resultado final obtido no cenário 1D-1



Fonte: autoria própria (2025).

O sistema desenvolvido nesse trabalho (figuras 3 e 4) encontrou uma combinação que utilizou 99,59 metros (eficiência de 99,59%, desperdício de 0,41 m): 1 Principal 7 (41,030 m) e 3 Fuga 6 (19,520 m cada). Foram testadas 20.348 combinações em 0,73 segundos (taxa de 27.874 combinações/segundo).

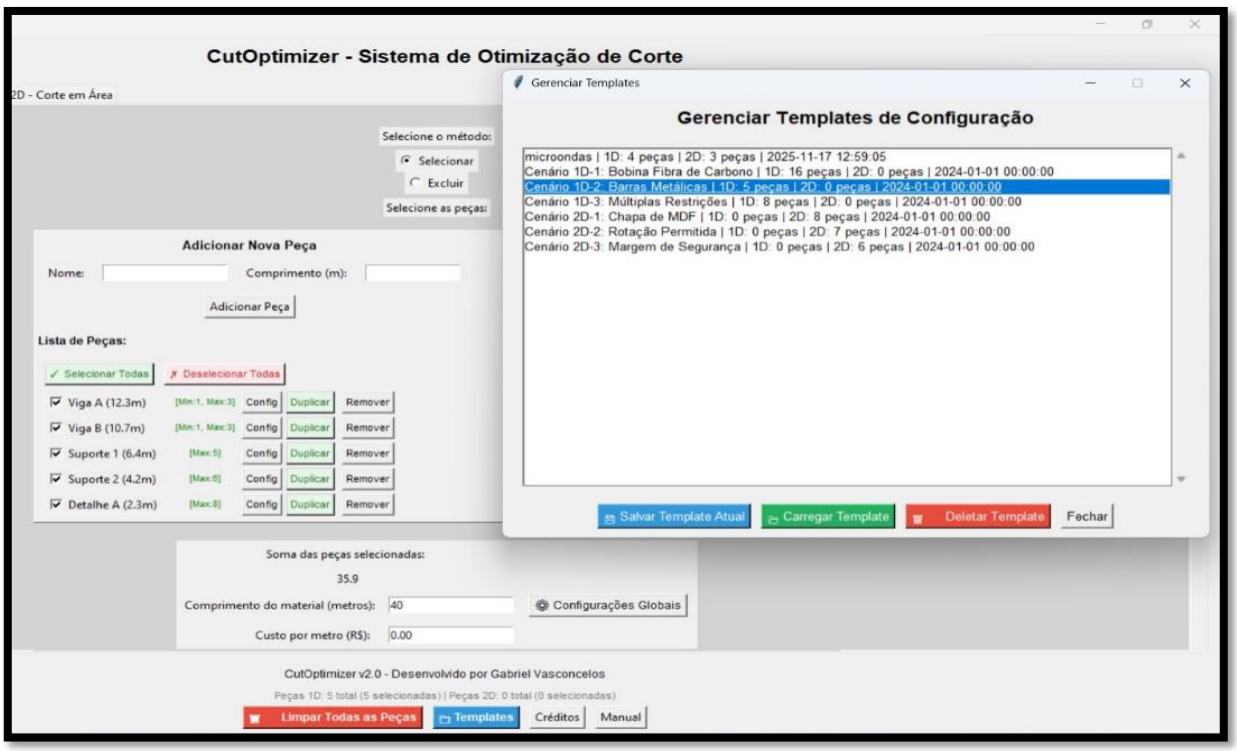
Figura 5- Visualização gráfica do layout de corte do teste 1D-1



Fonte: autoria própria (2025).

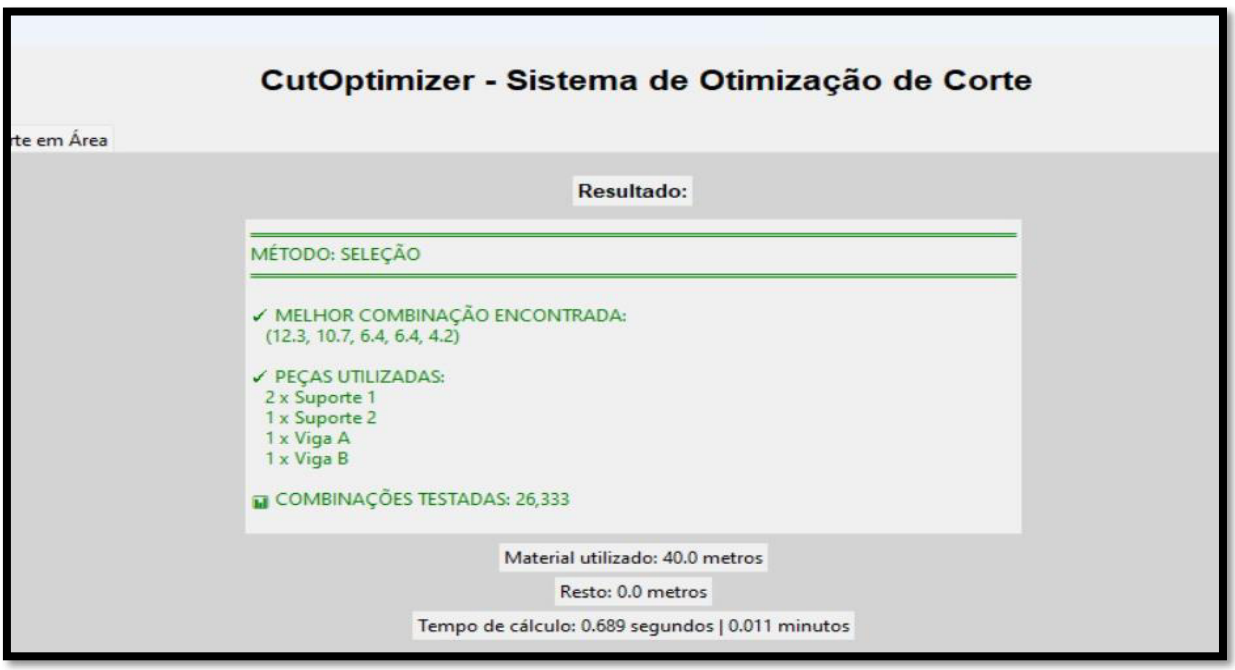
5.2.2 Resultados do Cenário 1D-2: Corte de Barras Metálicas com Restrição na Quantidade

Figura 6- Interface do software durante a configuração do cenário de teste 1D-2



Fonte: autoria própria (2025).

