

UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"
FACULDADE DE ENGENHARIA
CAMPUS DE ILHA SOLTEIRA

Gloria Patricia López Sepúlveda

SOLUÇÃO DO PROBLEMA DE CORTE BIDIMENSIONAL DE PEÇAS
RETÂNGULARES TIPO NÃO-GUILHOTINADO USANDO SIMULATED
ANNEALING

Ilha Solteira
2013

Gloria Patricia López Sepúlveda

SOLUÇÃO DO PROBLEMA DE CORTE BIDIMENSIONAL DE PEÇAS
RETÂNGULARES TIPO NÃO-GUILHOTINADO USANDO SIMULATED
ANNEALING

Dissertação apresentada a Faculdade de Engenharia do Campus de Ilha Solteira - UNESP, como parte dos requisitos para obtenção do título de Mestre em Engenharia Elétrica.

Especialidade: Automação.

Prof. Dr. Ruben Augusto Romero Lazaro
Orientador

Ilha Solteira
2013

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

L864s López Sepúlveda, Gloria Patricia .
Solução do problema de corte bidimensional de peças retangulares tipo não-guilhotinado usando simulated annealing / Gloria Patricia López Sepúlveda. -- Ilha Solteira: [s.n.], 2013
109 f. : il.

Dissertação (mestrado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de conhecimento: Automação, 2013

Orientador: Ruben Augusto Romero Lazaro
Inclui bibliografia

1. Cutting stock problem. 2. Não-guilhotinado. 3. Botton-left. 4. Simulated annealing.

CERTIFICADO DE APROVAÇÃO

TÍTULO: Solução do problema de corte bidimensional de peças retangulares tipo não-guilhotinado usando simulated annealing

AUTORA: GLORIA PATRICIA LOPEZ SEPULVEDA

ORIENTADOR: Prof. Dr. RUBEN AUGUSTO ROMERO LAZARO

Aprovada como parte das exigências para obtenção do Título de Mestre em Engenharia Elétrica ,
Área: AUTOMAÇÃO, pela Comissão Examinadora:



Prof. Dr. RUBEN AUGUSTO ROMERO LAZARO
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira



Prof. Dr. MARCOS JULIO RIDER FLORES
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira



Prof. Dr. EDUARDO NOBUHIRO ASADA
Departamento de Engenharia Elétrica / Escola de Engenharia de São Carlos-Usp

Data da realização: 25 de março de 2013.

*A Deus
A Hugo, a Valentina e minha mãe Carmen*

AGRADECIMENTOS

Principalmente gostaria de agradecer a Deus por tudo que recebo diariamente, saúde, paz, amor, tranquilidade, sabedoria e muito conhecimento.

A minha avó Dioselina e a minha mãe Carmen pelos seus ensinamentos.

Agradeço ao meu orientador de dissertação, o Professor Dr. Ruben Augusto Romero Lazaro, pela sua valiosa ajuda, pelo estímulo e excelente assessoramento no desenvolvimento deste trabalho.

A todos os professores do LaPSEE pelo apoio emocional no desenvolvimento deste trabalho.

Agradeço também a meus colegas do laboratório de pesquisa LAPSEE, os quais de alguma maneira têm contribuído no desenvolvimento desta tese.

Agradeço a minha família pelo apoio incondicional, especialmente a meu esposo Hugo Andrés Ruiz Flórez e a minha filha Valentina Ruiz López, que a cada dia me dá forças para prosseguir com os meus objetivos.

Ao CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) pelo apoio financeiro.

RESUMO

Os problemas de corte ótimo são considerados como clássicos dentro da pesquisa operacional, dado sua grande área de aplicação na indústria e sua alta complexidade matemática e computacional. Um dos problemas mais conhecidos desta classe é o chamado Cutting Stock Problem, o qual permite dividir uma placa em várias peças de diversos tamanhos, com a finalidade de obter a configuração que maximize a área da placa que está sendo utilizada. Dado o nível de dificuldade e a grande quantidade de aplicações deste problema em diversas áreas, existe muito interesse em criar novos procedimentos eficientes para resolver este tipo de problemas. Nesta dissertação, é apresentado o estado da arte dos diversos problemas de corte bidimensional de peças retangulares em uma única placa, em várias placas e em rolos, considerando a possibilidade de rotacionar as peças em um ângulo de 90° e com restrições de corte tipo não-guilhotinado. Além disso, são descritas as abordagens matemáticas aplicadas pelos diversos pesquisadores dedicados a resolver estes problemas. Este trabalho está focado em resolver exclusivamente o Cutting Stock Problem com e sem rotação de peças. Portanto, é proposto um tipo de codificação binária com vista a ser aplicada neste tipo de problema e resolvê-lo mediante o uso do algoritmo heurístico Bottom-Left, em conjunto com a metaheurística Simulated Annealing. Finalmente, para comprovar a eficiência da metodologia apresentada, foram utilizados alguns casos de estudo da literatura especializada, com diferentes níveis de complexidade. Para cada caso, são apresentados os resultados obtidos e é realizada uma comparação dos métodos de solução propostos para resolver o Cutting Stock Problem com e sem rotação de peças.

Palavras-chave: - Cutting stock problem. Não-guilhotinado. Bottom-left. Simulated annealing.

ABSTRACT

The problems of optimal cutting are considered as classic within the operational research, given their large area of application in industry and their high computational and mathematic complexity. One of the most well-known of this kind of problem is called the Cutting Stock Problem, which allows to divide a plate in several pieces of various sizes, in order to obtain the configuration that maximizes the area of the plate being used. Given the level of difficulty and the large number of applications of this problem in several areas, there is interest in creating new efficient procedures for solving this kind of problem. In this dissertation, it is presented the state of the art of the two-dimensional cutting problems of rectangular pieces on a single plate, in several plates and rollers, considering the possibility of rotating the pieces at an angle of 90° in which the cuts are restricted to non-guillotined type. Furthermore, the mathematical approaches applied by different researchers dedicated to solving these problems are described. This work is focused exclusively on solving the Cutting Stock Problem with and without rotating pieces. Therefore, a kind of binary encoding in order to be applied to this kind of problem and to solving it through the use of the Bottom-Left heuristic algorithm in conjunction with Simulated Annealing are proposed. Finally, to prove the efficiency of the presented methodology, some study cases from the specialized literature were used, with different levels of complexity. For each case, the results are presented and a comparison of the proposed solution methods for solving the Cutting Stock Problem with and without rotating pieces is performed.

Keywords: - Cutting stock problem. Non-guillotined type. Botton-left. Simulated annealing.

LISTA DE FIGURAS

Figura 1 - Classificação Básica	15
Figura 2 - Peças e placas demandadas para serem cortadas	16
Figura 3 - Peças demandadas e rolo de comprimento infinito	17
Figura 4 - Diversidade de peças e placa	17
Figura 5 - Corte tipo guilhotinado	19
Figura 6 - Padrão de corte tipo guilhotinado	19
Figura 7 - Corte tipo não-guilhotinado	20
Figura 8 - Padrão de corte tipo não-guilhotinado	20
Figura 9 - Exemplo de corte num rolo	30
Figura 10 - Disposição e corte das peças sobre uma placa	34
Figura 11 - Corte e distribuição das peças sobre as placas	35
Figura 12 - Corte e distribuição das peças sobre as placas	39
Figura 13 - Estratégia BL e BLF	46
Figura 14 - Heurística BL proposta em (Chazelle,1983)	47
Figura 15 - Heurística BL proposta em (Jakobs,1996)	48
Figura 16 - Heurística BL proposta em (Liu, 2002)	49
Figura 17 - Padrões de corte obtidos pela heurística em (Burke, 2004).	50
Figura 18 - Padrão de corte impossível com BL	51
Figura 19 - Padrão de corte impossível com DP	51
Figura 20 - Dificuldades dos algoritmos heurísticos.	54
Figura 21 - Comportamento das metaheurísticas populacionais no processo de otimização	56
Figura 22 - Comportamento das metaheurísticas de busca por vizinhança no processo de otimização.	57
Figura 23 - Representação gráfica de dois valores ótimos possíveis para uma função objetivo.	59
Figura 24 - Algoritmo da técnica de otimização Simulated Annealing.	66
Figura 25 - Função de aceitação	71
Figura 26 - Representação da Codificação binária	72
Figura 27 - Alocação da primeira peça	74
Figura 28 - Alocação da segunda peça	75
Figura 29 - Alocação final	76
Figura 30 - Vetor solução atual	78

Figura 31 - Vetor solução modificado	79
Figura 32 - Vetor solução modificado	81
Figura 33 - Vetor de rotação de peças	81
Figura 34 - Alocação da primeira peça	82
Figura 35 - Alocação da última peça	83
Figura 36 - Distribuição das peças No. 1	86
Figura 37 - Distribuição das peças No. 2	87
Figura 38 - Distribuição das peças No. 3	87
Figura 39 - Distribuição das peças No. 1	88
Figura 40 - Distribuição das peças No. 2	89
Figura 41 - Distribuição das peças No. 3	89
Figura 42 - Resultados obtidos pelos algoritmos propostos. a. Sem rotação de peças. b. Com rotação de peças	90
Figura 43 - Distribuição das peças do teste 2	93
Figura 44 - Distribuição das peças do teste 2	94
Figura 45 - Resultados obtidos pelos algoritmos propostos. a. Sem rotação de peças. b. Com rotação de peças	95
Figura 46 - Distribuição das peças do teste 3	97
Figura 47 - Distribuição das peças do teste 3	98
Figura 48 - Resultados obtidos pelos algoritmos propostos. a. Sem rotação de peças. b. Com rotação de peças	99

LISTA DE TABELAS

Tabela 1 - Base de dados inicial	72
Tabela 2 - Possíveis combinações para o problema com rotação de peças	80
Tabela 3 - Base de dados do caso teste 1	85
Tabela 4 - Parâmetros do algoritmo SA usados no teste 1	85
Tabela 5 - Dados gerais da simulação do caso teste 1	91
Tabela 6 - Base de dados do caso teste 2	92
Tabela 7 - Parâmetros do algoritmo SA usados no teste 2	93
Tabela 8 - Dados gerais da simulação do caso teste 2	95
Tabela 9 - Base de dados do caso teste 3	96
Tabela 10 - Valores dos parâmetros usados no teste 3	97
Tabela 11 - Dados gerais da simulação do caso teste 3	100

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Considerações preliminares	13
1.2	Definição do problema	22
1.3	Motivações e objetivos	23
1.4	Revisão bibliográfica	24
1.5	Estrutura do trabalho	26
2	GENERALIDADES DO PROBLEMA DE CORTE	28
2.1	Introdução	28
2.2	Aplicações na indústria	29
2.3	Introdução Histórica	30
2.4	Técnicas de solução do problema de corte bidimensional	31
2.5	Corte ótimo bidimensional de peças retangulares em uma única placa	33
2.6	Corte ótimo bidimensional de peças retangulares em várias placas	35
2.7	Corte ótimo bidimensional de peças retangulares em rolos	38
3	MODELAGEM MATEMÁTICA	41
3.1	Introdução	41
3.2	Modelagem matemática para o problema de corte bidimensional	41
3.3	Heurísticas para resolver problemas de otimização combinatória	42
3.4	Procedimentos heurísticos para resolver o problema de corte bidimensional	44
3.4.1	<i>Métodos Construtivos para o problema de corte bidimensional</i>	45
4	AS METAHEURÍSTICAS	53
4.1	Introdução	53
4.2	Classificação das técnicas metaheurísticas	55
4.3	Metaheurísticas como solução de um problema de otimização	57
4.4	Simulated Annealing	62
4.5	Descrição do Algoritmo do Simulated Annealing	64
4.5.1	<i>Algoritmo Simulated Annealing</i>	65
4.5.2	<i>Temperatura inicial</i>	65
4.5.3	<i>Comprimento da cadeia</i>	67
4.5.4	<i>Taxa de resfriamento</i>	67
4.5.5	<i>Definição da vizinhança</i>	68

4.5.6	<i>Critério de parada</i>	69
5	MÉTODO DE SOLUÇÃO PARA O PROBLEMA DE CORTE BIDIMENSIONAL	70
5.1	Introdução	70
5.2	Problema de corte bidimensional de peças retangulares sem rotação de peças	70
5.2.1	<i>Designação de valores dos parâmetros do SA</i>	70
5.2.2	<i>Base de Dados</i>	71
5.2.3	<i>Codificação e Decodificação de uma proposta de solução</i>	71
5.2.4	<i>Cálculo da função objetivo</i>	76
5.2.5	<i>Criação da configuração inicial</i>	77
5.2.6	<i>Definição da vizinhança</i>	78
5.2.7	<i>Atualização da solução incumbente</i>	78
5.2.8	<i>Algoritmo de corte bidimensional de peças retangulares tipo não-guilhotinado sem rotação de peças</i>	79
5.3	Problema de corte bidimensional de peças retangulares com rotação de peças	80
6	TESTES E RESULTADOS	84
6.1	Introdução	84
6.2	Caso teste 1	85
6.2.1	<i>Solução do problema de corte não guilhotinado sem rotação de peças</i>	85
6.2.2	<i>Solução do problema de corte não guilhotinado com rotação de peças</i>	88
6.2.3	<i>Comparação do desempenho dos algoritmos de corte propostos</i>	90
6.3	Caso teste 2	92
6.3.1	<i>Solução do problema de corte não guilhotinado sem rotação de peças</i>	93
6.3.2	<i>Solução do problema de corte não guilhotinado com rotação de peças</i>	94
6.3.3	<i>Comparação do desempenho dos algoritmos de corte propostos</i>	94
6.4	Caso teste 3	96
6.4.1	<i>Solução do problema de corte não guilhotinado sem rotação de peças</i>	97
6.4.2	<i>Solução do problema de corte não guilhotinado com rotação de peças</i>	98
6.4.3	<i>Comparação do desempenho dos algoritmos de corte propostos</i>	98
7	CONCLUSÕES E TRABALHOS FUTUROS	101
	REFERÊNCIAS	103

1 INTRODUÇÃO

1.1 Considerações preliminares

Em muitas aplicações da vida real relacionadas com os processos industriais surgem vários problemas que na área da Pesquisa Operacional são classificados como "*Problemas de corte e empacotamento*". Em termos gerais, nesses processos industriais é necessário produzir determinadas quantidades de placas de um tipo de material, por exemplo, madeira, metal, papel, plástico, vidro, ou tecido, entre outros, que depois devem ser cortadas em peças de tamanho menor. Nesse processo, o desperdício de material deve ser o mínimo possível.

Na literatura é comum encontrar que os problemas de *corte e empacotamento* são equivalentes, pois representam a mesma tarefa no espaço, ou seja, o fato de empacotar ou cortar faz referência à divisão de um espaço em partições menores onde serão alocados os itens a serem empacotados ou cortados. Nos anos 70's, Hinxman (1977) apresentou um esquema de classificação para o problema de corte e empacotamento. Anos depois em, Dyckhoff (1990) foi apresentada uma tipologia para o mesmo tipo de problemas. Essa tipologia é baseada em quatro itens:

- 1 Dimensionalidade:
 - (1) Uma dimensão.
 - (2) Duas dimensões.
 - (3) Três dimensões.
 - (N) Número de dimensões, com $N > 3$.

- 2 Tipo de atribuição:
 - (B) Todas as placas e uma porção das peças demandadas.
 - (V) Uma porcentagem das placas e todas as peças demandadas.

- 3 Tipo de placas armazenadas:
 - (O) Uma placa.
 - (I) Placas idênticas.

(D) Placas diferentes.

4 Diversidade de peças demandadas:

(F) Poucas peças de diferentes tamanhos.

(M) Muitas peças de muitos tamanhos.

(R) Muitas peças de poucas dimensões diferentes.

(C) Muitas peças idênticas.

Anos mais tarde, Lodi et al. (1990), apresenta-se uma forma de classificar o problema fazendo uso das características já existentes em Dyckhoff (1990). Os autores Salkyn e Dekluyver (1975), Christofides e Whitlock (1977) e Macleod et al. (1993), diferenciam dois tipos de problemas dependendo do valor das peças:

- O problema sem ponderação (*un-weighted*) onde o valor de cada peça é a área da mesma.
- O problema com ponderação (*weighted*) onde o valor da peça não depende da área dela, portanto, podem existir peças pequenas com valores maiores que os valores das peças grandes.

Assim, a classificação feita por Lodi et al. (1999) é a seguinte:

5 Restrições inerentes ao padrão de corte (tipo de corte):

(G) Exclusivamente cortes tipo guilhotinados.

(U) Exclusivamente cortes tipo não-guilhotinados.

6 Restrições inerentes à orientação das peças (possibilidade de ter ou não rotação das peças):

(T) As peças podem rotar 90°.

(A) As peças têm orientação fixa.

7 Valores das peças (o benefício obtido após cortar determinada peça):

(W) Peças com benefícios diferentes à sua área (*weighted*).

(Z) Peças com benefícios iguais à sua área (*un-weighted*)

8 Demanda das peças (quantidade de peças a serem cortadas):

(E) Peças com limite máximo de corte.

(I) Peças sem limites de corte.

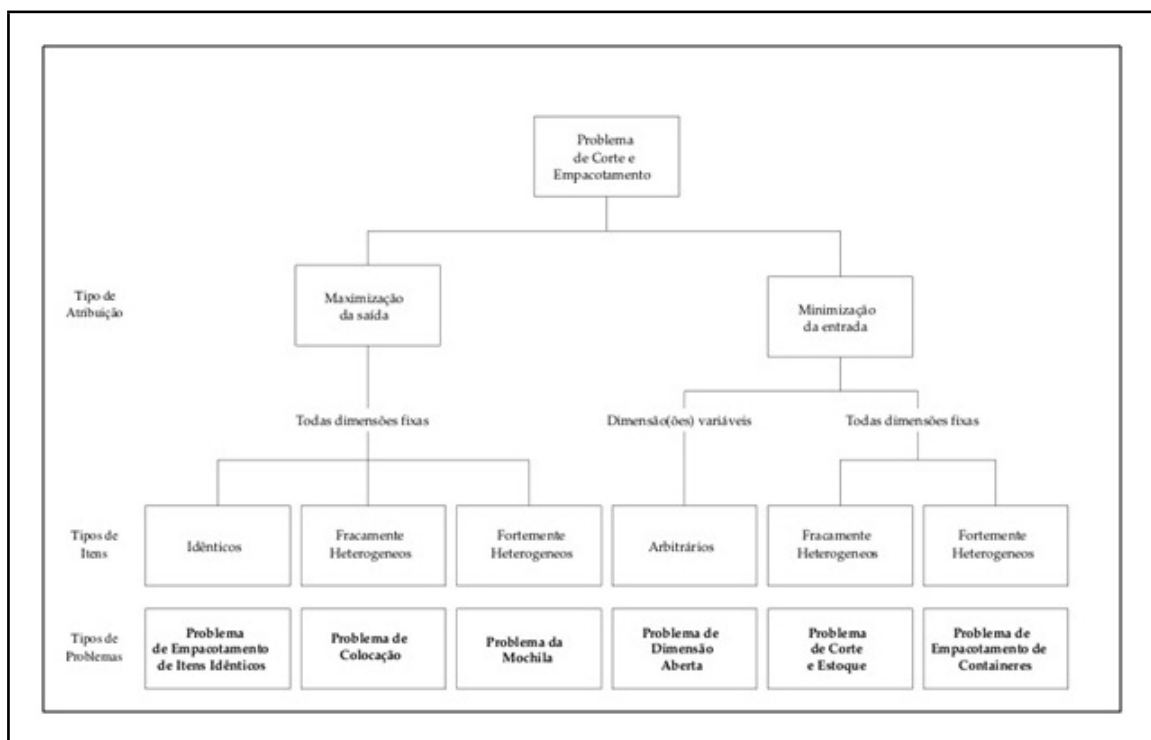
9 Forma das peças:

(L) Peças com forma regular (*retângulos, círculos, cubos, esferas, cilindros, etc.*)

(K) Peças com forma irregular.

Segundo Wascher et al. (2004), esta classificação apresenta algumas falhas. Neste trabalho foi apresentada uma extensão da tipologia de Dyckhoff (1990), detalhando de forma precisa, as especificações de cada tipo de problema. A classificação básica pode ser vista na Figura 1.

Figura 1 – Classificação Básica



Fonte: (WÄSCHER et al., 2004)

No entanto, para aplicações como a abordada neste trabalho, faz-se necessário destacar que os problemas de corte e empacotamento são diferentes. Assim, tem-se que o problema

de corte faz referência a um problema em duas dimensões, enquanto que no problema de empacotamento intervêm as três dimensões do espaço. O escopo deste trabalho consiste em encontrar a solução do problema de corte de peças.

Em termos gerais, o problema de corte tem sido amplamente difundido e estudado através do problema de corte de peças retangulares. Dentre as principais variantes dos problemas de corte destacam-se:

1 O caso onde se tem várias placas idênticas, com dimensões conhecidas (comprimento L e largura W). Portanto, é necessário distribuir todas as peças de tal forma que caibam dentro da placa. Assim o objetivo deste problema é encontrar a melhor distribuição das peças que permita minimizar o número de placas necessárias para o corte da totalidade das peças. Esse problema na literatura é chamado de *Bin Packing Problem* (BPP) e sua representação pode ser vista na Figura 2.

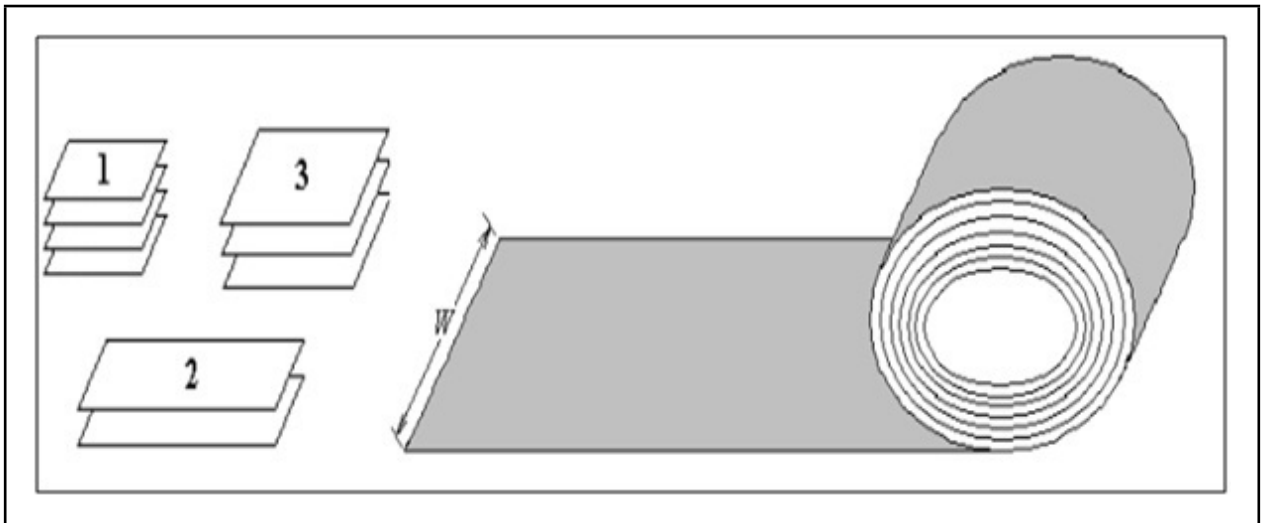
Figura 2 – Peças e placas demandadas para serem cortadas



Fonte: (ÁLVAREZ et al., 2009)

2 Outro tipo de problema é aquele em que só é conhecido o comprimento L da placa enquanto a largura W é um valor flexível. Neste problema o objetivo é minimizar W , distribuindo as peças da melhor forma possível. Essa variação do problema de corte faz referência ao caso em que se tem uma matéria prima disponível em rolos, como é o caso do papel e do tecido. Na literatura esse problema é conhecido como *Strip Packing Problem* (SPP) e sua representação pode ser vista na Figura 3.

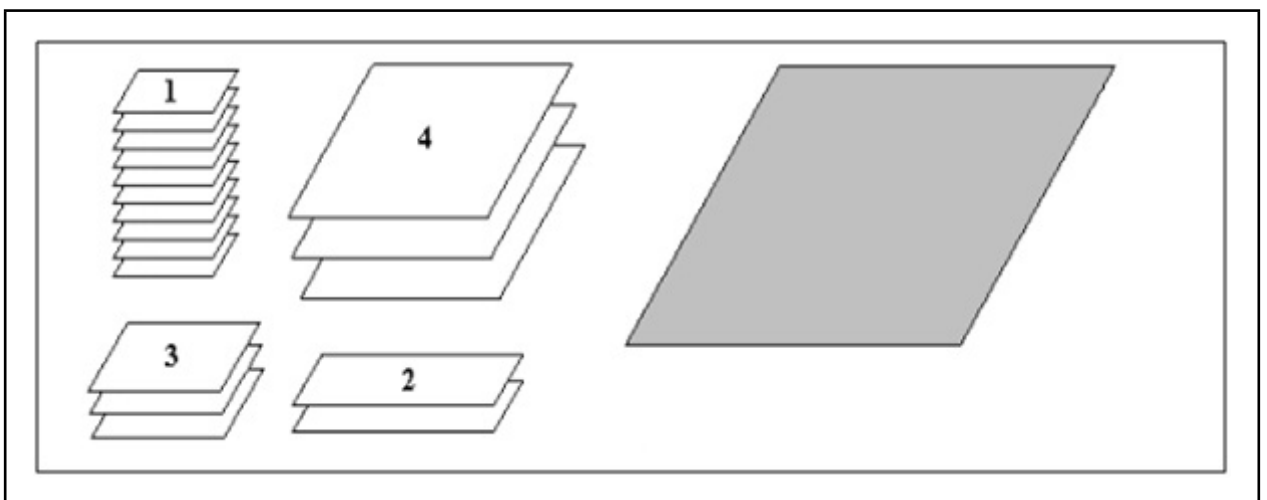
Figura 3 – Peças demandadas e rolo de comprimento infinito



Fonte: (ÁLVAREZ et al., 2009)

3 Por último, temos o problema onde têm-se só uma placa de dimensões (W, L), largura e comprimento respectivamente. Neste caso, pretende-se maximizar a área utilizada da placa, distribuindo na mesma, a maior quantidade de peças demandadas possíveis, e minimizar o desperdício. Este problema é conhecido na literatura como *Cutting Stock Problem* (CSP) e sua representação pode ser vista na Figura 4.

Figura 4 – Diversidade de peças e placa



Fonte: (ÁLVAREZ et al., 2009)

Os estudos do BPP e do SPP estão fora do escopo deste trabalho, portanto aborda-se o caso do CSP. Para esse caso, o problema consiste em encontrar a melhor distribuição de

um conjunto de n tipos de peças, cada uma com uma quantidade (q_i) e dimensões (l_i, w_i) conhecidas, sobre uma superfície S de dimensões (L, W), de modo que, a área usada em S seja a máxima possível, ou em outras palavras, que o desperdício de material seja o mínimo.

Matematicamente, o CSP pode ser formulado da seguinte forma:

$$\min f = (L * W) - \sum_{i=1}^n q_i * (l_i * w_i) \quad (1)$$

Ou

$$\max f = \sum_{i=1}^n q_i * (l_i * w_i) \quad (2)$$

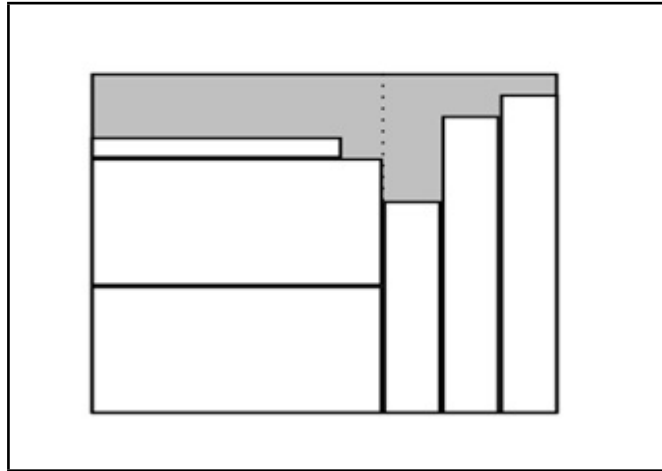
Sujeito a:

- i* q_i inteiro, $0 \leq q_i < (L * W)/(l_i * w_i)$.
- ii* $L \geq l_i$ e $W \geq w_i$.
- iii* $\sum_{i=1}^n q_i * (l_i * w_i) \leq (L * W)$
- iv* A distribuição das peças não deve ultrapassar as dimensões da placa.
- v* As peças não devem estar sobrepostas.

