

**UNIVERSIDADE NOVE DE JULHO - UNINOVE
PROGRAMA DE MESTRADO EM ENGENHARIA DE PRODUÇÃO**

MARISA CARLA VOIGT GAVA

**RESOLUÇÃO DO PROBLEMA DE CORTE BIDIMENSIONAL COM ITENS
IRREGULARES IDÊNTICOS USANDO ALGORITMOS GENÉTICOS E
PROCESSAMENTO DE IMAGENS DIGITAIS**

São Paulo

2016

MARISA CARLA VOIGT GAVA

**Resolução do problema de corte bidimensional com itens irregulares idênticos
usando algoritmos genéticos e processamento de imagens digitais**

**Dissertação apresentada ao Programa
de Mestrado em Engenharia de
Produção da Universidade Nove de
Julho - UNINOVE, como requisito
parcial para a obtenção do grau de
Mestre em Engenharia de Produção**

**Prof. Sidnei Alves de Araujo, Dr. -
Orientador, UNINOVE**

São Paulo

2016

Gava, Marisa Carla Voigt.

Resolução do problema de corte bidimensional com itens irregulares idênticos usando algoritmos genéticos e processamento de imagens digitais./ Marisa Carla Voigt Gava. 2016.

99 f.

Dissertação (mestrado) – Universidade Nove de Julho - UNINOVE, São Paulo, 2016.

Orientador (a): Prof. Dr. Sidnei Alves de Araújo.

1. Problema de corte. 2. Itens irregulares. 3. Meta-heurísticas. 4. Algoritmo genético. 5. Processamento de imagens digitais.

I. Araújo, Sidnei Alves de.

II. Título

CDU 658.5



PARECER DA COMISSÃO EXAMINADORA DE DEFESA DE DISSERTAÇÃO

DE

Marisa Carla Voigt Gava

Título da Dissertação: Resolução do Problema de Corte Bidimensional com Itens Irregulares Idênticos Usando Algoritmos Genéticos e Processamento de Imagens Digitais.

A Comissão Examinadora, Composta Pelos Professores Abaixo, Considero(a) o(a) candidato(a) Marisa Carla Voigt Gava Aprovada

São Paulo, 29 de fevereiro de 2016.

Sidnei Alves de Araújo (UNINOVE) - Orientador

Fernando Celso de Campos (UNIMEP) - Membro Externo

André Felipe Henriques Librantz (UNINOVE) - Membro Interno

Leonardo Junqueira (UNINOVE) - Membro Interno

*Dedico esse trabalho a Deus, meus pais,
meu marido Gava, meus filhos Tiago e Vitor e
ao meu irmão Cesar (em memória).*

AGRADECIMENTOS

A Deus, pela oportunidade de viver, pela saúde, sabedoria, força, pela coragem que nos concedeu, permanecendo ao meu lado em todos os percursos desta caminhada.

Agradeço à minha família que me apoiou quando me dediquei aos estudos.

Aos meus pais, que mesmo distantes fisicamente, plantaram a sementinha da curiosidade, perseverança e dedicação aos estudos.

Ao meu orientador Professor Dr. Sidnei Alves de Araújo, pela forma como me orientou, pela confiança em meu trabalho, pelas motivações, pela paciência em me atender e me orientar. Meu muito obrigado.

Aos docentes do Programa de Mestrado em Engenharia de Produção, agradeço o carinho e dedicação com o qual me ensinaram novos conteúdos que contribuíram para aumentar significativamente o meu aprendizado.

Ao Professor Dr. Leonardo Junqueira e ao Professor Dr. André Felipe H. Librantz, pelas sugestões apresentadas no momento do exame de qualificação.

Ao Professor Dr. Fabio Henrique Pereira por ter me auxiliado com o algoritmo genético.

Ao Professor Dr. João Carlos Curvelo Santana pelos ensinamentos referente ao planejamento fatorial.

À Universidade Nove de Julho - Uninove, eu agradeço a bolsa de estudos concedida e o apoio recebido dos coordenadores e diretor da informática.

A todos os amigos que compartilharam e conviveram comigo esta aventura da busca do conhecimento.

E, finalmente aos membros da banca pelo pronto atendimento ao convite.

"... porque para Deus nada é impossível."

Lucas (1:37)

RESUMO

O problema de corte consiste em cortar objetos maiores em itens menores com o objetivo de minimizar as sobras. Os objetos podem ser matérias-primas, tais como bobinas de papel, folhas de vidro, placas de metal, aço, alumínio ou madeira. Os itens representam o formato que deverá ser cortado e podem ser descritos como de geometrias irregulares côncavas ou convexas. O corte de matéria-prima é um processo industrial que tem atraído a atenção de muitos pesquisadores, visto que pode gerar grandes desperdícios, elevando o custo da produção. Não obstante, o conjunto de possíveis soluções para esse tipo de problema possui um grande número de combinações e, por esse motivo, sua complexidade computacional é considerada *NP-Hard*. Neste trabalho é proposta uma abordagem baseada em Algoritmo Genético (AG) e Processamento de Imagens Digitais para lidar com o problema de cortar placas retangulares (objetos) em peças idênticas (itens) com formas irregulares, categorizado na literatura como 2D-I-IIPP. O objetivo é maximizar o número de itens a serem cortados na área disponível do objeto, visando diminuir os desperdícios e, conseqüentemente, agregando ganhos econômicos ao processo de corte. Nesta abordagem tanto os objetos como os itens são representados como imagens digitais. O AG é responsável por gerar as possíveis soluções (conjuntos de translações e orientações dos itens). A avaliação de cada solução gerada pelo AG é realizada por um algoritmo de Processamento de Imagens Digitais que basicamente detecta as sobreposições entre os itens posicionados sobre o objeto e calcula a qualidade da solução. Para desenvolver a abordagem proposta foi utilizada a linguagem de programação C/C++, além das bibliotecas GAlib e Proeikon. Os resultados obtidos nos experimentos computacionais realizados indicam que a abordagem proposta é uma boa alternativa para solução do problema investigado.

Palavras-chave: Problema de Corte. Itens irregulares. Meta-Heurísticas. Algoritmo Genético. Processamento de Imagens Digitais.

ABSTRACT

The cutting problem involves cutting larger objects into smaller items with the aim of minimizing waste. The objects can be raw materials, such as rolls of paper, glass sheets, metal plates, steel, aluminum or wood. The items represent the shape to be cut and may be described as concave or convex irregular geometries. The cut of raw material is an industrial process which has attracted the attention of many researchers, since it can generate large waste, increasing the production cost. Nevertheless, the set of possible solutions to this problem has a large number of combinations and, therefore, its computational complexity is considered NP-Hard. In this work, we proposed an approach based on Genetic Algorithm (GA) and Digital Image Processing (DIP) to deal with the problem of to cut rectangular plates (objects) in equal parts (items) with irregular shapes, categorized in the literature as 2D-I-IIPP.

The aim is to maximize the number of items to be cut into the available area of the object in order to reduce waste and thus adding economic gains to the cutting process. In this approach the object and the items are represented as digital images. The GA is responsible for generating possible solutions (sets of translations and orientations of items). The evaluation of each solution generated by GA is performed by a RPID algorithm, which basically detects overlaps between the items placed on the object and calculates the quality of solution. To develop the proposed approach it was used the programming language C/C++ in addition to GAlib and Proeikon libraries. Based on computational experiments conducted the results indicate that the proposed approach is a good alternative to solve the problem investigated.

Keywords: Cutting Problem. Irregular items. Metaheuristic. Genetic Algorithm. Digital Image Processing.

LISTA DE ILUSTRAÇÕES

Figura 1 - Critérios de Categorização.....	27
Figura 2 - Visão Geral dos problemas de corte	28
Figura 3 - Classificação dos tipos de problemas	29
Figura 4 - Tipos de problemas básicos	31
Figura 5 - Relacionamento entre os problemas básicos e intermediários	33
Figura 6 - Representação cromossômica de um indivíduo	38
Figura 7 - Gráfico do quadro 1.	39
Figura 8 - Gráfico do quadro 2	41
Figura 9 - Exemplos de cruzamento.....	43
Figura 10 - Operador de Mutação.	43
Figura 11 - Estrutura básica do AG.....	44
Figura 12 - Representação matricial de uma imagem digital.	48
Figura 13 - Exemplo de imagem binária.....	48
Figura 14 - Exemplo de imagem em níveis de cinzas.....	48
Figura 15 - Composição das cores da tabela 5.....	49
Figura 16 - Visão geral das etapas metodológicas.....	53
Figura 17 – Visão geral do funcionamento da abordagem proposta	56
Figura 18 - Parte do cromossomo representando um item da solução	58
Figura 19 - Exemplo de imagem do item e do objeto.	59
Figura 20 - Exemplo de imagem do planejamento de corte	60
Figura 21 - Esquema de cores adotado para indicar as sobreposições dos itens.....	61
Figura 22 – Fluxograma detalhado do funcionamento da abordagem proposta	64
Figura 23 - Imagem das instâncias dos objetos	65
Figura 24 - Imagem das instâncias dos itens	66
Figura 25 - Gráfico comparativo com o tempo de processamento.	69
Figura 26 - Gráfico comparativo com o valor da FO.....	69
Figura 27 – Análise da influência sobre o coeficiente de sobreposição c_s	71
Figura 28 - Análise da influência sobre o coeficiente de quantidade de itens c_c	71
Figura 29 - Análise da influência sobre o coeficiente de distância c_d	72
Figura 30 - Resultados do experimento AG + RPID.....	74
Figura 31 - Resultados do experimento AG + RPID + HDE + SIF	76

Figura 32 - Resultados do experimento AG + RPID + HDE com o item_0.....	77
Figura 33 - Resultados do experimento AG + RPID + HDE com o item_1.....	78
Figura 34 - Resultados do experimento AG + RPID + HDE com o item_2.....	79
Figura 35 - Resultados do experimento AG + RPID + HDE com o item_3.....	80
Figura 36 - Resultados do experimento AG + RPID + HDE com o item_4.....	81
Figura 37 - Resultados do experimento AG + RPID + HDE com o item_5.....	82
Figura 38 - Resultados do experimento AG + RPID + HDE com o item_6.....	83
Figura 39 - Resultados do experimento AG + RPID + HDE com o item_7.....	84

LISTA DE QUADROS

Quadro 1 - Exemplo de seleção proporcional.	39
Quadro 2 - Exemplo de seleção por ranking.	41
Quadro 3 - Exemplo de seleção por torneio.	42
Quadro 4 - Síntese dos trabalhos abordando soluções para PCEs	50
Quadro 5 - Fatores, níveis e delta	67
Quadro 6 - Planejamento fatorial dos parâmetros	67

LISTA DE TABELAS

Tabela 1 - Termos relacionados na biologia e AG.	37
Tabela 2 - Valores padrão para os parâmetros.	46
Tabela 3 - Sugestão de valores dos parâmetros.	46
Tabela 4 - Valores padrão da literatura.	47
Tabela 5 - Composição de algumas cores no espaço RGB.	49
Tabela 6 - Parâmetros do AG utilizados no trabalho.	70
Tabela 7 - Resultados numéricos do experimento AG+RPID.	75
Tabela 8 - Resultados numéricos do experimento AG+RPID+HDE+SIF.	76
Tabela 9 - Resultados numéricos do experimento AG+RPID+HDE com o item_0.	77
Tabela 10 - Resultados numéricos do experimento AG+RPID+HDE com o item_1.	78
Tabela 11 - Resultados numéricos do experimento AG+RPID+HDE com o item_2.	79
Tabela 12 - Resultados numéricos do experimento AG+RPID+HDE com o item_3.	80
Tabela 13 - Resultados numéricos do experimento AG+RPID+HDE com o item_4.	81
Tabela 14 - Resultados numéricos do experimento AG+RPID+HDE com o item_5.	82
Tabela 15 - Resultados numéricos do experimento AG+RPID+HDE com o item_6.	83
Tabela 16 - Resultados numéricos do experimento AG+RPID+HDE com o item_7.	84

LISTA DE ABREVIATURAS E SIGLAS

1D CSP	<i>One-Dimensional Cutting Stock Problem</i>
2D ASSCSP	<i>Two-Dimensional Arbitrary Stock-Size Cutting Stock Problem</i>
2D CSP	<i>Two-Dimensional Cutting Stock Problem</i>
2D IIPP	<i>Two-Dimensional Identical Item Packing Problem</i>
2D-I-IIPP	<i>Two-Dimensional Irregular Identical Item Packing Problem</i>
AE	Algoritmos Evolucionários
AFSA	<i>Artificial Fish Swarm</i>
AG	Algoritmo Genético
AGs	Algoritmos Genéticos
BPP	<i>Bin Packing Problem</i>
CE	Computação Evolucionária ou Evolutiva
CPU	<i>Central Unit Process</i>
CSA	<i>Corner Space Algorithm</i>
CSP	<i>Cutting Stock problem</i>
DGA	<i>DemeGa</i>
DIP	<i>Digital Image Processing</i>
ExSearch	<i>Extended Search Algorithm</i>
FA	Função de aptidão
FLAOS	<i>Fitness Level based Adaptive Operator Selection</i>
FO	Função objetivo
FTSA	<i>Forest Tree Search Algorithm</i>
GA	<i>Genetic Algorithm</i>
GBA	<i>General Blocks Patterns Algorithm</i>
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
HDE	Heurística Descida de Encosta
HMCGA	<i>Hybrid Multi-Chromosome Genetic Algorithm</i>
ID	Número do processo no computador
IGA	<i>IncrementalGa</i>
IIPP	<i>Identical Item Packing Problem</i>
ILS	<i>Iterated Local Search</i>
KP	<i>Knapsack Problem</i>
LCD	<i>Liquid Crystal Display</i>
LP	<i>Linear Programming</i>
NFP	<i>No Fit Polygon</i>
<i>NP-Hard</i>	<i>No-deterministic Polynomial Time Hard</i>
ODP	<i>Open Dimension Problem</i>
PCE	Problema de Corte e Empacotamento

PCEs	Problemas de Corte e Empacotamento
PE	Problema de Empacotamento
PP	<i>Placement Problem</i>
PSO	<i>Particle Swarm Optimization</i>
RAM	<i>Random Access Memory</i>
RF	Resultado Final
RGB	<i>Red, Green, Blue</i>
RPID	Rotina de Processamento de Imagens Digitais
SBSBPP	<i>Single Bin Size Bin Packing Problem</i>
SGA	<i>SimpleGa</i>
SI	<i>Swarm Intelligence</i>
SIF	Solução Inicial Factível
SLOPP	<i>Single Large Object Placement Problem</i>
SRP	<i>Stock Reduction Problem</i>
SST	<i>Steady-State</i>
VNS	<i>Variable Neighborhood Search</i>
VLP	<i>Vehicle Loading Problem</i>

LISTA DE SIMBOLOS

<i>n</i>	Dimensões geométricas de um item ou objeto
<i>W</i>	Largura do Objeto
<i>H</i>	Altura do Objeto
<i>w</i>	Largura do Item
<i>h</i>	Altura do Item
<i>B</i>	Todos os objetos e seleção de itens
<i>V</i>	Todos os itens e seleção de objetos
<i>O</i>	Um objeto
<i>I</i>	Vários objetos com formatos idênticos
<i>D</i>	Vários objetos com formatos diferentes
<i>F</i>	Poucos itens com poucos formatos
<i>M</i>	Muitos itens com muitos formatos
<i>R</i>	Muitos itens com poucos formatos (não congruente)
<i>C</i>	Formatos congruentes
<i>s</i>	Objeto único
<i>id</i>	Objetos idênticos
<i>wh</i>	Objetos pouco heterogêneos
<i>sh</i>	Objetos muito heterogêneos
<i>p</i>	Quantidade total de Indivíduos de uma população
<i>Fitness_threshold</i>	Valor limite utilizado como critério de parada
<i>pRepl</i>	Percentual da população que será substituída a cada geração
<i>M</i>	Percentual da população que sofrerá mutação
<i>POP</i>	População
<i>Ps</i>	População temporária
<i>H</i>	Indivíduo de uma população
<i>pCross</i>	Percentual da população selecionada para cruzamento
<i>X</i>	Coordenada que representa as colunas da imagem
<i>Y</i>	Coordenada que representa as linhas da imagem
<i>M</i>	Número de linhas de uma imagem
<i>N</i>	Número de colunas de uma imagem
<i>IC</i>	Imagem colorida