Figura 7- Resultado final obtido 1D-2

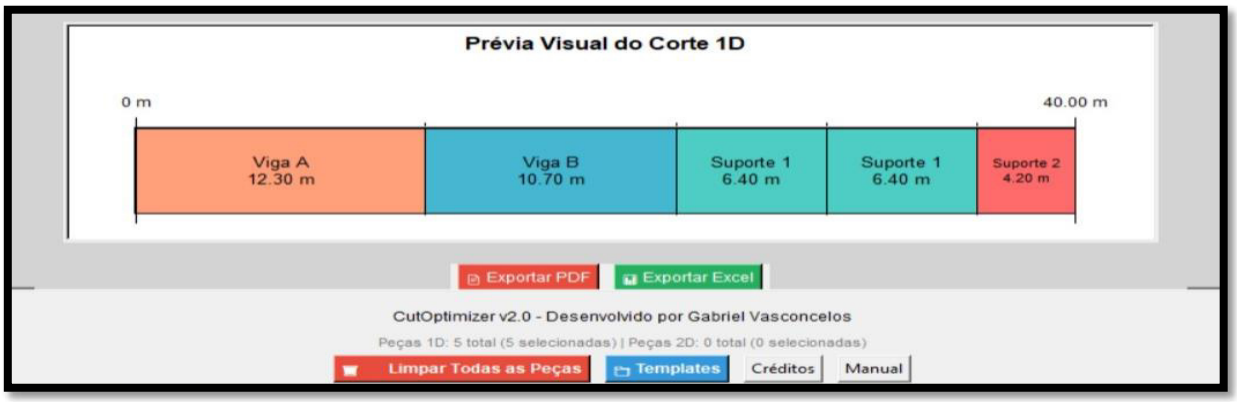


Fonte: autoria própria (2025).

O sistema desenvolvido nesse trabalho encontrou uma combinação que utilizou completamente o material (eficiência de 100%): 1 x Viga A (12,3 m), 1 x Viga B (10,7 m), 2 x Suporte 1 (6,4 m cada) e 1 x Suporte 2 (4,2 m). Foram testadas 26.333 combinações em 0,689 segundos (taxa de 38.219 combinações/segundo).

A combinação respeitou todas as restrições de quantidade configuradas. O algoritmo encontrou aproveitamento perfeito mesmo com valores decimais, demonstrando a eficácia da estratégia de busca.

Figura 8- Visualização gráfica do layout de corte obtido 1D-2



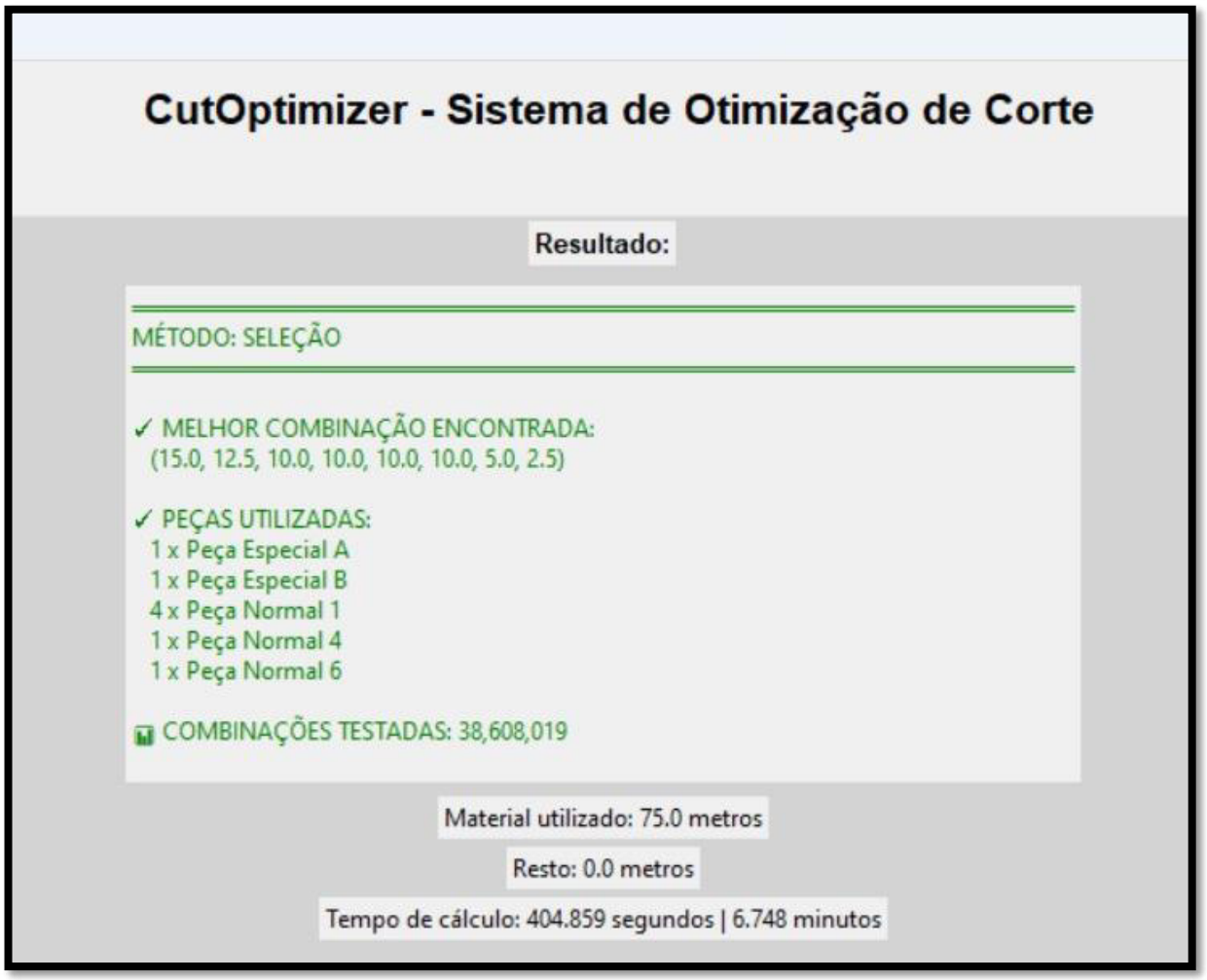
Fonte: autoria própria (2025).

5.2.3 Resultados do Cenário 1D-3: Corte com Múltiplas Restrições

Figura 9- Interface do software durante a configuração do teste 1D-3

Fonte: autoria própria (2025).

Figura 10- Resultado final obtido 1D-3

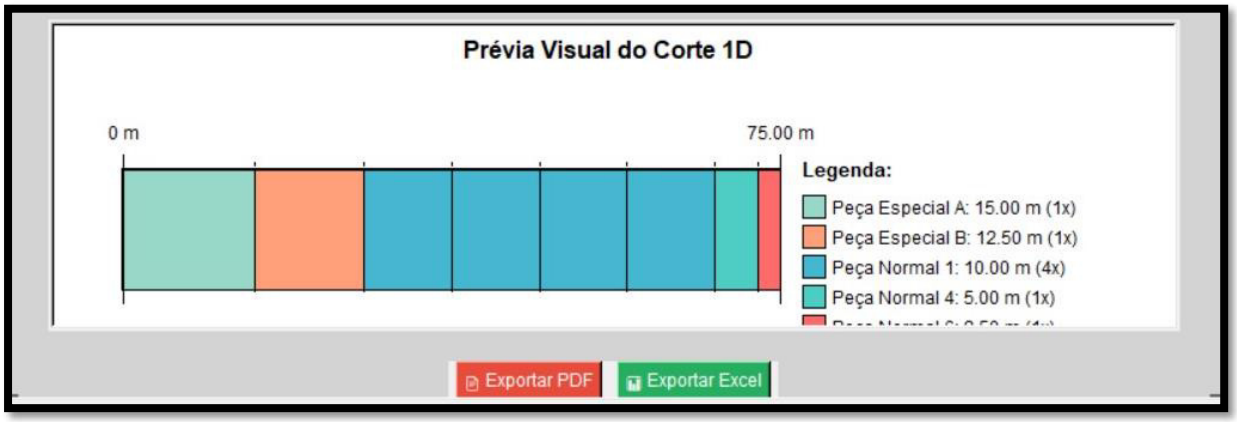


Fonte: autoria própria (2025).