O modelo matemático acima é o mais geral do CSP, pois existem outros tipos de restrições que podem ser inseridas ao modelo de acordo com as características do problema, entre as quais se destacam: as demandas das peças, o número de peças de um tipo que deve estar presente na solução, a rotação de peças e os tipos de corte, entre outras.

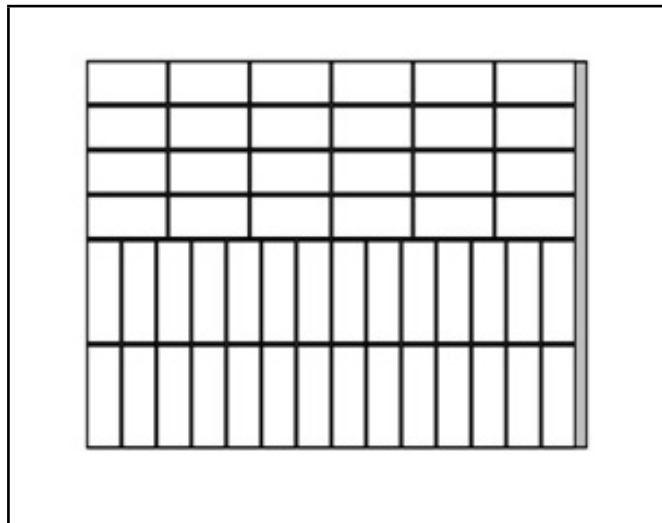
No BPP existem dois tipos de corte que podem ser efetuados para dividir as placas, conhecidos como corte tipo guilhotinado e corte tipo não-guilhotinado. O corte tipo guilhotinado se diferencia do corte tipo não-guilhotinado, em que este último atravessa a placa ou parte dela ortogonalmente de um extremo a outro. Em consequência, define-se um padrão tipo guilhotinado como à distribuição das peças dentro da placa ao serem divididas mediante sucessivos cortes do tipo guilhotinado, como pode ser visto nas Figuras 5 e 6.

Figura 5 – Corte tipo guilhotinado



Fonte: (PARREÑO, 2004)

Figura 6 – Padrão de corte tipo guilhotinado

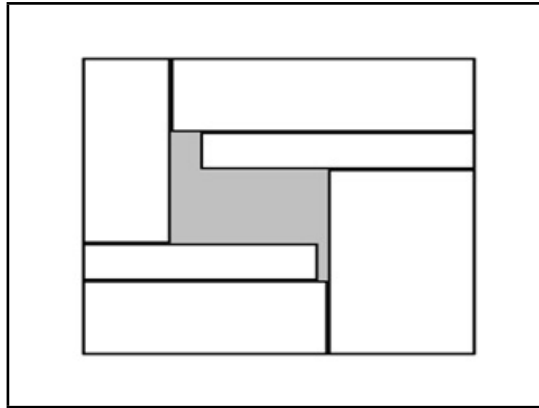


Fonte: (PARREÑO, 2004)

Por sua vez, o padrão de corte tipo não-guilhotinado corresponde à divisão das peças mediante sucessivos cortes tipo guilhotinados e tipo não-guilhotinados, como se mostra nas Figuras 7 e 8.

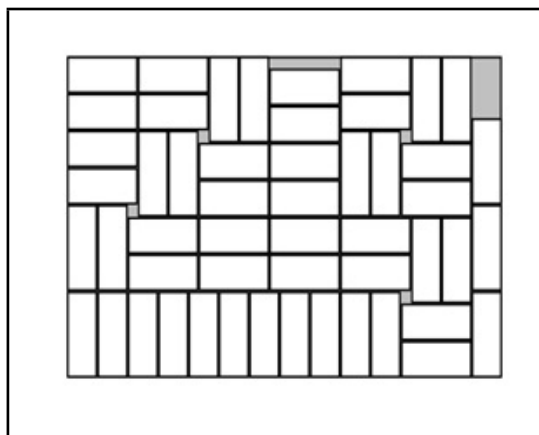
Cabe salientar que o modelo do CSP, detalhado em (1), para cortes tipo guilhotinados, pode ser considerado um problema mais difícil de resolver do que o problema para cortes tipo não-guilhotinados, pois no primeiro caso, deve ser considerada uma nova restrição

Figura 7 – Corte tipo não-guilhotinado



Fonte: (PARREÑO, 2004)

Figura 8 – Padrão de corte tipo não-guilhotinado



Fonte: (PARREÑO, 2004)

ao modelo matemático. Essa nova restrição deve garantir que os cortes sejam factíveis dada uma distribuição das peças dentro da placa, enquanto no segundo caso, qualquer arranjo ou acomodação das peças dentro da placa representa uma alternativa de solução do problema.

Dado que, de modo geral, o CSP apresenta uma complexidade matemática elevada e dado que ele representa um problema real na indústria, várias abordagens na área da pesquisa operacional têm sido propostas para encontrar a solução do CSP. Nesse contexto, destacam-se os algoritmos exatos, os algoritmos heurísticos, os algoritmos de otimização combinatória e a criação de modelos matemáticos.

No caso dos procedimentos exatos têm sido usados, entre outros, o método Branch-and-Bound (GROTSCHHEL et al., 1983), o método Branch-and-Cut (PADBERG; RINALDI, 1991) e o método Branch-and-Price (THIENEL, 1995), assim como alguns procedimentos baseados em estrutura de árvore (GILMORE; GOMORY, 1966; CHRISTOFIDES; WHITLOCK, 1977).

Por outro lado, na literatura especializada é comum encontrar várias estratégias heurísticas para posicionar as peças dentro da placa, sendo destaques os algoritmos Bottom-Left (BAKER et al., 1980), Bottom-Left-Fill (CHAZELLE, 1983), Difference-Process (LAI; CHAN, 1997), e as suas variações. Berkey e Wang (1987) propõem dois novos algoritmos que foram chamados de Finite-Best-Fit strip (FBS) e Finite-First-Fit (FFF), Lodi et al. (2002) apresenta uma variedade de métodos construtivos Next-Fit-Decreasing, First-Fit-Decreasing, Best-Fit-Decreasing, Hybrid First-Fit, entre outros.

Nos últimos anos, os algoritmos de otimização combinatória têm surgido como uma ferramenta matemática para explorar o espaço de soluções do problema de maneira eficiente, encontrando assim resultados de boa qualidade. Dentre esses algoritmos têm-se o Simulated Annealing (LAI; CHAN, 1997), os algoritmos genéticos (BEASLEY, 2004), busca Tabu (BURKE et al., 2004), GRASP (BELTRAN et al., 2004), Algoritmos Evolutivos (YEUNG; TANG, 2003), otimização por colônia de formigas (BURKE; KENDALL, 1999). Geralmente, esses algoritmos usam estratégias heurísticas para a alocação das peças na placa. Na literatura existem muitos outros tipos de estratégias que se enquadram dentro dessa classificação, os quais correspondem aos algoritmos usados no corte tipo guilhotinado, que dividem a placa em várias seções a fim de garantir os cortes factíveis, onde cada uma é preenchida de acordo com um procedimento heurístico. (TORO et al., 2008).

Finalmente, no que se refere aos modelos matemáticos, pode-se salientar os seguintes trabalhos: Cheng et al. (1995) apresentou um modelo matemático para o problema de empacotamento retangular em três dimensões; Gilmore e Gomory (1963) desenvolveram um modelo matemático para o problema de corte duas dimensões; Lodi et al. (2004) propõe um modelo de programação linear inteira (PLI) para o problema de corte bidimensional em níveis; Parreño (2004) apresenta um modelo do problema de corte bidimensional restrito.

1.2 Definição do problema

O CSP pertence à classe dos problemas NP (do inglês, non-polinomial) difíceis, devido a sua elevada complexidade matemática, principalmente, dado que para um número finito de peças o espaço de soluções corresponde ao número de combinações possíveis das peças a serem cortadas e essa quantidade de soluções incrementa-se exponencialmente à medida que aumenta o número de peças. Essa característica é uma séria limitante no momento de propor uma metodologia de solução, pois, dado um problema de complexidade matemática elevada, faz-se necessário contar com uma série de recursos computacionais que permitam avaliar muitas configurações no processo de busca de uma solução do problema.

O CSP faz parte dos problemas que são fáceis de definir intuitivamente, mas na hora da implementação computacional resultam as dificuldades de modelagem. Prova disto aparecem nas restrições iv e v do modelo mostrado em (1), além das restrições do corte factível no caso do corte tipo guilhotinado e os casos com rotação de peças. Essas características explicam os motivos da quantidade reduzida de modelos matemáticos propostos para resolver o CSP em comparação com outro tipo de abordagens desenvolvidas para resolver o problema.

Outro fator que aumenta o grau de complexidade do CSP é que as abordagens propostas são exclusivas para um tipo de problema, isto é, percebe-se que os algoritmos usados são exclusivos para um tipo de problema, ou seja, não existe um algoritmo padrão que consiga resolver o CSP e suas variações, especialmente nos casos de corte tipo guilhotinados e não-guilhotinados. Isto significa que cada versão do problema possui características específicas e diferenciadas.

Uma técnica de solução interessante, que tem sido amplamente estudada para resolver estes tipos de problemas, é através da utilização de algoritmos de otimização combinatória, também conhecidos como metaheurísticas. No entanto, muitas dessas abordagens

apresentam dificuldades ao definir o tipo de codificação do problema, já que, em muitos casos é indispensável usar vários vetores ou vetores de grande tamanho para representar as soluções do problema. Isto é uma condição que dificulta a definição da vizinhança e dos mecanismos de busca da solução.

Visando a utilização do algoritmo de otimização combinatória Simulated Annealing para resolver o CSP, deve-se considerar que, geralmente, o tempo que os mecanismos adotados por uma metaheurística para encontrar a solução de um problema depende do nível de complexidade do problema, ou seja, os mecanismos usados para resolver problemas de pequeno ou médio porte não necessariamente resultam ser os mais eficientes para encontrar a resposta de um problema de grande porte, considerando o amplo espaço de possíveis soluções. Para contornar essa dificuldade, no CSP têm sido usados mecanismos que permitem agrupar peças de características similares para reduzir o número de combinações possíveis. Também é frequente encontrar algoritmos heurísticos construtivos para gerar soluções iniciais de boa qualidade que ajudem a direcionar a busca de soluções em regiões com presença de ótimos locais.

Contudo, o CSP continua sendo um problema de difícil solução e, portanto, é necessária a escolha de uma técnica metaheurística que se adapte bem as suas características, para posteriormente, formular um procedimento capaz de lidar com as dificuldades inerentes do problema, e principalmente, que consiga resultados de boa qualidade.

1.3 Motivações e objetivos

O problema de corte é um tópico de pesquisa de grande interesse tanto no âmbito acadêmico como também no setor industrial. O papel desempenhado pelo problema de corte bidimensional de peças retangulares dentro da pesquisa operacional e no campo prático de produção das empresas é fundamental, pois um adequado planejamento e uma boa estratégia para realizar o corte de peças podem fazer com que as empresas possam diminuir os gastos e aumentar seus lucros.

O problema de corte, apesar de ser um tema bem difundido e amplamente estudado durante muitos anos e com diferentes formas de solução, seja através de modelos matemáticos ou técnicas de otimização combinatória, entre outros, ainda é motivo de estudo por parte de reconhecidos pesquisadores da área da engenharia de produção e da pesquisa operacional por ser um problema de alto nível de complexidade. É por isso que atualmente continuam sendo formuladas novas propostas de solução com o intuito de de-

desenvolver técnicas eficientes de otimização.

Neste contexto, são estabelecidos neste trabalho os seguintes objetivos:

- Através do estudo dos algoritmos heurísticos para alocação de peças, escolher uma estratégia que permita usar um tipo de codificação adequado para resolver o problema de corte bidimensional de peças tipo não-guilhotinado.
- Propor um tipo de codificação e definir a vizinhança para avaliar eficientemente o espaço de soluções do problema.
- Adaptar os mecanismos da metaheurística Simulated Annealing para resolver o problema de corte bidimensional de peças tipo não-guilhotinado.

1.4 Revisão bibliográfica

A maioria dos trabalhos desenvolvidos na área de corte estão diretamente relacionados com o problema de empacotamento, devido à grande semelhança entre os dois problemas. Uma revisão das propostas mais relevantes nesta área de pesquisa começa com Kantorovich (1939) e Brooks (1940), que apresentaram alguns estudos de interesse neste tema. Anos mais tarde, Gilmore e Gomory (1961), apresentam pela primeira vez uma aproximação para problemas reais, sendo esta publicação uma das mais importantes e de maior relevância dentro desta área de pesquisa, pois este trabalho incentivou muitos outros pesquisadores que abordam esse tema. Por isso, a cada ano aumenta o número de publicações que apresentam variações do problema.

Entre os anos 1940 e 1990, Sweeney e Paternoster (1992) apresentaram uma compilação de 400 livros, artigos e algumas outras publicações, onde se estudam os problemas de corte e empacotamento. Essa informação compilada foi classificada segundo as dimensões do problema, tipo de problema e tipo de aproximação da solução encontrada.

Anos mais tarde, Wascher et al. (2007) apresentou outra compilação feita das publicações entre os anos 1995 e 2005, com mais de 400 publicações. A documentação aqui apresentada mostra aplicações de problemas padrões que excluem variantes tipo multi-objetivo.

Os problemas de corte e empacotamento têm uma alta complexidade matemática, na literatura, sendo os mesmos considerados como NP-Difíceis e sendo demonstrados através da consideração do caso especial em que todos os itens têm o mesmo comprimento e

larguras diferentes. Esse problema é conhecido como problema de empacotamento unidimensional, (one-dimensional Bin packing). O problema de corte e empacotamento ótimo de uma dimensão foi provado como NP- Dificil por Garey e Johnson (1979) e Martello et al. (2003).

Também existem alguns métodos exatos propostos para resolver este tipo de problemas, por exemplo, Beasley (1985), Tsai et al. (1988) e Scheithauer e Terno (1993), apresentam formulações inteiras. Hadjiconsyantinos e Christofides (1995) propõem uma formulação que "relaxa"o problema e utiliza a programação dinâmica. Arenales e Morabito (1995) apresenta um algoritmo de busca em forma de árvore (*branch and bound*) baseado em um tipo especial de grafo (*AND/OR graph*). Fekete e Schepers (1997) criaram um procedimento baseado na representação e fazendo uso de grafos para mostrar a posição relativa das peças em um padrão possível. Finalmente, em Caprara (2004), apresenta-se um algoritmo com algumas melhorias daquele que foi proposto por Fekete e Schepers (1997).

Dentro das heurísticas construtivas, surge na década de 80, o procedimento mais usado para alocar peças: o algoritmo *Bottom-Left* (BL) apresentado em Chazelle (1983). Este algoritmo obtém padrões de corte tipo não-guilhotinado. Outro método utilizado na literatura para construir uma solução a partir do ordenamento das peças é o mecanismo denominado *Difference-Process* (DP) apresentado em Lai e Chan (1997). Esse algoritmo, similar ao BL, permite gerar padrões de corte tipo não-guilhotinado. O trabalho de Lodi et al. (2002) apresenta uma ampla variedade de métodos construtivos, como por exemplo: *Next-Fit-Decreasing*, *First-Fit-Decreasing*, *Best-Fit-Decreasing*, *Hybrid-First-Fit* entre outros, os quais foram usados para o problema de corte bidimensional em placas de largura definida e não definida, com padrões de corte tipo guilhotinado.

No trabalho de Beasley (2004), pode-se encontrar um algoritmo metaheurístico para a solução do problema de corte ótimo bidimensional tipo guilhotinado e sem restrições. Para o mesmo problema, só que considerando restrições, Lai e Chan (1997) propõem um algoritmo baseado em Simulated Anneling. Depois disto, Leung et al. (2001) utiliza a codificação usada por Lai e Chan (1997) para construir um algoritmo genético, em conjunto, com Simulated Anneling para demonstrar que existem padrões de corte que não podem ser encontrados pelos procedimentos BL e DP. Adicionalmente, em Beasley (2004) apresenta-se um algoritmo genético para o caso geral. Este algoritmo está baseado em uma nova formulação não linear. A formulação apresentada, permite resolver o problema quando se tem mais de uma placa, também quando se tem alguma região da placa que

não pode ser utilizada, e finalmente, permite resolver o problema quando é necessário a rotação das peças. Os resultados computacionais obtidos com esta formulação são os melhores até o momento, pois resolvem problemas com até 1000 tipos de peças e recopila a maior quantidade de testes usados na resolução desse tipo de problema. Além disso, gera uma nova base de dados de problemas baseados nos estudos de Fekete e Schepers (1997).

Para o problema de corte ótimo bidimensional tipo guilhotinado em placas com e sem rotação de peças, Cui (2007) apresenta um algoritmo que gera padrões de corte homogêneos em forma de T, enquanto para o mesmo problema sem rotação encontramos em Toro et al. (2008) um algoritmo híbrido usando busca em vizinhança variável e Simulated Anneling.

Na literatura existem muitos pesquisadores que têm-se dedicado a este ramo de pesquisa, e até hoje tem-se conhecimento de uma grande quantidade de publicações que abordam os problemas de corte e empacotamento, pois existem muitas aplicações na pesquisa operacional, na computação gráfica e na engenharia e tantas outras áreas. Portanto, nesta dissertação faz-se ênfase nas referências que permitiram evoluir as pesquisas na resolução desse tipo de problema. Algumas referências que são base para execução deste trabalho são: Dowsland e Dowsland (1992), Sweeney e Paternoster (1992), Wang e Washer (2002), Beasley (2004) e Leung et al. (2011) e aquelas que já foram mencionadas acima.

1.5 Estrutura do trabalho

O presente trabalho foi organizado em seis capítulos. Este primeiro capítulo é introdutório e apresenta o problema estudado, a técnica e os objetivos desta dissertação, além da revisão bibliográfica. A ele somam-se os seguintes capítulos:

No capítulo 2 é apresentado um panorama geral dos problemas de corte e empacotamento em duas dimensões.

No capítulo 3 é apresentada a modelagem matemática mais usada na literatura para o problema de corte ótimo bidimensional tipo não-guilhotinado. Adicionalmente são mencionados alguns procedimentos heurísticos (para distribuição das peças) e metaheurísticas para resolver o problema de corte de peças retangulares tipo não-guilhotinado em duas dimensões.

No capítulo 4 é apresentada uma descrição geral da técnica de otimização combinatória Simulated Annealing.

No capítulo 5 é apresentado um caso de estudo e uma análise detalhada dos resultados obtidos.

No capítulo 6 mostram-se as conclusões da pesquisa realizada, assim como algumas sugestões para trabalhos futuros.

2 GENERALIDADES DO PROBLEMA DE CORTE

2.1 Introdução

Neste capítulo, faz-se uma generalização dos conceitos básicos do problema de corte ótimo bidimensional, com a finalidade de estabelecer aspectos relevantes na definição dos diferentes problemas de interesse que tem a ver com o bom desenvolvimento deste trabalho.

O problema de corte bidimensional de peças dentro de uma placa é de grande importância na atividade industrial na manutenção de muitos materiais. Normalmente neste problema, com a finalidade de maior compreensão, cria-se um caso geral onde se dispõe de uma superfície S de determinado material que tem dimensão $L*W$. Além disso, se dispõem de um conjunto de n peças distintas, cada uma com dimensões diferentes dadas por $l_i * w_i$ e um benefício b_i associado a ela.

O problema consiste em encontrar a melhor distribuição das peças sobre a superfície S , dado um determinado número de peças de cada tipo, de tal forma, que se maximize o benefício obtido ou se minimize o desperdício do material.

Este problema pode ser formulado matematicamente da seguinte forma:

$$\max \sum_{i=1}^n b_i * x_i \quad (3)$$

Sujeito a:

$$\{R\}$$

Onde b_i, x_i é o benefício associado e o número de peças tipo i a serem cortadas respectivamente e $\{R\}$ representa o conjunto de restrições específicas de cada variante do problema, dado que as restrições podem mudar dependendo das características do problema em particular.

Na literatura, os problemas de corte são denominados como NP - Difíceis devido que os padrões de corte se incrementam exponencialmente com o número de peças a serem cortadas. Este é um dos motivos para usar técnicas metaheurísticas como ferramentas de

solução para este tipo de problema (JACKOBS, 1996).

O problema aqui descrito mostra a diversidade em todos os aspectos como, por exemplo, é possível ter vários tipos de placas, normalmente têm-se diferentes tipos de peças retangulares que estão definidas pelos seus tamanhos. Outra variante é que para a distribuição destas peças dentro de uma determinada placa, deve-se considerar a possibilidade ou não de rotar a peça num ângulo de 90° .

2.2 Aplicações na indústria

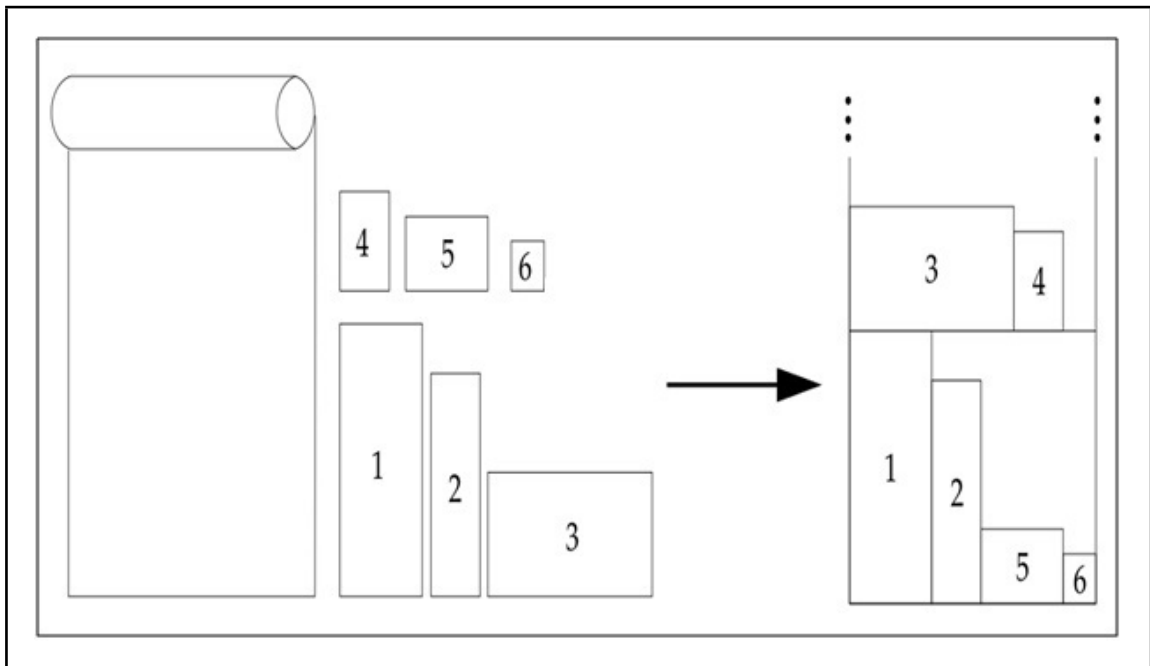
As aplicações deste problema na indústria são relevantes devido ao seu nível de importância dentro da produção de certos materiais que precisam de um corte especial para seu melhor aproveitamento, entre eles, o corte de madeira, papel, plástico, alumínio, couro, tecidos, vidro e alguns metais, de forma que os cortes são realizados sobre grandes superfícies destes materiais, para assim obter componentes retangulares. Encontrar a solução para este problema, também pode ajudar em outros contextos, como na alocação de mercadorias nos depósitos, onde cada produto tem que ser armazenado nas estantes; além disso, é usado nos jornais e artigos, para a alocação dos textos dentro do papel e obter o maior aproveitamento destes. Nos casos anteriores têm-se uma unidade de padronização que é o retângulo, e a função objetivo em comum entre estes problemas é alocar todas as peças requeridas num espaço menor. Este problema na literatura é conhecido como o problema bidimensional de Bin-Packing (2BP). Existem outros casos na indústria como é o corte de papel e tecidos, onde cada unidade de material é dada por rolos, e a função objetivo neste caso é cortar todas as peças, usando a menor área do rolo.

Imagine só grandes placas de vidro ou metal que devem ser cortadas em peças menores, para atender a demanda de diversos clientes. Neste caso, podemos nos preocupar apenas com a quantidade de placas a serem cortadas, já que as sobras do corte podem ser recicladas. Mas nem sempre é possível realizar o processo de reciclagem, como por exemplo, na indústria moveleira, onde as sobras de madeira são mais difíceis de serem recicladas, portanto tem-se a preocupação com o desperdício no corte.

Embora o problema de corte tenha em alguns casos as dimensões das placas bem definidas, existem outros problemas onde o tamanho da placa em uma das dimensões é ilimitado. Por exemplo, na indústria papelreira, se dispõe de grandes rolos de papel, como se mostra na Figura 9, que devem ser cortados em retângulos (peças) menores. Sua largura, em geral, é bem definida, mas seu comprimento, pode ser considerado in-

finito, devido que a soma dos comprimentos das peças a serem cortadas é muito menor que o comprimento do rolo. Este problema na literatura é conhecido como Problema de Empacotamento Bidimensional em Rolos (*Two Dimensional Strip Packing Problem*). Este problema é muito comum na indústria do papel, de polímeros e metalurgia, entre outros. Dada essa importância, muitos pesquisadores tem trabalhado na resolução deste problema, através do uso de metaheurísticas.

Figura 9 – Exemplo de corte num rolo



Fonte: (ANDRADE, 2006)

2.3 Introdução Histórica

O problema de corte bidimensional em placas (*Cutting and Packing Problem*) tem sido estudado há alguns anos atrás. No início foi apresentada uma proposta do problema pelo economista Kantoróvich no ano 1939 e por Brooks no ano 1940, mas as primeiras aproximações do problema à realidade, apareceu no ano 1961 com o Gilmore e Gomory, que em seu trabalho apresentou o problema de corte e empacotamento usando programação linear. Na história, esta publicação foi a mais significativa dentro desta área.

Nos anos 90's, Sweeney e Paternoster apresentam a recopilação mais completa de textos e trabalhos que tratam do tema de corte e empacotamento, permitindo ao leitor, ter

uma classificação dos estudos realizados até essa data, dividindo o texto segundo a dimensão do problema, tipo de problema e tipo de solução encontrada.

Gilmore e Gomory deram início a uma grande quantidade de trabalhos com diversos estudos que explicam as possíveis variações do problema. Embora, para alguns estudiosos do tema isto seja uma desvantagem, para outros, o fato de ter múltiplas opções de documentos que tratam do tema, torna o problema mais fácil de estudar, devido a extensa lista de trabalhos que faz com que o leitor interessado no tema, possa dar ênfases as variantes do problema, sem que perca o enfoque do tipo de problema a ser resolvido.

Dentro dos tópicos do problema de corte e empacotamento, surgem problemas onde a estrutura lógica e a finalidade são as mesmas, mas na literatura aparecem descritos com nomes diferentes, é por isto que Dyckhoff (1990), em um dos seus trabalhos, apresenta uma lista com os nomes mais usados para se referir a este tipo de problemas:

Problema de Corte e Empacotamento (Cutting and Packing Problems).

- *Marker Making Problems.*
- *Cutting Stock e Trim Loss Problems.*
- *Bin Packing, Dual Bin Packing, Strip Packing, Vector Packing e Knapsack Problem (Problema da Mochila).*
- *Problema de Carregamento do Container ou de um Veículo.*
- *Assortment, Dividing, Nesting ou Partitioning Problems.*
- *Line Balancing, Memory Allocation e Multiprocessor Scheduling Problems.*

2.4 Técnicas de solução do problema de corte bidimensional

Da mesma forma tem-se estudado diversas variantes do problema de corte, como também tem-se desenvolvido diferentes técnicas de solução do mesmo. Algumas das melhores aproximações surgidas são as seguintes:

- *Algoritmos de Programação Dinâmica.*
- *Métodos de Programação Inteira, para este caso, um dos algoritmos mais usados é a ramificação e delimitação, mais conhecidos como Branch and Bound.*

- *Métodos Heurísticos.*
- *Métodos de Inteligência Artificial como as Redes Neurais e sistemas especialistas.*
- *Métodos que usam o paralelismo.*

Os métodos anteriores se encontram descritos em Scheithauer e Terno (1993), onde são explicados para o caso da mochila unidimensional. Além disso, são apresentados os enfoques de cada uma das técnicas e finalmente são feitas comparações entre eles. Este trabalho faz um grande aporte aos pesquisadores do ramo, devido que apresenta a solução do problema de diferentes pontos de vista e permite ter uma visão geral dos diversos métodos aplicados.