L_2	Valor da componente G (verde) do sistema de cores RGB
IC_R	Cor vermelha
IC_G	Cor verde
IC_B	Cor Azul
θ	Ângulo de rotação
St	Status indicando se o item será ou não considerado no planejamento do corte
T_x	Translação horizontal do item a ser cortado
T_y	Translação vertical do item a ser cortado
dx	Passo da translação horizontal tx
dy	Passo da translação vertical ty
$d\theta$	Passo do ângulo θ (rotação)
Max_{tx}	Limite superior da coluna gerado pelo AG
Max_{ty}	Limite superior da linha gerada pelo AG
Max_{rot}	Limite superior de rotações gerada pelo AG
i	Índice do item
Img_O	Imagem do Objeto
Img_I	Imagem do Item
Img_P	Imagem do planejamento do corte
c_s	Coefficiente de sobreposição
c_c	Coefficiente de itens posicionados para o corte
c_d	Coefficiente de distância entre os itens
S	Quantidade de pixels com sobreposição
f	Coordenada referente à coluna do centro da imagem do item
g	Coordenada referente à linha do centro da imagem do item
T_i	Limite superior do número de itens
A_o	Área disponível no objeto
A_i	Área do item
Q_i	Quantidade real de itens que serão cortados
w_s	Peso atribuído ao coeficiente de sobreposição
w_c	Peso atribuído ao coeficiente de itens posicionados para o corte
w_d	Peso atribuído ao coeficiente de distância entre os itens
j	Índice auxiliar do item para cálculo da distância euclidiana

β	Número de combinações de dois itens sem repetição
$nGer$	Número de Gerações
Δ	Variação estimada dos fatores do planejamento fatorial
X_0	Ponto Central do planejamento fatorial
α	Pontos axiais do planejamento fatorial
k	Número de fatores utilizados no planejamento fatorial
$ImgObj$	Nome do arquivo da imagem do objeto
$ImgItem$	Nome do arquivo da imagem do item
$Indv$	Indivíduo de uma população do AG
$pConvergence$	Percentual de convergência como critério de parada
$nConvergence$	Número de gerações comparadas como critério de parada
$pConvergence_calc$	Percentual de convergência calculado na geração atual
$Qtd_itens_solucao$	Quantidade de itens com st igual a um
$Melhor_indv$	Indivíduo que possui maior valor da FO em uma população
$nGerações$	Número da geração atual
$Sem_melhoria$	Quantidade de gerações sem minimização no valor da FO
FO_0, FO_1	Valor da FO antes e depois da aplicação da HDE
Qtd	Quantidade de indivíduos selecionados para aplicar a HDE
$Acao_tx$	Incremento ou decremento da coordenada tx
$Acao_ty$	Incremento ou decremento da coordenada ty
$Acao_rot$	Incremento ou decremento do ângulo de rotação θ
Po	Contador de pixels na cor branca
Pi	Contador de pixels na cor preta
$Psbr$	Contador de pixels com sobreposição

SUMÁRIO

1	INTRODUÇÃO	19
1.1	PROBLEMA DA PESQUISA	22
1.2	OBJETIVOS	23
1.2.1	Objetivo Geral	23
1.2.2	Objetivos Específicos	23
1.3	JUSTIFICATIVA	24
1.4	ESTRUTURA DO TRABALHO	25
2	FUNDAMENTAÇÃO TEÓRICA	25
2.1	PROBLEMAS DE CORTE E EMPACOTAMENTO (PCEs)	26
2.1.1	Problemas Básicos	31
2.1.2	Problemas Intermediários	32
2.1.3	Problemas Refinados	33
2.2	ALGORITMOS DE OTIMIZAÇÃO COMBINATÓRIA	34
2.2.1	Algoritmo Genético (AG)	36
2.2.1.1	Representação	37
2.2.1.2	Seleção	39
2.2.1.3	Cruzamento	42
2.2.1.4	Mutação	43
2.2.1.5	Pseudocódigo básico do AG	44
2.2.1.6	Parâmetros do AG	46
2.3	PROCESSAMENTO DE IMAGENS DIGITAIS	47
2.4	TRABALHOS CORRELATOS	50
3	MATERIAIS E MÉTODOS	52
3.1	CARACTERIZAÇÃO METODOLÓGICA DA PESQUISA	52
3.2	METODOLOGIA EMPREGADA NO DESENVOLVIMENTO E AVALIAÇÃO DA ABORDAGEM PROPOSTA	53

3.3 MATERIAIS UTILIZADOS NO DESENVOLVIMENTO DA ABORDAGEM PROPOSTA	54
4 ABORDAGEM PROPOSTA.....	56
4.1 VISÃO GERAL DA ABORDAGEM PROPOSTA	56
4.2 REPRESENTAÇÃO CROMOSSÔMICA EMPREGADA NO AG.....	57
4.3 DETALHAMENTO DA ABORDAGEM PROPOSTA.....	59
5 RESULTADOS DOS EXPERIMENTOS COMPUTACIONAIS	65
5.1 DEFINIÇÃO DOS PARÂMETROS DE CONFIGURAÇÃO DO AG	66
5.2 DEFINIÇÃO DOS PESOS DOS COEFICIENTES QUE COMPÕEM A FO.....	70
5.3 AVALIAÇÃO DA ABORDAGEM PROPOSTA	73
5.3.1 Experimentos com AG + RPID	73
5.3.2 Experimentos com AG + RPID + HDE + SIF.....	75
5.3.3 Experimentos com AG + RPID + HDE	77
5.4 DISCUSSÃO DOS RESULTADOS	85
6 CONCLUSÕES E SUGESTÕES PARA A CONTINUIDADE DO TRABALHO .	87
REFERÊNCIAS.....	89
APÊNDICES	94

1 INTRODUÇÃO

Os Problemas de Corte e Empacotamento (PCEs) pertencem à área de Pesquisa Operacional e são classificados como problemas de otimização combinatória do tipo *NP-Hard* (*No-deterministic Polynomial Time Hard*), uma vez que possuem ordem de complexidade exponencial. Em outras palavras, para a resolução destes problemas é necessário um esforço computacional que cresce exponencialmente com o tamanho dos problemas (GAREY; JOHNSON, 1979). Além disso, esses problemas também têm se tornando comuns em outras áreas do conhecimento tais como a Ciência da Computação, Engenharia de Produção e Logística, dentre outras (PENG; CHU, 2010).

Os Problemas de Corte possuem várias aplicações práticas e surgem em uma grande variedade de processos industriais, tais como fabricação de vestuário, corte de placas metálicas ou plásticas, fabricação de móveis, fabricação de calçados, etc., e as matérias-primas (objetos) utilizadas nestes processos de fabricação são fornecidas em forma de bobinas de papel, folhas de vidro, placas de metal, alumínio ou madeira.

Sendo assim, os Problemas de Corte consistem em cortar esses objetos maiores em itens menores, com o objetivo de minimizar as sobras de matérias-primas, muitas vezes com intuito de maximizar os lucros. Desta forma, para se atingir esse objetivo é necessário determinar um padrão de corte que maximize a quantidade de itens menores cortados sobre o objeto.

Segundo Wäscher *et al.* (2007), tanto o Problema de Corte como o Problema de Empacotamento possuem estruturas idênticas, visto que, em ambos existem dois conjuntos de elementos que podem ter de uma a n dimensões geométricas. O conjunto dos elementos maiores é denominado conjunto de objetos e o conjunto dos elementos menores é denominado conjunto de itens. Os autores detalham que um ou mais elementos do conjunto de itens são selecionados e agrupados em um ou mais elementos do conjunto de objetos, conforme a capacidade e geometria dos objetos. Em outras palavras, os itens devem ser embalados (ou arranjados) inteiramente dentro de (ou sobre) um ou mais objetos, obedecendo restrições específicas ao problema.

Os objetos no qual os itens deverão ser cortados podem ter formas regulares ou irregulares. Regulares (na maioria dos casos retangulares) para as matérias-primas fornecidas em rolo, placas ou folhas, e irregulares para o caso de peles de couro ou placas metálicas, que podem, inclusive, apresentar regiões com defeitos, ou ainda apresentar regiões já cortadas (furos). Os itens podem ser descritos como polígonos convexos ou não, e ainda incluir bordas curvas (ALVAREZ-VALDEZ *et al.*, 2013).

Os PCEs tem sido alvo de muitas pesquisas e vêm despertando a atenção de vários pesquisadores. Segundo Wäscher *et al.* (2007), percebeu-se um aumento no número de publicações com novas abordagens para resolver os PCEs. Acredita-se que esse crescente interesse em solucionar tais problemas esteja ligado à minimização dos custos de produção. Desta forma, é possível encontrar na literatura, vários trabalhos que exemplificam a extensa diversidade de aplicações, materiais e recursos computacionais empregados nos Problemas de Corte.

Um exemplo de Problema de Corte com placas defeituosas retangulares foi apresentado por Vianna e Arenales (2006), no qual se empregou uma abordagem baseada em grafo E/OU.

Kallrath *et al.* (2014) discutiram um problema real em uma empresa de papel e celulose. Os problemas tratados se referiam a diferentes larguras de rolos de papel, ao número limitado destes em estoque e a problemas com a superprodução. Além disso, o Problema de Corte com folhas de vidro foi abordado por Lu e Huang (2015) em uma indústria de monitores LCD (*Liquid Crystal Display*). Nesse problema, o plano de produção restringia o corte da folha a apenas um tamanho de monitor, porém, existia a necessidade de cortar as folhas retangulares para diversos tamanhos de monitores. O grande desafio era determinar um plano de produção que visava diversificar o tamanho do item a ser cortado nas folhas, conforme a demanda dos clientes.

Observa-se que os pesquisadores têm proposto vários algoritmos que combinam um ou mais métodos, a fim de solucionar um determinado problema. Dentre os métodos computacionais empregados na solução dos PCEs estão: Algoritmo Genético (AG), *Branch and Bound*, *Branch-and-Price*, *Bottom Left*, *Simulated Annealing*, *Greedy Randomized Adaptive Search Procedure (GRASP)*, *Hiper-Heuristic*, *Genetic Programming*, *Colônia de Formigas*, *Fish Swarm*, *etc.*

Nota-se, na literatura, a ocorrência de vários trabalhos que utilizam recursos computacionais para solucionar os PCEs, entretanto a grande maioria desses estudos abordam Problemas de Corte apenas considerando itens regulares.

Um outro método pouco explorado, que pode ser utilizado na resolução dos PCEs é o Processamento de Imagens Digitais, o qual possui muitas aplicações como: análise de imagens médicas, processamento de imagens de documentos, reconhecimento de impressões digitais, reconhecimento de padrões em imagens de satélites, entre outras (MELLO, 2014). No caso dos PCEs, o Processamento de Imagens Digitais pode ser usado como um método de varredura para detectar sobreposições dos itens na região de corte.

Por se tratar de problemas do tipo *NP-hard*, Kallrath *et al.* (2014) descrevem que a maioria das abordagens descritas na literatura para resolver problemas deste tipo, são baseados em métodos heurísticos. Para Sedgewick (2002), dada a dificuldade de chegar a uma solução ótima para os PCEs em tempo hábil, em virtude do grande número de combinações a serem testadas, os pesquisadores têm optado por métodos heurísticos que não garantem soluções ótimas, mas que são capazes de encontrar boas soluções rapidamente.

Já as meta-heurísticas vão além das heurísticas, visto que englobam estratégias mais genéricas que se aplicam a um número maior de problemas (MELIÁN *et al.*, 2003). O AG, por exemplo, é uma das meta-heurísticas de otimização amplamente utilizada para gerar conjuntos de soluções factíveis para os PCEs. Segundo Belfiore e Fávero (2013), entende-se que uma solução factível é aquela que satisfaz todas as restrições de um problema. A ocorrência de qualquer violação às restrições leva a uma solução não factível. A solução factível que apresenta o melhor valor possível da função objetivo (FO) é chamada solução ótima. Percebe-se uma vasta gama de aplicações dos PCEs e estes implicam em muitas variantes diferentes de problemas. Sendo assim, Wäscher *et al.* (2007) apresentaram uma tipologia melhorada, quando comparada à tipologia de Dyckhoff (1990), cujos PCEs podem ser classificados em diferentes tipos de problemas básicos observando-se alguns critérios.

Neste trabalho empregou-se o AG e uma rotina de Processamento de Imagens Digitais para resolver o Problema de Corte bidimensional com itens irregulares idênticos. As próximas seções descrevem os procedimentos utilizados.

1.1 PROBLEMA DA PESQUISA

Conforme o exposto, os Problemas de Corte consistem em cortar objetos maiores em itens menores e são classificados como problemas de otimização combinatória com complexidade computacional do tipo *NP-Hard* (GAREY e JOHNSON, 1979). No presente trabalho, o objeto se refere a uma placa enquanto o item é uma peça que deverá ser cortada na placa. Ambos são representados por imagens digitais.

O objeto é retangular e sua imagem tem as dimensões fixas, representadas por W (largura) e H (altura). Considera-se ainda, que este objeto pode ser produto de sobra e apresentar furos localizados em diferentes posições da sua superfície. Ressalva-se que não é especificada a matéria-prima do objeto, porque a abordagem apresentada pode ser aplicada a qualquer material. Os itens possuem formatos irregulares e são idênticos, suas imagens têm dimensões w (largura) e h (altura), e podem ser rotacionados em uma angulação variando de 0° a 315° , com passo fixo de 45° .

Nesta situação, segundo Wäscher *et al.* (2007), o Problema de Corte abordado pode ser categorizado como sendo do tipo 2D-I-IIPP (*Two-Dimensional Irregular Identical Item Packing Problem*).

A dificuldade específica deste trabalho está em implementar algoritmos que empregam o AG e Processamento de Imagens Digitais, capazes de posicionar o maior número possível de itens idênticos irregulares sobre um objeto que pode apresentar furos, obedecendo às restrições de não sobreposição, bem como os ângulos considerados (0° , 45° , 90° , 135° , 180° , 225° , 270° e 315°).

Na literatura foram encontrados poucos estudos que apresentam soluções para resolver o Problema de Corte bidimensional com itens irregulares idênticos. Em adição, nenhum deles aborda o uso de AG combinado com o Processamento de Imagens Digitais para a resolução do problema 2D-I-IIPP e também não considera objetos contendo furos em suas superfícies. Diante disso, formulou-se a seguinte questão de pesquisa: É possível maximizar a quantidade de itens irregulares idênticos a serem cortados na área disponível de um objeto bidimensional que possui furos em sua superfície, por meio de uma abordagem que emprega AG e uma rotina de Processamento de Imagens Digitais como função objetivo?

1.2 OBJETIVOS

Este trabalho foi desenvolvido com intuito de apresentar uma abordagem para solução do problema classificado como 2D-I-IIPP, tanto no âmbito prático quanto acadêmico. No âmbito prático, pretendeu-se apresentar, por meio de uma imagem digital, o planejamento de corte de itens irregulares idênticos sobre um objeto. Já com relação ao acadêmico, pretendeu-se desenvolver uma nova abordagem empregando o AG combinado com uma rotina de Processamento de Imagens Digitais.

1.2.1 Objetivo Geral

Propor uma abordagem empregando AG e Processamento de Imagens Digitais para solucionar o problema de corte bidimensional com itens irregulares idênticos, categorizado como 2D-I-IIPP.