O sistema desenvolvido nesse trabalho encontrou uma combinação que utilizou completamente o material (eficiência de 100%): 1 x Peça Especial A (15,0 m), 1 x Peça Especial B (12,5 m), 4 x Peça Normal 1 (10,0 m cada), 1 x Peça Normal 4 (5,0 m) e 1 x Peça Normal 6 (2,5 m). Foram testadas 38.608.019 combinações em 404,859 segundos (6,748 minutos), resultando em uma taxa de 95.350 combinações por segundo.

A combinação respeitou todas as restrições: não repetição, prioridades e limites de quantidade.

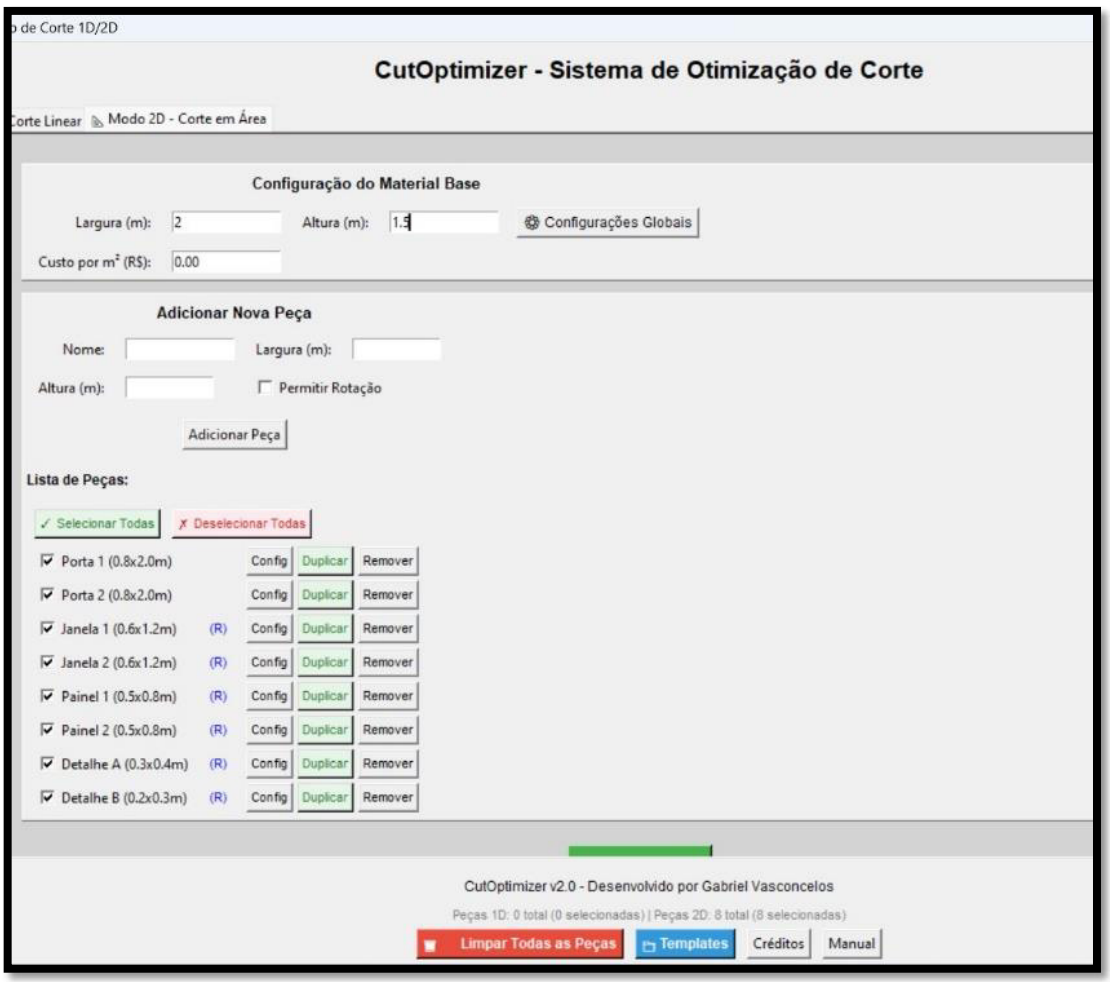
Figura 11- Visualização gráfica do layout de corte obtido no cenário 1D-3



Fonte: autoria própria (2025).

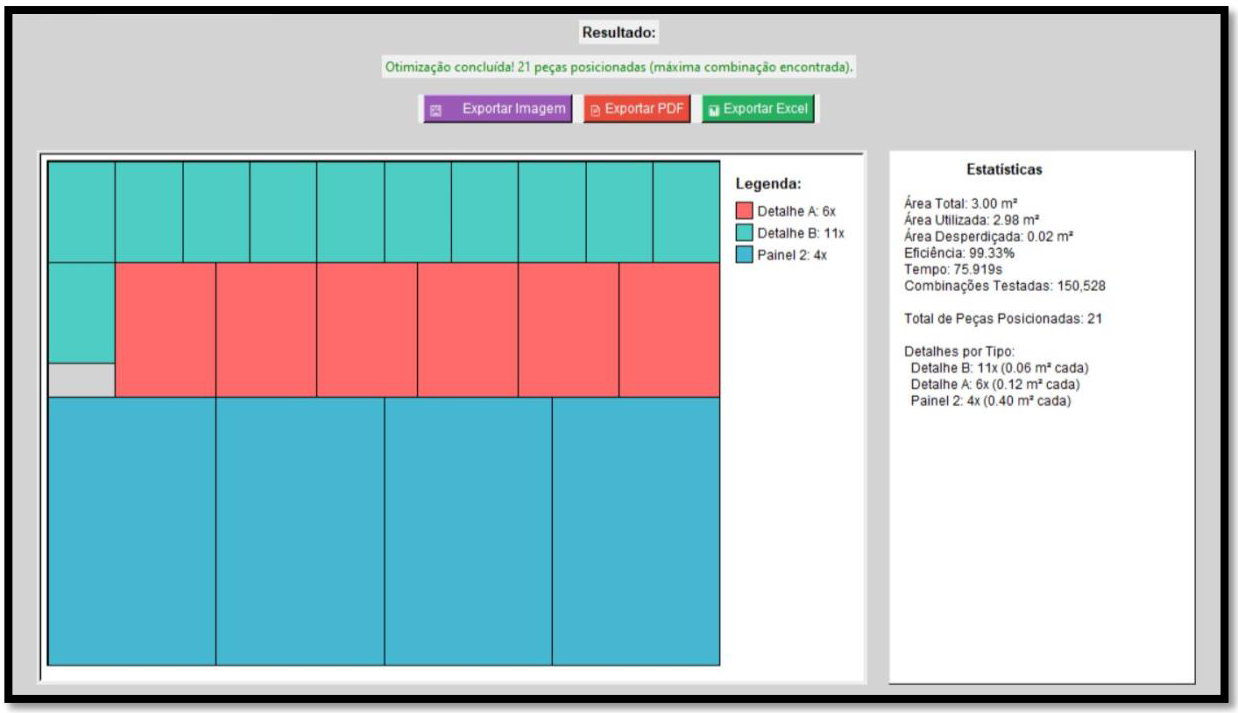
5.3.2 Resultados do Cenário 2D-1: Corte de Chapa de MDF

Figura 12- Interface do software durante a configuração do teste 2D-1



Fonte: autoria própria (2025).