Dowland e Dowland (1992), apresentou um trabalho de grande interesse, onde foi realizada uma seleção das variantes do problema de corte e empacotamento mais estudados, cada uma destas variantes foi analisada para conseguir uma estatística das soluções encontradas com maior relevância.

No mesmo ano, uma nova técnica foi apresentada por Cui et al. (2008). Neste trabalho apresenta-se um enfoque muito importante no desenvolvimento do problema de corte bidimensional. No documento foi apresentado um método que mistura procedimentos de ramificação e delimitação como Branch and Bound e técnicas de programação dinâmica. O funcionamento desta metodologia era o seguinte: tem-se uma placa para ser dividida em segmentos homogêneos verticais, onde as peças devem ser distribuídas para serem cortadas, e o padrão de corte é tipo guilhotinado. Para comprovar o bom funcionamento da metodologia, o autor fez 50 casos de prova, onde não se permite rotação das peças e cada caso permite 20 tipos de peças diferentes. Além disso, dá-se solução ao problema restringido onde não é necessário cortar todas as peças e o resultado final é apresentado baseado na porcentagem da área da placa utilizada.

Os trabalhos anteriores apresentados por diferentes autores despertaram o interesse no tema desenvolvido nesta dissertação, devido à grande diversidade de metodologias que se tem para resolver este tipo de problemas. Para obter um melhor entendimento do tema aqui apresentado, é necessário realizar uma classificação segundo o tipo de superfície que vai ser cortada, e por este motivo nos seguintes parágrafos se descrevem os diferentes problemas surgidos e que são de grande importância na literatura especializada.

2.5 Corte ótimo bidimensional de peças retangulares em uma única placa

Segundo as características do problema de corte, expressadas no Capítulo 1, as quais foram apresentadas na literatura por Dyckhoff (1990) e Lodi et al. (2002), é possível identificar o problema de corte ótimo bidimensional de peças retangulares em uma única placa, através dos seguintes identificadores (2,V,O,*,G,*,Z,E,L). Esta forma de representação reduzida do problema, é definido como:

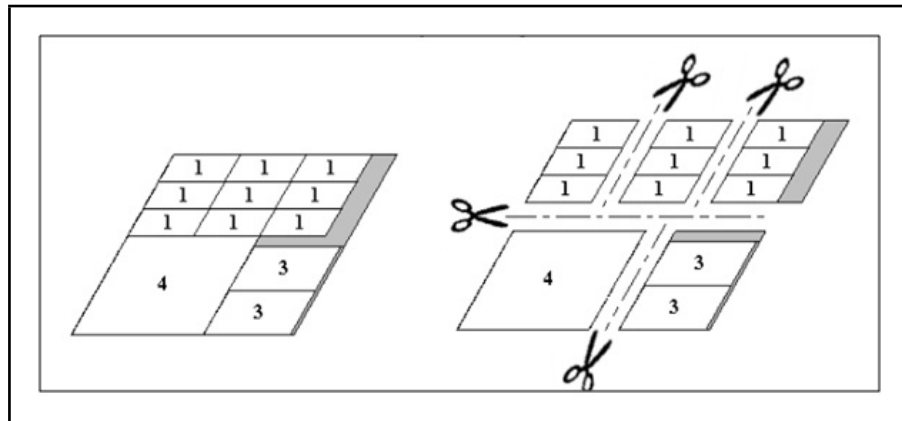
- (2) Está se tratando o problema de corte em duas dimensões..
- (V) Podem-se utilizar uma porcentagem das placas e todas as peças demandadas.
- (O) Utilização de uma única placa.
- (*) Refere a diversidade das peças demandadas.
- (G) quer dizer que as peças devem ser obtidas como resultado de aplicar sucessivos cortes tipo guilhotinados.
- (*) considera-se a opção de rotar as peças quando seja necessário, ou deixar as peças com orientação fixa.
- (Z) O benefício que se obtém pelo corte de cada peça é igual a sua área.
- (E) Têm peças com limite máximo de corte.
- (L) Têm peças de formas regulares.

A primeira formulação do problema (2,V,O,*,G,T,Z,E,L) é definida como: o corte de um retângulo que neste caso é denominado como placa de comprimento L e largura W . Um conjunto de retângulos de cardinalidade n que são denominadas como peças, de comprimento l_i e largura w_i , onde $(i = 1, \dots, n)$. Uma peça (l, w) é equivalente a uma peça (w, l) , como é mostrado na Figura 10.

Pode-se notar que para esta formulação do problema somente é modificada a sexta posição da nomenclatura usada anteriormente da seguinte maneira:

- (T) quer dizer que as peças podem rotacionar 90° .

Figura 10 – Disposição e corte das peças sobre uma placa



Fonte: (ÁLVAREZ et al., 2009)

Onde a função objetivo é dada pela seguinte equação:

$$\max \sum_{i=1}^n (l_i * w_i * z_i) \quad (4)$$

Sujeito a:

- Somatória das áreas das peças cortadas não devem superar a área total da placa.
- As peças não podem se sobrepor entre elas mesmas.
- A geração das peças deve ser dada pelo fato de realizar um corte de extremo à extremo sobre a placa de material ou subdivisões dela.

A segunda formulação do problema (2, V, O, *,G, A, Z, E, L) tem a mesma definição que o caso anterior, só que tem uma característica que fazem os dois problemas serem diferentes. Esta característica é a condição de orientação das peças. Isto quer dizer que uma peça (l, w) não é equivalente a uma peça (w, l) , como se apresenta a seguir:

(A) quer dizer que as peças têm orientação fixa.

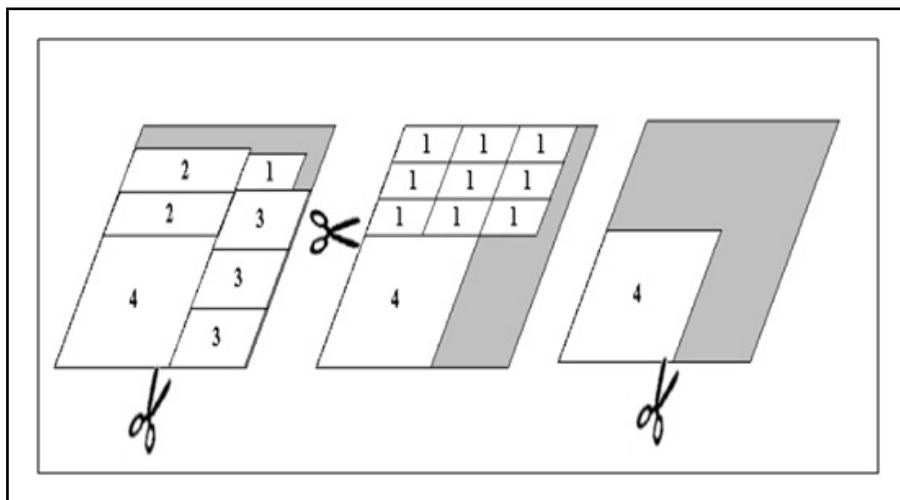
Os dois problemas definidos anteriormente têm sido trabalhados nos diferentes campos da otimização, como é a otimização exata e aproximada, também chamadas de heurísticas e metaheurísticas. Alguns dos trabalhos mais relevantes onde foram estudados estes dois tipos de problemas são apresentados na revisão bibliográfica no Capítulo 1.

2.6 Corte ótimo bidimensional de peças retangulares em várias placas

Com as características já apresentadas anteriormente no Capítulo 1, que descrevem a nomenclatura aqui utilizada, é possível representar o problema de corte ótimo em duas dimensões de peças retangulares em várias placas segundo os seguintes identificadores (2, B, I, *, G, *, Z, E, L). Este problema consiste no corte da totalidade de peças demandadas sobre placas idênticas bidimensionais, onde tem-se qualquer quantidade de peças. As peças devem ser obtidas depois de realizar sucessivos cortes tipo guilhotinados, além disso, se for necessário, permite-se ou não, a rotação das peças em um ângulo de 90° , dependendo do caso. O benefício que se obtém pelo corte de cada peça é igual a sua área, e finalmente os identificadores descrevem peças de formas regulares com limite máximo de corte.

A primeira instância do problema está dada por (2, B, I, *, G, T, Z, E, L) que se define como: cortar de um conjunto de retângulos que são denominados placas, um conjunto de retângulos menores de cardinalidade n que são denominados peças, de comprimento l_i e largura w_i , onde $(i = 1, \dots, n)$. Uma peça (l, w) é equivalente a uma peça (w, l) , como é mostrado na Figura 11.

Figura 11 – Corte e distribuição das peças sobre as placas



Fonte: (ÁLVAREZ et al., 2009)

Onde a função objetivo esta dada pela seguinte equação:

$$\min \sum_{k=1}^b L * W - \sum_{i=1}^n (l_i * w_i * z_i) \quad (5)$$

Sujeito a:

- Somatória das áreas das peças cortadas em cada placa não deve superar a área total da placa.
- As peças não podem se sobrepor entre elas mesmas.
- A geração das peças deve ser dada pelo fato de realizar um corte de extremo à extremo sobre a placa de material ou subdivisões dela.

A segunda instância do problema (2, B, I, *, G, A, Z, E, L), tem a mesma definição que o caso anterior, só que tem uma característica que fazem os dois problemas serem diferentes. Esta característica é a condição de orientação das peças. Isto quer dizer que uma peça (l, w) não é equivalente a uma peça (w, l) .

Os dois problemas definidos anteriormente têm sido trabalhados nos diferentes campos da otimização, como é a otimização exata e aproximada, também chamadas de heurísticas e metaheurísticas. Alguns dos trabalhos mais relevantes onde foram estudados estes dois tipos de problemas é apresentado a seguir.

Macleod et al. (1993) e Lodi et al. (1999), apresentam um resumo de avanços recentes nesta temática. Eles descrevem os limites disponíveis e os diferentes algoritmos exatos e aproximados. Um dos melhores trabalhos foi apresentado por Dell et al. (2002). Neste trabalho foi introduzido um limite inferior do problema com rotação das peças, sendo resolvido usando o algoritmo *branch and bound*.

A maior quantidade de algoritmos heurísticos aproximados que foram apresentados por Chung et al. (1982), Berkey e Wang (1987), Lodi et al. (1998) e Lodi et al. (1999), utilizam uma metodologia que consiste em duas etapas como é definido por Berkey e Wang (1987). A primeira etapa distribui todas as peças dentro do rolo de comprimento fixo e de largura infinita. A segunda etapa do problema utiliza a subdivisão do rolo em sub-rolos de comprimento e largura finita e finalmente as peças são alocadas em placas.

Berkey e Wang (1987) propõem dois novos algoritmos que foram chamados de *Finite-Best-Fit-Strip (FBS)* e *Finite-First-Fit (FFF)* com os quais pretendiam dar solução ao problema sem a rotação de peças. O algoritmo FBS aloca as peças seguindo a regra do melhor encaixe no rolo de forma decrescente, isto é, a peça é alocada no rolo atual, portanto a área total da peça pode estar contida dentro do rolo, caso contrário a peça deve ser alocada em outro rolo, alocando os sub-rolos resultantes dentro das placas com largura suficiente ou em uma nova placa se o sub-rolo não encaixa dentro de nenhuma das placas existentes. O algoritmo FFF, aloca as peças dentro de um rolo inferior ou em um novo rolo da primeira placa na qual possam ser alocadas as peças, caso contrário, na esquina inferior esquerda de uma nova placa.

Lodi et al. (1998) usa os algoritmos FBS e FFF para adaptar o problema de corte com rotação e apresenta uma aproximação *Floor Ceiling (FC)*. As peças são ordenadas de forma decrescente segundo a aresta mais curta, alocando uma peça de forma horizontal quando um novo rolo ou uma nova placa são criadas, permitindo assim, a alocação vertical em um rolo existente, sendo assim o corte factível. São modificadas as heurísticas: maior esquina inferior esquerda, inferior direita e a maior esquina superior direita do rolo, sendo desta forma considerada as restrições tipo guilhotinadas. Logo após desenvolvem a segunda etapa usando um algoritmo exato unidimensional *Bin packing problem (1BP)*. Assim, comprovam que o FC tem um melhor desempenho que o FBS e o FFF.

Também foram introduzidos na literatura alguns algoritmos aproximados, resultado da adaptação de diferentes técnicas metaheurísticas. Por exemplo, Lodi et al. (1991) usa um algoritmo de busca Tabu que explora o espaço de solução independente do problema de corte ou empacotamento, fazendo uso de vizinhança variável e de tamanho dinâmico. A busca é realizada através de uma heurística construtiva, que recombina um subconjunto de peças alocadas sobre um conjunto de placas, e uma peça alocada atualmente sobre uma placa vazia.

Para o problema de corte sem rotação das peças é utilizada uma heurística que pode ser resumida como um algoritmo de duas fases. A primeira fase de corte em rolos, requer a solução de vários problemas da mochila (*knapsack problem*) e a segunda fase do problema de corte em placas requer da solução do problema 1BP.

As heurísticas construtivas também têm sido adaptadas para o problema de corte com rotação de peças no trabalho de Lodi et al. (1991), onde introduzem o conceito de pseudo-peças. Além disso, é possível a orientação vertical de peças toda vez que o

problema permite.

Depois do aparecimento da proposta apresentada por Dowsland e Dowsland (1993), surge uma proposta feita por Faroe et al. (2003), quem utiliza um algoritmo heurístico aproximado para o problema de corte sem rotação de peças. Neste novo trabalho, é realizada uma busca local guiada, onde a vizinhança é explorada por meio de intercâmbio de peças. No processo de otimização, o algoritmo atribui de forma aleatória às peças alocadas em uma ou outra placa. Quando é encontrada uma solução infactível, uma nova função objetivo permite ver a totalidade de pares de peças que estão sobrepondo e deste modo se procede a minimizar a penalidade devido aos padrões que fazem que a função seja infactível. Quando o valor proposto para a incumbente é encontrado, ou um período de tempo fixado no do algoritmo é alcançado, termina-se a busca.

Outras heurísticas e metaheurísticas foram apresentadas na literatura para resolver variantes destes problemas. Um deles é o trabalho apresentado por Binkley e Hagiwara (2006) onde é descrita uma heurística de quatro esquinas que é usada junto com um algoritmo auto-adaptativo de Simulated Annealing e um algoritmo genético auto-adaptativo, para resolver o problema com padrão de corte tipo não guilhotinado. Um problema típico da indústria manufactureira de vidro é apresentado por Puchinger e Raidl (2007). Neste trabalho o número de cortes tipo guilhotinados permitidos não pode exceder três cortes. Para isto ser possível, foram criado dois modelos lineares inteiros de tamanho polinomial e um algoritmo *branch-and-price* baseado em uma formulação especial para o problema em duas dimensões.

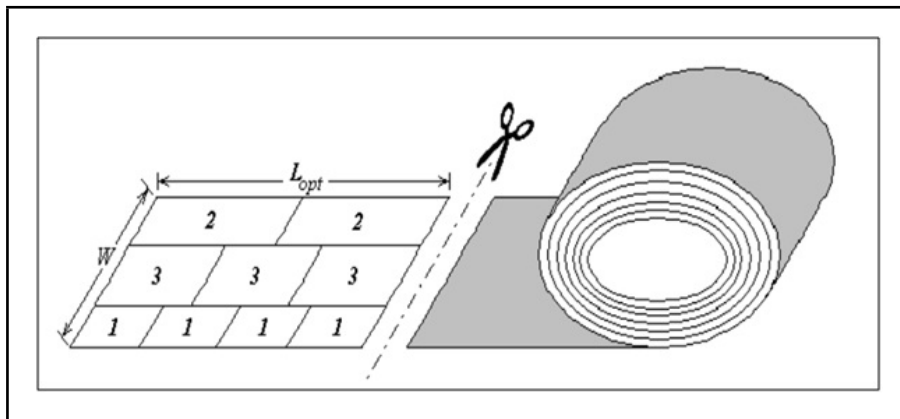
2.7 Corte ótimo bidimensional de peças retangulares em rolos

As características do problema de corte ótimo em duas dimensões, de peças retangulares em rolos, também conhecido como *Strip Packing Problem*, pode ser descrito usando os identificadores propostos por Dyckoff (1990) e Lodi et al. (2002), que foram apresentados no Capítulo 1. Para este tipo de problema corresponderia a seguinte nomenclatura (2, V, O, *, G, *, Z, E, L). Esta nomenclatura poderia ser expressa da seguinte forma: corte de todas as peças demandadas sobre um rolo de duas dimensões, onde tem-se qualquer tipo de peças. As peças devem ser obtidas depois de realizar sucessivos cortes tipo guilhotinados. Neste problema, igual ao problema de corte em placas, considera-se duas instâncias do problema, onde as peças podem rotar 90° e as peças têm orientação fixa, ou seja, não possuem rotação. O benefício que se obtém pelo corte de cada peça é igual a sua área, e finalmente os identificadores descrevem peças de formas retangulares com limite máximo

de corte.

A primeira parte do problema que corresponde a $(2, V, O, *, G, T, Z, E, L)$ sendo definida como: cortar de um retângulo que neste caso é denominado como rolo de comprimento infinito e largura W , um conjunto de retângulos de cardinalidade n que são denominadas como peças, de l_i e largura w_i , onde $(i = 1, \dots, n)$. Uma peça (l, w) é equivalente a uma peça (w, l) , como é mostrado na Figura 12.

Figura 12 – Corte e distribuição das peças sobre as placas



Fonte: (ÁLVAREZ et al., 2009)

Onde a função objetivo está dada pela seguinte equação:

$$\min L \quad (6)$$

Sujeito a:

- Todas as peças devem ser cortadas.
- As áreas das peças devem estar dentro do limite do rolo.
- As peças não podem se sobrepor entre elas mesmas.
- A geração das peças deve ser dada pelo fato de realizar um corte de extremo a extremo sobre a placa de material ou subdivisões dela.

A segunda instância do problema $(2, V, O, *, G, A, Z, E, L)$, tem a mesma definição que no caso anterior, só que tem uma característica que faz os dois problemas serem diferentes. Esta característica é a condição de orientação das peças. Isto significa que

uma peça de comprimento l e largura w não é equivalente a uma peça de largura l e de comprimento w .

A maioria dos trabalhos que tratam do problema de corte em rolos são algoritmos heurísticos aproximados. Fernandez de la Vega e Zissimopoulos (1998), Lesh et al. (2004), Kenyon e Rémila (1996), Zhang et al. (2006), apresentam algoritmos heurísticos aproximados para o problema *Strip Packing*. Bortfeldt (2006) e Beltran (2004), apresentam metaheurísticas para resolver este tipo de problema. Hopper e Turton (2001), apresenta uma descrição geral dos algoritmos metaheurísticos aplicados ao problema de corte ótimo em duas dimensões efetuado sobre rolos.

Na literatura existem poucos trabalhos que usam algoritmos exatos para resolver o problema de corte em duas dimensões em rolos. Hifi (1998) introduz o problema de corte e empacotamento realizando cortes tipo guilhotinados e propõe dois algoritmos baseados em *branch-and-bound*. Martello (2003), propõe um novo limite inferior e usa um algoritmo *branch-and-bound* para resolver o problema de corte em rolos restrito e realizando cortes tipo não-guilhotinados.

Recentemente têm sido propostos três trabalhos sobre o problema de corte e empacotamento em rolos com corte tipo guilhotinado. Em Cui et al. (2008) é proposto um algoritmo *branch-and-bound* recursivo para obter uma solução aproximada. Bekrar et al. (2007) introduz na literatura um novo limite inferior e um algoritmo *branch-and-bound*. Finalmente, Cintra et al. (2008) apresenta uma metodologia de geração de colunas e programação dinâmica.

3 MODELAGEM MATEMÁTICA E PROCEDIMENTOS HEURÍSTICOS PARA RESOLVER O PROBLEMA DE CORTE BIDIMENSIONAL DE PEÇAS RETANGULARES TIPO NÃO-GUILHOTINADO

3.1 Introdução

Considerando que existe uma quantidade significativa de publicações realizadas sobre o problema de corte bidimensional na literatura, neste capítulo é realizado o estudo de alguns trabalhos que representam o problema através de distintos modelos matemáticos e procedimentos heurísticos, com os quais é possível encontrar boas soluções devido à sua eficiência. Esta dissertação dedica este capítulo aos modelos matemáticos com os quais tem-se representado o problema de corte bidimensional e os procedimentos heurísticos que tem permitido o desenvolvimento de melhores técnicas de otimização para dar solução a este problema.

Inicialmente será feita uma abordagem geral dos modelos matemáticos que tem servido como base para resolver o problema de corte bidimensional de peças retangulares, logo após é apresentada as diversas técnicas heurísticas utilizadas na literatura para este problema.

3.2 Modelagem matemática para o problema de corte bidimensional

Os primeiros modelos matemáticos que apareceram na literatura para o problema de corte unidimensional foram apresentados em Gilmore e Gomory (1961) e Gilmore e Gomory (1963). Anos mais tarde os mesmos autores em Gilmore e Gomory (1965) e Gilmore e Gomory (1966) realizam um estudo detalhado do problema da mochila em uma e duas dimensões fazendo uso de modelagem matemática.

Outro estudo que resolve o problema de corte por meio de modelagem foi apresentado em Beasley (1985) de forma que se resolvia o problema para apenas uma placa bidimensional, sem restrições de corte tipo guilhotinado e onde a função objetivo pretendia maximizar o valor das peças alocadas. Neste trabalho Beasley propõe um modelo de programação linear inteira para definir as coordenadas discretas onde as peças podem ser alocadas. Outra modelagem similar foi proposta por Hadjiconstantinou e Christofides (1995).

Uma nova teoria foi introduzida na literatura para representar o problema de corte em Biro e Boros (1984). Utiliza-se a teoria de grafos para caracterizar os padrões de corte tipo não-guilhotinado. Além disso, a teoria de grafos também foi utilizada em Fekete e Schepers (1997) para determinar se a alocação das peças é factível, sem que as peças se sobreponha uma as outras e sem considerar as restrições de corte tipo guilhotinado.

Entre outros trabalhos, Lodi et al. (2002) e Lodi et al. (2004) apresentam um novo modelo que permite usar padrões formados por níveis. Este modelo considera somente restrições de corte tipo guilhotinados. Simultaneamente, Beasley (2004) apresenta uma formulação de corte tipo não-guilhotinado, usando modelagem matemática.

O problema de corte também tem sido abordado por uma ótica de modelagem matemática inteira mista em Cheng (1995). O trabalho apresenta o problema em três dimensões com contêineres, com um número de variáveis e restrições polinomiais. Este modelo tem sido tomado como a extensão da modelagem apresentada na literatura especializada por Onodera e Taniguchi (1991). Este modelo trata o problema de alocação de blocos em duas dimensões, baseando-se na enumeração de todas as possíveis alocações relativas de cada par de peças. Em Cheng (1995) não se consegue chegar a uma boa solução, devido que o modelo proposto tem algumas insuficiências na solução de problemas práticos de corte e empacotamento e os resultados computacionais foram os que o demonstraram. O mesmo modelo foi usado por Daniels et al. (1994), para ser usado na solução dos problemas de corte geral de polígonos em duas dimensões. Neste caso, comprova-se que o modelo foi ineficiente na prática.

Após a revisão bibliográfica geral dos modelos matemáticos apresentados para resolver o problema de corte e empacotamento, serão apresentados neste trabalho, alguns procedimentos heurísticos que tem sido relevantes para o desenvolvimento de melhores propostas de solução para o problema de corte bidimensional de peças retangulares tipo não guilhotinado.

3.3 Heurísticas para resolver problemas de otimização combinatória

A formulação geral dos problemas de pesquisa operacional podem ser de maximização ou minimização, e a função objetivo e as restrições podem ser lineares ou não lineares, sendo definido como:

Otimizar $f(x)$

Sujeito a:

$$h(x) = 0;$$

$$g(x) \leq B;$$

$$x \geq 0$$

Os métodos heurísticos estão fundamentados no fato da existência de problemas de otimização que pertencem à categoria denominada NP. Como já mencionamos anteriormente, para resolver este tipo de problema não existe um algoritmo de solução que seja polinomial, tendo assim um problema de otimização que pertence a esta categoria de problemas. Normalmente problemas deste tipo são resolvidos mediante o uso de técnicas heurísticas e metaheurísticas de otimização. Portanto, problemas onde a busca exaustiva não é suficiente, ou, onde a cardinalidade do espaço de busca aumenta exponencialmente com o tamanho do problema, tem sentido o uso de algoritmos heurísticos, como é o caso deste trabalho.

O uso de procedimentos heurísticos é recomendado quando se cumpre com uma ou mais de uma das seguintes condições:

- Problemas que não tem associado um método exato de solução, ou problema com método exato de solução que requer um alto nível de esforço computacional.
- Problemas que não requer encontrar uma solução ótima global, devido que é suficiente obter uma boa solução.
- Problemas onde tem-se dados pouco confiáveis, portanto, não é preciso encontrar o ótimo global para esses dados, já que soluções subótimas seriam equivalentes na prática.
- Quando existem limitantes de tempo para encontrar uma resposta, ou limitante nos recursos computacionais para efetuar os cálculos necessários.

- Quando o resultado fornecido pelo processo heurístico de otimização vai ser utilizado como solução inicial de um algoritmo exato, para reduzir notoriamente o número de iterações necessárias para encontrar uma boa solução.

A estratégia mais usada para realizar busca de forma heurística encontram-se dividida em quatro grupos:

- 1 **Heurísticas Construtivas:** sua finalidade é juntar componentes individuais das soluções iniciais até conseguir uma solução inicial factível.
- 2 **Heurísticas de melhoria de solução:** para este caso, se inicia com uma solução que em cada passo realiza uma busca dentro da vizinhança da mesma de uma melhor solução, se for encontrada, vai substituir a solução atual.
- 3 **Heurísticas de decomposição:** estes tipos de heurísticas dividem o problema em subproblemas mais fáceis de manejar, portanto, quando se tem a solução de cada subproblema, se obtém a solução do problema inicial, devido à integração das soluções obtidas em cada subproblema.
- 4 **Heurísticas de redução:** trata-se neste caso de identificar alguma característica que possivelmente pode ter a solução ótima, para assim poder simplificar o problema de busca.

As técnicas heurísticas podem ser criadas dependentes ou não do problema. Portanto a maioria delas são validas unicamente para um tipo de problema em particular. A diferença das técnicas metaheurísticas é que podem ser aplicadas a qualquer tipo de problema.

Considerando que o problema de corte bidimensional que está-se tratando neste trabalho tem um espaço de busca muito grande, justifica-se o uso de ferramentas heurísticas e metaheurísticas que permitam encontrar alternativas de solução para este problema, assim como é apresentado em Gilmore e Gomory (1966).

3.4 Procedimentos heurísticos para resolver o problema de corte bidimensional tipo não-guilhotinado de peças retangulares

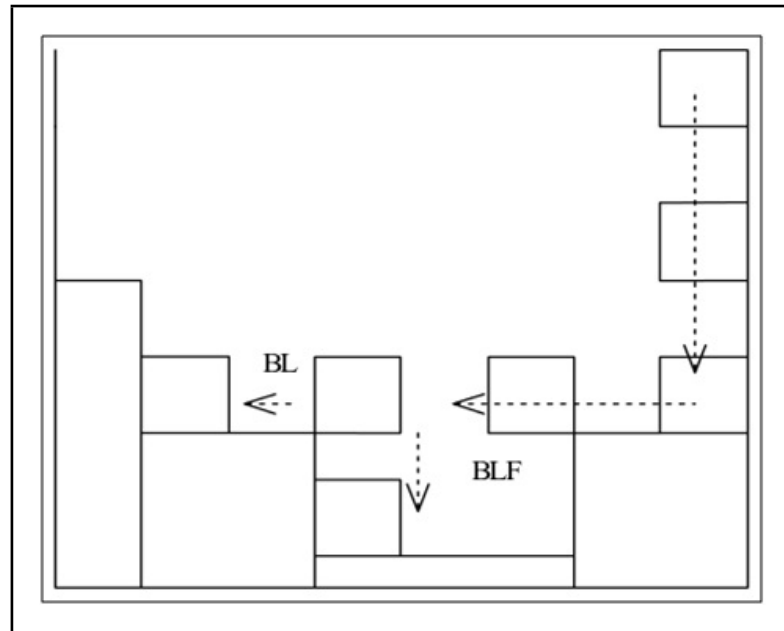
Para resolver o problema de corte tipo não-guilhotinado, tem surgido alguns métodos heurísticos que permitem fazer uma melhor distribuição das peças. Portanto serão descritos a continuação os métodos mais referenciados na literatura.