1.2.2 Objetivos Específicos

Para se atingir o objetivo geral, as seguintes etapas foram seguidas:

- a) Realizar um estudo bibliográfico para identificar os trabalhos propondo a solução de problemas correlatos ao investigado;
- b) Estudar e entender o funcionamento da meta-heurística AG e do Processamento de Imagens Digitais, além de compreender as bibliotecas GaLib e Proeikon;
- c) Desenvolver uma abordagem que emprega o AG e o Processamento de Imagens Digitais para resolver o Problema de Corte 2D-I-IIPP;
- d) Implementar os algoritmos envolvidos na abordagem proposta em linguagem C/C++ usando as bibliotecas GaLib e Proeikon;
- e) Criar instâncias (ou exemplares) do objeto e do item para execução de experimentos visando avaliar a abordagem proposta;
- f) Conduzir experimentos a fim de obter os valores adequados para os parâmetros do AG e para os pesos dos coeficientes da função objetivo;
- g) Conduzir uma série de experimentos visando avaliar a abordagem proposta.

1.3 JUSTIFICATIVA

Segundo Silva *et al.* (2014), os PCEs são problemas de otimização combinatória difíceis de serem resolvidos, que surgem no contexto de várias aplicações do mundo real. Eles estão presentes, por exemplo, em indústrias que lidam com diferentes tipos de matérias-primas como: papel, madeira, tecidos, metal entre outros. Segundo Wäscher *et al.* (2007), o objetivo dos PCEs é otimizar a utilização dessas matérias-primas (objetos) a fim de diminuir sobras ou desperdícios.

Desta forma, novas soluções que visam otimizar o Problema de Corte nas indústrias são justificadas pela necessidade de minimizar os desperdícios de matéria-prima. Os reflexos com a minimização desses desperdícios podem ser: a redução do custo de produção e, conseqüentemente, o aumento da competitividade e dos lucros.

Além desse aspecto prático, observa-se também que esse assunto tem despertado o interesse de muitos pesquisadores que atuam, principalmente, na área de Engenharia de Produção, visto que novas abordagens para resolver os diferentes tipos de PCEs podem trazer contribuições acadêmicas. Desse ponto de vista, pode-se vislumbrar contribuições diretas deste trabalho, já que dentre as publicações pesquisadas até fevereiro de 2016, não foram encontrados trabalhos empregando AG e o Processamento de Imagens Digitais para resolver o problema 2D-I-IIPP. Tampouco foram encontradas soluções para o problema de corte bidimensional com itens irregulares idênticos considerando objetos com furos em suas superfícies.

Soma-se a isso o fato de que o método *No Fit Polygon* (NFP), muito citado na literatura e empregado em conjunto com outros métodos para solução problemas de corte com itens irregulares, tem sido utilizado por pesquisadores apenas para um conjunto limitado de instâncias para as quais o NFP já foi calculado. Isso é decorrente da complexidade de sua implementação, o que significa um obstáculo para a utilização do NFP quando se pretende abordar um problema prático que inclui novos polígonos (EGEBLAD, 2008).

Dentre várias meta-heurísticas disponíveis, optou-se por utilizar o AG devido aos bons resultados apresentados na literatura para resolver problemas correlatos como os descritos em Shen e Zhang (2010), Lu e Huang (2015), Peng e Chu(2010) e Soto *et al.* (2013).

Com base no exposto, pode-se dizer que a contribuição acadêmica deste trabalho está em apresentar uma nova abordagem que utiliza o AG e o

Processamento de Imagens Digitais para resolução do 2D-I-IIPP, a qual poderá servir de base para novos modelos e ferramentas práticas.

1.4 ESTRUTURA DO TRABALHO

Este trabalho está organizado em 6 capítulos. No capítulo 1, apresentam-se a Introdução, o Problema da Pesquisa, os Objetivos e a Justificativa. No capítulo 2 é apresentada uma revisão bibliográfica sobre os problemas de corte e empacotamento, algoritmos de otimização combinatória, processamento de imagens digitais e trabalhos correlatos publicados na literatura. No capítulo 3, inicialmente é descrita a metodologia utilizada no desenvolvimento da pesquisa e na condução dos experimentos e, em seguida, descrevem-se os materiais utilizados. No capítulo 4, é descrita a abordagem proposta neste trabalho, envolvendo os procedimentos que a compõem. Também neste capítulo demonstra-se como a solução do problema investigado foi representada pelo cromossomo do AG e como a função objetivo é calculada a partir de uma rotina de processamento de imagens digitais. No capítulo 5 são apresentados e discutidos os resultados obtidos nos experimentos realizados para parametrização do AG, definição dos pesos da função objetivo e avaliação da abordagem proposta. Por fim, no capítulo 6, são apresentadas a conclusão e as sugestões para a continuidade do trabalho.

Além dos capítulos relacionados, fazem parte deste trabalho: as Referências Bibliográficas e os Apêndices A e B que apresentam, respectivamente, o detalhamento dos pseudocódigos implementados e uma lista dos trabalhos resultantes desta dissertação.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo é constituído por conteúdos que permitem ao leitor entender os diversos conceitos abordados adiante no trabalho. Apresenta-se, inicialmente, uma revisão bibliográfica que aborda os Problemas de Corte e Empacotamento, incluindo suas categorizações e detalhamentos. Em seguida, são abordados os algoritmos de otimização combinatória detalhando as características das heurísticas e meta-heurísticas, computação evolucionária e algoritmos evolucionários. Destaca-se, neste capítulo, o algoritmo genético, sua representação cromossômica, os operadores de seleção, cruzamento e mutação, seu pseudocódigo básico e enfatiza-se a importância dos valores atribuídos aos seus parâmetros de configuração. Não obstante, as características técnicas de manipulação de imagens bem como as diferenças entre os vários tipos de imagens digitais também são abordadas. Por fim, apresentam-se trabalhos publicados na literatura que tratam dos Problemas de Corte e Empacotamento.

2.1 PROBLEMAS DE CORTE E EMPACOTAMENTO (PCEs)

O Problema de Corte consiste em cortar objetos grandes (bobinas de papel, folhas de vidro, placas de aço, alumínio ou madeira) em itens menores visando minimizar as sobras. Sendo assim, é necessário que um padrão de corte maximize a quantidade cortada de itens menores e minimize o desperdício de materiais. De forma muito parecida, o Problema de Empacotamento (PE) consiste em colocar a maior quantidade de itens menores dentro de objetos maiores

Segundo Wäscher *et al.* (2007), tanto o Problema de Corte como o Problema de Empacotamento possuem uma estrutura comum: um conjunto de objetos maiores (*input ou supply*) e um conjunto de itens menores (*output ou demand*). Independente do problema, todos os itens menores deverão ser posicionados inteiramente sobre ou dentro do objeto maior obedecendo a certas restrições. Dessa maneira, ambos os problemas pertencem a uma classe mais genérica conhecida na literatura como PCEs.

Os PCEs foram categorizados inicialmente em 1990 por Dyckhoff, o qual dividiu tais problemas em quatro categorias considerando-se algumas características como: dimensão, tipo de associação entre os objetos e os itens e a variedade de objetos e

itens. A figura 1 apresenta um resumo com as principais características desta tipologia:

Figura 1 - Critérios de Categorização.

1.	Dimensão	3.	Variedade de Objetos
	(1) - Unidimensional		(O)- Um objeto
	(2) - Bidimensional		(I)- Formato idêntico
	(3) - Tridimensional		(D)- Formato diferente
	(N) - N-dimensional (N>3)		
2.	Tipo de associação	4.	Variedade de Itens
	(B)- Todos os objetos e seleção de itens		(F)- Poucos itens com poucos formatos
	(V)- Todos os itens e seleção de objetos		(M)- Muitos itens com muitos formatos
			(R)- Muitos itens com poucos formatos (não congruente)
			(C)- Formatos congruentes

Fonte: Adaptado de Dyckhof (1990)

Pode-se observar que na primeira categoria Dyckhoff (1990) não limitou-se a apenas uma dimensão, ele previu várias dimensões. A segunda categoria refere-se ao tipo de atribuição entre itens e objetos, indicado pelas letras B e V, ambas referenciando palavras de origem alemã. O tipo B indica que todos os objetos devem ser utilizados e alguns itens deverão ser selecionados. No outro caso, V significa que todos os itens deverão ser utilizados e alguns objetos selecionados.

O terceiro critério refere-se ao número de objetos. Foram considerados os tipos O, I e D. O primeiro refere-se a somente um objeto, o segundo refere-se a vários objetos idênticos e o último a vários objetos diferentes.

A última categoria classifica os itens. Os tipos F, M, R e C foram apresentados, sendo o primeiro para indicar poucos itens com diferentes geometrias, M representa vários itens muito diferentes entre si. R também representa vários itens, mas com uma diversidade menor e por fim a letra C significa figuras congruentes, ou seja, mesma forma e tamanho (DYCKHOFF, 1990).

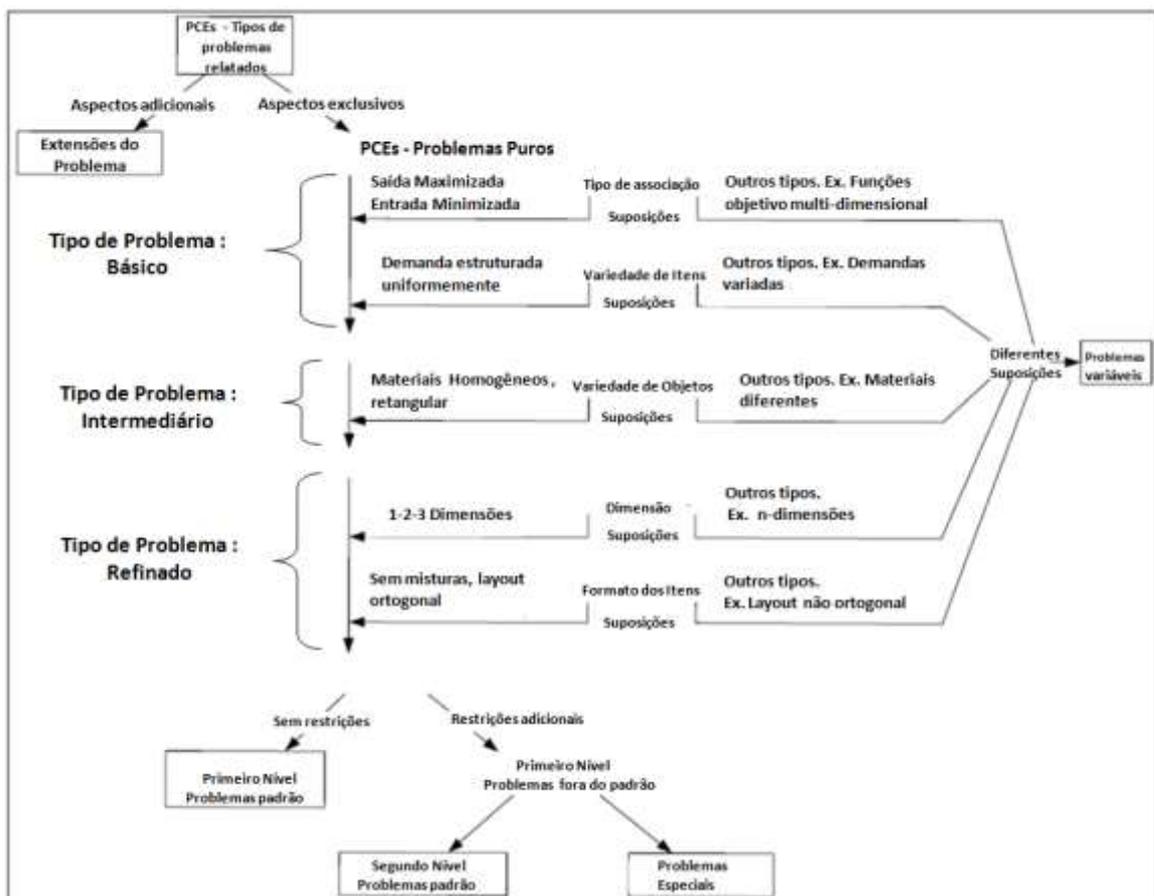
Apesar desta tipologia unir as áreas de corte e empacotamento, a comunidade científica alegou alguns problemas quanto à sua codificação, além de verificar que alguns PCEs não podiam ser categorizados adequadamente ou possuíam dupla categorização. Um exemplo está no problema de *Vehicle Loading Problem (VLP)*, o qual foi categorizado como 1/V//F e 1/V//M, ou seja, a aplicação da tipologia apresentou resultados confusos sendo considerada parcialmente inconsistente.

Wäscher *et al.* (2007) apresentaram uma tipologia melhorada, comparando-se à apresentada por Dyckhoff (1990). Nesta nova abordagem, considera-se que os problemas podem ser divididos em tipos básicos, podendo ser combinados entre si.

Notou-se que o objetivo foi criar uma identificação para os problemas e desta forma servir de base para a investigação científica. O desenvolvimento de modelos, algoritmos, geradores de problemas e a padronização para a classificação na literatura são alguns exemplos de sua aplicação.

Cinco critérios foram utilizados por Wäscher *et al.* (2007), para a definição dos tipos básicos de problemas: dimensão, tipo de associação entre os objetos e os itens, variedade de objetos, variedade de itens e formato dos itens. A figura 2 apresenta os tipos de problemas resultantes da aplicação desses critérios:

Figura 2 - Visão Geral dos problemas de corte



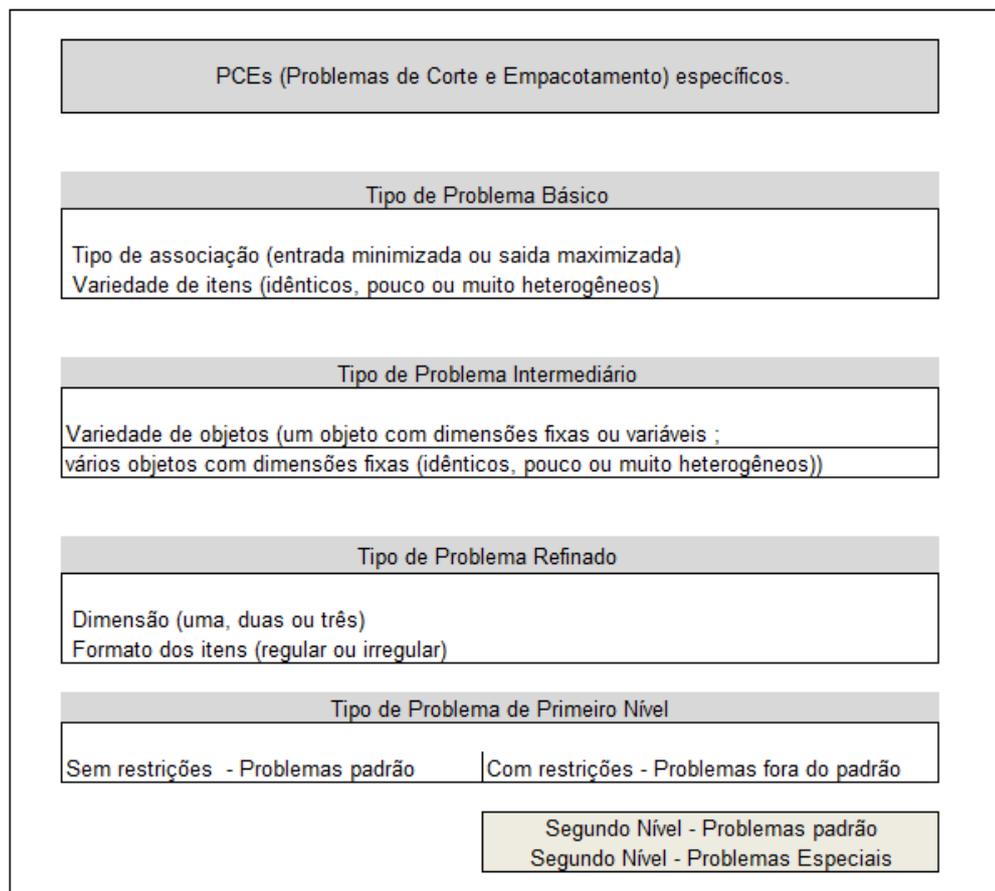
Fonte: Adaptado de Wäscher *et al.* (2007)

Verifica-se que, extensões do problema e problemas puros dos PCEs são as primeiras subdivisões apresentadas. A primeira trata de aspectos adicionais que

ampliam a visão do problema para além das áreas de corte e empacotamento. Este tipo de problema não foi tratado na tipologia apresentada. O segundo se refere ao problema específico de corte ou empacotamento de itens em objetos.