Figura 13- Resultado final obtido 2D-1



Fonte: autoria própria (2025).

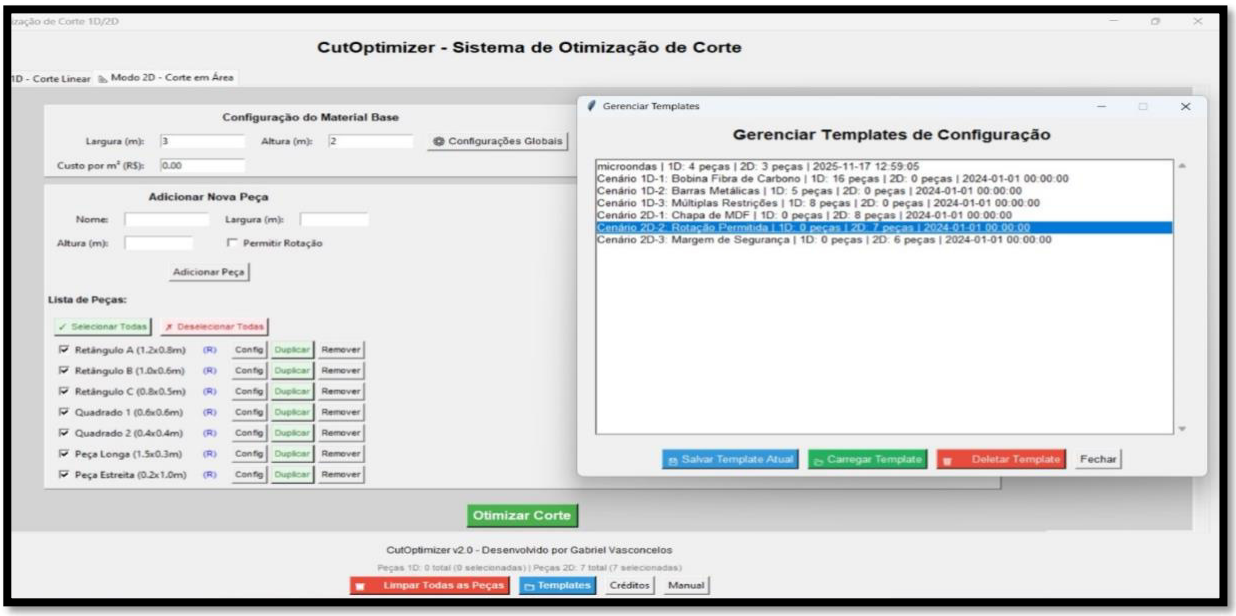
Figura 14- Estatísticas do resultado final obtido 2D-1

Fonte: autoria própria (2025).

O sistema desenvolvido nesse trabalho encontrou uma combinação que utilizou 2,98 m² (eficiência de 99,33%, desperdício de 0,02 m²): 11 x Detalhe B, 6 x Detalhe A e 4 x Painel 2, totalizando 21 peças. Foram testadas 150.528 combinações em 75,919 segundos (1,265 minutos), resultando em uma taxa de 1.983 combinações por segundo.

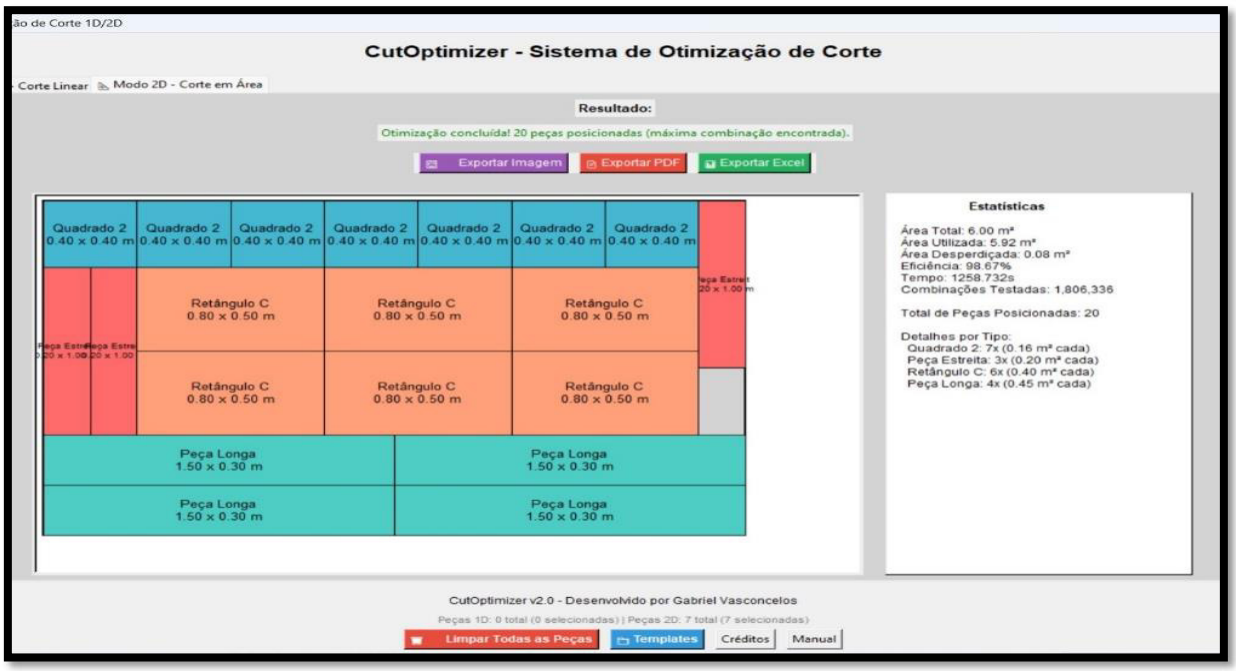
5.2.5 Resultados do Cenário 2D-2: Corte com Rotação Permitida

Figura 15- apresenta a interface do software durante a configuração do teste 2D-2



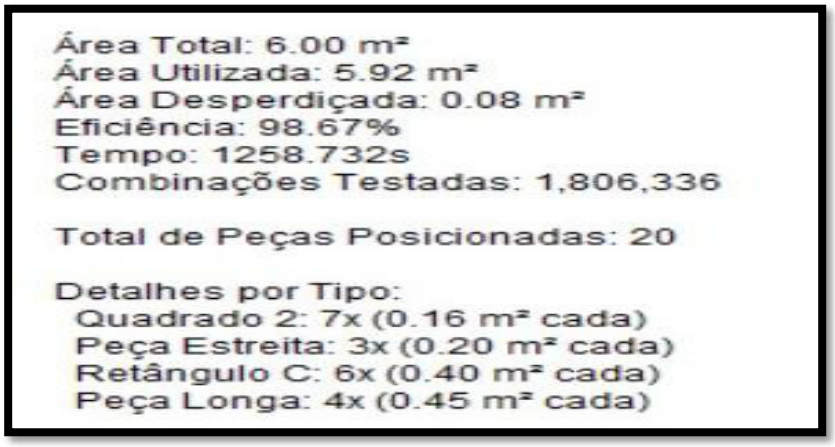
Fonte: autoria própria (2025).