3.4.1. Métodos Construtivos para o problema de corte bidimensional

Um dos métodos heurísticos clássicos mais documentados na literatura é a estratégia *Bottom-Left* (BL) proposta por Baker et al. (1980). Nas últimas décadas têm sido propostas algumas variantes deste método. O algoritmo BL apresenta as seguintes propriedades:

- O primeiro elemento é um limite superior do número de padrões possíveis do corte. Neste caso, o algoritmo BL calcula o limite da seguinte forma: para um problema onde se tem n retângulos, o limite superior passa a ser $(2^n \cdot n!)$. Isto deve-se ao fato que o problema de corte está catalogado como um problema de permutação. Além disso, tem-se sequências de peças retangulares, onde cada retângulo aloca-se em duas direções, as quais tem que ser paralelas às coordenadas (x,y) . Segundo Gilmore e Gomory (1966), na prática, menos de $(2^n \cdot n!)$ soluções possíveis podem ser geradas pelo algoritmo BL.
- Segundo a estratégia BL, cada peça alocada, tem-se que posicionar inicialmente na esquina superior direita da placa. Após isto, desloca-se até a posição mais inferior da placa, e finalmente deve-se deslocar novamente até a posição mais a esquerda possível, portanto, a solução é dada pela ordem em que são introduzidas as peças e sua possível rotação. O grande problema que esta estratégia apresenta, é que gera as soluções com muito desperdício da placa que vai ser cortada. Devido a isto, foi proposto um novo algoritmo, que basicamente, é o BL modificado. Este novo método foi proposto em Chazelle (1983), sendo chamado de *Bottom-Left-Fill* (BLF). O objetivo geral do uso do algoritmo BLF, é minimizar o desperdício gerado pelo BL. Este algoritmo tem muitas semelhanças com o algoritmo BL, mais tem uma diferença que faz com que o BLF supere o BL. Esta característica que os diferencia, é que o algoritmo BLF antes de alocar cada uma das peças, comprova que a mesma não cabe em nenhum dos espaços gerados até o momento. Na Figura 13, apresenta-se graficamente a solução do problema a partir da óptica dos dois algoritmos.
- A estratégia BLF consiste em encontrar um espaço livre para cada peça, começando na esquina inferior esquerda da placa, isto é, na coordenada $(0,0)$. Na alocação de cada peça, o algoritmo é o encarregado de fazer o teste em cada nível da placa, para ver a possibilidade de alocar a peça em algum dos espaços livres, começando no nível mais baixo. Em caso de não ser possível alocar uma das peças em um dos níveis, a peça passa a ser alocada no seguinte nível superior. Todo o processo

Figura 13 – Estratégia BL e BLF



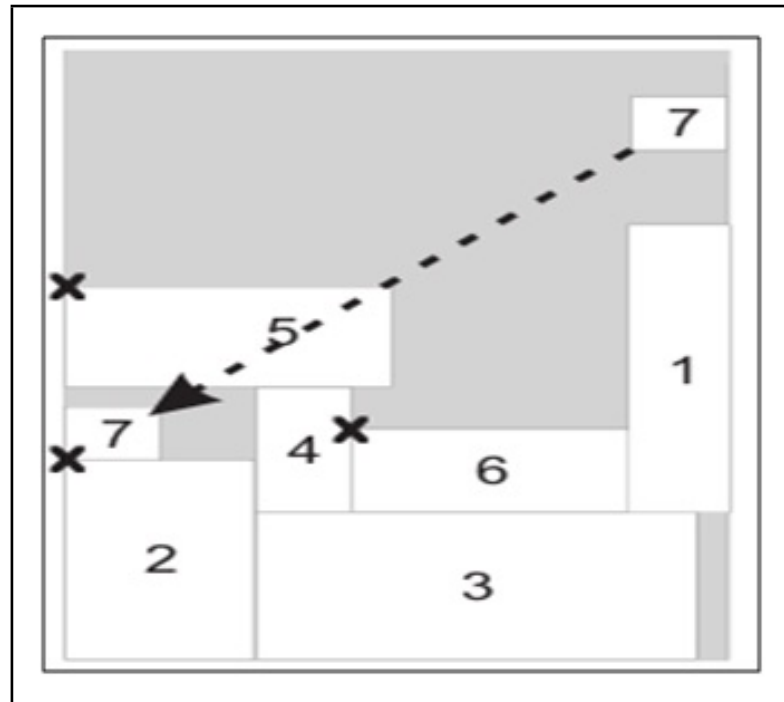
Fonte: (TORO et al., 2008)

que foi descrito, é repetido até conseguir fazer a melhor distribuição de todas as peças, obtendo-se assim, a partir do procedimento construtivo, uma solução inicial do problema.

Uma das heurísticas clássicas mais documentadas é a heurística *bottom-left* (BL) proposta por Baker et al. (1980). Nos últimos anos, tem sido propostas algumas variações deste método. Segundo Baker, temos que a heurística BL que aloca sucessivamente as peças retangulares na posição da placa mas profunda possível; após isto, desloca-as totalmente até a esquerda. Portanto, as soluções estão determinadas pela ordem em que as peças são introduzidas dentro da placa e a possibilidade de serem rotacionadas. Este método foi melhorado por Chazelle (1983) e foi chamado de *bottom-left-fill* (BLF). Este novo método pode encher os espaços produzidos no padrão de corte que se apresenta na Figura 14.

Jacobs (1996) utiliza outro método BL. Neste caso, cada peça é alocada inicialmente na esquina superior direita da placa. Após isto, é deslocada até a posição mais profunda possível. Uma vez nesta posição, a peça é deslocada para a esquerda até onde seja possível, repetindo esta rotina até que a peça esteja presa em uma posição. Uma das desvantagens deste método está na criação de áreas vazias no padrão de corte, como é apresentado na Figura 15.

Figura 14 – Heurística BL proposta em (Chazelle,1983)



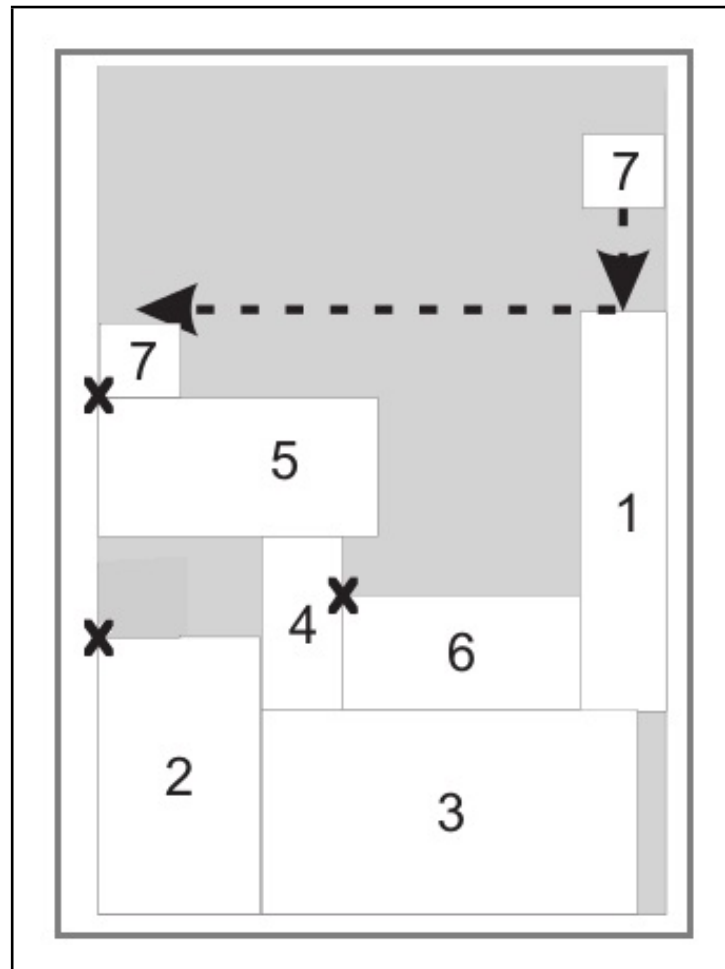
Fonte: (CHAZELLE, 1983)

Anos mais tarde, Liu e Teng (2002), desenvolveram outra heurística BL similar ao apresentado por Jakobs. Nesta metodologia, o deslocamento para abaixo tem prioridade, por tanto, a peça se desloca para a esquerda só se o o deslocamento para abaixo não é possível, como pode ser visto na Figura 16.

Na literatura, encontram-se variações da heurística BL, sendo que as mais representativas são as apresentadas em Chazelle (1983), Jackobs (1996) e Liu e Teng (2002). Nestes trabalhos, são feitas algumas modificações do algoritmo BL, mas o objetivo continua sendo o mesmo. Nas figuras 14, 15 e 16 são apresentados um exemplo dos padrões de corte obtidos pela heurística BL, segundo as variações realizadas pelos pesquisadores já mencionados.

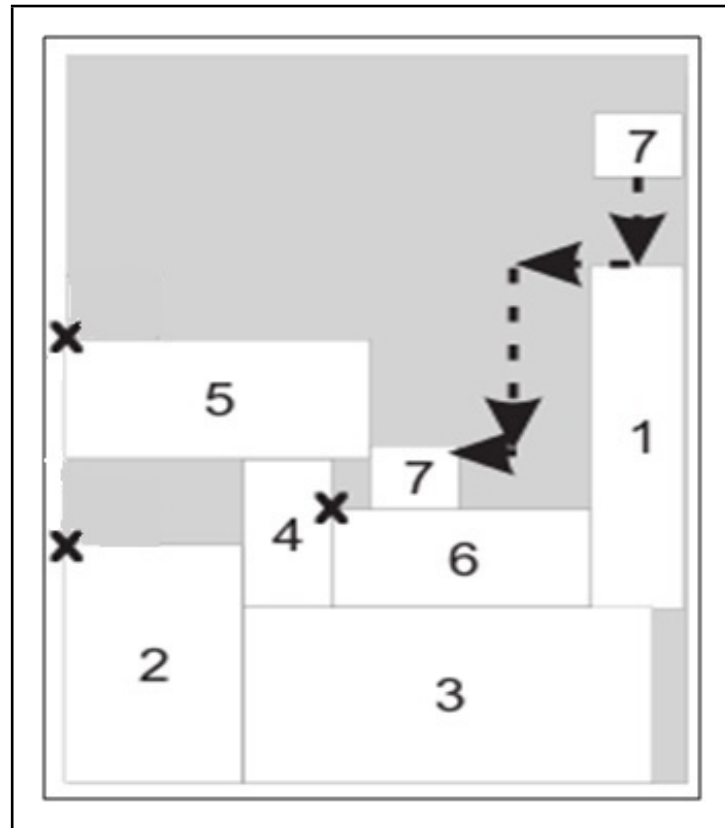
- Na busca por heurísticas que proporcionaram melhores soluções, aparece uma nova proposta feita em Burke et al. (2004). Esta nova heurística é chamada de *Best-Fit* (BF), sendo o funcionamento deste algoritmo baseado em um ordenamento dinâmico das peças retangulares em contraposição de uma permuta de peças. O algoritmo faz uma análise do conjunto de espaços disponíveis, começando pela posição mais a

Figura 15 – Heurística BL proposta em (Jakobs,1996)



Fonte: (JACKOBS, 1996)

Figura 16 – Heurística BL proposta em (Liu, 2002)



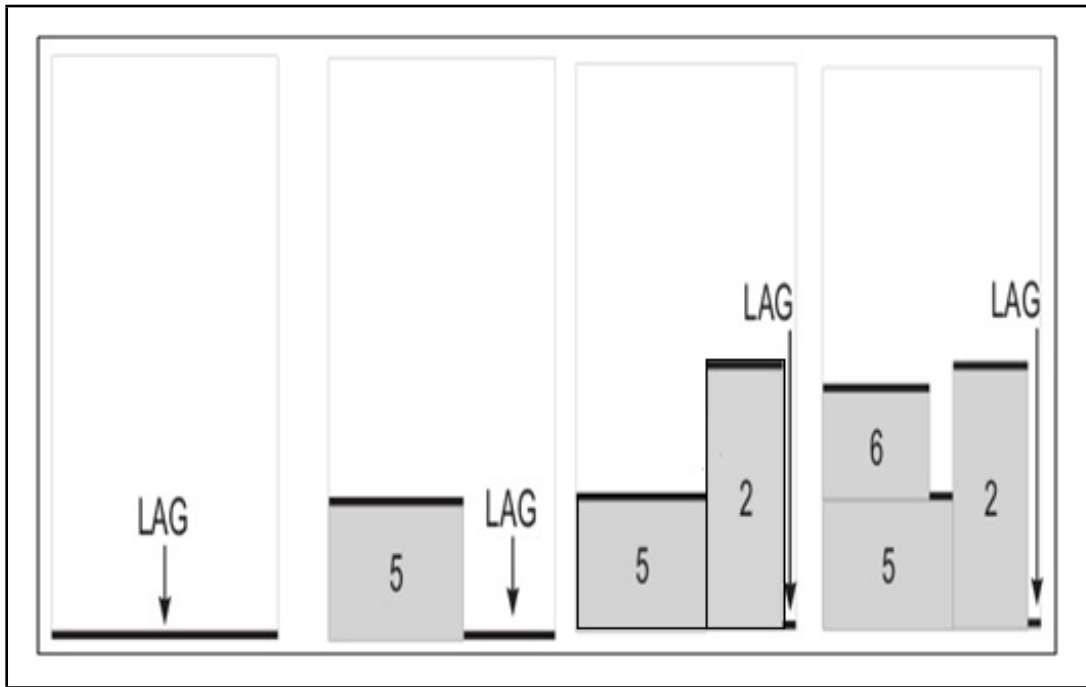
Fonte: (LIU; TENG, 2002)

esquerda e inferior possível, tal que esta posição foi denominada pelos autores como (LAG). Após esse procedimento passa-se a selecionar para cada espaço vazio a peça que melhor se encaixe, se existir alguma.

Portanto, o algoritmo permite tomar decisões, acerca da ordem de distribuição das peças, já que, informa qual será a seguinte peça e onde terá que ser alocada. Na Figura 17, mostra-se um exemplo do procedimento realizado por esta heurística, e mostra em cada caso a posição LAG. Com o uso de algumas metaheurísticas como TS, SA e AG, esta heurística tem desenvolvido alguns métodos para resolver o problema de corte tipo não-guilhotinado, os quais têm obtido resultados de boa qualidade.

- Com os resultados obtidos mediante o uso da heurística BL, têm sido desenvolvidas novas estratégias, entre elas encontramos *Difference-Process* (DP), apresentada em Lai e Chan (1997). Esta estratégia permite alocar cada peça na esquina que esteja

Figura 17 – Padrões de corte obtidos pela heurística em (Burke, 2004).



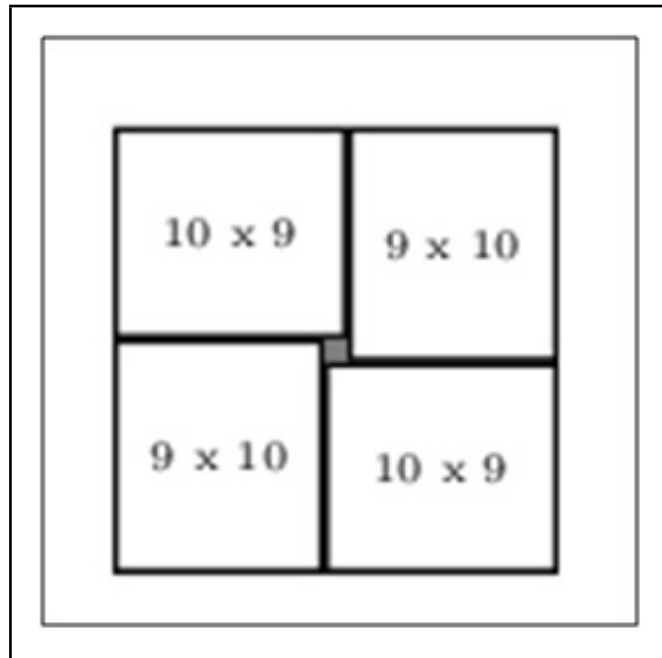
Fonte: (BURKE et al., 2004)

mais perto da esquina inferior esquerda da placa. Para medir a posição em que fica cada peça é usada a distancia euclidiana.

Todas as estratégias apresentadas na literatura permitem encontrar resultados de boa qualidade, mais deve-se lembrar, que ainda existem padrões de corte que não é possível ser alcançado com estas estratégias, assim como é mostrado nas Figuras 18 e 19.

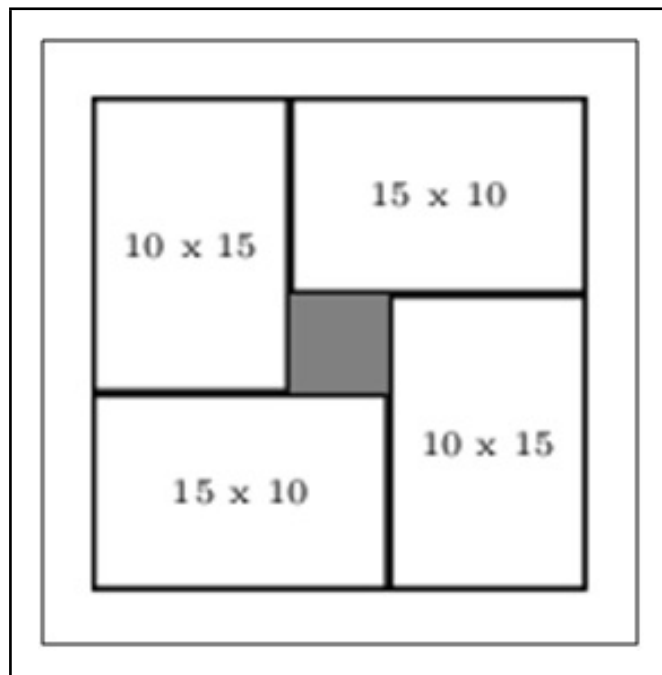
- Uma metodologia diferente para distribuir as peças é apresentada por Wu (2002). Neste trabalho propõe-se um procedimento determinista no qual tenta-se alocar as peças inicialmente nas esquinas e posteriormente no centro da placa. O algoritmo original foi desenvolvido para o problema especial onde: $P_i = Q_i \forall_i$, onde o número de peças de cada tipo i tem que estar dentro de unos limites P_i, Q_i com $(0 \leq P_i \leq Q_i)$, e é permitida a rotação das peças. O algoritmo corta as peças de uma em uma, na esquina do espaço vazio após realizar cada corte. Para escolher a peça a ser cortada e a esquina na qual será realizado o corte, é gerada uma função que estima o benefício que pode ser obtido ao completar o corte de cada peça.

Figura 18 – Padrão de corte impossível com BL



Fonte: (PARREÑO, 2004)

Figura 19 – Padrão de corte impossível com DP



Fonte: (PARREÑO, 2004)

Na literatura especializada é possível encontrar muitas formas de resolver o problema de corte bidimensional com e sem rotação de peças. No desenvolvimento

desta dissertação foi usada uma das heurísticas BL apresentada neste capítulo, em conjunto com uma técnica metaheurística SA que é apresentada a seguir.

4 AS METAHEURÍSTICAS

4.1 Introdução

O termo otimização encontra-se definido na literatura como o procedimento mediante o qual é avaliado um número finito de possíveis soluções que cumprem as características requeridas para resolver determinado problema, permitindo assim, encontrar a melhor solução de todas as que têm sido avaliadas. Os algoritmos de otimização combinatória tem sido classificados em duas categorias, as metaheurísticas populacionais e as metaheurísticas de busca por vizinhança. Portanto, é possível encontrar problemas de maximização e minimização até obter um valor específico, o qual é conhecido como função objetivo e seu valor depende de um número finito de variáveis de entrada. As variáveis de entrada podem ser independentes ou podem estar relacionadas com uma ou mais restrições. Gallego et al. (2008), define-se que, um problema de tipo inteiro pode ter uma grande quantidade de soluções factíveis e infactíveis, surgindo assim o problema de explosão combinatória, o qual é resolvido mediante o uso de técnicas heurísticas ou metaheurísticas.

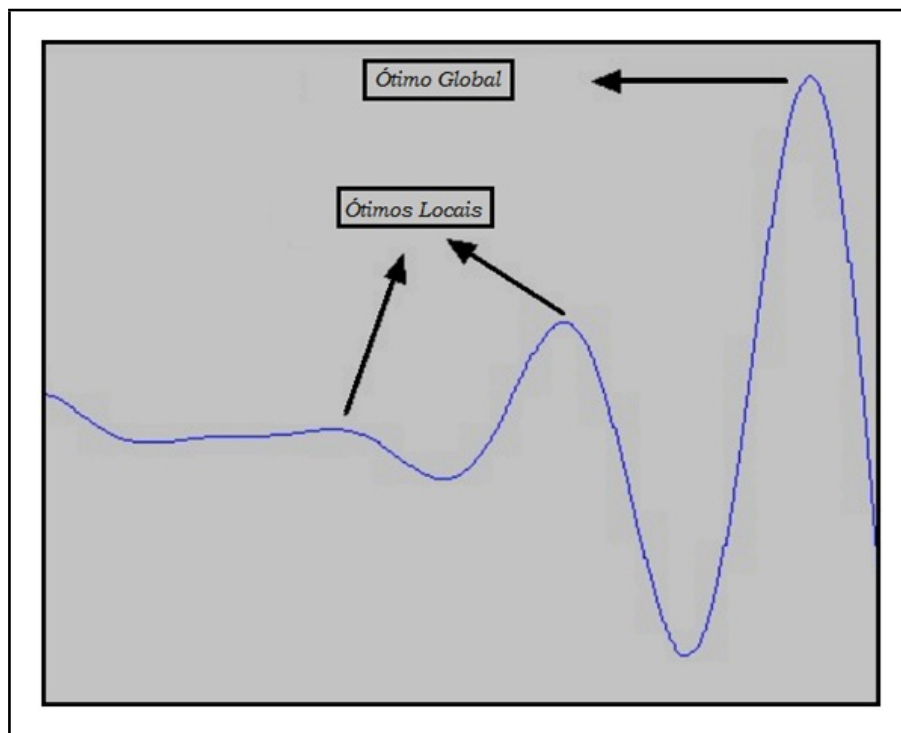
Uma das maneiras mais usuais de obter a solução de um problema combinatório, também chamado de *solução ótima*, consiste em avaliar todas as possíveis soluções do problema. Quando se tem problemas de grande porte e alta complexidade matemática, é requerido tempos computacionais muito altos (dias, meses ou anos) para resolvê-lo. No entanto, podem ser usadas dispositivos computacionais sofisticados e máquinas operando em paralelo, mas mesmo assim não resolvem o problema de tempo computacional. Uma alternativa de solução para este problema, é dada pelo uso de algoritmos que permitem encontrar soluções em tempo computacional razoável. É por isto, que foram desenvolvidos diversos métodos na literatura conhecidos como "*técnicas exatas*", as quais permitem diminuir a quantidade de tempo computacional para resolver um problema, dado que não avalia explicitamente todo o espaço de soluções, mas sim alguns pontos e subespaços do problema, além de avaliar de forma implícita os subespaços que ainda não foram analisados. O único problema que apresentam estas técnicas, é que só podem ser usadas para problemas pequenos e de baixa complexidade matemática. No caso de serem usadas em problemas de grande porte, estas técnicas precisam de tempo de processamento muito elevado, e em alguns casos não consegue encontrar a solução ótima ou soluções sub-ótimas do problema, assim como é apresentado em Gallego et al. (2008).

O surgimento dos algoritmos heurísticos foi dado pela necessidade de encontrar soluções de qualidade em tempos computacionais razoáveis. A palavra heurística é um

termo que tem suas raízes na palavra grega "*heuriskein*", que quer dizer "*encontrar*", e normalmente encontra-se relacionada com a capacidade de resolver de forma inteligente problemas reais utilizando o conhecimento disponível. É por isto, que tem sido desenvolvido grandes quantidades de procedimentos heurísticos para encontrar a solução de problemas de otimização específicos, logrando extrair estas soluções iniciais para serem usadas em outras aplicações ou problemas mais complexos.

A grande dificuldade que apresentam os algoritmos heurísticos é o fato de cair em ótimos locais, sendo que o algoritmo não consegue escapar dos mesmos. Esta característica é muito comum em problemas onde se tem vários pontos ótimos locais, como pode ser observado na Figura 20. Para encontrar a solução deste problema, têm surgido os procedimentos conhecidos como *metaheurísticas* ou *heurísticas modernas*.

Figura 20 – Dificuldades dos algoritmos heurísticos.



Fonte: Elaboração do autor

Metaheurística é um termo derivado de duas palavras gregas: *heurísticas* (explicada anteriormente) e o sufixo *meta*, que quer dizer de nível superior. Portanto, é possível definir as metaheurísticas como estratégias inteligentes capazes de melhorar os procedimentos heurísticos gerando resultados de alta qualidade.

Glover (1986), apresentou pela primeira vez o termo metaheurística, dando início a uma variedade de técnicas e critérios diferentes para resolver uma grande quantidade de problemas. O êxito na aplicação das técnicas metaheurísticas a um problema em particular, depende em grande medida do tipo de problema e do jeito em que são usados os distintos operadores das técnicas de otimização. Os diferentes tipos de metaheurísticas e os requerimentos necessários para modelar um procedimento como um problema de otimização é apresentado a seguir.

4.2 Classificação das técnicas metaheurísticas

A classificação das técnicas metaheurísticas está dividida em dois grupos segundo os trabalhos apresentados por Glover (2003), Cruz (2005), Pantrigo (2005) e Sucupira (2007). Nestes trabalhos define-se que muitos aspectos podem ser considerados ao realizarmos a classificação dos métodos metaheurísticos. Portanto, é possível dividir as metaheurísticas em duas classes, segundo as características que se estejam analisando. Devido à inexistência de uma categorização padronizada, esses métodos estão classificados como é mostrado a seguir: metaheurísticas populacionais e as metaheurísticas de busca por vizinhança.

a. Metaheurísticas populacionais.

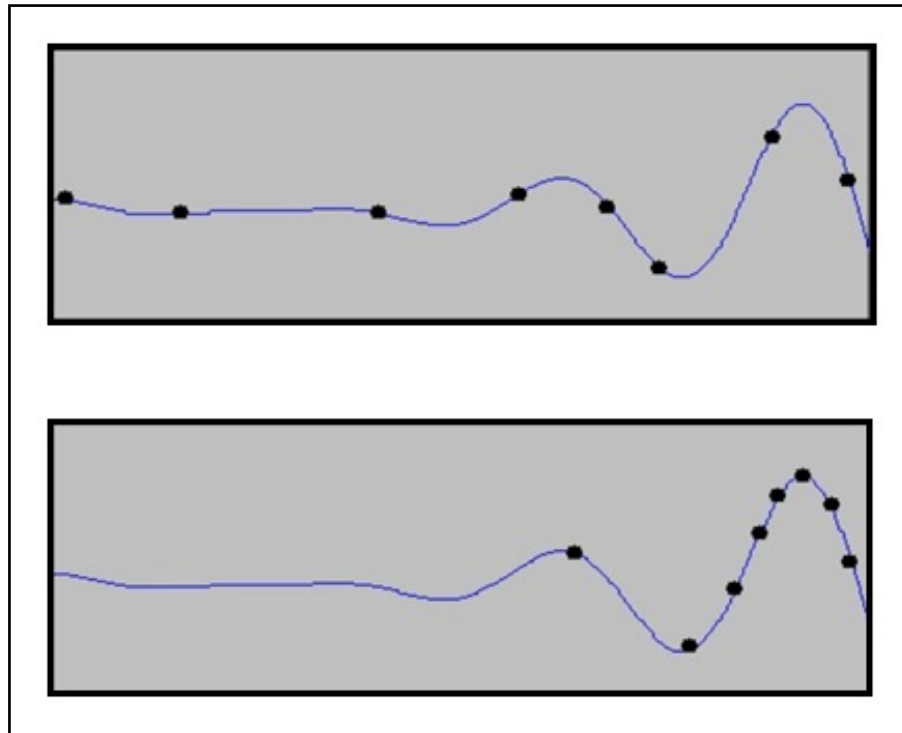
As metaheurísticas populacionais realizam a busca de uma possível solução, conservando de forma simultânea um conjunto de possíveis soluções, como pode-se observar na Figura 21.