Constata-se que os critérios tipo de associação e variedade de itens menores foram utilizados em conjunto para designar um tipo de problema básico. O critério variedade de objetos maiores resultou em um problema denominado intermediário. As características quanto à dimensão e ao formato de itens deram origem a um tipo de problema denominado como refinado. Este problema possui ainda outras subdivisões no qual restrições adicionais podem ou não ser consideradas (WÄSCHER *et al.*, 2007). A figura 3 apresenta um resumo dos tipos de problemas e os critérios utilizados para a sua categorização:

Figura 3 - Classificação dos tipos de problemas



Fonte: O Autor

Pode-se observar na figura 3 que os critérios foram utilizados individualmente ou combinados para categorizar os tipos de problemas.

O critério tipo de associação apresenta-se dividido em duas situações: saída maximizada ou entrada minimizada. Segundo Wäscher *et al.* (2007), na primeira situação tem-se um grande número de itens que deverão ser associados a uma pequena quantidade de objetos. Como a quantidade de objetos é insuficiente, existe a necessidade de uma seleção dos itens. O problema da mochila (*Knapsack Problem*) exemplifica essa situação. Tem-se um objeto (mochila) e vários itens (peças). O objetivo é maximizar a saída inserindo o maior número de peças na mochila (KATAOKA; YAMADA, 2014).

Na segunda situação, a quantidade de objetos é suficiente para acomodar todos os itens. Sendo assim todos os itens serão associados à menor quantidade de objetos. Um exemplo é o corte de tecidos (*Bin Packing Problem*), no qual tem-se vários objetos (bobinas de tecido) e itens (moldes). Neste caso, o objetivo é obter todos os moldes utilizando a menor largura do rolo de tecido, ou seja, minimizando a entrada (BALDACCI *et al.*, 2014).

Variedade de itens é outro critério abordado e apresenta algumas sub-divisões: itens idênticos (*Identical Small Items*), pouco heterogêneos (*weakly heterogeneous assortment*) e muito heterogêneos (*strongly heterogeneous assortment*). Os itens idênticos apresentam as mesmas dimensões. Pouco heterogêneos permite formar pequenos grupos que apresentam itens com o mesmo formato e tamanho, enquanto muito heterogêneos apresentam a característica de que poucos itens possuem o mesmo formato e tamanho (WÄSCHER *et al.*, 2007).

Outro critério especificado por Wäscher *et al.* (2007), é a variedade de objetos, sendo apresentado com subdivisões: somente um objeto ou vários objetos. No primeiro caso, pode apresentar todas as dimensões fixas ou apresentar uma ou mais dimensões variáveis. No segundo caso, todas as dimensões são fixas. A diversidade está no formato dos objetos, podendo ser: idênticos, pouco ou muito heterogêneos.

Quanto à dimensão, observa-se que podem existir uma, duas ou três, e as dimensões superiores a três são tratadas como problemas variantes.

O último critério se refere ao formato dos itens. Esse critério faz parte do tipo de problema refinado e considera itens com até três dimensões. O formato do conjunto de itens pode ser regular ou irregular. Retângulos, círculos, caixas, cilindros, esferas, dentre outros, são consideradas formas regulares (WÄSCHER *et al.*, 2007).

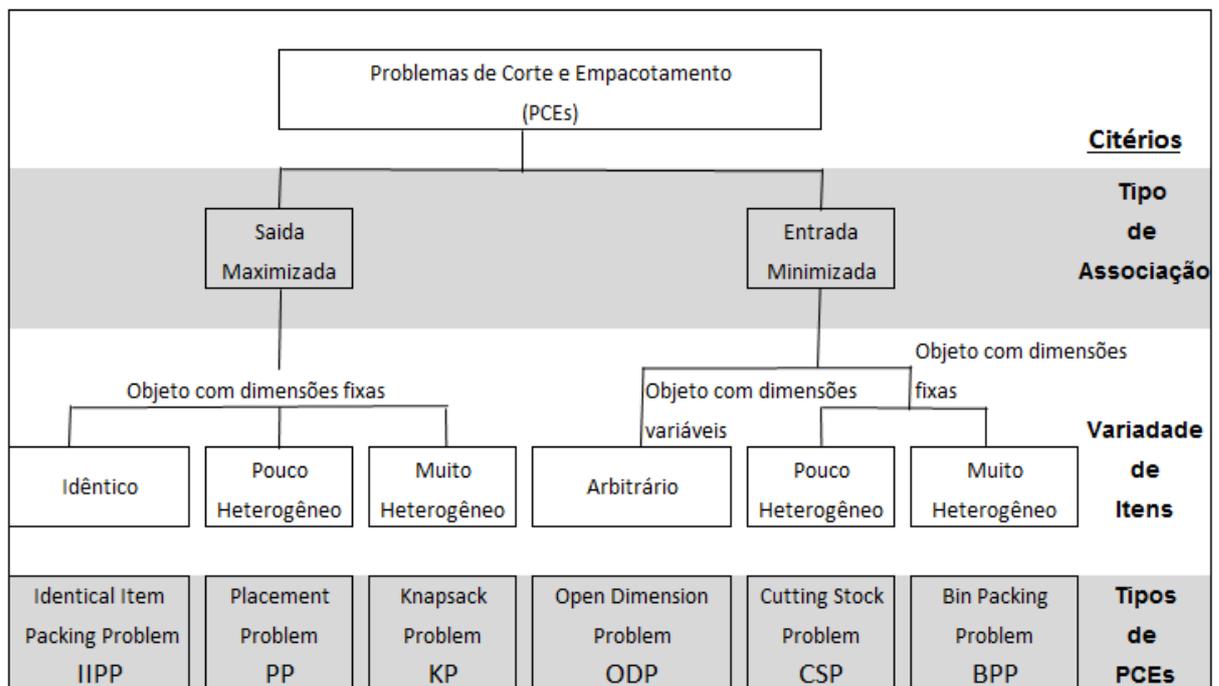
Ressalva-se que para todos os critérios, quando o conjunto de itens ou objetos que estão sendo analisados não se enquadram em alguma categoria, os autores tratam como variantes (*variants*).

2.1.1 Problemas Básicos

Observa-se que as nomenclaturas foram mantidas na língua inglesa por se tratar de termos usuais na literatura.

Os critérios tipo de associação e variedade dos itens foram combinados e deram origem aos tipos de problemas básicos, como ilustrado na figura 4.

Figura 4 - Tipos de problemas básicos



Fonte: Adaptado de Wäscher *et al.* (2007)

Wäscher *et al.* (2007), consideraram o critério tipo de associação com saída maximizada e obtiveram três problemas básicos:

a) *Identical Item Packing Problem (IIPP)*: consiste em atribuir o maior número possível de um conjunto de itens idênticos a um número limitado de objetos. Nesse caso, a otimização está relacionada à melhor organização dos itens em cada objeto conforme a geometria do item.

b) *Placement Problem* (PP): consiste em associar um conjunto de itens pouco heterogêneos a um conjunto limitado de objetos. O objetivo é maximizar o número de itens ou minimizar o desperdício.

c) *Knapsack Problem* (KP): a principal característica que difere este problema do PP é a grande variedade de itens que necessitam ser associados aos objetos.

Ao analisar-se a entrada minimizada, Wäscher *et al.* (2007) verificaram outros três problemas básicos:

a) *Open Dimension Problem* (ODP): consiste em um problema em que um conjunto de itens deve ser totalmente acomodado em um objeto. Um valor referente à uma ou mais dimensões, comprimento, tamanho e volume do objeto são desconhecidos (variáveis). Assim, esse problema consiste em minimizar o valor de entrada destas variáveis.

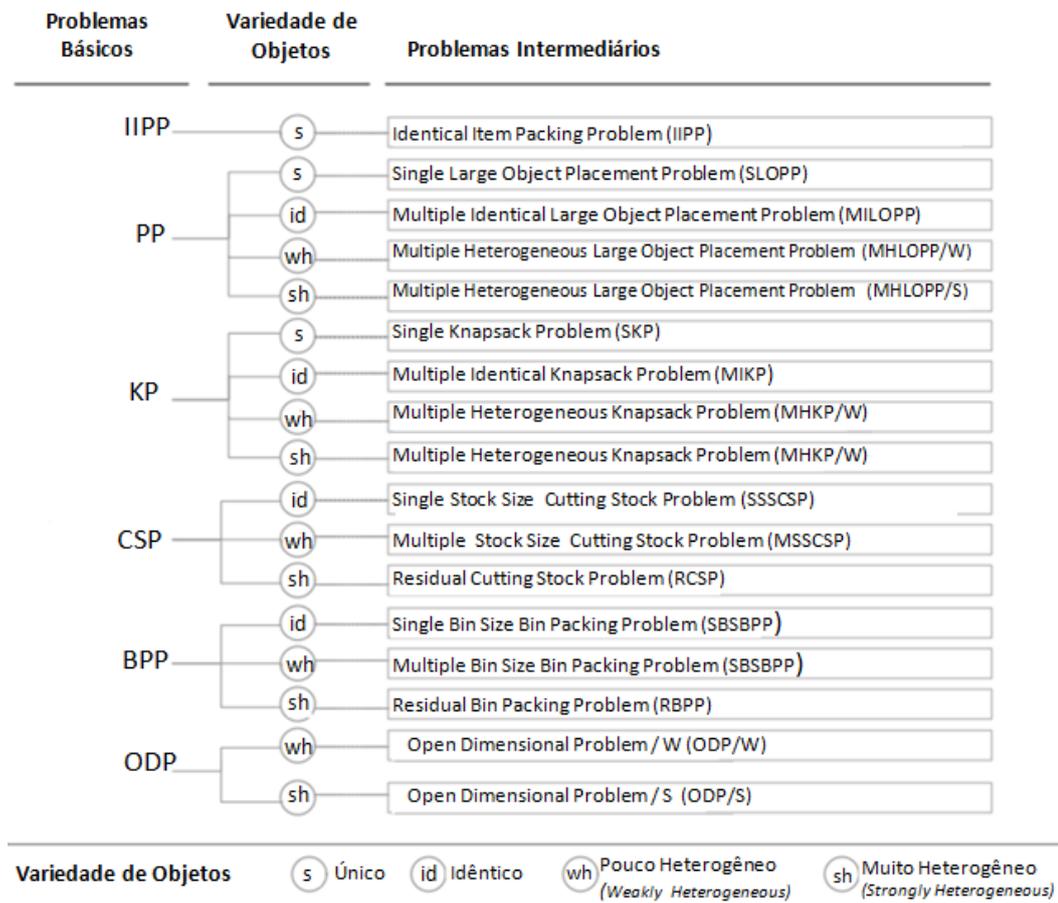
b) *Cutting Stock problem* (CSP): considera-se que a extensão para os objetos é fixa em todas as dimensões. Neste problema requisita-se que um conjunto de itens pouco heterogêneos sejam completamente alocados em objetos devidamente selecionados. O resultado deve ser a minimização de valores, número ou tamanho total dos objetos (desperdício).

c) *Bin Packing Problem* (BPP): no que se refere aos itens são muito heterogêneos e as demais características assemelham-se ao CSP.

2.1.2 Problemas Intermediários

Os tipos de problemas básicos recebem uma nomenclatura adicional quando o critério variedade de objetos é analisado. Segundo Wäscher *et al.* (2007), o critério variedade de objetos é dividido em quatro tipos: objetos únicos (*s*), idênticos (*id*), pouco (*wh*) ou muito heterogêneos (*sh*). Na figura 5 observa-se que adicionando esse novo critério obtém-se os problemas intermediários:

Figura 5 - Relacionamento entre os problemas básicos e intermediários



Fonte: Adaptado de Silva *et al.* (2014)

2.1.3 Problemas Refinados

Os critérios de dimensão (uma, duas ou três) combinados com o formato dos itens promovem a criação dos problemas refinados. Adjetivos são adicionados às nomenclaturas advindas dos problemas intermediários. Um modelo que pode ser usado é descrito por Wäscher *et al.* (2007), sendo a atribuição dos números 1,2,3 para a dimensão seguido do formato geométrico dos itens.

Na literatura encontra-se um problema referenciado como *Nesting Problem* que não faz parte da categorização de Wäscher *et al.* (2007), mas que segundo Bennell e Oliveira (2008) e Alvarez-Valdez *et al.* (2013), se refere a um problema refinado 2D irregular ODP.

O problema abordado neste trabalho pode ser classificado como um problema básico e intermediário do tipo *Identical Item Packing Problem* (IIPP) e como um

problema refinado do tipo 2D irregular IIPP. Sendo assim, ele pode ser referenciado como 2D-I-IIPP.

2.2 ALGORITMOS DE OTIMIZAÇÃO COMBINATÓRIA

Segundo Weise (2008), os algoritmos de otimização geralmente podem ser divididos em duas classes: determinísticos e probabilísticos. Para o primeiro tipo, os mesmos dados de entrada sempre produzirão os mesmos resultados de saída visto que o algoritmo executará sempre as mesmas instruções. No segundo tipo uma possível solução muda aleatoriamente. Esses algoritmos possuem a desvantagem de poder produzir resultados diferentes a cada execução, mesmo considerando a mesma entrada de dados (WEISE, 2008).

Normalmente os algoritmos probabilísticos são utilizados para resolução de problemas que possuem alta complexidade ou dimensões do espaço de busca elevados. Entende-se como espaço de busca o conjunto de todas as soluções de um problema (COELHO, 2003).

Branch & Bound, Greedy Search e Busca A* são alguns dos exemplos de algoritmos determinísticos, enquanto Monte Carlo, Subida de Encosta (*Hill Climbing*) com reinício aleatório, ILS (*Iterated Local Search*), GRASP (*Greedy Randomized Adaptive Search Procedure*) e AGs são alguns dos exemplos de algoritmos probabilísticos (WEISE, 2008).

Segundo Weise (2008), as heurísticas e meta-heurísticas são algoritmos eficientes de otimização, que encontram boas soluções em espaços de busca com dimensões elevadas.

A palavra *heuristic* tem origem na língua grega e significa descobrir ou encontrar (BARBOSA, 2011). As heurísticas são funções de otimização que avaliam a utilidade de um determinado elemento e auxiliam na tomada de decisão sobre qual elemento, de um conjunto de possíveis soluções, será examinado em seguida (WEISE, 2008). Deste modo, as heurísticas podem ser utilizadas como medidas aproximadas para avaliar a qualidade de uma solução candidata. Em adição, elas podem fornecer informações para direcionar uma determinada pesquisa no sentido de encontrar o seu objetivo.

O algoritmo subida de encosta (*Hill-Climbing*) é uma das técnicas de busca local mais básicas. Em cada passo do algoritmo, a solução associada ao estado corrente é substituída pela solução representada pelo melhor vizinho. O algoritmo encerra quando alcança um pico, ou seja, quando não encontra uma solução vizinha melhor que a solução representada pelo estado corrente (RICH; KNIGHT, 1994; RUSSEL; NORVIG, 2004). Analogamente, em um problema de minimização o algoritmo encerra quando um vale é encontrado. Neste caso, segundo Caudill e Butler (2000), ele é denominado descida de encosta (*Hill-Descent*) ou DownHill segundo Raska e Ulrych (2014).