Figura 16- Resultado final obtido 2D-2



Fonte: autoria própria (2025).

Figura 17- Estatísticas do resultado final obtido 2D-2

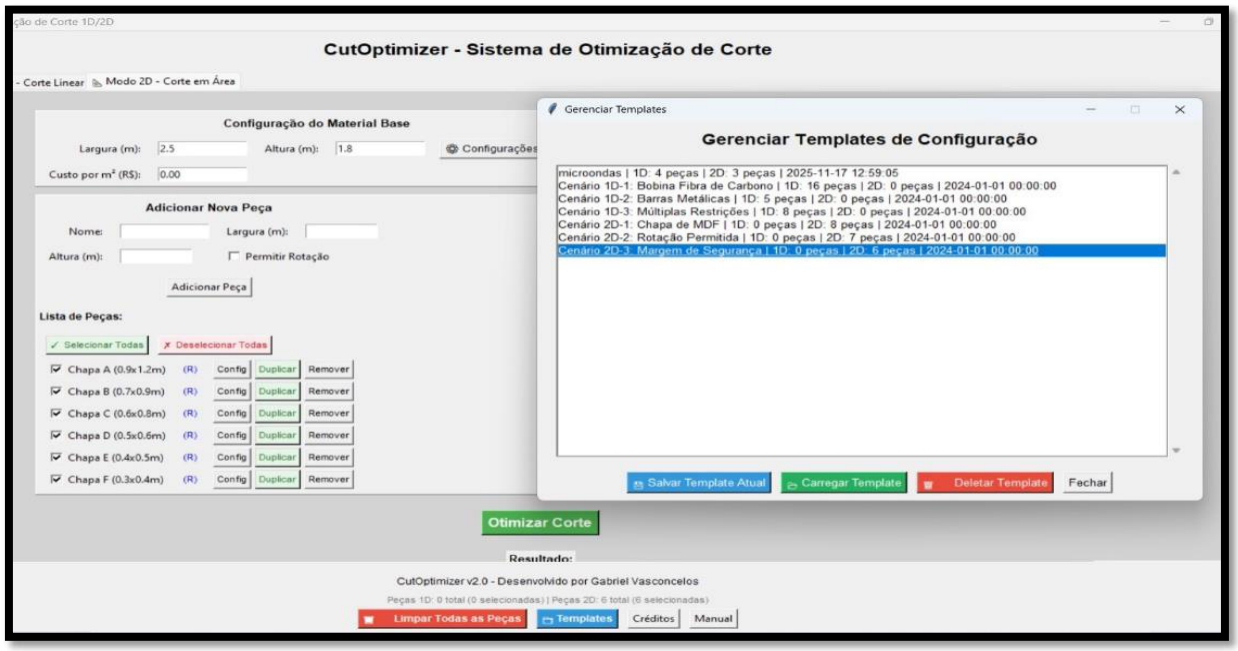


Fonte: autoria própria (2025).

O sistema desenvolvido nesse trabalho encontrou uma combinação que utilizou 5,92 m² (eficiência de 98,67%, desperdício de 0,08 m²): 7 x Quadrado 2, 3 x Peça Estreita, 6 x Retângulo C e 4 x Peça Longa, totalizando 20 peças. Foram testadas 1.806.336 combinações em 1.258,732 segundos (20,979 minutos), resultando em uma taxa de 1.435 combinações por segundo. As peças retangulares foram rotacionadas quando necessário para melhor aproveitamento.

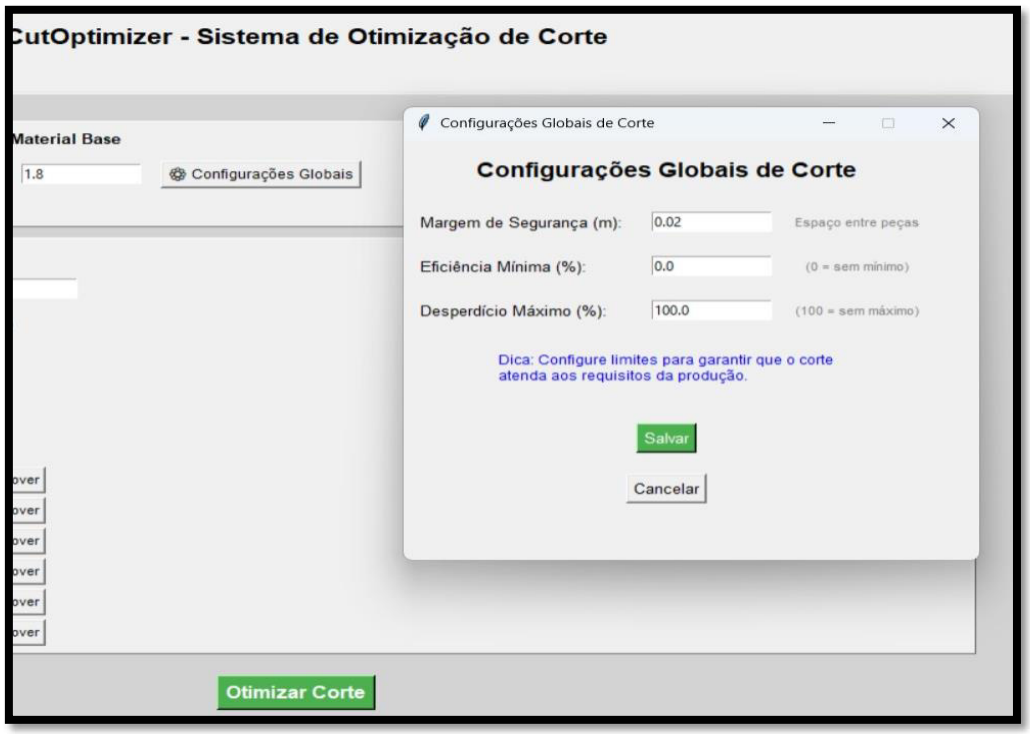
5.2.6 Resultados do Cenário 2D-3: Corte com Margem de Segurança