Estes algoritmos se diferenciam dos demais, pela capacidade de tomar a informação fornecida pelos indivíduos da população, para combiná-las e posteriormente obter novas soluções através da evolução da informação inicial. Em outras palavras, pode-se dizer que as metaheurísticas populacionais trabalham com uma população, que é catalogada como um conjunto de soluções, as quais evoluem através de processos individuais ou de iterações entre os indivíduos, visando o aumento da qualidade média sem comprometer a diversidade do conjunto.

Dentro da classificação das metaheurísticas populacionais, destacam-se:

- *Algoritmos Genéticos.*
- *Algoritmos Meméticos.*

Figura 21 – Comportamento das metaheurísticas populacionais no processo de otimização



Fonte: Elaboração do autor

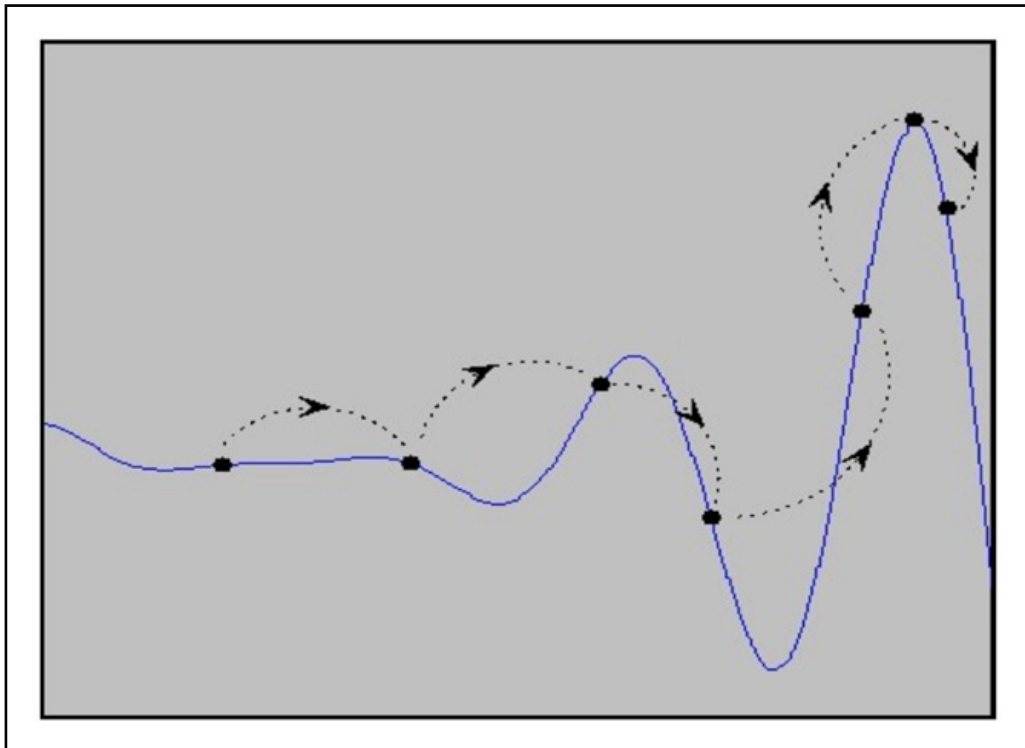
- *Otimização com Formigas Artificiais.*
- *Método Particle Swarm Optimization.*
- *Busca Dispersa.*
- *Reconexão de Caminhos.*
- *Algoritmos de Estimação de Distribuição.*

b. **Metaheurísticas de busca por vizinhança.**

Esta classe de metaheurísticas de busca por vizinhança trabalha em cada iteração com um conjunto de soluções, usualmente chamadas de *vizinhança*. Desta forma, os algoritmos baseados em vizinhança provêm uma forma natural e intrínseca de explorar o espaço de busca.

Cada solução encontrada é avaliada enquanto se realiza a busca de novas soluções e finalmente é selecionada a solução de melhor qualidade das soluções encontradas durante o processo de busca. Na Figura 22, apresenta-se uma melhor compreensão deste tipo de metaheurística:

Figura 22 – Comportamento das metaheurísticas de busca por vizinhança no processo de otimização.



Fonte: Elaboração do autor

Dentro da classificação das metaheurísticas de busca por vizinhança, destacam-se:

- *Recozimento Simulado (Simulated Annealing).*
- *Busca Tabu.*
- *Busca Local Iterada.*
- *GRASP (Greedy Randomized Adaptive Search Procedures).*
- *Busca Reativa.*
- *Busca Local Guiada.*
- *Busca por Vizinhança Variável.*
- *Hill-Climbing.*

4.3 Metaheurísticas como solução de um problema de otimização

Para que um problema de otimização combinatória possa ser modelado a partir de qualquer metodologia metaheurística, é preciso ter em conta uma série de passos, para

encontrar uma solução em particular que proporciona o maior benefício possível para o problema que está se resolvendo. Estes passos são enumerados a seguir:

***i* Representação de uma proposta de solução.**

A forma consistente de representar uma alternativa de solução de um problema dentro do espaço de soluções de dito problema, é através da codificação. O fato de se ter uma boa representação de uma proposta de solução de um problema, é um fator muito importante que permite encontrar sucesso na obtenção de resultados de excelente qualidade.

Devido ao fato de que propostas de solução podem ser factíveis ou infactíveis, deve-se criar uma estratégia que permita penalizar as propostas de solução que resultem ser infactíveis, garantindo assim, que a busca de uma boa solução, seja realizada em maior proporção dentro da região factível, obtendo-se assim uma solução real do problema.

Para verificar se uma proposta de solução encontra-se codificada de uma forma adequada, é necessário observar se a informação que a representa, permite calcular o valor da função objetivo do problema e determinar se a proposta é factível ou infactível. Além disso, uma proposta de solução é eficientemente codificada, quando é utilizada dentro de uma técnica metaheurística e permite o desenvolvimento dos operadores próprios de cada algoritmo.

Uma proposta de solução pode ser representada de formas diferentes, isto é, depende do modelo matemático do problema que está se resolvendo, e do tipo de valores que pode tomar cada variável, já que podem ser valores de tipo binários, reais, inteiros ou misto entre eles.

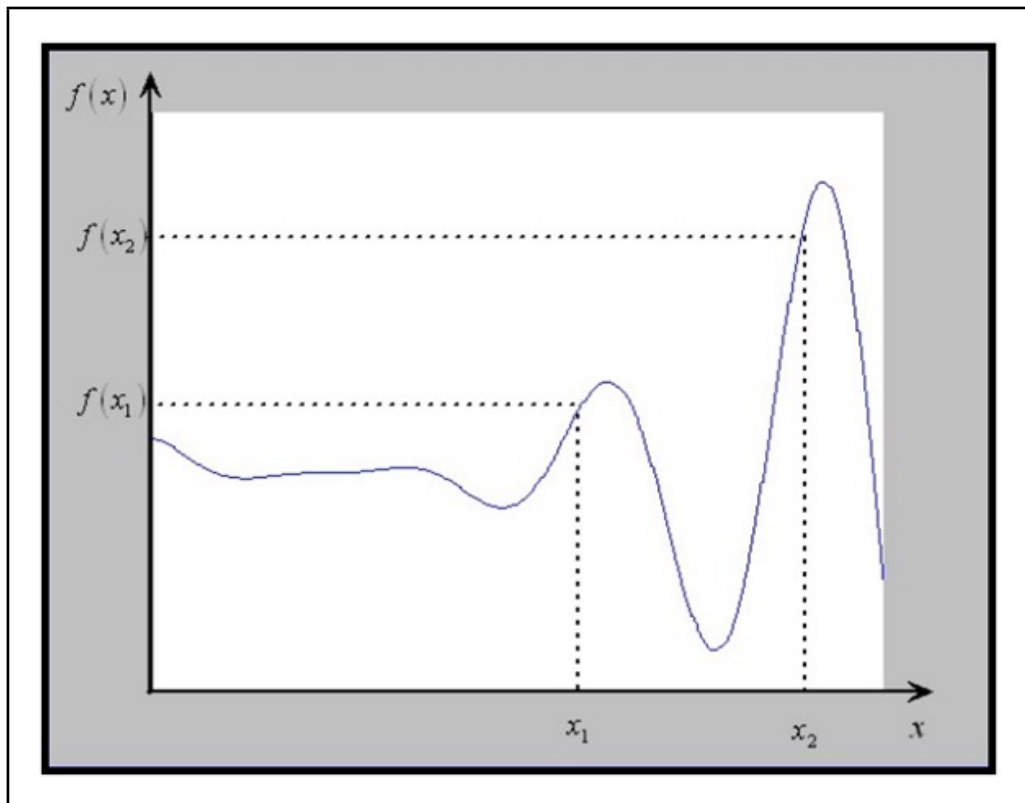
***ii* Formulação das restrições.**

As restrições do problema devem ser cumpridas pelas soluções, de forma que sejam consideradas como factíveis. As restrições se caracterizam por cumprir com um valor pré-estabelecido, e geralmente encontra-se expressa com uma desigualdade ($>$, \geq , $<$, \leq), ou por meio de uma igualdade ($=$) quando a restrição tem que cumprir com um valor estritamente indicado.

iii Definição da função objetivo.

Dependendo do problema a ser resolvido deve-se definir a forma matemática da função objetivo a ser otimizada. A função objetivo geralmente denominada $f(x)$, permite avaliar a qualidade das alternativas de solução, para assim, compará-la entre uma solução **A** e uma solução **B**, e determinar qual é a melhor segundo o contexto do problema. Na trajetória do processo de otimização, pretende-se encontrar uma solução para a função objetivo do problema que está sendo tratado, mas por ser um problema de maximização ou minimização, busca-se um valor máximo ou mínimo que cumpra com as restrições do problema e, além disso, que represente a função objetivo do mesmo, como ilustrado na Figura 23.

Figura 23 – Representação gráfica de dois valores ótimos possíveis para uma função objetivo.



Fonte: Elaboração do autor

iv Especificação dos limites das variáveis.

As variáveis podem ter limites que permitem diminuir o espaço de busca, para encontrar a solução ótima do problema em uma menor quantidade de tempo. Geralmente estes limites são definidos no início da resolução do problema e podem ser

maiores ou menores do que um valor dado. Mas também existem problemas onde as variáveis têm um valor máximo e um valor mínimo que são fixos, denominadas de variáveis canalizadas.

v Criar o modelo matemático. Os problemas de otimização podem ser representados da seguinte forma:

Otimizar uma função objetivo
Sujeito a
Um conjunto de restrições

Portanto, a representação matemática do modelo usando relações matemáticas simples terá a seguinte forma:

$$\min f(x)$$

Sujeito a:

$$g_i(x) \leq 0 \quad i = 1, \dots, l$$

$$h_i(x) = 0 \quad i = 1, \dots, m$$

$$x_i \leq k$$

$$L_{inf} \leq x_j \leq L_{sup}$$

vi Definir a vizinhança.

O bom desempenho do algoritmo está sujeito em grande proporção a estrutura da vizinhança, a qual tem varias formas de ser definida dependendo do problema que se esteja resolvendo. Para definir o tipo de vizinhança é necessário contar com vários aspectos que influenciam na obtenção de bons resultados, sendo dois destes aspectos: o tipo de problema e a técnica de solução selecionada para resolver o problema. As diversas metodologias de otimização geralmente iniciam o processo de solução com uma solução inicial que pode ser factível ou infactível, e mediante um processo de transição, é escolhida uma solução vizinha da solução atual. Este processo é repetitivo, de acordo com a lógica de cada algoritmo o qual passa de uma solução

(configuração) a outra solução (configuração) vizinha, até cumprir com o critério de parada. No percurso do processo de transição entre as diversas soluções, se mantém uma solução chamada de solução atual, além disso, também é armazenada a melhor solução encontrada, enquanto são realizadas as transições. O processo de busca termina quando se obtém como resposta do algoritmo a melhor solução encontrada, a qual é denominada de incumbente.

***vii* Encontrar uma solução.**

O melhor valor encontrado num espaço de possíveis soluções, que permita maximizar ou minimizar uma função objetivo, segundo seja o caso, é definida como o ponto ótimo de um problema de otimização. Mas como todos os problemas tem relacionado uma dimensionalidade e uma função objetivo, podem ser encontrados múltiplos pontos ótimos locais que representam somente o melhor valor dentro de uma vizinhança de um espaço de busca.

Dado o caso de otimização sem restrições, qualquer solução gerada dentro dos limites das variáveis, que tenha melhor valor da função objetivo, segundo a natureza do problema, pode ser avaliada e escolhida, ao contrário da otimização com restrições, onde a solução que é escolhida como a melhor, além de estar dentro dos limites das variáveis, tem que ser factível, isto é, tem que cumprir com as restrições de igualdade e desigualdade do problema.

***viii* Definir o critério de parada.**

Dado que os algoritmos de otimização precisam de algum critério para decidir em que momento finalizar a exploração do espaço de soluções, têm sido criados vários critérios de parada. O critério de parada depende do problema em que se esteja trabalhando e cada pesquisador pode definir seu próprio critério de parada segundo o tipo do problema.

Alguns exemplos de critérios de parada mais utilizados na literatura para resolver problemas de otimização são os seguintes:

- Quando no transcurso de um número dado de iterações não se consegue melhorar a solução incumbente, sendo assim a mesma é escolhida e se finaliza o processo de busca.

– Número máximo de iterações.

A seguir, é descrito o tipo de metaheurística proposto como metodologia de solução do problema tratado nesta dissertação.

4.4 Simulated Annealing

A técnica metaheurística chamada de Simulated Annealing (SA), é uma técnica introduzida na literatura nos inícios dos anos 80 por Kirkpatrick e Vecchi (1983), o qual se inspirou em um procedimento termodinâmico utilizado na metalurgia, com o objetivo de passar um objeto do estado sólido a alta temperatura para um corpo perfeito chamado de cristal e, neste processo, a temperatura é diminuída preservando o equilíbrio térmico (KIRKPATRICK; VECCHI, 1983; PETER; ARTS, 1992). A técnica de annealing está dividida em três fases: a primeira é uma fase de aquecimento que depende do tipo de material e de seu nível de deformação; na segunda fase, a temperatura é mantida constante permitindo assim, uma realocação molecular em estados de energia mínima; na terceira e última fase, é realizado um esfriamento controlado, para diminuir gradualmente a temperatura até obter um sólido onde as partículas são encontradas num estado cristalino perfeito. No início da fase de esfriamento, deve ser permitido o alcance do ponto de equilíbrio térmico para cada valor de temperatura, caso contrário, o sólido pode apresentar uma estrutura amorfa e não a estrutura cristalina de mais baixa energia.

No ano 1953 foi proposto o algoritmo chamado de "*Algoritmo de Metrópolis*", o qual era usado para estudar as propriedades de equilíbrio nas análises do comportamento microscópico dos corpos. Este algoritmo deu origem a uma das metodologias de otimização combinatória mais usadas na literatura como é o SA.

O algoritmo de Metrópolis tem suas bases nas técnicas de simulação de Monte Carlo, onde se geram sequências de estado de um sólido, isto é, dado um sólido em um estado i e com um nível de energia E_i , pode ser gerado o seguinte estado j com nível de energia E_j mediante o uso de um mecanismo que conduz ao estado seguinte através de uma pequena perturbação. Se a diferença dos níveis de energia $E_j - E_i \leq 0$, o estado j é aceito, caso contrário, o estado j é aceito com determinada probabilidade, sendo dada por:

$$p = e^{\left\{ \frac{E_j - E_i}{k_b T} \right\}} \quad (6)$$

Onde T representa a temperatura e k_b representa a constante de *Boltzmann*. Esta regra de aceitação é conhecida como *critério de Metrópolis*.

Portanto, para cada valor de temperatura T , o material em estado sólido chega a um estado de equilíbrio térmico, o qual é representado através da distribuição de probabilidade de Boltzmann, como mencionado anteriormente, onde a probabilidade de um material em estado sólido esteja num estado i com um nível de energia E_i a uma temperatura T , é dado pela equação a seguir:

$$p\{X = 1\} = \frac{1}{Z(t)} e^{\left(\frac{-E_i}{k_b T}\right)}$$

Na equação anterior:

- X representa uma variável estocástica do estado atual do sólido.
- $Z(t) = \sum_j e^{\left(\frac{-E_j}{k_b T}\right)}$ é um fator de normalização, comumente denominado como função de partição.
- k_b é a constante de Boltzmann.
- $e^{\left(\frac{-E_i}{k_b T}\right)}$ é conhecido como fator de Boltzmann.

Generalizando, é possível dizer que o processo annealing encontra-se composto por duas fases:

- i Incrementa-se a temperatura T até chegar a um valor máximo.
- ii Diminui-se lentamente a temperatura de T e preserva-se o equilíbrio termodinâmico até que as partículas consigam-se reacomodar em um estado de energia mínima, o qual é denominado "*estado fundamental do sólido*".

Na área de otimização combinatória tem sido desenvolvido um algoritmo que aplica conceitos similares ao processo do recozimento dos sólidos. Este algoritmo consegue ser eficiente na hora de realizar a busca de soluções de boa qualidade em problemas com complexidade matemática e de grande porte.

As primeiras provas realizadas com este algoritmo foram aplicadas ao problema do caixeiro viajante (PCV), mas na atualidade, existem múltiplas aplicações na área de pesquisa operacional e engenharia. Isso se deve à qualidade dos resultados obtidos que dependem

unicamente da definição dos operadores do processo de otimização: a vizinhança e o programa de esfriamento.

4.5 Descrição do Algoritmo do Simulated Annealing

As características principais do algoritmo de otimização SA são: a capacidade de realizar buscas de soluções ótimas através de uma única solução capaz de explorar sua vizinhança através de mínimas modificações e, além disso, consegue escapar de pontos ótimos locais. Isso deve-se a uma definição análoga com a probabilidade de aceitação dada pela equação (3). Esta definição estabelece que:

"Dada uma instância de um problema de otimização combinatória e denotados como i e j duas soluções com custo $f(i)$ e $f(j)$ respectivamente, o critério de aceitação determina se j é aceito a partir de i ao aplicar a seguinte probabilidade de aceitação:"

$$P_{\text{aceit}} = \begin{cases} 1, & \text{se } f(j) \leq f(i) \\ e^{\frac{f(i)-f(j)}{k \cdot T}}, & \text{se } f(j) > f(i) \end{cases} \quad (6)$$

Onde, neste caso, T tem as unidades de custo, isto é, as mesmas unidades de $f(i)$.

Na equação anterior é possível verificar que a probabilidade de aceitar uma nova configuração de pior qualidade ($f(j) > f(i)$), é o resultado de comparar a expressão correspondente com um número aleatório gerado dentro de uma distribuição de probabilidade uniforme no intervalo $[0,1]$.

Dentro do processo de otimização combinatória SA, existe uma alternativa de solução que é análoga ao estado atual de um sistema físico. Além disso, o valor da sua função objetivo é equivalente à energia deste estado. A seguir é apresentada uma descrição de cada um dos parâmetros que atuam dentro do algoritmo.

- *Algoritmo Simulated Annealing.*
- *Temperatura inicial.*
- *Comprimento da cadeia.*

- *Taxa de resfriamento.*
- *Definição da vizinhança.*
- *Critério de parada.*

A seguir serão descritos os itens anteriormente mencionados.

4.5.1. *Algoritmo Simulated Annealing*

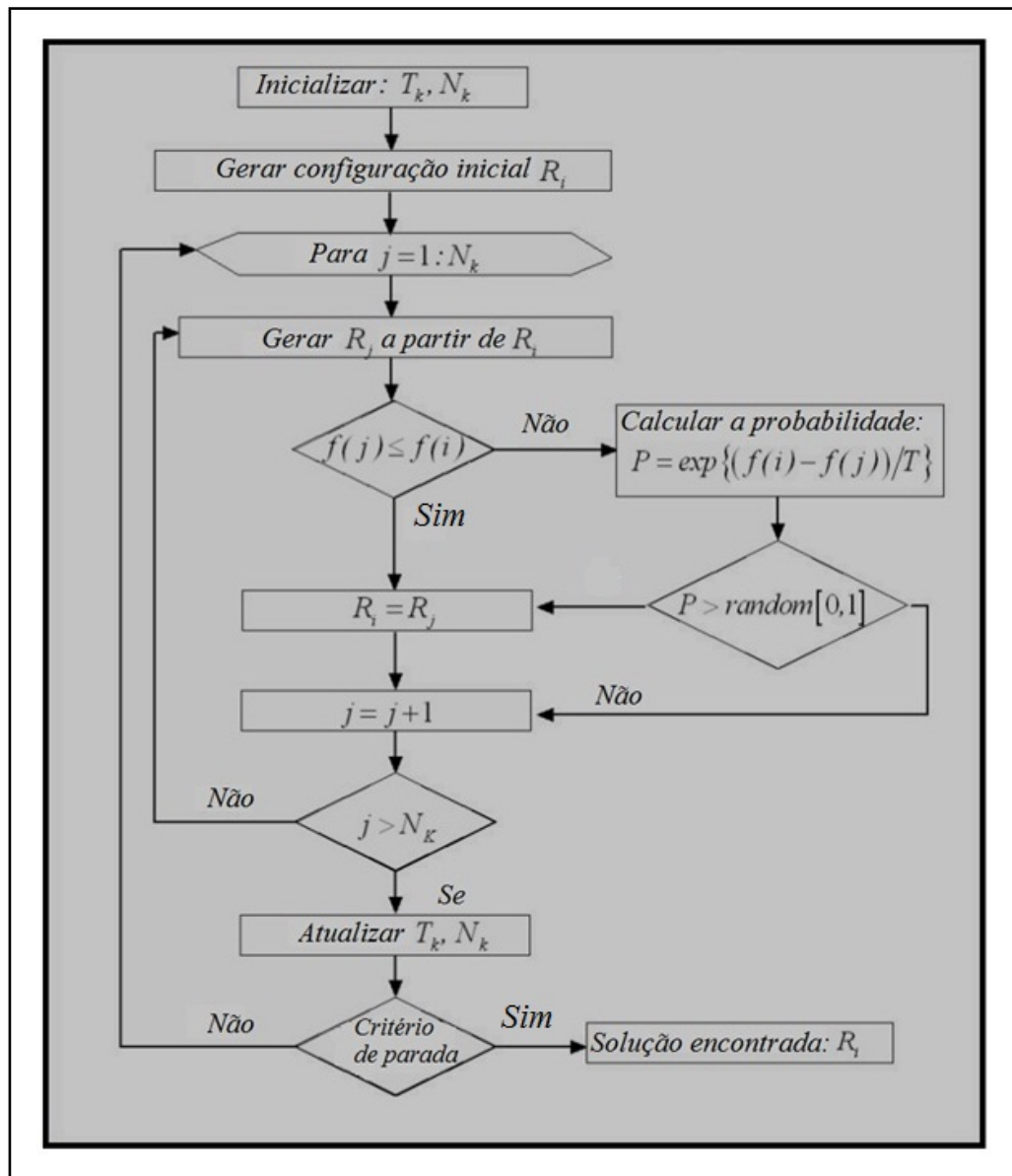
Uma melhor panorâmica do algoritmo SA pode ser apreciada no diagrama de fluxo apresentado na Figura 24, onde se descreve o algoritmo geral para resolver um problema de minimização.

4.5.2. *Temperatura inicial*

Para conseguir bons resultados na exploração das fases do algoritmo SA, é necessário determinar um valor adequado para a temperatura. Portanto, usa-se uma estratégia para que nos primeiros passos do algoritmo, este permita realizar uma transição entre soluções de pior qualidade, devido que encontra-se longe de um ponto ótimo global. Esta estratégia consiste em atribuir a temperatura inicial em um valor alto, dado que no algoritmo a temperatura vai diminuindo gradualmente, como também diminui a possibilidade de aceitação de soluções de má qualidade, devido que a exploração de sua vizinhança é realizada em regiões que estão perto do ponto ótimo do problema. O valor da temperatura inicial (T_0) é obtido mediante o uso de um processo construtivo, onde são simuladas as transições feitas no início do processo para criar a primeira cadeia de Markov. Dado que a Cadeia de Markov e a temperatura têm uma relação inversa, quando são usadas cadeias com comprimentos maiores, a temperatura inicial é menor.

Dentro do algoritmo SA o conceito físico de temperatura não representa um significado real, mas sim encontra-se catalogado como um parâmetro cujas unidades são as mesmas de $f(x)$, de grande importância que é usado para calibrar o algoritmo. O valor deste parâmetro depende principalmente da complexidade do problema tratado, devido que, valores muito altos geram um esforço computacional maior dentro do processo e os valores que são baixos fazem com que o algoritmo no início do processo, fique encerrado dentro de soluções de baixa qualidade.

Figura 24 – Algoritmo da técnica de otimização Simulated Annealing.



Fonte: (RUIZ, (2009))

4.5.3. Comprimento da cadeia

Dado que, a cada diminuição da temperatura, a probabilidade de aceitação também é afetada, já que a mesma diminui, é necessário aumentar o número de possibilidades a serem avaliadas. Este número de possibilidades em cada passo do processo iterativo é denominado Cadeia de Markov ou número de tentativas N_k . O comprimento da Cadeia de Markov N_k é usada para definir o número de vizinhos para serem explorados em cada nível de temperatura T_k . Para chegar a uma exploração mais intensa das temperaturas baixas, cria-se uma regra que permite manipular uma cadeia com comprimento crescente, ao tempo que a temperatura diminui.

Inicialmente para definir o comprimento da cadeia N_0 , é escolhido um valor proporcional ao tamanho do problema, por exemplo, k vezes o número de variáveis, portanto, $N_0 = K * num_{var}$. Para calcular a longitude da cadeia nas iterações seguintes, são usadas duas expressões matemáticas, apresentadas a seguir:

$$N_{k+1} = \beta N_0$$

Onde N_0 representa o comprimento inicial da cadeia e β tem um valor maior do que 1. A outra expressão matemática é a seguinte:

$$N_{k+1} = \rho N_k$$

Onde o comprimento da nova cadeia (N_{k+1}) depende do valor anterior de (N_k) e ρ é um valor constante que permite o crescimento constante da cadeia em uma porcentagem certa. As expressões anteriores permitem realizar o uso de valores da cadeia de forma constante ou variável respectivamente.

4.5.4. Taxa de resfriamento

O valor da taxa de resfriamento (T) é de grande importância dentro do algoritmo SA, dado que permite a diminuição do valor da função objetivo. Este processo é conhecido como programa de resfriamento e é representado mediante o uso de uma taxa η como se mostra a seguir:

$$T_{k+1} = \beta * T_k$$

A taxa de resfriamento recomendada na literatura encontra-se delimitada pelo seguinte intervalo $[0.8, 0.99]$ que corresponde a um resfriamento lento, que permite uma exploração intensa a temperaturas baixas. Portanto, o valor da cadeia N_k tem um valor tão grande que permite que o sistema alcance seu estado de equilíbrio para o nível de temperatura T_k . O valor de N_k pode ser considerado como crescente quando T_k é decrescente. Embora, esta regra não é geral, e para cada aplicação pode ser criado um programa de resfriamento específico. Uma solução sugerida por diferentes autores para sair de pontos ótimos locais e conseguir realizar a exploração de novas regiões, é o uso de uma técnica denominada "*aquecimento*", para melhorar as soluções geradas na busca atual.

O critério anteriormente apresentado, não é o único que existe para determinar a taxa de diminuição de temperatura. Na literatura pode-se encontrar outras alternativas que avaliam o desempenho do processo no nível de temperatura T_k , e assim estabelecer o valor T_{k+1} . Uma das alternativas é apresentada a seguir:

$$T_{k+1} = \frac{T_k}{\left[1 + \frac{\ln(1+\delta)T_k}{3\sigma(T_k)}\right]}$$

Onde:

- $\sigma(T_k)$: representa o desvio padrão da função de custo das configurações geradas na temperatura (T_k) e $\delta \in [0.001, 0.20]$

4.5.5. Definição da vizinhança

A vizinhança apresenta um grau de importância muito significativo, dado que, sua definição depende da geração da nova configuração a partir da configuração atual. Uma vizinhança está definida por aquelas configurações que podem ser geradas através de pequenas modificações na configuração atual, isto é, por exemplo, retirar um elemento ativo, adicionar um elemento, ou trocar um elemento que esteja presente na configuração atual por um que não esteja presente.