As meta-heurísticas são métodos utilizados para resolver uma ampla classe de problemas. Muitas meta-heurísticas são inspiradas em fenômenos naturais da natureza ou processos físicos. Elas combinam medidas como funções objetivo, funções de aptidão ou heurísticas (WEISE, 2008).

A função de aptidão deve ser idealizada para cada problema a ser resolvido. Esta deverá retornar um valor numérico que seja proporcional a uma habilidade ou utilidade do indivíduo representado (BEASLEY *et al.*, 1993). A função objetivo determina a qualidade da solução e consiste em maximizar ou minimizar uma determinada função matemática (BELFIORE; FÁVERO, 2013).

No grupo das meta-heurísticas, destaca-se a Computação Evolucionária ou Evolutiva (CE), que surgiu nos anos 1950 e permaneceu inexplorada por aproximadamente três décadas. Entre os anos 1980 e 1990, os avanços no desempenho das plataformas computacionais propiciaram a aplicação da CE e o número de publicações acerca dessa temática tem aumentado significativamente (COELHO, 2003).

Swarm Intelligence (SI) e Algoritmos Evolucionários (AEs) fazem parte da CE. O SI se inspira no comportamento de formigas, peixes ou pássaros e copia da natureza a forma como esses agentes traçam pequenas rotas eficientemente, na busca pelos alimentos. Os AEs copiam o processo da evolução natural, tratando soluções candidatas de um problema como indivíduos de uma população que competem e se reproduzem em um ambiente virtual. A cada geração esses indivíduos se tornam melhores neste ambiente.

Segundo Soto *et al.* (2013), os AEs são poderosos métodos de otimização, inspirados na evolução biológica como o cruzamento, mutação e seleção. Goldberg

(1989) enfatiza que AEs são técnicas eficientes e robustas para procurar soluções em espaços de busca irregulares de problemas complexos e sem restrição de dimensão.

Dentre algumas vantagens de se utilizar os AEs, Goldberg (1989) cita a necessidade de se conhecer apenas o valor da função objetivo de cada indivíduo da população, os ruídos e descontinuidades nos dados possuem pouco impacto sobre a eficiência, os conceitos são fáceis de serem compreendidos e as transições que ocorrem são probabilísticas. Coelho (2003) ressalva algumas limitações dos AEs, como as diferenças no desempenho que podem ocorrer em cada execução e a dificuldade para a determinação de um ótimo global quando não se utiliza uma metodologia de otimização local.

Os AEs possuem várias abordagens, visto que foram desenvolvidos independentes uns dos outros. Dentre elas estão: AG, Programação Evolutiva, Estratégias Evolutivas e Programação Genética. Soto *et al.* (2013) relatam que o AG é o método mais popular devido ao sucesso na resolução de problemas do mundo real, considerados difíceis. Desta forma, reforça-se a utilização do AG na abordagem proposta neste trabalho. Não obstante, de acordo com a observação de Coelho (2003), é importante que se tenha associado ao AG algum algoritmo de otimização local tal qual o *Hill-Climbing/Hill-Descent*.

2.2.1 Algoritmo Genético (AG)

As primeiras pesquisas com AGs tiveram início nos anos 1950. Na década de 1970 o pesquisador John Henry Holland publicou um livro que foi considerado o ponto inicial dos AGs. Mais tarde, nos anos 1980, outro pesquisador David. E. Goldberg, que era ex-aluno de Holland, apresentou a primeira aplicação industrial e também publicou um livro no qual apresentava o AG de forma mais detalhada. Estes fatos fortaleceram a utilização dos AGs para a resolução de problemas de otimização e aprendizado de máquinas.

O AG foi inspirado nas teorias de Darwin e Mendel. Para Darwin a evolução biológica ocorre por meio de um mecanismo de seleção natural no qual os indivíduos mais bem adaptados ao ambiente possuem maior probabilidade de reprodução e longevidade. Observa-se que esse mecanismo corresponde ao operador de seleção e à função de aptidão (*fitness*) do AG que será tratado adiante. Mendel defende a teoria da hereditariedade e dominância que ocorre por meio dos genes, teoria

devidamente comprovada. A troca de material genético (hereditariedade) acontece por meio de cruzamentos e mutações dos genes (COELHO, 2003).

Para Koza (1997), o AG tenta encontrar uma boa solução, ou a melhor, de forma genética, por meio da criação de populações de indivíduos com várias séries de gerações. Cada indivíduo desta população representa uma solução candidata para um determinado problema. O AG transforma uma população de indivíduos, conforme um valor de aptidão, em uma nova geração utilizando a seleção, cruzamento e mutação.

Dentre as bibliotecas disponíveis para implementar o AG está a GAlib. Ela disponibiliza quatro tipos de AG: *SimpleGa* (SGA), *Steady-State* (SST), *IncrementalGa* (IGA) e *DemeGa* (DGA). O SGA troca toda a população em cada geração e possui a opção de elitismo, o SST substitui uma parte da população por meio de um percentual de substituição informado via parâmetro, o IGA permite personalizar o método de substituição definindo-se como deve ser cada novo integrante da população, enquanto o DGA manipula múltiplas populações em paralelo migrando indivíduos entre as populações (WALL, 1996).

2.2.1.1 Representação

Os AGs são inspirados na genética e na evolução, sendo assim, os termos utilizados na programação do AG são muito parecidos com os termos da biologia. A tabela 1 apresenta uma analogia entre os termos utilizados na linguagem natural da biologia e os termos utilizados em um AG.

Tabela 1 - Termos relacionados na biologia e AG.

Linguagem Natural	AG
Cromossomo	Indivíduo, string de bits
Gene	Características
Alelo	Valor
<i>Locus</i>	Posição
Genótipo	Estrutura
Fenótipo	Conjunto de parâmetros

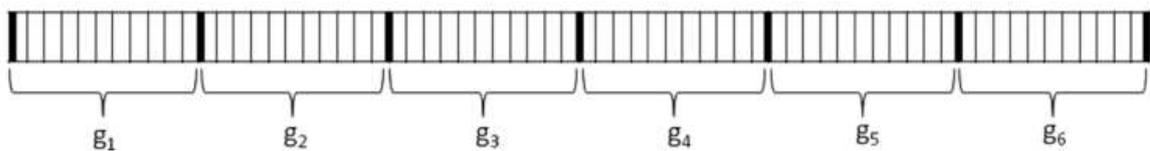
Fonte: Adaptado de Linden (2008).

Conforme observado na tabela 1, um cromossomo representa um indivíduo de uma população. Normalmente este indivíduo é implementado no AG como uma estrutura de dados do tipo *string* de *bits*, a qual representa uma solução candidata que é descrita por Mitchell (1997) como uma hipótese.

Um cromossomo é formado por genes que representam características do indivíduo. Na resolução de um problema, um gene representa uma variável de decisão. Os possíveis valores que podem ser atribuídos ao gene são denominados alelos. A posição de cada gene no cromossomo é denominada *locus* e a estrutura de um cromossomo é denominada genótipo. O fenótipo corresponde ao conjunto de parâmetros do algoritmo decodificados (LINDEN, 2008).

Para representar um indivíduo, estão disponíveis várias estruturas de dados como: números reais, inteiros, vetores, *strings* e árvores. Cada problema a ser resolvido possui a sua própria estrutura de dados. Essa representação (denominada representação cromossômica) deve ser simples mas completamente expressiva (WALL, 1996). A figura 6 mostra um exemplo de representação cromossômica de um indivíduo (cromossomo):

Figura 6 - Representação cromossômica de um indivíduo



Fonte: Librantz *et al.* (2011).

O cromossomo ilustrado na figura 6 emprega uma estrutura do tipo *string* de *bits* com 66 alelos divididos em 6 genes, cada um representando uma variável do problema a ser resolvido.

Linden (2008) destaca que a representação cromossômica é fundamental para o AG, visto que consiste na tradução do problema real em uma maneira viável que o computador possa tratar. Para Mitchell (1997) as hipóteses são representadas frequentemente por *string* de *bits* e desta forma podem ser manipuladas pelos operadores genéticos. Weise (2008) considera uma população de *string* de *bits*, que são alteradas por meio de operadores genéticos, como AGs canônicos.

Segundo Coelho (2003), os operadores usualmente empregados em AGs são: seleção, cruzamento e mutação, os quais são apresentados a seguir.

2.2.1.2 Seleção

Para Weise (2008), a seleção em AEs consiste em escolher indivíduos da população atual de acordo com o seu valor de aptidão (*fitness*). Segundo Mitchell (1996), a função de aptidão define um critério para ordenar potenciais hipóteses e, por meio das seleções probabilísticas, essas hipóteses podem ser incluídas na próxima geração. Mitchell (1997) define seleção como a maneira de escolher indivíduos em uma população com o objetivo de gerar descendentes para a próxima geração e como esses descendentes serão criados. Ou seja, o propósito da seleção é filtrar indivíduos em uma população e esperar que seus descendentes, por sua vez, tenham ainda maiores valores de aptidão. Coelho (2003) entende que o objetivo do operador de seleção é enfatizar as melhores soluções que constituem uma população, selecionando as soluções relativamente aptas e removendo as soluções remanescentes.

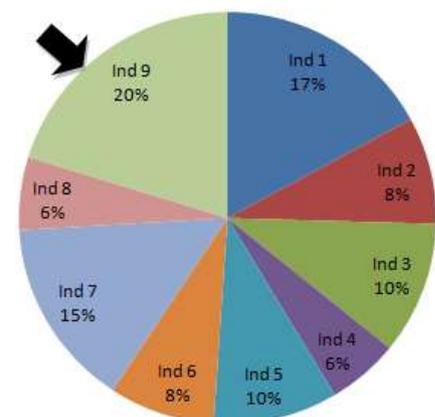
Vários métodos de seleção têm sido apresentados na literatura, sendo os mais comuns: seleção proporcional (*Roulette Wheel Selection*), seleção por torneio, elitismo e seleção por *ranking*.

A seleção proporcional foi o método aplicado no AG original, introduzido por Holland, sendo assim é considerado o esquema de seleção mais antigo (WEISE, 2008). No quadro 1 e figura 7 pode-se observar como funciona esse método de seleção:

Quadro 1 - Exemplo de seleção proporcional. Figura 7 - Gráfico do quadro 1.

Função aptidão: $f(x) = \sin(x) + \sqrt{x}$						
População - P		Fenótipo			Prob.	Ind.
		x	f(x)			
1	0	0	0	8	3,82	17% 1
0	0	1	1	3	1,87	8% 2
0	0	1	0	2	2,32	10% 3
0	1	0	0	4	1,24	6% 4
0	1	1	0	6	2,17	10% 5
0	0	0	1	1	1,84	8% 6
0	1	1	1	7	3,30	15% 7
0	1	0	1	5	1,28	6% 8
1	1	1	1	15	4,52	20% 9
Total:					22,37	100%

Fonte: O autor



Fonte: O autor

O valor da aptidão é calculado para cada indivíduo, por intermédio de uma função de aptidão (*fitness*). Por meio deste cálculo, obtém-se o valor de aptidão individual e o acumulado da população. No exemplo, demonstrado no quadro 1, o objetivo é a maximização do valor de aptidão. Para calcular o valor da probabilidade para que o indivíduo seja selecionado, basta dividir o valor da aptidão individual pelo valor da aptidão da população. Cada indivíduo é associado a uma fatia da roleta. O tamanho de cada fatia é proporcional ao seu valor de aptidão. Observa-se que o indivíduo nove possui a maior probabilidade de ser escolhido, visto que possui a maior fatia da roleta. Observa-se que um mesmo indivíduo poderá ser selecionado mais de uma vez.

Segundo Mitchell (1996), a roleta se moverá de acordo com a quantidade de indivíduos da população. O indivíduo que estiver na fatia selecionada fará parte do novo grupo de pais para a próxima geração.

O ato de mover a roleta é descrito por Linden (2008) como um processo de geração de números aleatórios. O número escolhido pode estar entre 0 e 100 que corresponde ao percentual de probabilidade, entre 0° e 360° relativos aos ângulos do círculo ou ainda entre 0 e a soma total das aptidões.

A seleção por *ranking* é uma variante da seleção proporcional, criada com o propósito de prevenir convergências rápidas e a dominância de um superindivíduo (MITCHELL, 1996). Esse método requer somente o valor da função de aptidão para mapear as soluções em um conjunto parcialmente ordenado (COELHO, 2003). O método consiste em ordenar todos os indivíduos de acordo com a sua função de aptidão e utilizar este ranking como base para a seleção, ao invés de utilizar-se diretamente o valor da função de aptidão (LINDEN, 2008). Desta forma evita-se que um superindivíduo domine as futuras gerações desta população.

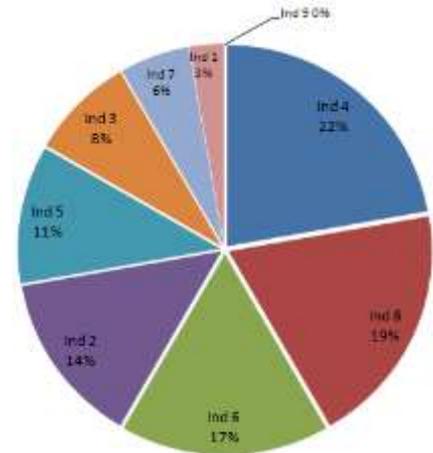
Sendo assim, a probabilidade de um indivíduo ser selecionado é proporcional à sua posição (*rank*) em uma lista ordenada com todos os indivíduos de uma população (WEISE, 2008). Cada indivíduo é posicionado em ordem crescente pelo valor da sua aptidão. Essa posição determina a probabilidade de seleção. Atribui-se 1 (um) para a pior solução ou menos apto e p (quantidade total de Indivíduos) para a melhor solução ou melhor adaptado, desta forma, ocorre uma ordenação do pior para o melhor (MITCHELL, 1996). No quadro 2 e figura 8, pode-se observar como ocorre esta ordenação pelo valor da aptidão, a atribuição do *rank* e o cálculo da probabilidade.

Quadro 2 - Exemplo de seleção por ranking.

Fenótipo		Escala			
x	f(x)	Indivíduo	Rank	Norm. 2.0	Probabilidade
4	1,24	4	1	2,00	22%
5	1,28	8	2	1,75	19%
1	1,84	6	3	1,50	17%
3	1,87	2	4	1,25	14%
6	2,17	5	5	1,00	11%
2	2,32	3	6	0,75	8%
7	3,30	7	7	0,50	6%
8	3,82	1	8	0,25	3%
15	4,52	9	9	0,00	0%
Total					100%

Fonte: O autor

Figura 8 - Gráfico do quadro 2



Fonte: O autor

No método anterior, o indivíduo 8 possuía uma probabilidade de 6% de ser escolhido. Neste método, o mesmo indivíduo possui probabilidade de 19% de ser selecionado. Isto ocorre porque ele ocupa a segunda posição no *rank* possuindo maior probabilidade de ser selecionado.

Sendo assim, pode-se observar que esse método permite que outros indivíduos tenham maiores chances de ser selecionados, não permitindo que o melhor indivíduo tenha a dominância da seleção. Os indivíduos menos aptos podem apresentar características genéticas capazes de encontrar a melhor solução e desta forma não podem ser descartados. Vale destacar que o melhor indivíduo fará parte da próxima geração por meio do elitismo.

Segundo Mitchell (1996), o elitismo é um complemento para muitos métodos de seleção porque obriga o AG a reter um certo número de melhores indivíduos em cada geração. Sendo assim, o elitismo copia o melhor indivíduo para a próxima geração garantindo a sua sobrevivência para as sucessivas gerações. Como no exemplo da seleção por ranking, o melhor indivíduo seria perdido se não fosse selecionado para reproduzir. Alguns pesquisadores descobriram que o elitismo melhora de forma significativa a performance do AG (MITCHELL, 1996).