Figura 18- Interface do software durante a configuração do teste 2D-3



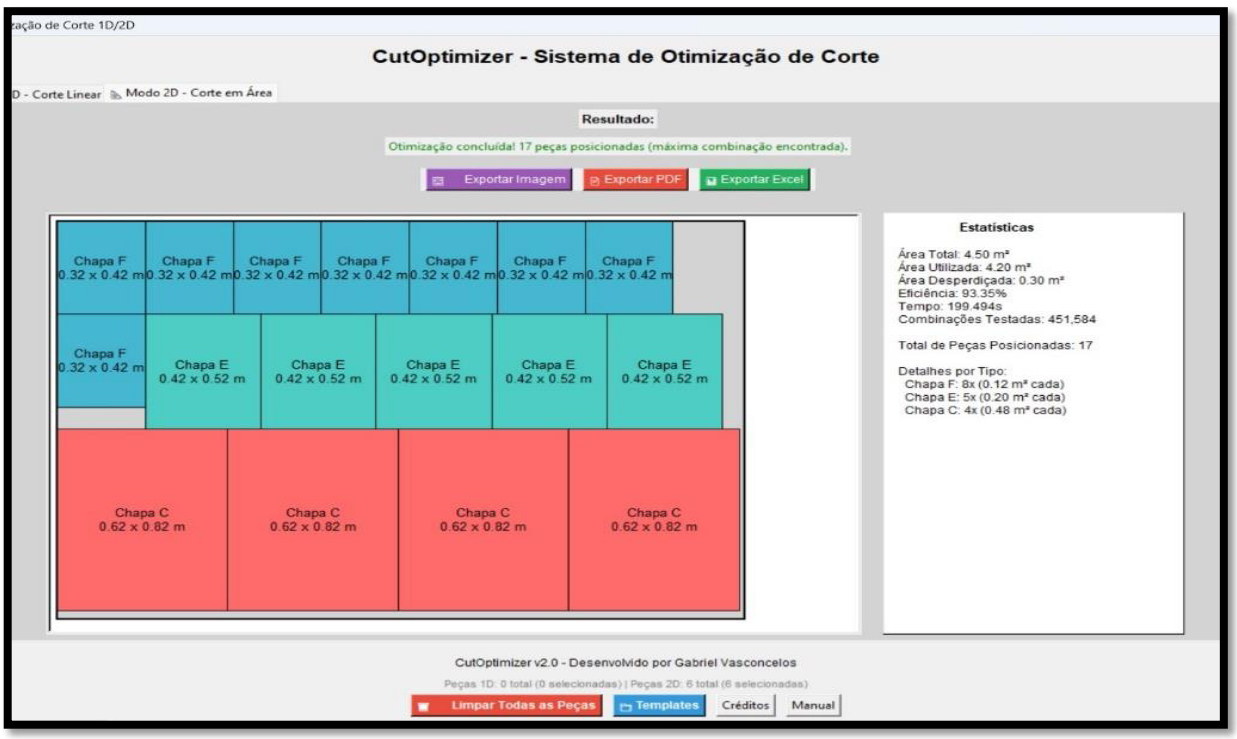
Fonte: autoria própria (2025).

Figura 19- Interface do software durante a configuração da margem de segurança 2D-3



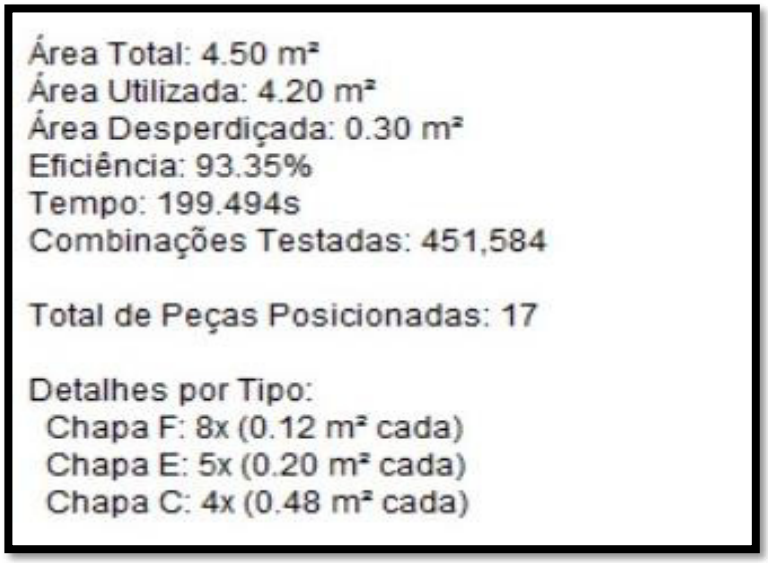
Fonte: autoria própria (2025).

Figura 20- Resultado final obtido 2D-3



Fonte: autoria própria (2025).

Figura 21- Estatísticas do resultado final obtido 2D-3



Fonte: autoria própria (2025).

O sistema desenvolvido nesse trabalho encontrou uma combinação que utilizou 4,20 m² (eficiência de 93,35%, desperdício de 0,30 m²): 8 x Chapa F, 5 x Chapa E e 4 x Chapa C, totalizando 17 peças. Foram testadas 451.584 combinações em 192,94 segundos (3,216 minutos), resultando em uma taxa de 2.343 combinações por segundo.

A combinação prioriza peças menores para preencher espaços. A margem de segurança de 2 centímetros reduz o espaço utilizável, gerando uma eficiência menor que nos outros cenários que foram testados anteriormente sem margem, mas ainda representa um aproveitamento elevado.

5.3 Análise e Discussão dos Resultados

5.3.1 Análise do Cenário 1D-1

O resultado obtido demonstra uma alta eficiência em aplicações sem restrições complexas. A eficiência de 99,59% demonstra um elevado aproveitamento do material, com desperdício de apenas 0,41%. O tempo pequeno de 0,73 segundos, prova a baixa complexidade do cenário testado, permitindo que em cenários assim o uso em tempo real seja aplicado, já que a velocidade para se obter o resultado é alta, fornecendo o resultado de forma quase instantânea.

5.3.2 Análise do Cenário 1D-2

O resultado obtido nesse cenário demostra a elevada eficiência para esse tipo de corte, mesmo com restrições de quantidade. Os valores das restrições também foram aplicados em valores decimais, o que adiciona um ponto de maior atenção com os valores, mas mesmo assim o a eficiência foi de 100%, o que indica aproveitamento total do material cortado. O tempo de 0,689 segundos mostra que mesmo com as restrições aplicadas é possível obter o resultado em tempo quase nulo. Possivelmente esse tempo menor também foi fruto do menor número de peças utilizados (5 versus 16).

5.3.3 Análise do Cenário 1D-3

O resultado demonstra que o software desenvolvido tem capacidade de encontra soluções mesmo com várias restrições ao mesmo tempo. Novamente foi obtido uma eficiência de 100%, ou seja, o material foi aproveitado totalmente. As combinações que foram feitas apresentam o comportamento de heurística da priorização: as peças com elevadas prioridades foram incluídas (peças especiais), e a ferramenta também considerou tanto a prioridade como a capacidade de preenchimento das peças. O tempo de 6,7 minutos para testar 38,6 milhões de combinações representa um desempenho aceitável para a complexidade do cenário que foi testado. A estratégia de geração iterativa mostrou elevada capacidade para evitar problemas de memória, no decorrer do desenvolvimento esses problemas eram constantes ao simular cenários mais complexos, mas esse problema foi resolvido e validado nesse teste.