Para escolher uma nova configuração na estrutura da vizinhança é realizado um processo de forma aleatória, embora, é possível incrementar a probabilidade de seleção, para aquelas configurações que possuem um índice de sensibilidade que faz com que sejam

mais interessantes, de forma que melhore o valor da função objetivo e diminua os tempos computacionais.

4.5.6. Critério de parada

O critério de parada para o algoritmo SA encontra-se relacionado com o nível de temperatura com a qual o sistema físico se encontra frio ($T_f = 0$). Mas, a probabilidade de que uma configuração de pior qualidade seja aceita, antes de chegar a este valor, é quase zero. Portanto, o valor recomendado para o parâmetro (T_f) tem que ser maior do que zero, mas deve estar perto deste valor.

Normalmente os critérios de parada são diferentes para todos os tipos de problemas, mas estes critérios encontram-se baseados no término do processo e se a melhora esperada no valor da função objetivo é pequena. Na literatura pode-se encontrar diferentes procedimentos para estabelecer o critério de parada, os mais comuns são os seguintes:

- (1) Fixar um número determinado de valores de T_k para os quais será realizado o procedimento iterativo.
- (2) Parar a busca de um nível de temperatura T_k quando não seja possível encontrar uma solução que supere a melhor solução encontrada até esse momento.
- (3) Parar a busca quando no último nível de temperatura não seja possível encontrar um número mínimo de aceitações.
- (4) Depois de realizar todas as alternativas para um determinado nível de temperatura T_k , são comparadas as diferenças entre os valores máximos e mínimos das funções objetivo, entre as transações aceitadas neste nível de temperatura. Se as diferenças são muito próximas uma da outra, não é necessário o uso do algoritmo SA, portanto, $T_{k+1} = 0$. O processo de otimização pode terminar com uma busca local usando um algoritmo de melhoria iterativa simples.

O algoritmo SA, exposto até o momento, foi utilizada na proposta feita para resolver o problema de corte bidimensional de peças retangulares, tipo não-guilhotinado com e sem rotação de peças que será apresentado no capítulo seguinte.

5 MÉTODO DE SOLUÇÃO PARA O PROBLEMA DE CORTE BIDIMENSIONAL DE PEÇAS RETANGULARES TIPO NÃO-GUILHOTINADO

5.1 Introdução

O emprego de metaheurísticas para resolver problemas de alta complexidade matemática, tem demonstrado a possibilidade de encontrar resultados de boa qualidade em tempos computacionais aceitáveis. Por esse motivo, é comum encontrar diversas aplicações das diferentes áreas do conhecimento modeladas como um problema de otimização combinatória.

Para obter bons resultados na solução de um problema através desses algoritmos de otimização, é necessário conhecer em profundidade o problema. Para isto, é indispensável definir seus objetivos, assim como a representação das soluções mediante um tipo de codificação. Posteriormente, de acordo com as características do problema, escolhe-se o algoritmo de otimização a ser usado, tendo em conta que cada algoritmo possui mecanismos diferentes para explorar o espaço de soluções, sendo necessário calibrar seus parâmetros.

Nesse contexto, a metodologia proposta para resolver o problema de corte bidimensional de peças retangulares tipo não-guilhotinado consiste em usar o algoritmo heurístico Bottom-Left, juntamente com a metaheurística Simulated Annealing, para encontrar a melhor distribuição das peças dentro da placa. Neste capítulo, são apresentadas em detalhe as principais características dos dois procedimentos propostos para resolver o problema de corte não-guilhotinado com e sem rotação de peças.

5.2 Problema de corte bidimensional de peças retangulares sem rotação de peças

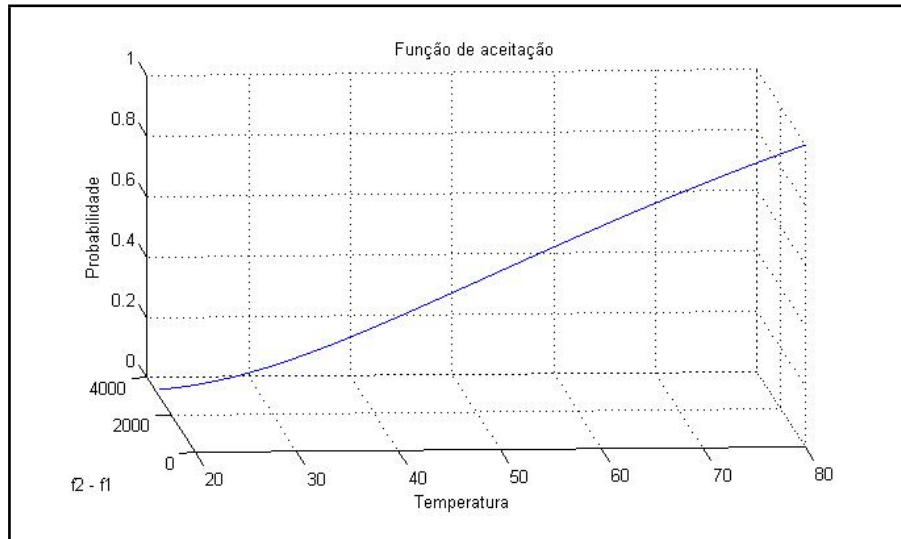
Os principais aspectos a serem tidos em conta na solução do problema de corte bidimensional de peças retangulares são listados a seguir.

5.2.1. Designação de valores dos parâmetros do SA

No caso do algoritmo SA, faz-se necessário definir os valores das temperaturas inicial e final T_0 e T_{min} , o comprimento da cadeia N_k , entre outros. Em termos gerais, a escolha desses parâmetros depende em grande proporção das características e do conhecimento do problema. No entanto, um critério que ajuda consideravelmente na atribuição de alguns destes parâmetros é denominado probabilidade de aceitação, devido que, como

é de conhecimento, para valores altos de temperatura, a probabilidade de aceitar uma solução com um valor da função objetivo de péssima qualidade deve ser alta enquanto que para valores menores de temperatura a probabilidade de aceitação das soluções de péssima qualidade vão diminuindo, como pode ser visto na Figura 25.

Figura 25 – Função de aceitação



Fonte: Elaboração do autor

5.2.2. Base de Dados

Faz-se necessário ter uma base de dados que contenha as dimensões da placa e das peças, (comprimento L e largura W), assim como a quantidade Q de peças de cada tipo conforme é apresentado na Tabela 1.

5.2.3. Codificação e Decodificação de uma proposta de solução

A codificação e a decodificação de uma proposta de solução fazem parte das estratégias fundamentais para que um problema seja resolvido da melhor forma quando são utilizadas técnicas de otimização heurísticas e metaheurísticas. Segundo Gallego et al. (2008), a codificação e a decodificação de um problema são importantes porque além de representar e definir o espaço de soluções, também é possível avaliar e quantificar a qualidade das diferentes configurações encontradas no processo de otimização.

Em termos gerais, dada uma solução do problema é necessário utilizar dentro do algoritmo um mecanismo que permita decodificar esta configuração, com o intuito de determinar se a proposta de solução é fatível ou infatível e calcular o valor da sua função objetivo.

Tabela 1 – Base de dados inicial

Placa: L=20, W=15				
Peças	l_i	w_i	Q_i	Área
T_1	7	5	1	35
T_2	7	10	1	70
T_3	4	12	1	48
T_4	3	8	1	24
T_5	5	12	1	60
T_6	9	8	1	72
T_7	11	14	1	154
T_8	8	9	1	72
T_9	15	6	1	90
T_{10}	11	7	1	77

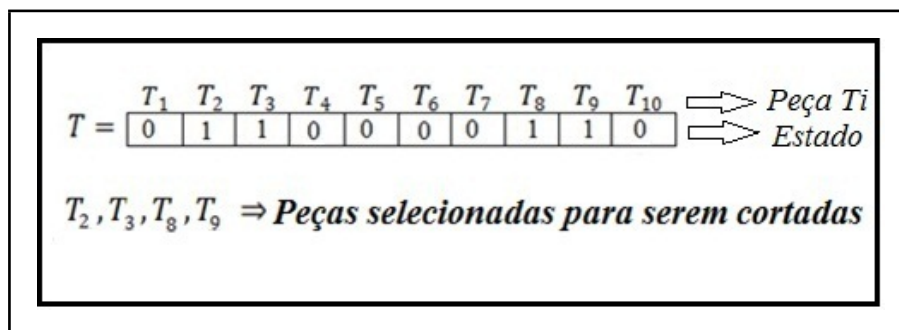
Fonte: (PARREÑO, 2004)

A seguir são mencionados vários aspectos relacionados com o tipo de codificação do problema assim como o mecanismo adotado para decodificar as alternativas de solução do problema.

Codificação do problema

Para o caso do problema de corte bidimensional de peças tipo não-guilhotinado sem rotação, propõe-se uma codificação de tipo binária para representar uma possível solução do problema, conforme apresentado na Figura 26.

Figura 26 – Representação da Codificação binária



Fonte: Elaboração do autor

Portanto:

$$T_i = \begin{cases} 0, & \text{se a } i\text{-ésima peça não é alocada dentro da placa} \\ 1, & \text{se a } i\text{-ésima peça é alocada dentro da placa} \end{cases}$$

Uma vez definido o tipo de codificação, se dá início ao processo de alocação de peças dentro da placa, como é descrito a seguir.

Alocação das peças dentro da placa: Processo de Decodificação

Para realizar a alocação das peças dentro da placa, utiliza-se uma variante do algoritmo heurístico BL, onde a ordem de alocação das peças dentro da placa depende dos valores das áreas (ordem decrescente).

Este tipo de estratégia adotada para resolver o problema de corte bidimensional sem rotação permite alocar de uma única forma as peças dentro da placa, pois, de acordo com a Figura 26, não é igual alocar as peças usando o BL na ordem T_2 , T_3 , T_8 e T_9 do que na ordem T_3 , T_8 , T_9 e T_2 , e assim por diante. Isto representa uma grande vantagem, pois diminui o número de combinações possíveis no processo de alocação das peças.

Os principais passos do mecanismo de alocação de peças são descritos a seguir:

- De acordo com a Tabela 1 e dado o vetor solução da Figura 26, ordenam-se as peças de forma decrescente segundo a área.

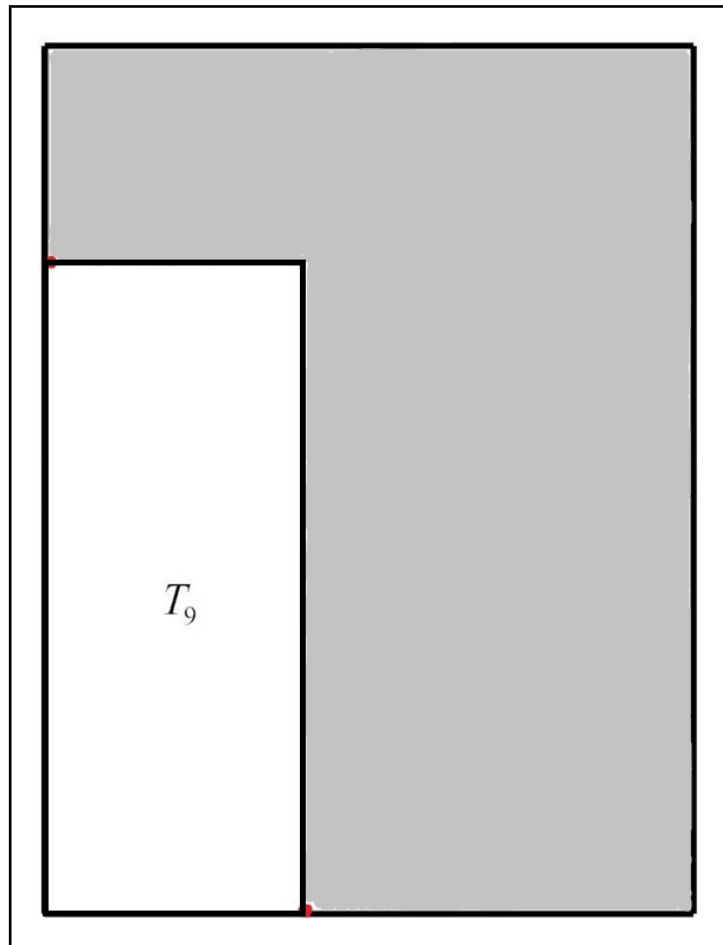
$$T_9 \quad - \quad T_8 \quad - \quad T_2 \quad - \quad T_3$$

- A primeira peça a ser alocada é aquela com maior área, a qual é fixada na parte inferior esquerda da placa, isto é, na coordenada $(x_i^1, y_i^1) = (0, 0)$. A coordenada final da peça é $(x_f^1, y_f^1) = (w^1, l^1)$, como pode ser visto na Figura 27.

Adicionalmente, podem ser identificados em cor vermelho os dois pontos candidatos onde a seguinte peça pode ser alocada.

- A peça seguinte é escolhida como a peça de maior área entre as quais ainda não pertencem à configuração atual. Segundo a estratégia do BL, a nova peça é alocada

Figura 27 – Alocação da primeira peça



Fonte: Elaboração do autor

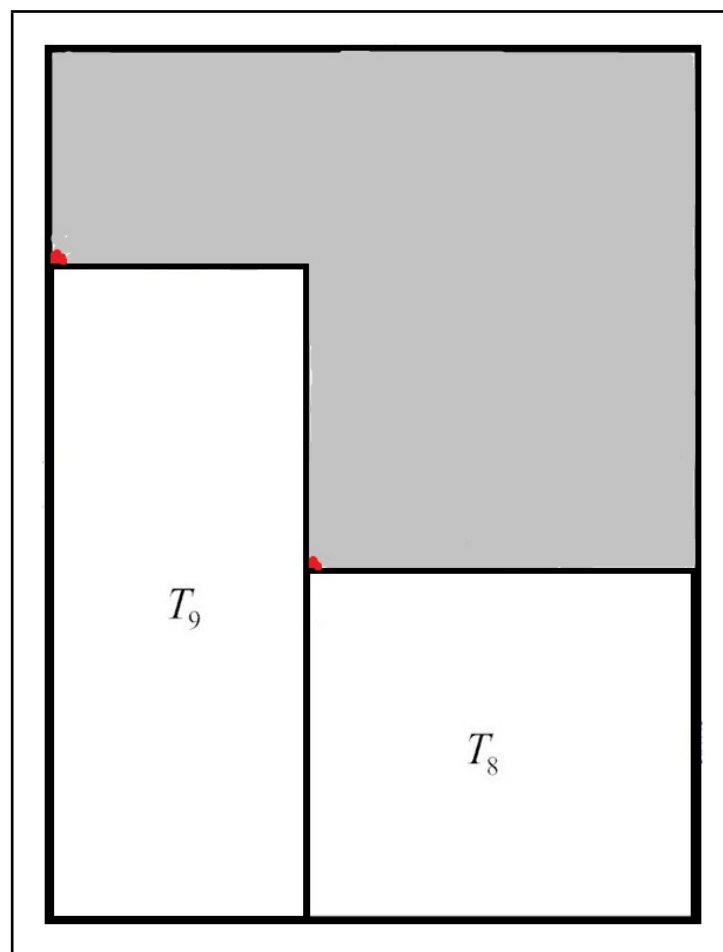
de acordo com o seguinte critério:

$$(x_i^2, y_i^2) = (0, w^1), \quad \text{se } w^1 + w^2 \leq W.$$

$$(x_i^2, y_i^2) = (l^1, 0), \quad \text{caso contrário.}$$

Na Figura 28 é mostrado a alocação da segunda peça de acordo com o critério anterior.

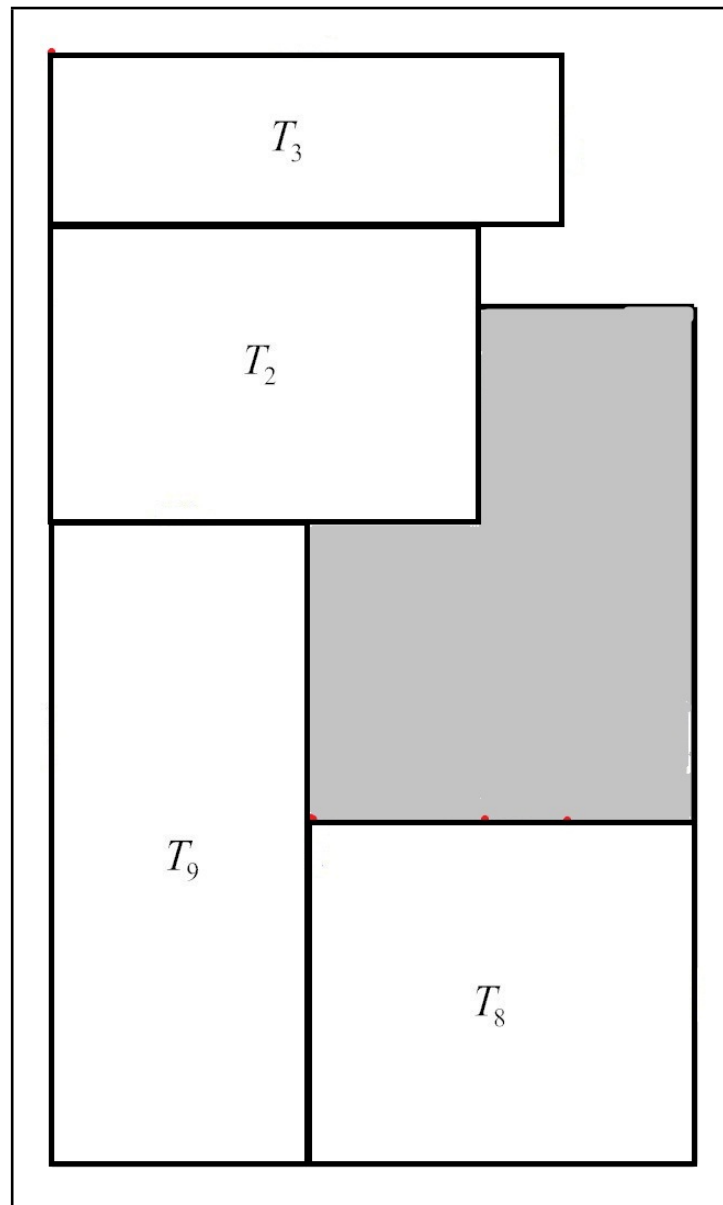
Figura 28 – Alocação da segunda peça



Fonte: Elaboração do autor

- O procedimento segue até alocar a última peça, de acordo com a Figura 29, onde pode-se verificar que a proposta de solução é infactível.

Figura 29 – Alocação final



Fonte: Elaboração do autor

5.2.4. Cálculo da função objetivo

Para avaliar a qualidade de uma configuração dada e com a finalidade de estabelecer uma comparação entre as diferentes soluções encontradas, é realizado o cálculo da função objetivo. Nesta proposta, visa-se encontrar a área efetiva (A^{efec}) usada da placa, isto é, a área da placa que está sendo ocupada pelas peças presentes no vetor solução.

De acordo com a Figura 29, a A^{efec} está dada pelo somatório das áreas das peças T_9 ,

T_8 , parte da peça T_2 e parte de T_3 .

Considerando que cada peça alocada tem uma coordenada inicial (x_i, y_i) e uma coordenada final (x_f, y_f) , a expressão para calcular a função objetivo pode ser representada como:

$$\max \quad f.o. = A^{efec} - \alpha * infac \quad (7)$$

Sendo, $infac$ a infactibilidade da alternativa de solução e α é a percentagem usada para penalizar a infactibilidade. O valor da infactibilidade é calculada da seguinte forma:

$$infac = \begin{cases} 0, & \text{se } A^{efec} \leq \sum_{\forall k \in \Omega S} (x_f - x_i) * (y_f - y_i). \\ \sum_{\forall k \in \Omega S} (x_f - x_i) * (y_f - y_i) - A^{efec}, & \text{caso contrário.} \end{cases}$$

A equação anterior significa que uma solução é considerada factível, se todas as peças não ultrapassam as dimensões da placa. Caso contrário a configuração deve ser penalizada, como pode-se verificar na Figura 29.

5.2.5. Criação da configuração inicial

De modo geral, a configuração inicial pode ser obtida de várias maneiras: aleatoriamente, usando algum tipo de lógica devido ao conhecimento do problema ou através de um algoritmo constructivo usando fatores de sensibilidade. Em alguns casos dado o nível de complexidade do problema, resulta vantajoso estabelecer um mecanismo para criar uma configuração de início. Isto se deve ao fator de que a configuração inicial pode ser considerada um fator de importância relevante na obtenção de bons resultados na resolução de problemas, principalmente quando o espaço de soluções é muito amplo e são usados os algoritmos de otimização combinatória.

Na configuração inicial adotada para solucionar o problema de corte bidimensional de peças retangulares tipo não-guilhotinado, realiza-se uma seleção aleatória, onde as peças com maiores áreas têm maior probabilidade de conformar a solução de início do que as peças com menor área.

5.2.6. Definição da vizinhança

Na literatura existem várias estratégias para definir a vizinhança de uma solução, sendo que cada pesquisador pode definir sua própria estratégia segundo o problema abordado. Para o caso especial do problema de corte bidimensional de peças retangulares tipo não-guilhotinado sem rotação de peças, adota-se uma estratégia que permite modificar o vetor de solução atual, por outro vetor quase idêntico, com uma mudança em uma de suas posições. Isto é, se uma posição do vetor tem uma peça selecionada para ser cortada "1", pode ser trocada por "0", que representa a peça não selecionada e vice-versa. Para esta aplicação o esquema de vizinhança é o seguinte:

- Adicionar uma peça no vetor solução.
- Retirar uma peça selecionada do vetor solução.
- Trocar duas peças ao mesmo tempo no vetor solução. Isto é, inserir uma peça e ao mesmo tempo retirar uma peça que já tinha sido selecionada.

A seleção da posição do vetor que vai ser modificado é escolhida através de um processo aleatório. As possíveis modificações que são feitas no vetor solução podem ser vistas nas Figuras 30 e 31.

Figura 30 – Vetor solução atual

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
0	1	1	0	0	0	0	1	1	0

$T_2, T_3, T_8, T_9 \Rightarrow$ **Peças selecionadas para serem cortadas**

Fonte: Elaboração do autor

5.2.7. Atualização da solução incumbente

Considerando que o algoritmo de SA possui uma estratégia para sair de ótimos locais que consiste em substituir a configuração atual por outra configuração com valor de qualidade inferior na função objetivo, faz-se necessário atualizar constantemente a solução incumbente do problema. Para isto, são usados os seguintes critérios para atualizar a solução incumbente:

Figura 31 – Vetor solução modificado

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
0	1	1	0	0	0	0	1	1	1

$T_2, T_3, T_8, T_9, T_{10} \Rightarrow$ **Peças selecionadas para serem cortadas**

Fonte: Elaboração do autor

- Se a solução incumbente e a solução atual são infactíveis, mas a solução atual é menos infactível e tem melhor valor da função objetivo.
- Se a solução incumbente é infactível e a solução atual é factível.
- Se a solução incumbente e a solução atual são factíveis, mas a solução atual tem melhor valor da função objetivo.

5.2.8. Algoritmo de corte bidimensional de peças retangulares tipo não-guilhotinado sem rotação de peças

- i Entrada de dados.
- ii Definição dos valores dos parâmetros: $T_0, T_{min}, \rho, N_0, \beta, k, iter = 0, .$
- iii Criação da configuração inicial, T .
- iv Cálculo da função objetivo, f_{o1} . (Decodificação)
- v Se $T_k > T_{min}$, fazer $iter = iter + 1, j = 0$ e ir ao passo vi. Do contrário ir ao passo xiii.
- vi Se $j \leq N_k$, fazer $j = j + 1$ e ir ao passo vii. Em caso contrário ir ao passo xii.
- vii Criar uma configuração vizinha, T_2 .
- viii Cálculo da função objetivo da configuração vizinha, f_{o2} . (Decodificação)
- ix Se $f_{o2} > f_{o1}$ substituir a configuração atual, $T = T_2$ e $f_{o1} = f_{o2}$. Em caso contrário ir ao passo x.
- x Calcular o valor da função de aceitação, f^{acei} . Se $f^{acei} > random[0, 1]$, substituir a configuração atual, $T = T_2$ e $f_{o1} = f_{o2}$.

- xi Se $iter = 1$ armazenar a configuração atual como a solução incumbente, $incum = T$ e $fo_{incum} = fo_1$. Caso contrário, atualizar a solução incumbente e ir ao passo vi.
- xii Atualizar os valores da temperatura $T_k = T_k * \beta$ e do comprimento da cadeia $N_k = N_k * \rho$. Ir ao passo v.
- xiii Finalizar.
- xiv Saída de resultados, $incum$ e fo_{incum} .

5.3 Problema de corte bidimensional de peças retangulares com rotação de peças

Diferente do método proposto sem rotação de peças onde para cada vetor solução somente existe uma única forma de alocar as peças dentro da placa, e portanto, só existe um único valor da $f.o.$, no caso do método que considera a rotação das peças esta condição deixa de ser válida porque cada peça pode ser posicionada de duas formas diferentes, resultando um total de 2^k possíveis formas de alocar as peças, sendo k o número de peças a serem distribuídas dentro da placa.

Isto significa que para a solução mostrada na Figura 31 ($k = 5$), existem 32 formas diferentes de distribuir as peças na placa e igual valor de funções objetivo, como pode ser visto na Tabela 2. Por convenção adota-se que 0 corresponde ao tipo de peça que não tem sido rotacionada e vice-versa

Tabela 2 – Possíveis combinações para o problema com rotação de peças

T_9	T_{10}	T_8	T_2	T_3	$f.o$
0	0	0	0	0	$f.o_1$
0	0	0	0	1	$f.o_2$
0	0	0	1	0	$f.o_3$
0	0	0	1	1	$f.o_4$
0	0	1	0	0	$f.o_5$
0	0	1	0	1	$f.o_6$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
1	1	1	1	0	$f.o_{31}$
1	1	1	1	1	$f.o_{32}$

Fonte: Elaboração do autor

Para contornar esta dificuldade e dado que resulta inviável avaliar as 2^k configurações para cada vetor solução dentro do processo de otimização e com o intuito de não criar

dois algoritmos diferentes para resolver os problemas de corte bidimensional com e sem rotação de peças, incorpora-se no processo de decodificação um procedimento, antes de ordenar as peças de acordo com os valores das áreas, que consiste basicamente em escolher aleatoriamente as peças que irão ser rotacionadas, e posteriormente realizar o processo de alocação das peças.

Os principais aspectos da estratégia de decodificação usada no problema de corte com rotação de peças, é explicada através do seguinte exemplo:

- Considerando como solução atual a configuração mostrada na Figura 32.

Figura 32 – Vetor solução modificado

T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
0	1	1	0	0	0	0	1	1	0

$T_2, T_3, T_8, T_9 \Rightarrow$ *Peças selecionadas para serem cortadas*

Fonte: Elaboração do autor

- Cria-se um vetor binário de forma aleatória para escolher as peças do vetor solução que irão ser rotacionadas. Como pode ser visto na Figura 33, o tamanho deste vetor é igual ao número de peças presentes na solução atual.