Weise (2008) considera a seleção por torneio um dos mais populares e eficientes esquemas de seleção. Neste método, dois indivíduos são selecionados aleatoriamente da população e comparados. Aquele que possuir o maior valor de aptidão fará parte do novo grupo de pais. Goldberg e Deb (1991) enfatizam que este método pode ser implementado com processamento paralelo, visto que não há

necessidade dos dados globais para efetuar os cálculos. O quadro 3 ilustra o funcionamento deste método:

Quadro 3 - Exemplo de seleção por torneio.

População - P	Fenótipo		Indivíduo	Torneio		Pais Selecionados
	x	f(x)				
1 0 0 0	8	3,82	1	Ind 1	Ind 4	Ind 1
0 0 1 1	3	1,87	2	Ind 4	Ind 3	Ind 3
0 0 1 0	2	2,32	3	Ind 3	Ind 7	Ind 7
0 1 0 0	4	1,24	4	Ind 4	Ind 4	Ind 4
0 1 1 0	6	2,17	5	Ind 9	Ind 2	Ind 9
0 0 0 1	1	1,84	6	Ind 6	Ind 4	Ind 6
0 1 1 1	7	3,30	7	Ind 6	Ind 5	Ind 5
0 1 0 1	5	1,28	8	Ind 8	Ind 1	Ind 1
1 1 1 1	15	4,52	9	Ind 2	Ind 5	Ind 5

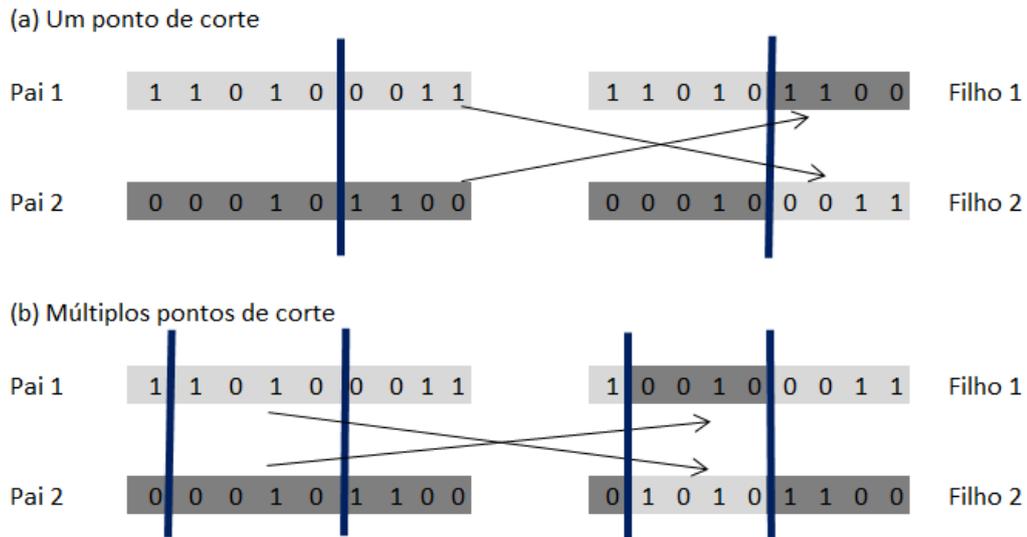
Fonte: O Autor

Neste exemplo o torneio é realizado entre dois indivíduos (binário). Os indivíduos foram selecionados de forma aleatória. Na coluna torneio, o indivíduo com a cor cinza representa o vencedor. Observa-se que o indivíduo quatro é o menos apto, mas participou do torneio várias vezes, competindo inclusive com ele mesmo em um torneio. Os vencedores do torneio farão parte do novo grupo de pais.

2.2.1.3 Cruzamento

Crossover, cruzamento ou recombinação são termos comumente encontrados na literatura. Segundo Coelho (2003), é o principal operador do AG e consiste na troca de partes dos cromossomos entre os indivíduos. Este operador escolhe aleatoriamente um *locus* e copia as partes de dois cromossomos pais formando um par de cromossomos filhos (MITCHELL, 1996).

Pode haver um, dois ou múltiplos pontos de corte. Na figura 9 pode-se observar como ocorre a troca de material genético entre dois cromossomos pais e a geração de dois indivíduos filhos.

Figura 9 - Exemplos de cruzamento.

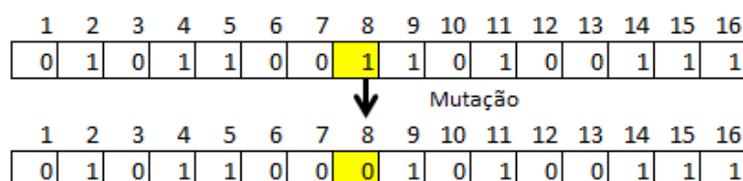
Fonte: Adaptado de Coelho (2003)

2.2.1.4 Mutaç o

Para Weise (2008), o operador de muta o   extremamente importante para preservar a diversidade da popula o por meio de pequenas mudan as. Para Coelho (2003), este operador previne converg ncias prematuras para  timos locais, por meio de amostragens de novos pontos de busca.

Segundo Wall (1996), converg ncia refere-se   similaridade nos valores da fun o objetivo e pode, ainda, ser utilizada como um crit rio de parada. Na biblioteca GALib, observa-se que as medidas de converg ncia utilizam a aptid o do melhor indiv duo. A converg ncia ocorre quando o melhor indiv duo das i - simas gera es anteriores possui a mesma aptid o do melhor indiv duo da gera o atual.

Em *string* de *bits*, a muta o ocorre com a simples invers o do valor de um bit (alelo) conhecido como *bit-flip mutation*. Na figura 10 pode-se observar a aplica o do operador de muta o. Neste exemplo, o *locus* 8 foi escolhido aleatoriamente e o gene teve seu alelo alterado de 1 para 0.

Figura 10 - Operador de Muta o.

Fonte: O autor

Neste trabalho empregou-se como operadores do AG a mutação, cruzamento de um ponto e a seleção proporcional. Para determinar o tipo de AG, daqueles presentes na biblioteca GAlib, foram efetuados vários experimentos entre o SST e o SGA e optou-se por utilizar o SST, devido aos melhores resultados obtidos.

2.2.1.5 Pseudocódigo básico do AG

Mitchell (1997) menciona que existem várias implementações do AG que apresentam pequenas diferenças, mas a estrutura básica segue o padrão ilustrado na figura 11. Nesta estrutura, pode-se observar o emprego dos operadores de mutação, cruzamento e seleção.

Figura 11 - Estrutura básica do AG.

<p>Variáveis</p> <p>POP: População</p> <p>Ps: População temporária</p> <p>h: Indivíduo de uma população</p> <p>hi : i-ésimo indivíduo</p> <p>Parâmetros</p> <p>Fitness : Função que avalia e pontua a aptidão de um indivíduo</p> <p>Fitness_threshold : Valor limite utilizado como critério de parada</p> <p>p : Quantidade de indivíduos de uma população</p> <p>pRepl : Percentual da população que será substituída a cada geração</p> <p>pCross : Percentual da população selecionada para cruzamento</p> <p>m : Percentual da população que sofrerá mutação;</p> <p>Algoritmo GA (Fitness, Fitness_threshold, p ,pRepl ,pCross,m)</p> <p>Início</p> <ul style="list-style-type: none"> • Inicializa a população POP { POP ← Gerar p indivíduos randomicamente } • Avaliar a aptidão de cada indivíduo h em POP { Execute a função Fitness(h)} • Enquanto (maior aptidão de um indivíduo h) < (Fitness_threshold) faça <ul style="list-style-type: none"> • Criar uma nova geração Ps: <ol style="list-style-type: none"> 1- Seleção : Probabilisticamente selecionar (1-pRepl)p indivíduos de POP e adicionar a Ps. A probabilidade Pr(hi) para selecionar um indivíduo hi da população POP é obtida pela fórmula: $Pr(h_i) = \frac{Fitness(h_i)}{\sum_{j=1}^p Fitness(h_j)}$ (Utiliza o valor relativo da aptidão em relação ao valor total da aptidão da população) 2- Cruzamento : Selecionar probabilisticamente (pCross*p)/2 pares de indivíduos de POP. Utilizar a fórmula Pr(hi) Para cada par (h1,h2) produzir dois novos indivíduos Adicionar todos os novos indivíduos para Ps. 3- Mutação: Escolher m percentual de indivíduos de Ps com probabilidade uniforme. Para cada indivíduo selecionado, inverter randomicamente um bit em sua representação. 4- Sobrepor a população POP {POP ← Ps} 5- Avaliar a aptidão de cada indivíduo h em POP { Execute a função Fitness(h)} • Retornar o indivíduo de POP com a maior aptidão. <p>Fim</p>

Fonte: Adaptado de Mitchell (1997).

O algoritmo inicia com a criação da população inicial (geração 0). O valor do parâmetro p determina a quantidade de indivíduos que serão criados randomicamente. Para gerar esses números randômicos faz-se necessária uma semente, a qual depende do gerador utilizado. A biblioteca GALib possui uma função que gera a semente inicial por meio da multiplicação da hora atual pelo número do processo (ID) do computador. Quando o computador não possui ID utiliza-se apenas a hora (WALL, 1996).

Além da geração randômica dos indivíduos, é possível inicializar uma população com indivíduos factíveis, ou seja, com soluções geradas previamente por um outro algoritmo. Essa técnica é conhecida como *seeding* e tem por objetivo acelerar o AG na busca pela melhor solução (OMAN, 2001).

Para avaliar como está a adaptação do indivíduo em relação ao problema, utiliza-se a função de aptidão. Um indivíduo bem adaptado é aquele que possui alta probabilidade de seleção (COELHO, 2003). No exemplo da figura 11, a função de aptidão é denominada *Fitness*.

Uma vez conhecido o valor de aptidão de cada indivíduo, faz-se necessário selecionar quais os indivíduos que farão parte da próxima geração. Os indivíduos são selecionados conforme o método de seleção escolhido e conforme o valor do parâmetro $pRepl$, que determina a quantidade de indivíduos que serão substituídos.

O próximo passo é selecionar os indivíduos, de acordo com o método de seleção desejado, para formar o grupos dos pais. Observa-se que, no exemplo da figura 11, empregou-se a seleção proporcional.

No operador de cruzamento, pares de pais são selecionados e geram novos descendentes. Os novos indivíduos são adicionados a uma população temporária. Essa população temporária sofrerá a ação do operador de mutação.

A população atual (geração 0) será substituída pela população temporária, iniciando-se desta forma uma nova geração (geração 1). O ciclo se reinicia com cálculo da aptidão de cada indivíduo.

Esse processo acontecerá até que um critério de parada seja satisfeito. Nesta estrutura, utilizou-se o valor do parâmetro *Fitness_threshold*, sendo assim o programa será executado até que o valor de aptidão seja maior que esse parâmetro.

A biblioteca GALib possui dois critérios de parada: quando se atinge um certo número de gerações ou quando se alcança uma determinada porcentagem de convergência (WALL, 1996).

2.2.1.6 Parâmetros do AG

A figura 11 demonstra a estrutura padrão de um AG. Pode-se observar que o AG utiliza alguns parâmetros numéricos. Esses valores representam o tamanho da população p , o percentual de cruzamento $pCross$ e o percentual da taxa de mutação m . O ajuste desses parâmetros é de extrema importância, visto que influenciam diretamente na qualidade e no tempo de processamento (COELHO *et al.*, 2015).

Mitchell (1997) apresenta alguns valores típicos, utilizados em seus experimentos, exibidos na tabela 2. Ele observa que esses valores dependem da tarefa de aprendizagem a ser executada.

Tabela 2 - Valores padrão para os parâmetros.

Parâmetros	Valores
Taxa de cruzamento ($pCross$)	0,6
Taxa de mutação (m)	0,001
Tamanho da população (p)	100 a 1000

Fonte: Mitchell (1997)

Devido ao impacto que esses parâmetros podem ter sobre os resultados do AG, alguns pesquisadores têm apresentado estudos relevantes. Após a execução de vários experimentos, para um problema específico, COELHO *et al.* (2015), sugeriram os valores apresentados na tabela 3:

Tabela 3 - Sugestão de valores dos parâmetros.

Parâmetros	Valores
Taxa de cruzamento ($pCross$)	0,55
Taxa de mutação (m)	0,01
Tamanho da população (p)	1000

Fonte: Coelho *et al.* (2015).

Pinho *et al.* (2007), apresentaram um trabalho na qual a técnica de planejamento de experimentos é empregada para obter os valores dos parâmetros do AG. Rosa (2011) obteve os valores dos parâmetros por meio da execução de uma série de dez simulações com a variação de apenas um parâmetro e mantendo os

outros fixos. A análise, para a definição dos parâmetros, ocorreu sobre a média de dez simulações para cada experimento.

Segundo Coelho (2003), a escolha da probabilidade de cruzamento e probabilidade de mutação é um problema complexo de otimização não linear. De acordo com o autor, não existe uma regra única para determinar o tamanho da população e os valores dos parâmetros. Coelho (2003) apresenta os valores padrão encontrados na literatura conforme a tabela 4:

Tabela 4 - Valores padrão da literatura.

Parâmetros	Valores
Taxa de cruzamento (pCross)	0,5 a 1.0
Taxa de mutação (m)	0,001 a 0,05
Tamanho da população (p)	30 a 200

Fonte: Coelho (2003)

Em resumo, pode-se entender que a determinação dos valores dos parâmetros é um fator que merece toda a atenção do pesquisador. Observa-se que os valores apresentados nas tabelas 2 a 4 não seguem um padrão. Desta forma, considera-se que o pesquisador deve obter os valores dos parâmetros empregando-se algum método confiável e que seja específico para o problema tratado. Neste trabalho é empregado o planejamento fatorial na determinação dos valores dos parâmetros supracitados para a execução dos experimentos.

2.3 PROCESSAMENTO DE IMAGENS DIGITAIS

De acordo com Pedrini e Schwartz (2008), o Processamento de Imagens Digitais consiste em um conjunto de técnicas para capturar, representar e transformar imagens com auxílio do computador. Ainda de acordo com os autores, tais técnicas permitem extrair e identificar informações das imagens, além de prover melhoria de qualidade visual e interpretação automática de seus conteúdos.

Yadav e Yadav (2009) definem uma imagem como uma função bidimensional $f(x,y)$ em que x e y são coordenadas espaciais em um determinado plano. Uma imagem é considerada digital quando (x,y) e os valores de amplitude de f são finitos em quantidades discretas. A amplitude de f no ponto (x,y) , denominado *pixel*, é

chamada de intensidade ou escala de cinza da imagem neste ponto, sendo normalmente representada como potência de 2.

Uma imagem digital é então representada por uma matriz de dimensões $M \times N$, em que M representa as linhas e N as colunas. As coordenadas de origem são $(0,0)$ e as coordenadas da primeira linha são: $(0,0)$, $(0,1)$, $(0,2)$, ..., $(0,N-1)$. Cada *pixel* está localizado em uma coordenada (x,y) , como ilustra a figura 12:

Figura 12 - Representação matricial de uma imagem digital.

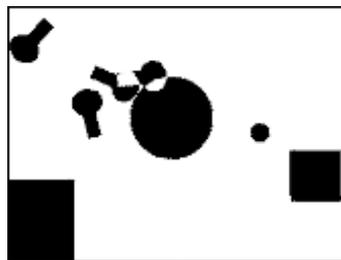
$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Fonte: (YADAV; YADAV, 2009).

As imagens podem ser classificadas em quatro tipos: preto e branco ou imagem binária (*binary*), níveis de cinzas (*grayscale*), imagens coloridas e imagens multiespectrais (GONZALEZ; WOODS, 2002).

Nas imagens binárias os pixels podem assumir apenas dois valores 0 (zero) ou 1 (um), sendo o valor 0 associado ao preto e o valor 1 ao branco. A figura 13 apresenta um exemplo deste tipo de imagem.