5.3.4 Análise do Cenário 2D-1

O resultado desse cenário que foi testado demonstra a eficácia do algoritmo *bottom-left* modificado. A eficiência de 99,33% com desperdício de apenas 0,67% indica aproveitamento quase completo do material que foi cortado. O tempo de 75,919 segundos (1,3 minutos) reflete já uma maior complexidade computacional dos problemas 2D, pois nos cenários 2D já é exigido verificações como verificação de colisões e posicionamento espacial que não existem nas combinações 1D. A taxa de

1.983 combinações/segundo é menor que nos cenários 1D, como esperado. As peças menores foram muito utilizadas, demonstrando a capacidade de preencher espaços vazios no material. As portas não foram incluídas nas combinações pois sua altura (2,0 m) ultrapassa a altura do material que está sendo cortado (1,5 m), isso valida a verificação de correta das dimensões das peças.

5.3.5 Análise do Cenário 2D-2

O resultado que foi obtido nesse teste prova o funcionamento do algoritmo de rotação automática. A eficiência de 98,67% (desperdício de 1,33%) indica aproveitamento quase completo. O tempo de 1.258,732 segundos (21 minutos) demonstra a complexidade extra de se testar múltiplas orientações. A taxa de 1.435 combinações/segundo é menor que no 2D-1 por causa da necessidade de testar orientações rotacionais quando necessário, provando a eficácia da estratégia de rotação automática.

5.3.6 Análise do Cenário 2D-3

O resultado que foi obtido demostra o sucesso no uso da ferramenta em combinações 2D com margem de segurança aplicada. A eficiência de 93,35% indica aproveitamento da maior parte da área do material cortado, mesmo com a restrição que foi adicionada. O tempo de 192,94 segundos (3,2 minutos) reflete complexidade intermediária. A taxa de 2.343 combinações/segundo é superior ao 2D-2, muito possivelmente devido ao menor número de peças e menor área que foram usadas nesse cenário. As peças menores foram muito utilizadas, demonstrando eficiência para preencher espaços com margens. A margem reduz o espaço utilizável, resultando em eficiência menor que os cenários sem margem, mas ainda representando aproveitamento elevado considerando a restrição que foi aplicada.

5.3.7 Comportamento das Restrições

A validação das restrições funcionou como esperado. A não repetição garantiu que peças especiais aparecessem no máximo uma vez. O sistema de prioridades influenciou a ordem de processamento. Os limites de quantidade máxima foram

respeitados, e o algoritmo encontrou um equilíbrio entre seguir os limites e maximizar o aproveitamento.

5.3.8 Comparação entre Cenários Simples e Complexos

Os três cenários 1D apresentam diferenças consideráveis: 1D-1 (0,73 s, 20.348 combinações), 1D-2 (0,689 s, 26.333 combinações) e 1D-3 (404,859 s, 38.608.019 combinações). Com base nesses dados é possível ver o impacto das restrições na quantidade de combinações geradas e no tempo necessário para obter os resultados. A maior quantidade de combinações está diretamente ligada a um maior uso computacional, como consequência é necessário maior tempo para obter os resultados. O 1D-2 foi processado em tempo parecido ao do 1D-1, possivelmente devido ao menor número de peças. O 1D-3 mostrou um aumento exponencial no tempo (587 vezes mais lento que 1D-2). Em todos os cenários foi obtido eficiências muito elevadas (99,59%, 100% e 100%), demonstrando capacidade de encontrar soluções de alta qualidade, embora com custos computacionais diferentes.

5.3.9 Comparação entre Cenários 2D

Para os cenários 2D foi encontrado diferenças significativas: 2D-1 (75,919 s, 150.528 combinações, 99,33%), 2D-2 (1.258,732 s, 1.806.336 combinações, 98,67%) e 2D-3 (192,94 s, 451.584 combinações, 93,35%). O 2D-2 precisou de 16,6 vezes mais tempo que o 2D-1 pois precisou testar múltiplas orientações. O 2D-3 foi processado em tempo menor que o 2D-2, muito possivelmente devido ao menor número de peças e menor área do material cortado. Sobre a eficiência, foi possível observar uma redução progressiva: 99,33% → 98,67% → 93,35%, sendo a redução mais significativa entre 2D-2 e 2D-3, refletindo o impacto da margem de segurança.

5.3.10 Comparação com Métodos Manuais

A determinação manual de qual combinação a melhor combinação para o corte de um material com várias restrições e peças seria extremamente trabalhosa, demorada e com altas chances de erro. No cenário 1D-3, o algoritmo testou mais de 38 milhões de combinações em 6,7 minutos, um operador humano não conseguiria

testar milhões de combinações em um tempo aceitável, é humanamente impossível. A ferramenta automatiza esse processo, permitindo inclusive visualizar os cortes para um melhor entendimento do operador.

5.4 Comparação entre Corte 1D e Corte 2D

A comparação entre o corte 1D e corte 2D apresentam como principal diferença a complexidade computacional. No corte 1D a complexidade está mais relacionada ao número de combinações, no corte 2D é adicionado uma verificação de colisões e também a parte de posicionamento espacial bidimensional de cada peça no material a ser cortado.

Os tempos de processamento mostram o aumento da complexidade: 1D-1 (0,73 s, 20.348 combinações) versus 2D-1 (75,919 s, 150.528 combinações), 2D-2 (1.258,732 s, 1.806.336 combinações) e 2D-3 (192,94 s, 451.584 combinações). A taxa de processamento (combinações por segundo) também apresentou uma diferença considerável: 1D processou aproximadamente 27.874 combinações/segundo, enquanto 2D processou entre 1.435 e 2.343 combinações/segundo (redução de 12 a 19 vezes).

Sobre a eficiência, observou-se uma redução progressiva: 99,59% (1D-1) → 99,33% (2D-1) → 98,67% (2D-2) → 93,35% (2D-3). A redução entre 1D e 2D é pequena, indicando que o algoritmo 2D mantém alta qualidade. Todos os métodos alcançaram eficiências muito elevadas (acima de 93%), mostrando uma elevada capacidade de encontrar soluções de alta qualidade, embora com custos computacionais diferentes.

5.5 Limitações Identificadas

5.5.1 Limitações do Algoritmo

O tempo de processamento pode se tornar proibitivo para problemas com maior número de tipos de peças ou dimensões muito grandes. A estratégia de busca exaustiva possui complexidade exponencial, limitando a escalabilidade. A limitação do espaço de busca através de limites dinâmicos pode potencialmente excluir combinações ótimas em alguns casos. No algoritmo 2D, a verificação de colisões e

teste de múltiplas orientações aumentam significativamente o tempo de processamento. A aplicação de margens de segurança reduz a eficiência, representando um trade-off necessário entre eficiência e requisitos práticos.

5.5.2 Limitações do Ambiente de Teste e dos Dados

Os cenários que foram testados utilizam dados simulados, não representando a variabilidade de situações reais. Os cenários que foram testados possuem números relativamente pequenos de peças e dimensões moderadas. Problemas com maior diversidade ou configurações mais complexas podem apresentar desafios adicionais não avaliados.