Figura 33 – Vetor de rotação de peças

T_9	T_8	T_2	T_3	⇒	<i>Vetor de peças selecionadas para serem cortadas.</i>
1	1	1	1		
T_9	T_8	T_2	T_3	⇒	<i>Vetor de peças selecionadas para serem rotacionadas</i>
0	0	1	0		

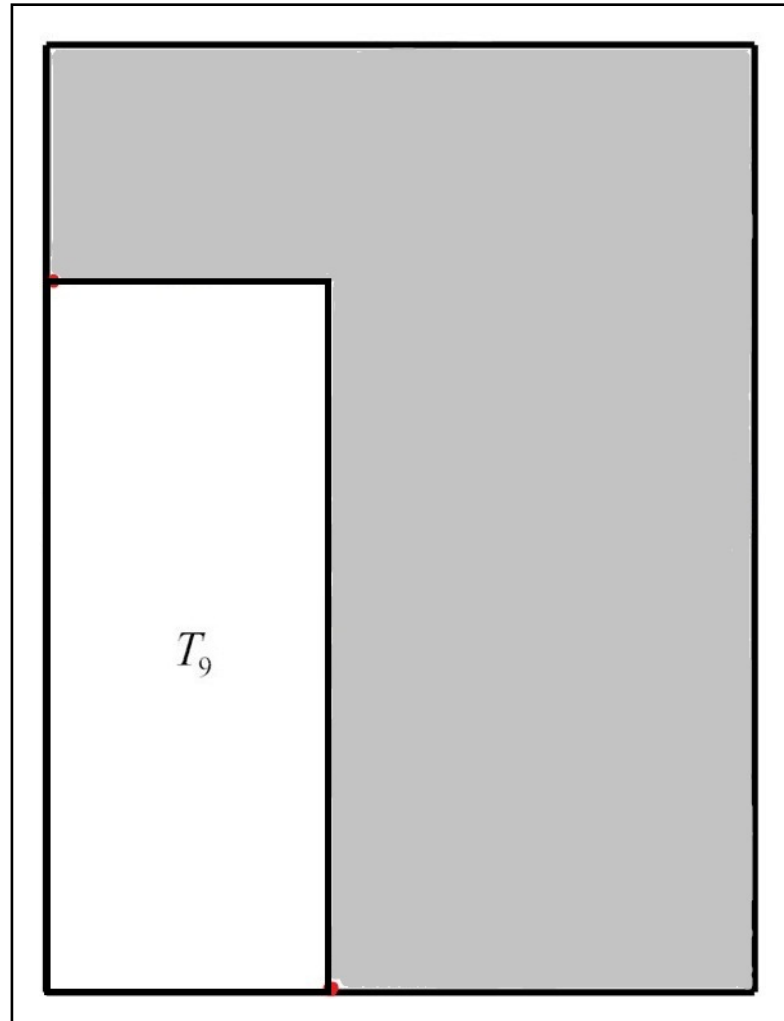
Fonte: Elaboração do autor

- Ordenam-se as peças de forma decrescente segundo o valor da área.

$$T_9 - T_8 - T_2 - T_3$$

- Aloca-se a primeira peça, a qual é fixada na parte inferior esquerda da placa, isto é, na coordenada $(x_i^1, y_i^1) = (0, 0)$. A coordenada final da peça é $(x_f^1, y_f^1) = (w^1, l^1)$, como pode ser visto na Figura 34.

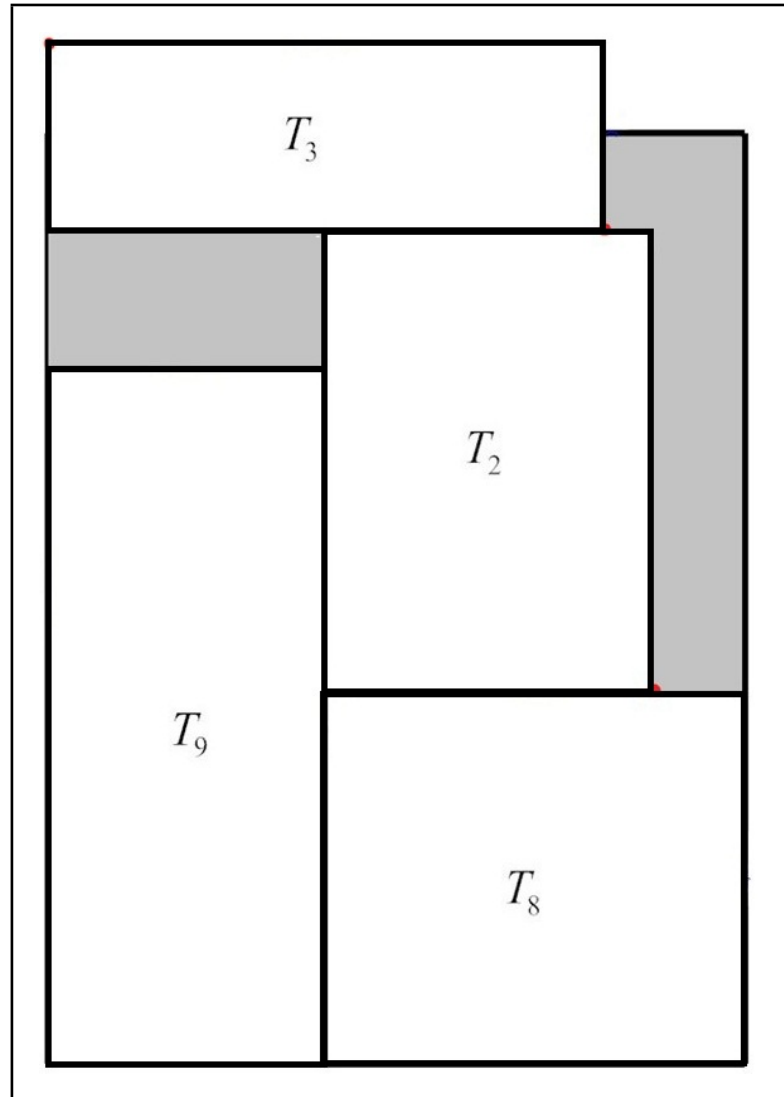
Figura 34 – Alocação da primeira peça



Fonte: Elaboração do autor

- O procedimento continua até alocar a última peça, conforme é apresentado na Figura 35.

Figura 35 – Alocação da última peça



Fonte: Elaboração do autor

Em resumo, o algoritmo proposto para resolver o problema de corte bidimensional de peças retangulares com rotação de peças é o mesmo usado para o problema sem rotação de peças, com a adição de uma rotina dentro do processo de decodificação encarregada de selecionar as peças a rotar.

6 TESTES E RESULTADOS

6.1 Introdução

Para validar os métodos de solução propostos do problema de corte não-guilhotinado com e sem rotação de peças são usados três sistemas testes da literatura especializada que de acordo com a sua quantidade de peças podem ser classificados como problemas de nível de complexidade baixo, médio e alto, respectivamente.

Cada caso de estudo contém a base de dados com a informação correspondente as dimensões da placa e dos diferentes tipos de peças. Além disso, mostram-se os valores dos principais parâmetros do algoritmo Simulated Annealing usados durante as simulações. Esses valores foram escolhidos tendo em conta os valores da probabilidade de aceitação ao início e ao final do processo iterativo, assim como o número de iterações totais dos algoritmos propostos para cada teste.

Adicionalmente, são apresentadas para cada caso de estudo as soluções gráficas com a distribuição das peças dentro da placa obtidas para os problemas com rotação e sem rotação de peças. Realiza-se também uma análise dos resultados alcançados, onde através de um gráfico que mostra-se a porcentagem de ocupação da placa para cada tipo de problema, é comparado o desempenho dos algoritmos propostos para um total de cinquenta simulações. Finalmente, são apresentados os dados gerais obtidos nas simulações onde aparecem os valores máximo, mínimo, médio da função objetivo e outros valores numéricos e estatísticos que medem o sucesso dos algoritmos propostos.

6.2 Caso teste 1

O primeiro caso teste é composto por um total de 21 peças distribuídas em 10 tipos diferentes. A informação com as dimensões da placa e das peças pode ser vista na Tabela 3.

Tabela 3 – Base de dados do caso teste 1

<i>Placa: L=10, W=10</i>				
<i>Peças</i>	l_i	w_i	Q_i	<i>Área total</i>
T_1	2	3	2	12
T_2	2	7	3	42
T_3	2	4	2	16
T_4	2	6	3	36
T_5	1	9	2	18
T_6	4	8	1	32
T_7	1	4	2	8
T_8	10	1	1	10
T_9	7	3	3	63
T_{10}	5	4	2	40
<i>Total</i>			21	277

Fonte: (PARREÑO, 2004)

Na Tabela 4 aparecem os valores dos parâmetros do algoritmo SA escolhidos para fazer as simulações do caso teste 1.

Tabela 4 – Parâmetros do algoritmo SA usados no teste 1

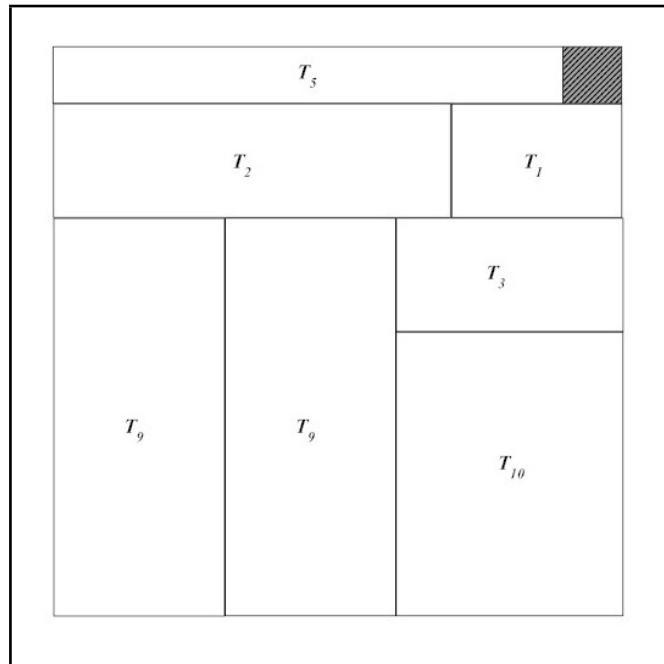
T_0	80
N_0	20
ρ	0,85
β	1,12
T_{min}	5
k	1
α	1

Fonte: (PARREÑO, 2004)

6.2.1. Solução do problema de corte não guilhotinado sem rotação de peças

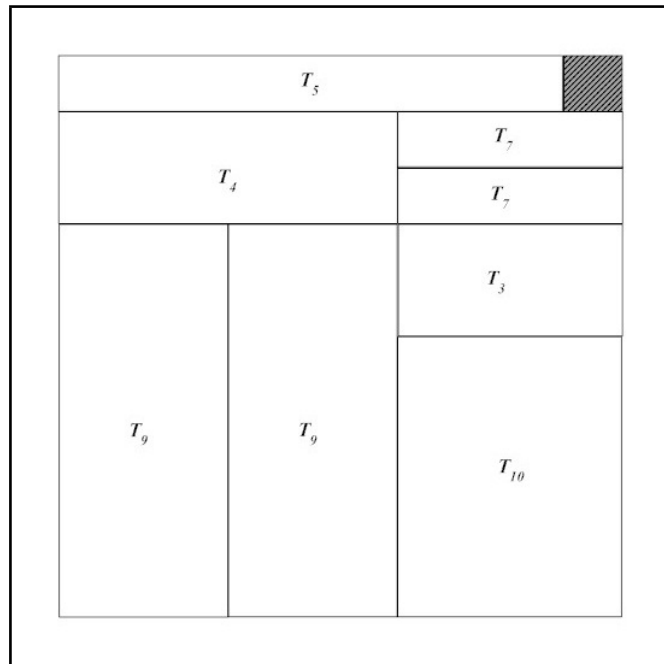
Para este caso de estudo foram encontradas três soluções diferentes com uma porcentagem de ocupação da área da placa igual a 99%. A gráfica com a distribuição das peças para cada solução encontrada é mostrada a seguir.

Figura 36 – Distribuição das peças No. 1



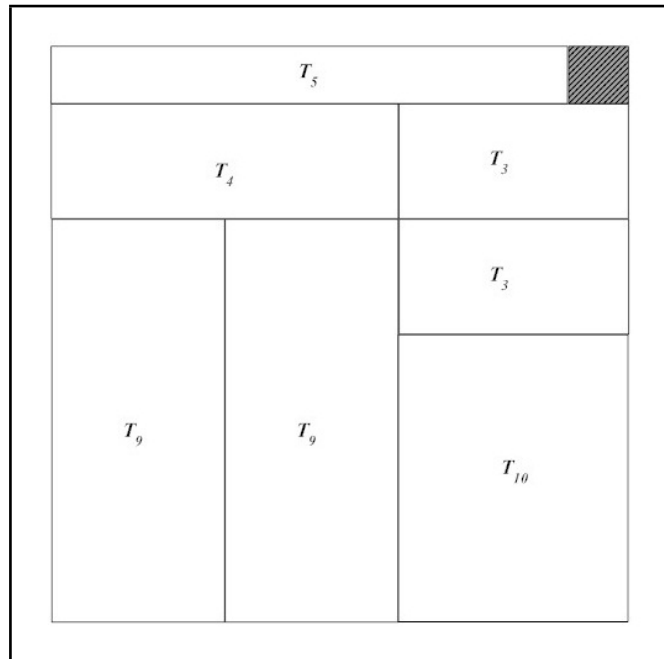
Fonte: Elaboração do autor

Figura 37 – Distribuição das peças No. 2



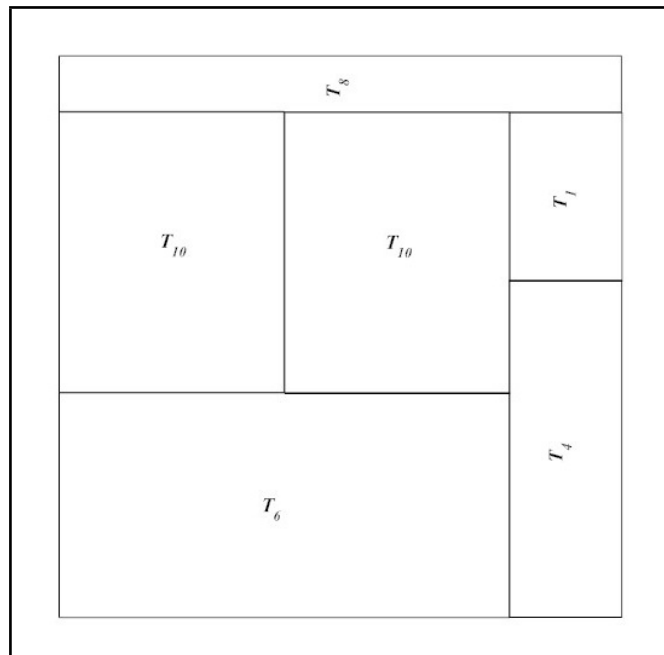
Fonte: Elaboração do autor

Figura 38 – Distribuição das peças No. 3



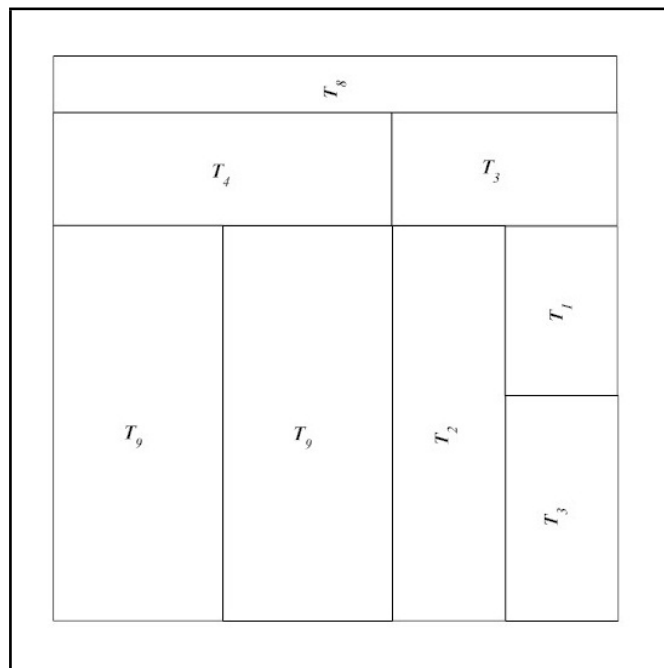
Fonte: Elaboração do autor

Figura 40 – Distribuição das peças No. 2



Fonte: Elaboração do autor

Figura 41 – Distribuição das peças No. 3



Fonte: Elaboração do autor

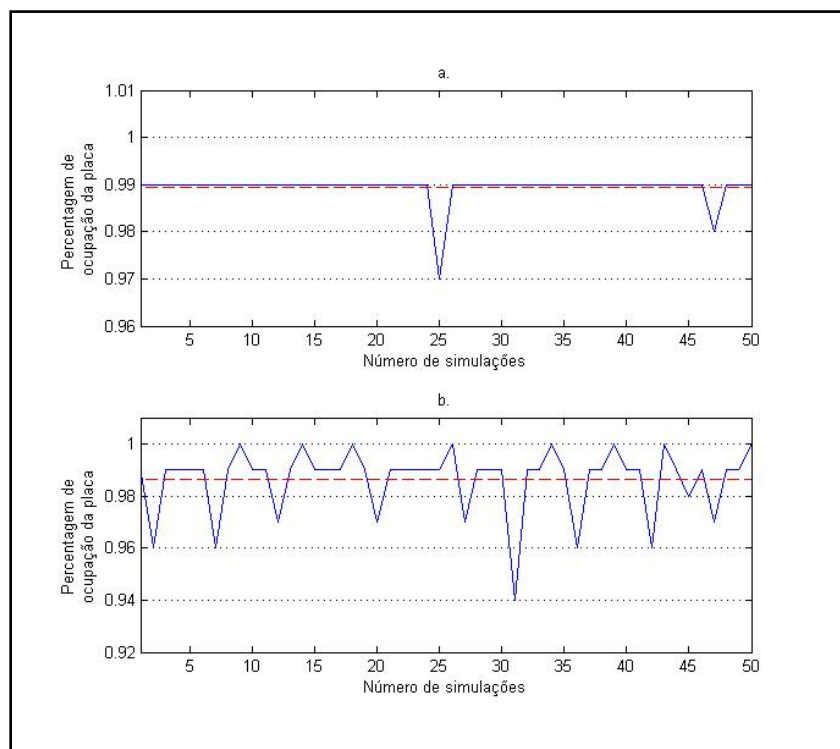
6.2.3. Comparação do desempenho dos algoritmos de corte propostos

Os resultados correspondentes ao caso teste No. 1, classificado como um problema com um nível de complexidade baixo, mostram que para este caso teste não existe uma única solução, pois nos problemas com rotação e sem rotação de peças, os algoritmos propostos conseguem encontrar diferentes configurações como solução do problema.

Neste caso teste foram testados duas estratégias para criar a configuração inicial, aleatória equiprovável e aleatória não-equiprovável, mas essa escolha não teve um impacto considerável nos resultados finais.

Adicionalmente, na Figura 42 é apresentado o desempenho dos dois algoritmos propostos, onde é possível apreciar que as soluções encontradas estão bem próximas de ocupar o 100% da área da placa.

Figura 42 – Resultados obtidos pelos algoritmos propostos. a. Sem rotação de peças. b. Com rotação de peças



Fonte: Elaboração do autor

É importante salientar que no problema com rotação de peças as soluções encontradas são mais dispersas quando comparadas com as soluções sem rotação de peças. Isto é devido, principalmente, ao fato da maior quantidade de configurações possíveis presentes no espaço de solução do problema com rotação de peças.

Na Tabela 5 aparecem registrados os valores de interesse da função objetivo encontrados nas simulações, assim como o número total de iterações e o tempo médio de duração para cada simulação.

Tabela 5 – Dados gerais da simulação do caso teste 1

<i>f.o.</i>	<i>Sem rotação</i>	<i>Com rotação</i>
<i>Valor máximo (%)</i>	99,00	100,00
<i>Valor mínimo (%)</i>	97,00	94,00
<i>Valor médio (%)</i>	98,94	98,64
<i>Desvio padrão (%)</i>	00,31	1,27
<i>Número iterações</i>	1108	
<i>Tempo médio [s]</i>	3,3998	

Fonte: Elaboração do autor

Segundo a informação contida na Tabela 5, aprecia-se que em média as soluções encontradas pelos algoritmos de corte bidimensional propostos com e sem rotação de peças têm valores de quase o 99% de ocupação da placa.

6.3 Caso teste 2

O segundo teste é composto por um total de 48 peças distribuídas em 24 tipos diferentes de peças. Essa base de dados foi modificada especificamente para criar um caso de estudo com um nível de complexidade maior do que o primeiro sistema teste. A informação com as dimensões da placa e das peças pode ser vista na Tabela 6.

Tabela 6 – Base de dados do caso teste 2

<i>Placa: L=200, W=300</i>				
<i>Peças</i>	l_i	w_i	Q_i	<i>Área total</i>
T_1	38	39	3	4446
T_2	67	74	1	4958
T_3	71	67	1	4757
T_4	33	71	3	7029
T_5	31	40	2	2480
T_6	56	25	1	1400
T_7	70	57	1	3990
T_8	34	33	3	3366
T_9	31	58	1	1798
T_{10}	61	41	2	5002
T_{11}	28	51	3	4284
T_{12}	28	56	3	4704
T_{13}	72	64	1	4608
T_{14}	35	58	1	2030
T_{15}	73	37	1	2701
T_{16}	73	54	1	3942
T_{17}	54	51	1	2754
T_{18}	27	63	1	1701
T_{19}	54	28	1	1512
T_{20}	65	26	1	1690
T_{21}	25	69	3	5175
T_{22}	48	27	1	1296
T_{23}	31	48	4	5952
T_{24}	33	32	8	8448
<i>Total</i>			48	94361

Fonte: (MORABITO, (2008))

Dado que este sistema teste possui características diferenciadas em relação ao primeiro caso de estudo, é necessário assignar novos valores para alguns dos parâmetros do algoritmo SA de acordo com os critérios expostos na introdução deste capítulo. Esses valores aparecem na Tabela 7.

Tabela 7 – Parâmetros do algoritmo SA usados no teste 2

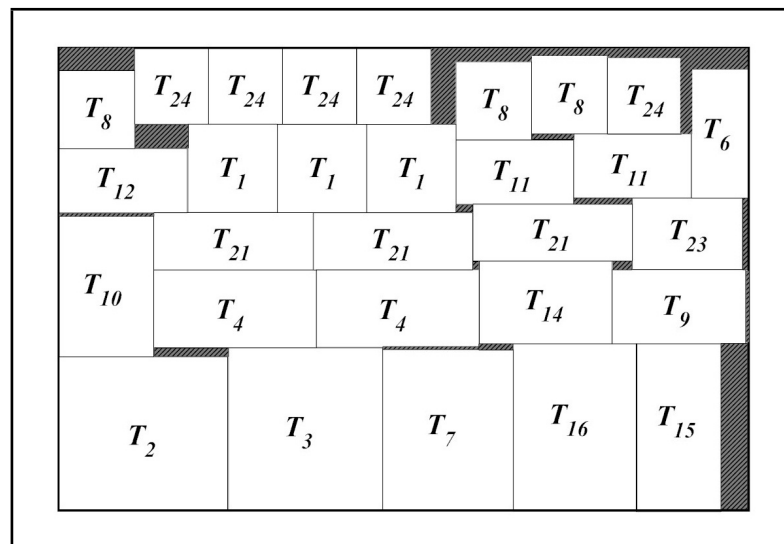
T_0	80
N_0	90
ρ	0,90
β	1,12
T_{min}	20
k	40
α	1

Fonte: Elaboração do autor

6.3.1. Solução do problema de corte não guilhotinado sem rotação de peças

A melhor solução encontrada ao longo do processo iterativo possui uma porcentagem de ocupação da área da placa igual a 94,90%. A solução gráfica para esta configuração é mostrada na Figura 43.

Figura 43 – Distribuição das peças do teste 2

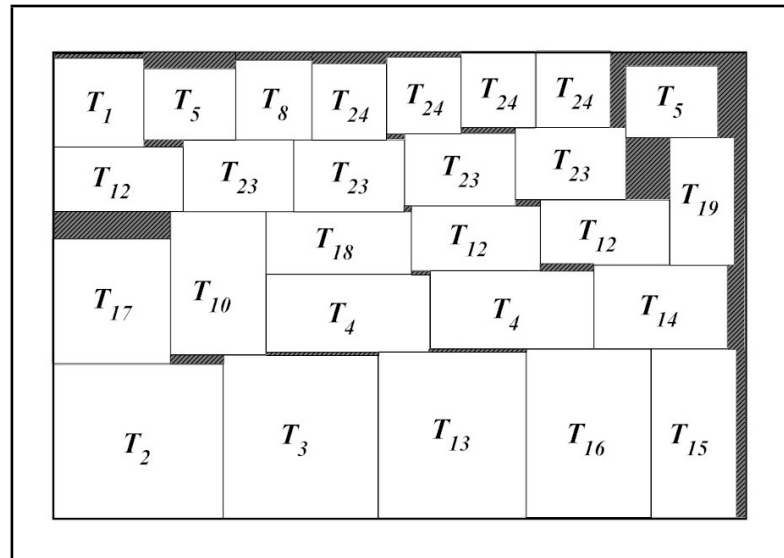


Fonte: Elaboração do autor

6.3.2. Solução do problema de corte não guilhotinado com rotação de peças

A melhor solução encontrada para este caso em particular possui uma porcentagem de ocupação da área da placa igual a 93,52%. A solução gráfica para esta configuração é mostrada na Figura 44.

Figura 44 – Distribuição das peças do teste 2



Fonte: Elaboração do autor

6.3.3. Comparação do desempenho dos algoritmos de corte propostos

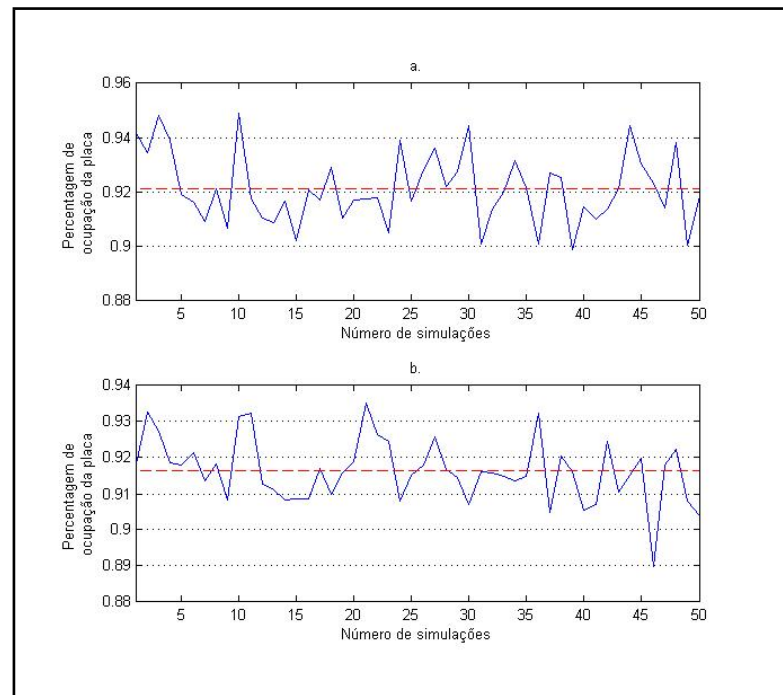
Os resultados do segundo caso de estudo, que possui um nível de complexidade médio, apresentam valores de ocupação da área da placa aceitáveis, pois em média foram alcançados valores próximos a 92%.

O grau de dispersão das soluções encontradas durante as simulações indica que existe uma maior quantidade de soluções dentro do espaço de busca.

Nesta simulação o ponto de início torna-se um fator de peso a ser considerado para a obtenção de bons resultados. Portanto, os melhores resultados foram obtidos para configurações iniciais não-equiprováveis e com uma porcentagem de ocupação da placa entre 50% e 60%.

O desempenho dos dois algoritmos propostos para este caso de estudo podem ser vistos na Figura 45.

Figura 45 – Resultados obtidos pelos algoritmos propostos.
a. Sem rotação de peças. b. Com rotação de peças



Fonte: Elaboração do autor

Na Tabela 8 aparecem registrados os valores de interesse da função objetivo encontrados nas simulações, assim como o número total de iterações e o tempo médio de duração para cada simulação.

Tabela 8 – Dados gerais da simulação do caso teste 2

<i>f.o.</i>	<i>Sem rotação</i>	<i>Com rotação</i>
<i>Valor máximo (%)</i>	94,90	93,52
<i>Valor mínimo (%)</i>	89,87	88,98
<i>Valor médio (%)</i>	92,12	91,62
<i>Desvio padrão (%)</i>	1,30	0,87
<i>Número iterações</i>	2907	
<i>Tempo médio [s]</i>	62,3182	

Fonte: Elaboração do autor

6.4 Caso teste 3

O terceiro teste é composto por um total de 106 peças distribuídas em 20 tipos diferentes de peças. Este problema é classificado como um problema de alto nível de complexidade dado o número elevado de possíveis soluções presentes no espaço de busca. A informação com as dimensões da placa e das peças pode ser vista na Tabela 9.