Figura 13 - Exemplo de imagem binária.



Fonte: O autor

Em imagens em níveis de cinzas cada pixel normalmente ocupa um *byte* (8 *bits*) que representa a variação do brilho de 0 (preto) até 255 (branco), como pode ser visto na figura 14.

Figura 14 - Exemplo de imagem em níveis de cinzas.



Fonte: (GONZALEZ; WOODS, 2002).

Em uma imagem colorida ou multibanda a cor de um pixel normalmente é representada por um conjunto de três ou quatro valores, dependendo do espaço de cores empregado. Segundo Jayaraman *et al.* (2009), o espaço de cores mais popular é o RGB (**R**ed, **G**reen, **B**lue), que é utilizado para reproduzir as cores em monitores de computador ou televisão e emprega um conjunto de três valores para representação de cada pixel, ou seja, $f(x,y)=(R,G,B)$ na qual R representa o vermelho, G representa o verde e B o azul (GONZALEZ; WOODS, 2002).

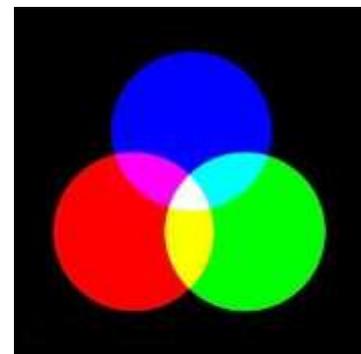
Em resumo, dada uma imagem colorida IC no espaço de cores RGB , esta pode ser representada por três planos independentes IC_R , IC_G e IC_B , sendo que cada um deles pode ser interpretado (isoladamente) como uma imagem em níveis de cinzas. Como pode ser visto na figura 15, as cores são obtidas por meio da mistura das componentes R , G e B , em determinadas quantidades (o valor de cada componente varia de 0 a 255). Na tabela 5 apresenta-se a composição de algumas cores:

Tabela 5 - Composição de algumas cores no espaço RGB.

Cor	Quantidade		
	Vermelho (R)	Verde (G)	Azul (B)
Branco	255	255	255
Preto	0	0	0
Vermelho	255	0	0
Verde	0	255	0
Azul	0	0	255
Amarelo	255	255	0
Magenta	255	0	255
Ciano	0	255	255

Fonte: O Autor

Figura 15 - Composição das cores da tabela 5



Fonte: (PEDRINI e SCHWARTZ, 2008)

Por fim, as imagens multiespectrais são imagens de um mesmo objeto adquiridas em múltiplas bandas que podem representar diferentes grandezas, tais

como frequência, temperatura e pressão (PEDRINI; SCHWARTZ, 2008). Esse tipo de imagem pode ainda conter informações imperceptíveis para o ser humano como: infravermelho, ultravioleta e outras bandas no espectro eletromagnético (JAYARAMAN *et al.*, 2009).

Por fim, cabe ressaltar que o Processamento de Imagens Digitais é utilizado neste trabalho com a finalidade de extrair e identificar informações das imagens dos objetos e dos itens a serem cortados, bem como propiciar a visualização do planejamento do corte e fornecer um valor numérico que indica a qualidade desse planejamento.

2.4 TRABALHOS CORRELATOS

Esta seção tem por objetivo apresentar alguns trabalhos publicados na literatura que tratam dos PCEs. Em princípio, a ideia foi selecionar somente trabalhos que lidam com o 2D-I-IIPP abordado neste trabalho, porém pouco foi encontrado sobre a solução desse problema na literatura. Desta forma, são apresentados, no quadro 4 a seguir, trabalhos abordando métodos e técnicas aplicados às soluções dos diversos PCEs

Quadro 4 - Síntese dos trabalhos abordando soluções para PCEs

Tipo de Problema	Método/Técnica/Algoritmo utilizado	Autor (ano)
2D retangular IIPP	<i>Branch and bound</i>	Birgin e Lobato (2010)
<i>Nesting Problem com itens irregulares</i>	<i>Branch and bound</i>	Alvarez-Valdez <i>et al.</i> (2013)
2D retangular CSP	<i>Mixed-Integer Programming e Branch-and-price</i>	Furini e Malaguti (2013)
<i>2D CSP</i>	<i>Branch-and-price</i>	Malaguti <i>et al.</i> (2013)
2D retangular ASSCSP (<i>Two-Dimensional Arbitrary Stock-Size Cutting Stock Problem</i>)	GBA (<i>General Blocks Patterns Algorithm</i>)	Cui <i>et al.</i> (2014)
<i>Stock Reduction Problem (SRP)</i>	Algoritmo híbrido que combina <i>Linear Programming (LP)</i> e AG	Shen e Zhang (2010)
2D retangular CSP	AG, CSA (<i>Corner Space Algorithm</i>) e <i>Linear Programming (LP)</i>	Lu e Huang (2015)

2D SLOPP e 2D Knapsack Problem	<i>Greedy Algorithms, Hybrid Genetic Algorithm</i>	Hadjiconstantinou e Iori (2007)
1DCSP (One-Dimensional Cutting Stock Problem),	<i>Hybrid Multi-chromosome Genetic Algorithm (HMCGA)</i>	Peng e Chu (2010)
1DCSP (One-Dimensional Cutting Stock Problem)	<i>Fitness Level based Adaptive Operator Selection (FLAOS)</i>	Zhang et al. (2014)
2D Strip Packing Problem	<i>Hiper-Heuristic Genetic Programming</i>	Burke et al. (2010)
2DCSP	<i>Hiper-Heuristic Ant-Q</i>	Khamassi et al. (2011)
2D Strip Nesting Problem	Colônia de formigas e <i>No Fit Polygon (NFP)</i>	Yang (2014)
2D Strip Packing Problem	AG, <i>Greedy Bottom-Left</i> , NFP	Pinheiro et al. (2015)
3D retangular IIPP	Busca tabu simples com heurísticas de blocos	Poli e Pureza (2012), Pureza e Morabito (2006)
3D retangular IIPP	Programação linear inteira, GAMS e CPLEX	Junqueira et al. (2012)
2D Knapsack Problem e 2D CSP.	GRASP, <i>Extended Search Algorithm (ExSearch)</i> , <i>Exact dynamic programming</i> , <i>reduced raster points</i> , NFP, <i>Column Generation Algorithm</i>	Del Vale et al. (2012)
SBSBPP (Single Bin Size Bin Packing Problem)	<i>Forest Tree Search Algorithm (FTSA)</i> , funções phi e GRASP	Han et al. (2013)
2D CSP	<i>Artificial Fish Swarm (AFSA)</i>	Song et al. (2013)
1D CSP	<i>Particle Swarm Optimization (PSO)</i> .	Lagha et al. (2014)

Fonte: O autor

Pode-se perceber que os PCEs são problemas cujas soluções permitem o emprego de várias técnicas computacionais. Diante da quantidade de técnicas ainda não exploradas, acredita-se que seja uma área com grande potencial de crescimento. Vale ressaltar que não foram encontradas evidências da utilização da abordagem proposta neste trabalho em outras pesquisas descritas na literatura. Não obstante, também não foram encontrados trabalhos abordando o 2D-I-IIPP considerando que os objetos (placas) podem conter furos em suas superfícies.

3 MATERIAIS E MÉTODOS

Neste capítulo é realizada uma descrição detalhada das características metodológicas da pesquisa seguida da apresentação das etapas realizadas conforme a ordem de execução. Por fim, são descritos os materiais utilizados no desenvolvimento da abordagem proposta.

3.1 CARACTERIZAÇÃO METODOLÓGICA DA PESQUISA

Este trabalho tem o objetivo de gerar informações e caracteriza-se por seu interesse prático na solução de problemas reais. Sendo assim, para Marconi e Lakatos (2010) pode-se classificá-lo como sendo de natureza aplicada. A abordagem proposta emprega técnicas computacionais para gerar dados numéricos. Sendo assim, do ponto de vista da forma de abordar o problema, essa pesquisa pode ser classificada como axiomática quantitativa normativa, visto que, para Bertrand e Fransoo (2002), produz conhecimento sobre o comportamento de certas variáveis do modelo e os pesquisadores desta linha olham para o problema em questão pelo viés de modelos matemáticos que podem ser analisados a fim de melhorar resultados existentes na literatura.

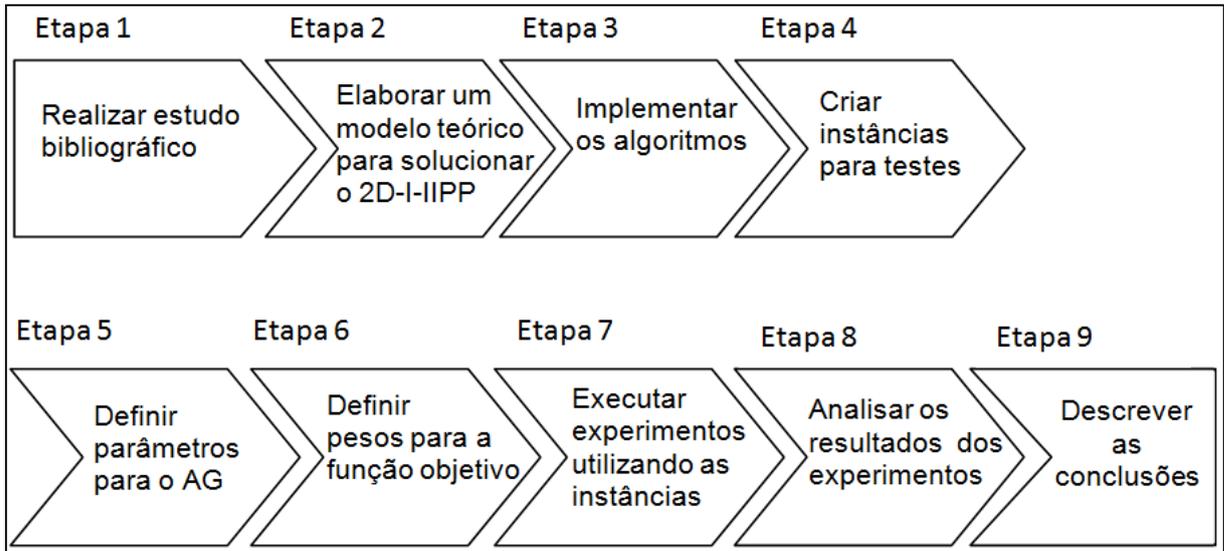
Do ponto de vista dos objetivos, trata-se de uma pesquisa exploratória, visto que para Gil (2010) envolve levantamento bibliográfico e proporciona maior familiaridade com o problema com vistas a torná-lo explícito ou a construir hipóteses.

O método exploratório foi empregado a fim de aumentar a familiaridade do pesquisador com o ambiente (AG, Processamento de Imagens Digitais e PCEs) e o método experimental foi utilizado para determinar os valores dos parâmetros necessários para configurar o AG, bem como os pesos atribuídos aos coeficientes que compõem a função objetivo, além de ser utilizado para demonstrar os resultados da abordagem proposta (MARCONI; LAKATOS, 2010).

3.2 METODOLOGIA EMPREGADA NO DESENVOLVIMENTO E AVALIAÇÃO DA ABORDAGEM PROPOSTA

O desenvolvimento e a avaliação da abordagem proposta foram realizados em 9 etapas, as quais estão apresentadas na figura 16 a seguir.

Figura 16 - Visão geral das etapas metodológicas



Fonte: O autor

Na etapa 1, uma pesquisa bibliográfica foi realizada visando subsidiar a fundamentação teórica e a análise de trabalhos correlatos. As fontes utilizadas foram: artigos científicos oriundos de periódicos e de conferências nacionais e internacionais, dissertações, livros e *web sites*. Deu-se preferência para as publicações com data superior ao ano de 2009 até fevereiro de 2016, mas publicações de anos anteriores também foram consideradas para conceituar e definir termos importantes. As consultas foram feitas principalmente nas bases: Portal Capes, IEEE Explore, ProQuest e Science Direct, levando em consideração os seguintes termos (em português ou inglês, dependendo da base): Problema de corte, Itens irregulares, Itens idênticos, IIPP, Algoritmo Genético e Processamento de imagens. Nas pesquisas bibliográficas realizadas não foram encontrados indícios da utilização de uma abordagem que empregasse AG e Processamento de Imagens Digitais para a resolução de PCEs categorizados como 2D-I-IIPP.

Na etapa 2 foi proposta uma abordagem para a solução do 2D-I-IIPP na qual se empregou o AG em conjunto com uma rotina de Processamento de Imagens

Digitais e, na etapa seguinte, os algoritmos que constituem esta abordagem foram implementados em linguagem C/C++.

Na etapa 4 foram criadas instâncias (exemplares) de objetos com furos e de itens irregulares. Tais instâncias são representadas por imagens, as quais foram utilizadas para executar os experimentos descritos nas etapas seguintes.

Na etapa 5, por meio de um planejamento fatorial, foram estimados os principais parâmetros de configuração do AG.

A fim de definir, exemplificar e identificar a relevância dos pesos dos coeficientes que compõe a função objetivo, na etapa 6, foram executados experimentos no qual cada variável que compõe a função objetivo foi testada isoladamente.

Na etapa 7 foram executados experimentos para avaliar a abordagem proposta, os quais foram divididos em três grupos. No primeiro grupo de experimentos objetivou-se avaliar os resultados dos algoritmos inicializando-se toda a população do AG de forma aleatória. No segundo grupo, injetou-se uma solução inicial factível (SIF) na população inicial do AG e empregou-se a heurística descida de encosta (HDE). Por fim, no terceiro, inicializou-se a população do AG de forma aleatória e empregou-se a HDE. Os resultados desses experimentos são apresentados, no capítulo 5, em forma de imagens e por meio de dados numéricos.

Os resultados obtidos nos experimentos foram discutidos na etapa 8 enquanto as conclusões acerca das discussões foram elaboradas na nona e última etapa deste trabalho.

3.3 MATERIAIS UTILIZADOS NO DESENVOLVIMENTO DA ABORDAGEM PROPOSTA

Os algoritmos desenvolvidos nesse trabalho foram implementados em linguagem de programação C/C++, utilizando o compilador Dev C++. Para manipular a meta-heurística AG foi empregada a biblioteca GALib¹ versão 2.4.7. Já para a manipulação e processamento das imagens digitais utilizou-se a biblioteca PROEIKON (KIM, 2012), que se encontra disponível na internet². As instâncias (imagens) do problema investigado foram criadas arbitrariamente utilizando-se o

¹ <http://lancet.mit.edu/ga/>

² <http://www.lps.usp.br/hae/software/proeikon.html>

software Microsoft Paint. A linguagem de programação, o compilador e as bibliotecas foram escolhidos por dispor de todos os recursos necessários ao desenvolvimento dos algoritmos e também por serem amplamente utilizadas em aplicações correlatas e citadas na literatura.

Considerando que a solução para problema de corte abordado demanda alto custo computacional, os experimentos foram executados em um computador com a seguinte configuração: processador Intel(R) Core(TM) i7-4790 CPU 3.60 GHz, memória RAM 16,0 GB, sistema operacional Windows 10 Pro, 64 bits, disco rígido de 1 TB.