5.5.3 Impacto das Limitações

As limitações não comprometem os resultados para os cenários que foram testados, indicam apenas áreas para melhorias futuras. Antes de qualquer veredito, ó o uso constante vai dizer se é necessário ou não melhorias no algoritmo, se for necessário, algoritmos mais complexos podem lidar com esses problemas maiores.

6 CONCLUSÃO

No decorrer desse trabalho foi apresentado o desenvolvimento de um sistema de otimização de corte de matérias gerais para indústria, uma ferramenta computacional que possui algoritmos para corte linear (1D) e corte em área (2D) juntamente com uma interface gráfica intuitiva com várias configurações de corte disponíveis para o usuário.

O objetivo geral foi satisfeito através do desenvolvimento completo do software, desenvolvimento feito em Python com a interface gráfica em *Tkinter*, fazendo uso de algoritmos de busca exaustiva com heurísticas. A validação da ferramenta foi feita através de seis cenários de teste e demonstrou plenas capacidades de encontrar soluções eficientes, com todas as eficiências obtidas sendo superiores a 93% em todos os cenários de teste.

Todos os objetivos específicos foram atingidos. No decorrer do trabalho foram implementados algoritmos para corte linear utilizando busca exaustiva com geração iterativa de combinações, para corte em área foi utilizado o algoritmo em *bottom-left* modificado com rotação automática. O sistema de restrições suporta quantidade mínima e máxima, prioridades, repetir ou não uma peça, e margens de segurança. As heurísticas que foram implementadas demonstraram eficácia em reduzir o tempo de processamento das combinações, e a interface gráfica oferece todas essas possibilidades de configurações, juntamente com visualização e exportação dos resultados obtidos no uso da ferramenta.

As principais contribuições que esse trabalha entrega são três. A contribuição científica é que o trabalho contribuiu para a aplicação prática de algoritmos de otimização combinatória, com isso foi possível demonstrar a viabilidade de abordagens heurísticas para problemas NP-difíceis. A contribuição tecnológica está no desenvolvimento de uma ferramenta completa que possui várias funcionalidades, oferecendo suporte para ambos os tipos de corte. A contribuição prática está na disponibilidade de uma ferramenta acessível que pode ser usado em vários setores industriais para melhorar o aproveitamento dos seus materiais e reduzir desperdícios, onde a eficiência do software desenvolvido supera os métodos manuais.

As limitações que foram identificadas incluem a complexidade exponencial da busca exaustiva, que pode exigir um tempo de processamento muito elevado para problemas muito grandes, e também a possibilidade de exclusão de combinações

ótimas devido a limitação dinâmica do espaço de busca que é usado nas combinações. No algoritmo 2D, a verificação de colisões e teste de múltiplas orientações aumentam consideravelmente o tempo de processamento. Os cenários testados utilizaram dados simulados com números relativamente pequenos de peças, não representando necessariamente a variabilidade de situações reais.

As limitações da ferramenta não comprometem a validade dos resultados obtidos para os cenários testados, mas indicam apenas áreas para melhorias no futuro. Para trabalhos futuros, pode ser implementado algoritmos metaheurísticos, integração de técnicas de paralelização computacional, validação em ambientes industriais reais, desenvolvimento de uma versão web com integração em sistemas de gestão, e também o desenvolvimento de um novo algoritmo para suportar corte tridimensional.

Em resumo, o trabalho atendeu aos objetivos que foram apresentados, tendo como resultado final um sistema funcional que demonstrou capacidade de encontrar soluções eficientes. As contribuições científicas, tecnológicas e práticas, junto com as limitações encontradas e às sugestões para trabalhos futuros, deixam uma base sólida para evoluções do sistema que foi desenvolvido.

7 REFERÊNCIAS

ABUABARA, A. **Otimização no corte de tubos estruturais: aplicação na indústria aeronáutica agrícola**. 2006. Dissertação (Mestrado em Engenharia de Produção) – Departamento de Engenharia de Produção, Universidade Federal de São Carlos, São Carlos, 2006. Disponível em: https://repositorio.ufscar.br/items/65421134-0705-4e7a-8829-9dc0854f7ffd. Acesso em: 18 de nov. 2025.

ARENALES, M. N.; MORABITO, R.; YANASSE, H. H. Linear and nonlinear integer models for constrained two-stage two-dimensional knapsack problems. **Pesquisa Operacional,** Rio de Janeiro, v. 19, n. 2, p. 107-299, ago. 1999. Disponível em: https://prod.org.br/doi/10.1590/S0103-65132013005000023. Acesso em: 11 de nov. 2025.

BOGUE, E. T. **Algoritmos exatos e heurísticos para variações de problemas de roteamento, empacotamento e corte guilhotinado**. 2020. Tese (Doutorado em Ciência da Computação) – Universidade Federal de Minas Gerais, Belo Horizonte, 2020. Disponível em: https://repositorio.ufmg.br/handle/1843/33972. Acesso em: 15 dez. 2025.

CHERRI, A. C. **O problema de corte de estoque com reaproveitamento das sobras de material**. 2006. Dissertação (Mestrado em Engenharia de Produção) – Departamento de Engenharia de Produção, Universidade Federal de São Carlos, São Carlos, 2006. Disponível em: http://www.teses.usp.br/teses/disponiveis/55/55134/tde-17052006-131244/publico/DisMestradoDri.pdf. Acesso em: 15 dez. 2025.

GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: a guide to the theory of NP-completeness**. New York: W. H. Freeman, 1979. Disponível em: https://perso.limos.fr/~palafour/PAPERS/PDF/Garey-Johnson79.pdf. Acesso em: 15 dez. 2025.

GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. **Operations Research**, Baltimore, v. 9, n. 6, p. 849-859, nov./dez. 1961. Disponível em: https://www.researchgate.net/publication/266478800_A_Linear_Programming_Approach_to_the_Cutting_Stock_Problem_I. Acesso em: 15 dez. 2025.

NEMHAUSER, G. L.; WOLSEY, L. A. **Integer and combinatorial optimization**. New York: Wiley, 1988. Disponível em: https://www.academia.edu/15152518/Integer_and_Combinatorial_Optimization. Acesso em: 15 dez. 2025.

WOLSEY, L. A. **Integer programming**. New York: Wiley, 1998.

MORABITO, R. Modelos de otimização para o problema de corte nas indústrias de papel e papelão e de móveis. **Gestão & Produção**, São Carlos, v. 1, n. 1, p. 59-76,

jun. 1994. Disponível em: https://www.scielo.br/j/gp/a/bztQCRGBTcn94kBM8CPNQrD/?lang=pt. Acesso em: 03 dez. 2025.

RIBEIRO, M. V.; MORALES, G. **Otimização Combinatória: Programação Linear Inteira, Algoritmos e o Problema de Roteirização**. In: CONGRESSO NACIONAL DE INICIAÇÃO CIENTÍFICA EM INFORMÁTICA, 2020, Rio de Janeiro. Anais [...]. Rio de Janeiro: IFF, 2020. Disponível em: https://editoraessentia.iff.edu.br/index.php/confict/article/view/1033. Acesso em: 03 dez. 2025.