Tabela 9 – Base de dados do caso teste 3

<i>Placa: L=1001, W=2002</i>				
<i>Peças</i>	l_i	w_i	Q_i	<i>Área total</i>
T_1	67	53	1	3551
T_2	194	135	1	26190
T_3	407	119	7	339031
T_4	382	160	5	305600
T_5	410	194	8	636320
T_6	86	205	6	105780
T_7	319	297	3	284229
T_8	414	449	8	1487088
T_9	223	81	2	36126
T_{10}	78	195	9	136890
T_{11}	172	400	4	275200
T_{12}	216	429	1	92664
T_{13}	217	170	4	147560
T_{14}	275	386	8	849200
T_{15}	324	51	4	66096
T_{16}	77	424	9	293832
T_{17}	416	92	4	153088
T_{18}	111	240	5	133200
T_{19}	160	383	8	490240
T_{20}	203	269	9	491463
<i>Total</i>			106	6353348

Fonte: (MORABITO, (2008))

Da mesma forma que nos teste anteriores, faz-se necessário ajustar valores aos parâmetros do algoritmo SA, com a finalidade de explorar uma quantidade razoável de configurações de acordo com a natureza do problema. Esses valores aparecem na Tabela 10.

Tabela 10 – Valores dos parâmetros usados no teste 3

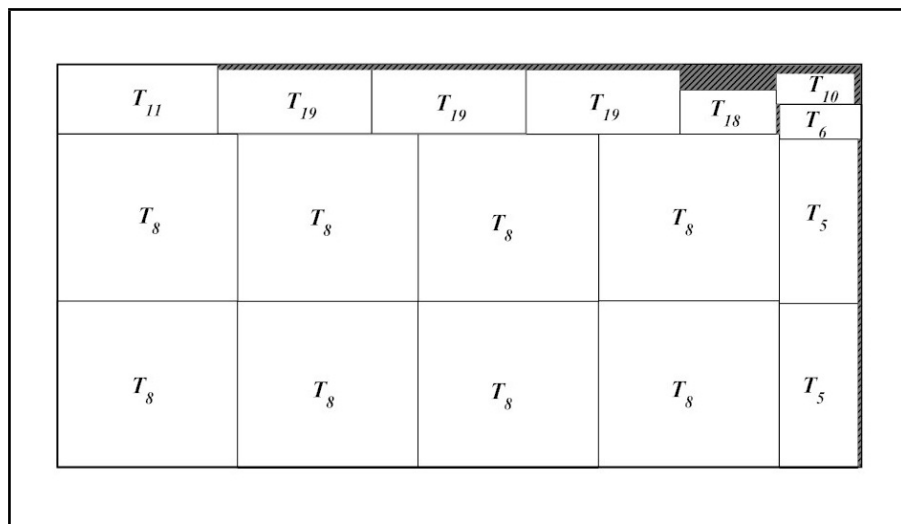
T_0	250
N_0	105
ρ	0,90
β	1,2
T_{min}	60
k	350
α	1

Fonte: Elaboração do autor

6.4.1. Solução do problema de corte não guilhotinado sem rotação de peças

A melhor solução encontrada para este caso de estudo possui uma porcentagem de ocupação da área da placa igual a 97,72%. A solução gráfica correspondente para esta configuração é mostrada na Figura 44.

Figura 46 – Distribuição das peças do teste 3

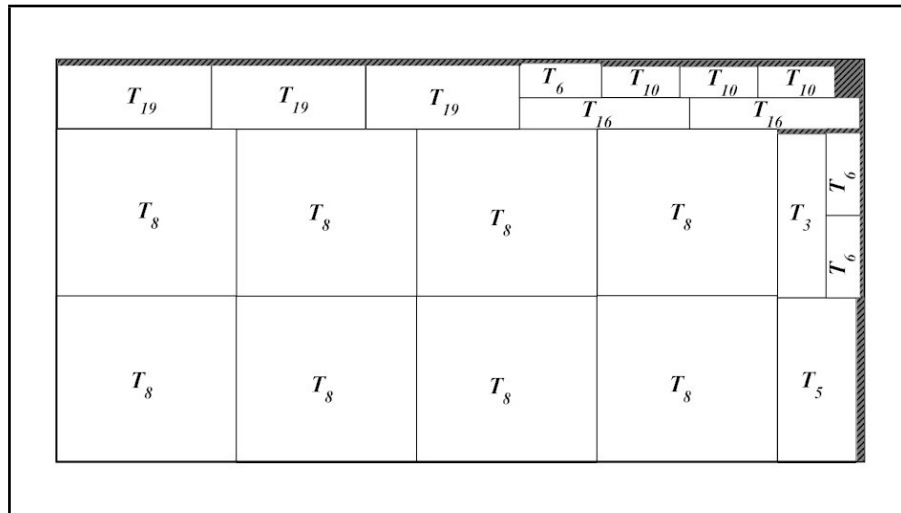


Fonte: Elaboração do autor

6.4.2. Solução do problema de corte não guilhotinado com rotação de peças

A melhor solução encontrada para este caso de estudo possui uma porcentagem de ocupação da área da placa igual a 97,94%. A solução gráfica correspondente para esta configuração é mostrada na Figura 44.

Figura 47 – Distribuição das peças do teste 3



Fonte: Elaboração do autor

6.4.3. Comparação do desempenho dos algoritmos de corte propostos

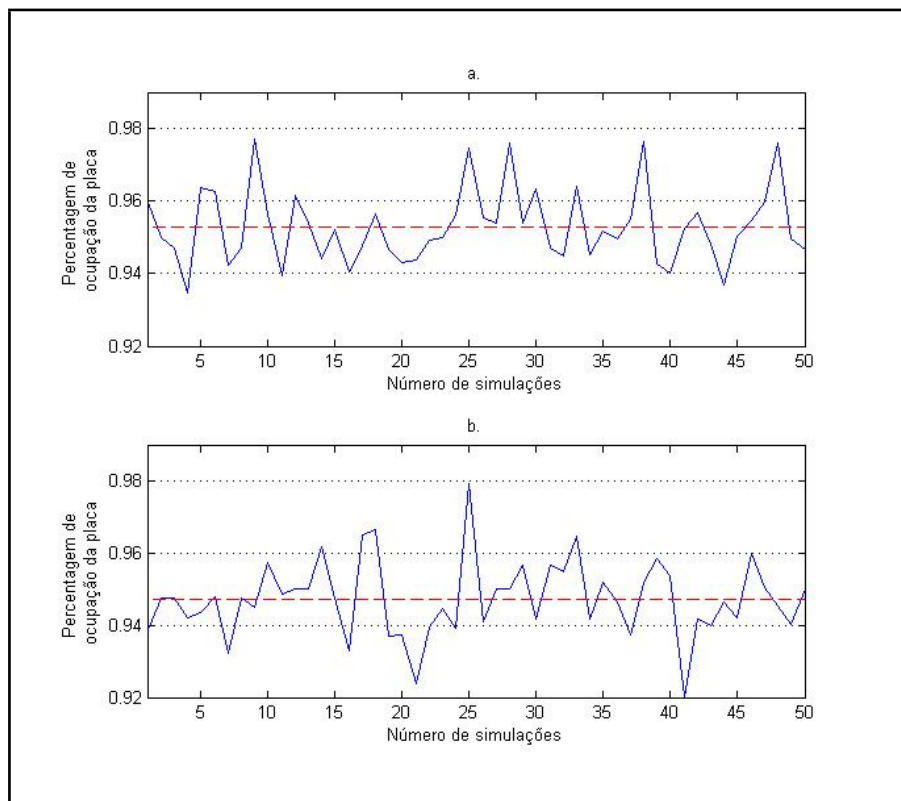
Os resultados do terceiro caso de estudo, que possui um nível de complexidade alto, apresentam valores de ocupação da área próximos ao 98%, resultados satisfatórios para este tipo de problema.

As diferentes soluções encontradas durante as simulações são muito mais dispersas que nos dois casos anteriores, posto que o número de possíveis soluções tem sido aumentado consideravelmente.

A configuração inicial nesta simulação continua sendo importante na obtenção de resultados de boa qualidade. Sendo que um bom ponto de início corresponde a uma solução que esteja entre 50% e 60% da ocupação da área da placa.

O desempenho dos dois algoritmos propostos para este caso de estudo podem ser vistos na Figura 48.

Figura 48 – Resultados obtidos pelos algoritmos propostos. a. Sem rotação de peças. b. Com rotação de peças



Fonte: Elaboração do autor

Na Tabela 11 aparecem registrados os valores de interesse da função objetivo encontrados nas simulações, assim como o número total de iterações e o tempo médio de duração para cada simulação.

Tabela 11 – Dados gerais da simulação do caso teste 3

<i>f.o.</i>	<i>Sem rotação</i>	<i>Com rotação</i>
<i>Valor máximo (%)</i>	97,72	97,94
<i>Valor mínimo (%)</i>	93,47	92,03
<i>Valor médio (%)</i>	95,31	94,73
<i>Desvio padrão (%)</i>	1,04	1,06
<i>Número iterações</i>	6211	
<i>Tempo médio [s]</i>	70,4213	

Fonte: Elaboração do autor

7 CONCLUSÕES E TRABALHOS FUTUROS

- Neste trabalho formulou-se um novo procedimento, que usa de forma conjunta o algoritmo combinatório Simulated Annealing e um procedimento heurístico de alocação de peças denominado Botton-Left, capaz de resolver eficientemente o problema de corte bidimensional de peças retangulares tipo não-guilhotinado com e sem rotação de peças.
- Com esta pesquisa foi possível ampliar o entendimento dos principais conceitos usados dentro do problema de corte de peças retangulares os quais servem como o ponto de início para a formulação de novos procedimentos de fácil implementação e que podem melhorar os resultados alcançados nesta pesquisa.
- Os resultados obtidos nesta pesquisa comprovam a capacidade de resolver eficientemente o problema de corte bidimensional de peças retangulares tipo não-guilhotinado através da metodologia proposta, pois nos casos de estudo foi possível encontrar soluções que alcançam porcentagens de ocupação da placa entre 90% e 98%.
- Durante a pesquisa foram validados três objetivos de maneira satisfatória: a proposta de um tipo de codificação binária para representar uma solução do problema, a escolha de um procedimento heurístico para alocação de peças e a adaptação do algoritmo Simulated Annealing para resolver o problema de corte bidimensional não-guilhotinado.
- Os resultados alcançados durante a pesquisa mostram a viabilidade de implementar de maneira conjunta algoritmos heurísticos de alocação de peças e o algoritmos de otimização combinatória através de uma codificação binária para avaliar eficientemente o amplo espaço de soluções próprias dos problemas de corte com altos níveis de complexidade matemática e a obtenção de resultados de boa qualidade.
- Demonstrou-se durante esta pesquisa que cada caso de estudo possui características particulares que fazem com que seja necessário adotar estratégias, como por exemplo na criação da configuração de início e na calibração dos parâmetros, entre outras,

com a finalidade de garantir a obtenção de resultados satisfatórios.

- Uma grande dificuldade que apareceu durante as simulações está relacionada com a exploração eficiente do espaço de busca dado a ampla quantidade de soluções possíveis principalmente no momento de se considerar a rotação de peças. Para contornar esta dificuldade deve-se estabelecer um mecanismo que permita agrupar aleatoriamente várias tipos de peças para formar um novo bloco. Desta maneira, estaria sendo reduzido consideravelmente a quantidade de possíveis soluções do problema.
- Outro trabalho futuro que poderia ser implementado, consiste em usar outro procedimento heurístico de alocação de peças que possa preencher desperdiços de uma maneira mais eficiente do que a estratégia adotada pela metodologia Bottom-Left.

REFERÊNCIAS

- ÁLVAREZ, D.; TORO, E.; GALLEGO, R. Empaquetamiento óptimo bidimensional con rotación de piezas usando un algoritmo híbrido de búsqueda en vecindad variable y recocido simulado. **Scientia Et Technica**, Pereira, v. 1, n. 41, p. 279-284, 2009
- ÁLVAREZ, D.; TORO, E.; GALLEGO, R. Empaquetamiento óptimo bidimensional en rollos infinitos usando un algoritmo híbrido de búsqueda en vecindad variable y recocido simulado. **Scientia Et Technica**, Pereira, v. 2, n. 42, p. 205-211, 2009.
- ÁLVAREZ-VALDES, R.; PARREÑO, F.; TAMARIT, J. A Branch and Cut Algorithm for the Pallet Loading Problem. **Computers and Operations Research**, Oxford, v. 32, n. 11, p. 3007-3029, 2005 .
- ANDRADE C. E. **Um algoritmo exato para o Problema de Empacotamento Bidimensional em Faixas**. Dissertação (Mestrado em Ciência da Computação) - Instituto de Computação, Universidade Estadual de Campinas, Campinas, 2006.
- ARENALES, M.; MORABITO, R. An AND/OR-graph approach to the solution of two-dimensional non-guillotine cutting problems. **European Journal of Operational Research**, Philadelphia, v. 84, n. 3, p. 599-617, 1995.
- BAKER, B. S., COFFMAN, E. G., RIVEST, R. L. Orthogonal packing in two dimensions. **SIAM Journal on Computing**, Auckland, v. 9, n. 4, p. 846-855, 1980.
- BEASLEY, J. E. An exact two-dimensional non-guillotine cutting tree search procedure. **Operational Research**, Catonsville, v. 33, n.1, p. 49-64, 1985.
- BEASLEY, J. E. A population heuristic for constrained two-dimensional non-guillotine cutting. **European Journal of Operational Research**, London, v. 156, n. 3, p. 601-627, 2004.
- BEKRAR, A.; KACEM, I.; CHU, C. A comparative study of exact algorithms for the two dimensional strip packing problem. **Journal of Industrial and Systems Engineering**, Binghamton, v. 1, n. 2, p. 151-170, 2007.
- BELTRAN, J. D. et al. GRASP/VNS hybrid for the strip packing problem. In.: INTERNATIONAL WORKSHOP ON HIBRID METAHEURISTICS (HM -04), 1., 2004, Valencia. **Proceedings...** Valencia: Universidad de la Laguna, 2004. p. 79-90.

BERKEY, J.; WANG, P. Two-dimensional finite bin packing algorithms. **Journal of the Operational Research Society**, Basingstoke, v. 38, n. 5, p. 423-429, 1987.

BINKLEY, K. B. e HAGIWARA, M. Applying self-adaptive evolutionary algorithms to two-dimensional packing problems using a four corners-heuristic. **European Journal of Operational Research**, London, v. 183, n. 3, p. 1230-1248, 2007.

BIRO, M.; BOROS, E. Network flows and non-guillotine cutting patterns. **European Journal of Operational Research**, London, v. 16, n. 2, p. 215-221, 1984.

BORTFELD, A. A genetic algorithm for the two-dimensional strip packing problem with rectangular pieces. **European Journal of Operational Research**, London, v. 172, n. 3, p. 814-837, 2006.

BURKE, E.; KENDALL, G. Applying ant algorithms and the no fit polygon to the nesting problem. In.: AUSTRALIAN JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 12., 1999, Sydney. **Proceedings...** Sydney: AI99 LNCS, 1999. p. 453-464.

BURKE, E.; KENDALL, G.; WHITWELL, G. Metaheuristic enhancements of the best-fit heuristic for the orthogonal stock cutting problem. **INFORMS Journal on Computing**, California, v. 52, n. 4, p. 655-671, 2004.

CAPRARA, A.; MONACI, M. On the two dimensional knapsack problem. **Operations Research Letters**, Amsterdam, v. 32, n.1, p. 5-14, 2004.

CHAZELLE, B. The bottom-left bin packing heuristic: an efficient implementation. **IEEE Transactions on Computers**, Sydney, v. 32, n. 8, p. 697-707, 1983.

CHEN, C. S.; LEE, S. M.; SHEN, Q. S. An analytical model for the container loading problem. **European Journal of Operational Research**, London, v. 80, n. 1, p. 68-76, 1995.

CHRISTOFIDES, N.; WHITLOCK, C. An algorithm for two-dimensional cutting problems. **Operations Research**, Catonsville, v. 25, n. 1, p. 30-44, 1977.

CHUNG, F. K. R.; GAREY, M. R.; JOHNSON, D. S. On packing two-dimensional bins. **SIAM J. of Algebraic and Discrete Methods**, Auckland, v. 3, n. 1, p. 66-76, 1982.

CINTRA, G. F. et al. Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation. **European Journal of Operational Research**, London, v. 191, n. 1, p. 61-85, 2008.

CRUZ, C. **Estrategias coordinadas paralelas basadas en soft-computing para la solución de problemas de optimización**. Tesis (Doctorado en Ciencias de la Computación e Inteligencia artificial) - Universidad de Granada, Granada, 2005.

CUI, Y. An exact algorithm for generating homogenous T-shape cutting patterns. **Computers and Operations Research**, Oxford, v. 34, n. 4, p. 1107-1120, 2007.

CUI, Y. et al. A recursive branch-and-bound algorithm for the rectangular guillotine strip packing problem. **Computers and Operations Research**, Chicago, v. 35, n. 4, p. 1281-1291, 2008.

DANIELS, K.; MILENKOVIC, V. J.; LI, Z. Multiple containment methods. **Center for Research in Computing Technology**, Cambridge, v. 37, n. 6, p. 12-94, 1994.

DELL AMICO, M.; MARTELLO, S.; VIGO, D. A lower bound for the nonoriented two-dimensional bin packing problem. **Discrete Applied Mathematics**, Amsterdam, v. 118, n. 2, p. 13-24, 2002.

DOWSLAND, K. A.; DOWSLAND, W. Packing problems. **European Journal of Operational Research**, London, v. 56, n. 1, p. 2-14, 1992.

DOWSLAND, K.A. Some experiments with simulated annealing techniques for packing problems. **European Journal of Operational Research**, London, v. 68, n. 3, p. 389-399, 1993.

DYCKHOFF, H. A typology of cutting and packing problems. **European Journal of Operational Research**, London, v. 44, n. 2, p. 145-159, 1990.

FAROE, O.; PISINGER, D.; ZACHARIASEN, M. Guided local search for the three-dimensional Bin-Packing Problem. **INFORMS Journal on Computing**, California, v. 15, n. 3, p. 267-283, 2003.

FEKETE, S. P.; SCHEPERS, J. **On more-dimensional packing III: exact algorithms**. Germany: University of Köln, 1997.

FERNANDEZ DE LA VEGA, W.; ZISSIMOPOULOS, V. An approximation scheme for strip packing of rectangles with bounded dimensions. **Discrete Applied Mathematics**, Amsterdam, v. 82, n. 1-3, p. 93-101, 1998.

GALLEGO, R.; ESCOBAR A.; ROMERO, R. **Programacion Lineal Entera**. Pereira: Universidad Tecnológica de Pereira, 2008.

- GALLEGO, R.; ESCOBAR, A. H.; TORO, E. M. **Técnicas metaheurísticas de optimización**. Pereira: Universidad Tecnológica de Pereira, 2008.
- GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: a guide to the theory of NP-completeness**. San Francisco: W. H. Freeman, 1979.
- GILMORE, P.; GOMORY, R. A linear programming approach to the cutting stock problem. **Operations Research**, Catonsville, v. 9, n. 6, p.849-859, 1961.
- GILMORE, P.; GOMORY, R. A linear programming approach to the cutting stock problem - part II. **Operations Research**, Catonsville, v. 11, n. 6, p. 863-888, 1963.
- GILMORE, P.; GOMORY, R. Multi-stage cutting stock problems of two and more dimensions. **Operations Research**, Catonsville, v. 13, n. 1, p. 94-120, 1965.
- GILMORE, P.; GOMORY, R. The theory and computation of knapsack functions. **Operations Research**, Catonsville, v. 14, n. 6, p. 1045-1074, 1966.
- GLOVER F. **The Combinatoric Explosion in Artificial Intelligence and Operations Research: reassessment**. Greenberg: North-Holland Publishing, 1986.
- GLOVER, F. **Handbook of Metaheuristics**. Norwell: Kluwer Academic Publishers, 2003.
- GROTSCHHEL, M., JUNGER, M., REINELT, G. A cutting plane algorithm for the linear ordering problem. **Operations Research**, Catonsville, v. 32, n. 6, p. 1195-1220, 1983.
- HADJICONSYANTINOUS, E.; CHRISTOFIDES, N. An exact algorithm for general, orthogonal, two dimensional knapsack problems. **European Journal of Operational Research**, London, v. 83, n. 1, p. 39-56, 1995.
- HIFI, M. Exact algorithms for the guillotine strip cutting/packing problem. **Computers and Operations Research**, Oxford, v. 25, n. 11, p. 925-940, 1998.
- HINXMAN, A. I . A two dimensional trim-loss problem with sequencing constraints. **Advanced Paper of IJCAI-77 MTI** , v. 2, n. 4, p. 859-864, 1977.
- HOPPER, E.; TURTON, B. C. H. A review of the application of meta-heuristic algorithms to 2D strip packing problems. **Artificial Intelligence Review**, Cardiff, v. 16, n. 4, p. 257-300, 2001.
- JACKOBS, S. On Genetic Algorithms For The Packing Of Polygons. **European Journal Of Operational Research**, London, v.88, n.1, p. 165-181, 1996.

KENYON, C.; RÉMILA, E. Approximate strip packing. In.: ANNUAL SYMPOSIUM ON FOUNDATION OF COMPUTER SCIENCE (FOCS 96), 37., 1996, Burlington. **Proceedings...** Burlington: IEEE Computer Society, 1996. p. 31-36.

KIRKPATRICK, S.; VECCHI, M. Optimization by simulated annealing. **Science, New Series**, Boston, v. 220, n. 4598, pp 671-680, 1983.

LAI, K. K.; CHAN, J. W. M. An evolutionary algorithm for the rectangular cutting stock problem. **International Journal of Industrial Engineering**, Bruselas, v. 4, n. 5 , p. 130-139, 1997.

LAI, K. K.; CHAN, J. W. M. Developing a simulated annealing algorithm for the cutting stock problem. **Computers and Industrial Engineering**, Philadelphia, v. 32, n. 1, p. 115-127, 1997.

LESH, N. et al. Exhaustive approaches to 2D rectangular perfect packings. **Information Processing Letters**, Amsterdam, v. 90, n. 1 , p. 7-14, 2004.

LEUNG, T.W.; YUNG, C. H.; TROUTT, M.D. Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem. **Computers and Industrial Engineering**, Philadelphia, v. 4, n. 3, p. 201-214, 2001.

LEUNG, C. H.; ZHANG, D.; SIM, K. M. A two-stage intelligent search algorithm for the two-dimensional strip packing problem. **European Journal of Operational Research Society**, London, v. 215, n. 1, p. 57 - 69, 2011.

LIU, R. et al. Module placement with boundary constraints using o-tree representation. In.: IEEE International Symposium on Circuits and Systems (ISCAS2002), 2., 2002, Phoenix. **Proceedings...** Phoenix: IEEE, 2002. p. 871-874.

LODI, A.; MARTELLO, S.; VIGO, D. Neighborhood search algorithm for the guillotine non-oriented two-dimensional bin packing problem. In.: META-HEURISTICS: ADVANCES AND TRENDS IN LOCAL SEARCH PARADIGMS FOR OPTIMIZATION, 1., 1998, Boston. **Proceedings...** Boston: Kluwer Academic Publishers, 1998. p. 125-139.

LODI, A.; MARTELLO, S.; VIGO, D. Heuristic and metaheuristic approaches for a class of two-dimensional bin packing problems. **INFORMS Journal of Computing**, California, v. 11, n.4, p. 345-357, 1999.

LODI, A.; MARTELLO, S.; VIGO, D. Approximation algorithms for the oriented two-dimensional bin packing problem. **European Journal of Operational Research**, London, v. 112, n. 1, p. 158-166, 1999.

LODI, A.; MARTELLO, S.; MONACI, M. Two-dimensional packing problems: A survey. **European Journal of Operational Research**, London, v. 141, n. 2, p. 241-252, 2002.

LODI, A.; MARTELLO, S.; VIGO, D. Models and bounds for two-dimensional level packing problems. **Journal of Combinatorial Optimization**, Amsterdam, v. 8, n. 1, p. 363-379, 2004.

MACLEOD, B. et al. An algorithm for the 2D guillotine cutting stock problem. **European Journal of Operational Research**, London, v. 68, n. 3, p. 400-412, 1993.

MARTELLO, S.; MONACI, M.; VIGO, D. An exact approach to the strip-packing problem. **INFORMS Journal on Computing**, California, v. 15, n. 3, p. 310-319, 2003.

ONODERA, H.; TANIGUCHI, Y.; TAMARU, K. Branch-and-bound placement for building block layout. In.: ACM/IEEE DESIGN AUTOMATION CONFERENCE, 28., 1991, New York. **Proceedings...** New York: ACM, 1991. p. 433-439.

PADBERG, M. W.; RINALDI, G. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. **Society for Industrial and applied mathematics**, Auckland, v. 33, n. 1, p. 60-100, 1991.

PANTRIGO, J. **Resolucion de problemas de optimizacion dinámica mediante la hibridación de partículas y metaheurísticas poblacionales**. Tesis (Doctorado en Sistemas Telemáticos e Informáticos) - Universidad Rey Juan Carlos, Madrid, 2005.

PARREÑO, F. **Algoritmos heurísticos y exactos para problemas de corte no guillotina en dos dimensiones**. Tesis (Doctorado en Probabilidad y Estadística) - Universidad De Valencia, Valencia, 2004.

PETER, J.; AARTS, E. **Simulted annealing:theory and applications**. Boston: Kluwer Academic Publishers, 1992.

PUCHINGER, J.; RAIDL, G.R. Models and algorithms for three-stage two-dimensional bin packing. **European Journal of Operational Research**, London, v. 183, n. 3, p. 1304-1327, 2007.

RUIZ, H. **Estimación de estado de sistemas eléctricos usando algoritmos de optimizacion**. Tesis (Maestria en Planeamiento de Sistemas Eléctricos de Potência) - Universidad Tecnológica de Pereira, Pereira, 2009.

SALKIN, H. M.; DEKLUYVER, C. A. The knapsack Problem: A Survey. **Naval Research Logistics Quartely**, Maldem, v. 22, n. 1, pp. 127-144, 1975.

SCHEITHAUER, G.; TERNO, J. Modeling of packing problems. **Optimization**, Halle, v. 28, n. 1, p.63-84, 1993.

SUCUPIRA, I. **Um Estudo Empírico de Hiper-Heurísticas**. Dissertação (Mestrado em Ciência da Computação) - IME/USP, São Paulo, 2007.

SWEENEY, P. E.; PATERNOSTER, R. Cutting and Packing Problems: a categorized, application-orientated research bibliography. **Journal of the Operational Research Society**, Basingstoke, v. 43, n. 7, p. 691-706, 1992.

THIENEL, S. **ABACUS A Branch And Cut System, Version 2.3**. Colonia: OREAS, 1995.

TORO, E.; GARCÉS, A.; RUIZ, H. Solución al problema de empaquetamiento bidimensional usando un algoritmo híbrido constructivo de búsqueda en vecindad variable y recocido simulado. **Revista Facultad de Ingeniería**, Medellín, v. 46, n. 1, p. 119-131, 2008.

TSAI, R. D; MALSTROM, E. M.; MEEKS, H. D. A two dimensional palletizing procedure for warehouse loading operations. **IIE Transactions**, Tempe, v. 20, n. 4, p.418-425, 1988.

WANG, P. Y.; WASCHER, G. Cutting and packing. **European Journal of Operational Research**, London, v. 141, n. 2 , p. 239-469, 2002.

WASCHER, G.; HAUBNER, H.; SCHUMANN, H. **An Improved Typology of Cutting and Packing Problems**. Magdeburg: Faculty of Economics and Management, 2004.

WASCHER, G.; HAUBNER, H.; SCHUMANN, H. Cutting and packing. **European Journal of Operational Research**, London, v. 141, n. 2, p. 239-469, 2007.

Wu, Y. L. et al. An effective quasi-human based heuristic for solving rectangle packing problem. **European Journal of Operational Research**, London, v. 141, n. 1, p. 341-358, 2002.

YEUNG, L.H.W.; TANG, W.K.S. A hybrid genetic approach for garment cutting in the clothing industry. **IEEE Transactions on Industrial Electronics**, Auburn, v. 50, n. 3, p. 449-455, 2003.

ZHANG, D.; KANG, Y.; DENG, A. A new heuristic recursive algorithm for the strip rectangular packing problem. **Computers and Operations Research**, Chicago, v. 33, n.1, p. 2209-2217, 2006.