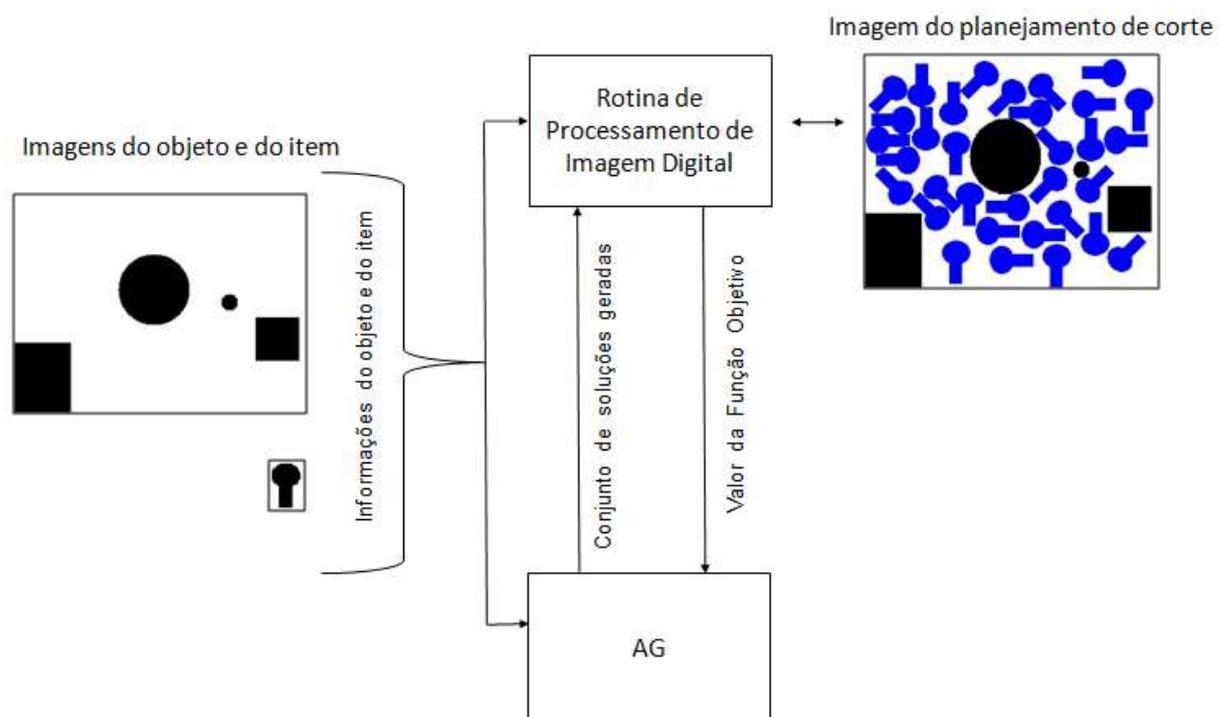
4 ABORDAGEM PROPOSTA

Neste capítulo, primeiro são descritos a estrutura e o funcionamento da abordagem proposta neste trabalho, envolvendo os procedimentos e algoritmos que a compõem. Também neste capítulo demonstra-se como a solução do problema investigado foi codificada (representada) pelo cromossomo do AG e como a função objetivo é calculada a partir de uma rotina de processamento de imagens digitais.

4.1 VISÃO GERAL DA ABORDAGEM PROPOSTA

A abordagem proposta neste trabalho, ilustrada na figura 17, emprega o AG e uma rotina de Processamento de Imagens Digitais para solucionar o problema de cortar itens idênticos com formas irregulares em objetos maiores, classificado como 2D-I-IIPP. Os itens são peças enquanto os objetos podem ser placas, por exemplo de madeira, metal ou plástico. Tanto o objeto quanto o item são representados como imagens digitais binárias. Já a imagem que ilustra a solução do problema, ou seja, o planejamento de corte, é uma imagem colorida no espaço RGB.

Figura 17 – Visão geral do funcionamento da abordagem proposta



Fonte. O autor

Conforme se observa na figura 17, inicialmente algumas informações referentes às imagens do objeto e do item são enviadas ao AG e à rotina de Processamento de Imagens Digitais, para inicialização dos seus parâmetros de configuração.

O AG é o responsável por gerar um conjunto de soluções, baseadas nessas configurações iniciais. Cada possível solução, contém as coordenadas x e y , ângulo de rotação e um status de posicionamento do item sobre o objeto. Esse conjunto de soluções é enviado à rotina de Processamento de Imagens Digitais para que seja avaliado.

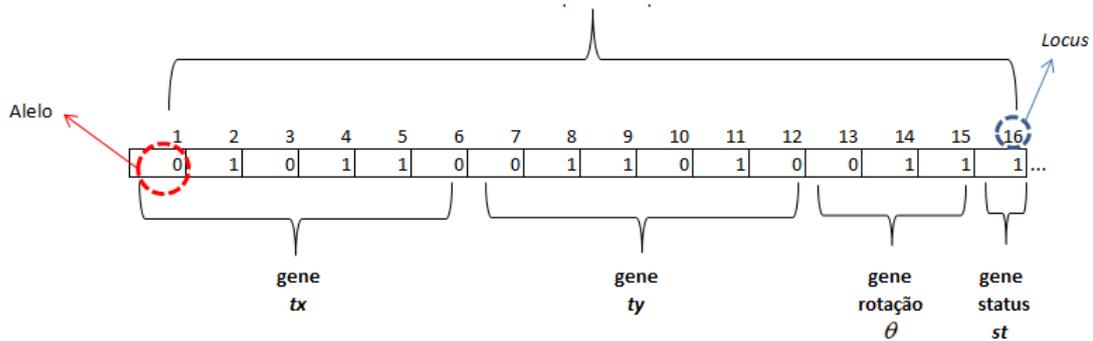
A avaliação de cada solução gerada pelo AG, é realizada por um algoritmo de Processamento de Imagens Digitais que, inicialmente posiciona os itens sobre o objeto, gerando uma imagem de planejamento de corte. Neste momento, também são calculados os coeficientes que compõem a função objetivo.

Para calcular o valor da função objetivo, a rotina de Processamento de Imagens Digitais posiciona os itens na imagem de planejamento de corte e detecta as sobreposições, itens fora das bordas e itens posicionados sobre os furos. Além disso, ainda ocorre o computo da quantidade de itens posicionados para corte e o cálculo da distância entre esses itens. Esses valores numéricos compõem a função objetivo e indicam ao AG a qualidade da solução gerada. Por meio dos operadores de seleção, cruzamento e mutação o AG evolui suas soluções até encontrar o ponto de parada.

4.2 REPRESENTAÇÃO CROMOSSÔMICA EMPREGADA NO AG

A fim de encontrar uma solução para o problema descrito, por meio da implementação do AG, faz-se necessário transformar as variáveis deste problema para uma forma codificada. Para tanto, adotou-se a representação cromossômica ilustrada de forma resumida na figura 18.

Figura 18 - Parte do cromossomo representando um item da solução



Fonte. O autor

Na figura 18 é possível identificar como os 4 genes (tx , ty , θ e st), que representam apenas um item, estão distribuídos ao longo do cromossomo. Nesta representação, o cromossomo ou indivíduo é representado por uma *string* de *bits*.

Observa-se que o conjunto de alelos entre as posições 1 e 6 representam a translação do item com relação ao eixo x (tx). O conjunto de alelos entre as posições 7 e 12 representam a translação do item com relação ao eixo y (ty). Os valores tx e ty correspondem às coordenadas e são utilizados para posicionar um item no objeto. O conjunto de alelos entre as posições 13 e 15 representam o ângulo de rotação do item (θ). E o gene que está na posição 16 representa o status (st), que indica se o item será posicionado ou não no objeto.

Percebe-se que são necessários 16 *bits* para representar um item no cromossomo, sendo 6 *bits* para cada tx e ty , 3 *bits* para rotação (θ) e 1 bit para status (st). Para calcular a quantidade ideal de *bits* para cada gene, utilizaram-se as dimensões fixas da imagem do objeto (W e H) e as variáveis dx , dy e $d\theta$.

As variáveis dx e dy representam os passos utilizados para as translações nos eixos x (colunas) e y (linhas). Já a variável $d\theta$ representa o passo do ângulo de rotação do item. Essas variáveis foram criadas a fim de diminuir o espaço de busca e definidas com os valores $dx=5$, $dy=4$ e $d\theta=45$. Com base nessas variáveis é possível identificar os valores máximos de armazenamento nos genes e definir a quantidade de *bits* necessária.

As equações 1, 2 e 3 descrevem como identificar os limites superiores do ângulo de rotação (θ) e das coordenadas tx e ty :

$$Max_tx = \frac{W}{dx} - 1, \text{ na qual } W = 320 \quad (1)$$

$$Max_ty = \frac{H}{dy} - 1, \text{ na qual } H = 256 \quad (2)$$

$$Max_rot = \frac{360}{d\Theta} \quad (3)$$

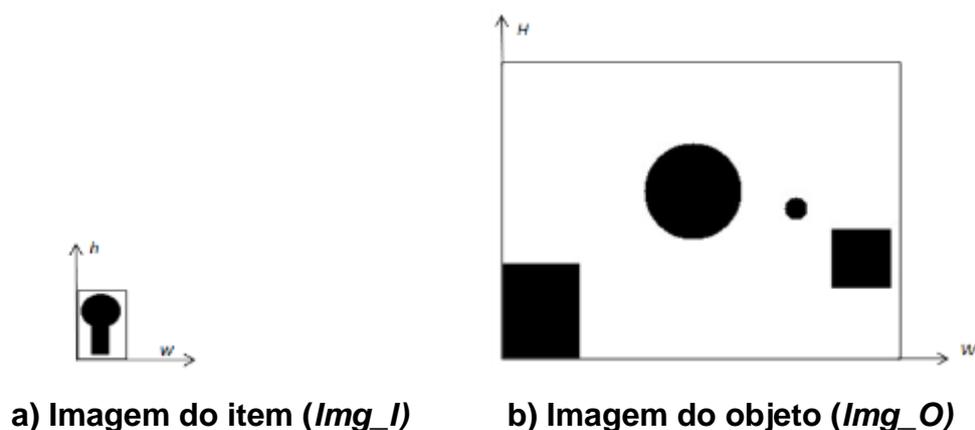
Tanto Max_tx como Max_ty são computadas com o valor 63, que corresponde ao limite superior da coluna (tx) e da linha (ty). Já o limite superior de rotações (Max_rot) é igual a 8, variando de 0° a 315° em intervalos de 45° .

4.3 DETALHAMENTO DA ABORDAGEM PROPOSTA

Conforme descrito na seção anterior, tanto o objeto (placa) quanto o item (peça) são representados por imagens digitais binárias (figura 19). Elas são denotadas, respectivamente, por Img_O e Img_I . Vale ressaltar que os objetos podem conter furos sinalizando cortes anteriores (placas para reaproveitamento) e os itens são idênticos com geometrias irregulares côncavas ou convexas.

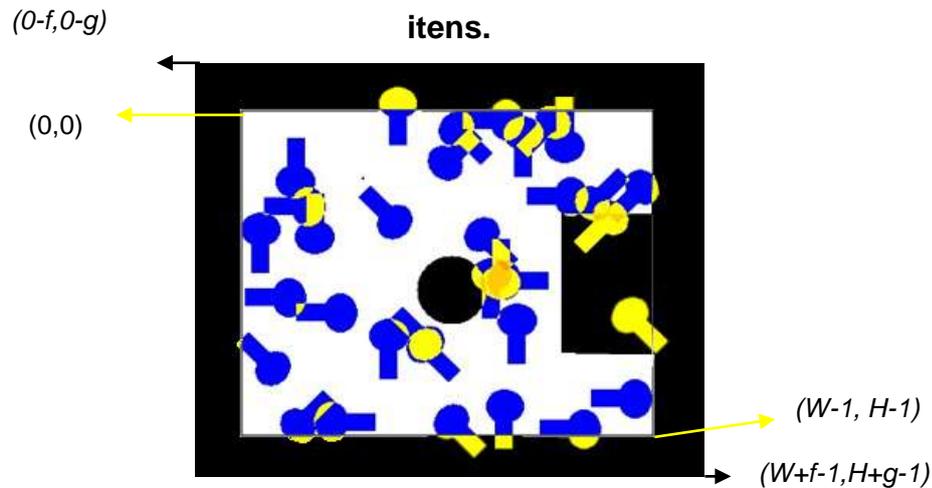
As imagens dos objetos possuem tamanho fixo (320×256), sendo 320 a largura (W) e 256 a altura (H), enquanto as imagens dos itens possuem tamanhos variados. Neste caso, a largura (w) pode estar entre 31 e 56, e altura (h) entre 32 e 65.

Figura 19 - Exemplo de imagem do item e do objeto.



Fonte: O Autor

Figura 21 - Esquema de cores adotado para indicar as sobreposições dos itens.



A rotina de Processamento de Imagens Digitais é responsável por gerar a imagem Img_P e fornecer valores numéricos que indicam a qualidade do planejamento de corte. Esses valores são utilizados como indicadores para medir a qualidade de uma solução gerada pelo AG e compõe a função de aptidão (FA) ou Função Objetivo (FO). Os indicadores são: coeficiente de sobreposição (c_s), coeficiente de itens posicionados para o corte (c_c) e o coeficiente de distância entre os itens (c_d).

Com base nessas considerações, a quantidade de pixels com sobreposição (S) na imagem Img_P pode ser obtido da seguinte forma:

$$S = \sum_{x=-f}^{W+f-1} \sum_{y=-g}^{H+g-1} Psbr, \text{ sendo } Psbr = \begin{cases} 1, & \text{se } Img_P(tx, ty) = (255, L_2, 0) \\ 0, & \text{caso contrário} \end{cases} \quad (4)$$

na qual: W e H são, respectivamente, a largura e altura da imagem Img_P , tx e ty indicam as coordenadas da imagem Img_P , $f=w/2$ e $g=h/2$ indicam as coordenadas do centro da imagem do item Img_I e $0 \leq L_2 \leq 255$ indica o valor da componente G do pixel $Img_P(tx, ty)$. Quanto menor for esse valor, maior é número de ocorrências de sobreposições neste ponto. Pode-se perceber, na equação 4, que uma certa região em torno da imagem Img_P é considerada. Então, se alguma parte do item for posicionada fora do domínio de Img_P , ela será considerada como sobreposição uma

vez que o background da imagem Img_P , assim como o da imagem Img_O , é representado pela cor preta (0,0,0).

O número máximo de itens (T_i) que podem ser cortados em um objeto é dado pela seguinte equação:

$$T_i = \frac{A_o}{A_i} \quad (5)$$

na qual: A_o (equação 6) representa a área disponível no objeto (somente pixels brancos) e A_i (equação 7) a área do item, composta por pixels pretos.

$$A_o = \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} P_o, \text{ na qual } P_o = \begin{cases} 1, & \text{se } Img_O(x, y) = (255, 255, 255) \\ 0, & \text{caso contrário} \end{cases} \quad (6)$$

$$A_i = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} P_i, \text{ na qual } P_i = \begin{cases} 1, & \text{se } Img_I(x, y) = (0, 0, 0) \\ 0, & \text{caso contrário} \end{cases} \quad (7)$$

Na verdade, T_i não representa a quantidade de itens que realmente serão cortados no objeto, mas o limite superior do número de itens que podem ser posicionados para corte. É o AG, por meio dos genes contidos no cromossomo e destinados para status (0 ou 1), quem determina a quantidade real de itens $Q_i \leq T_i$ a serem cortados na área útil do objeto.

A FO apresentada na equação 8 é utilizada para medir a qualidade de uma solução gerada pelo AG e é baseada nos seguintes indicadores: coeficiente de sobreposição (c_s), coeficiente de itens posicionados para o corte (c_c) e o coeficiente de distância entre os itens (c_d), os quais são descritos nas equações 9 a 11. Em resumo, a FO corresponde à média ponderada dos três coeficientes e quanto menor for o seu valor, maior será a qualidade do planejamento do corte:

$$FO = \text{Min } FA = \frac{c_s w_s + c_c w_c + c_d w_d}{w_s + w_c + w_d} \quad (8)$$

na qual: $w_s + w_c + w_d$ representam os pesos atribuídos aos respectivos coeficientes.

$$c_s = \frac{S}{A_i T_i} \quad (9)$$

$$c_c = 1 - \left(\frac{Q_i}{T_i} \right) \quad (10)$$

O coeficiente c_s exprime o percentual de pixels não factíveis, ou seja, com sobreposição. Já o coeficiente c_c indica o percentual de itens posicionados sobre o objeto, independente da existência de sobreposição ou não.

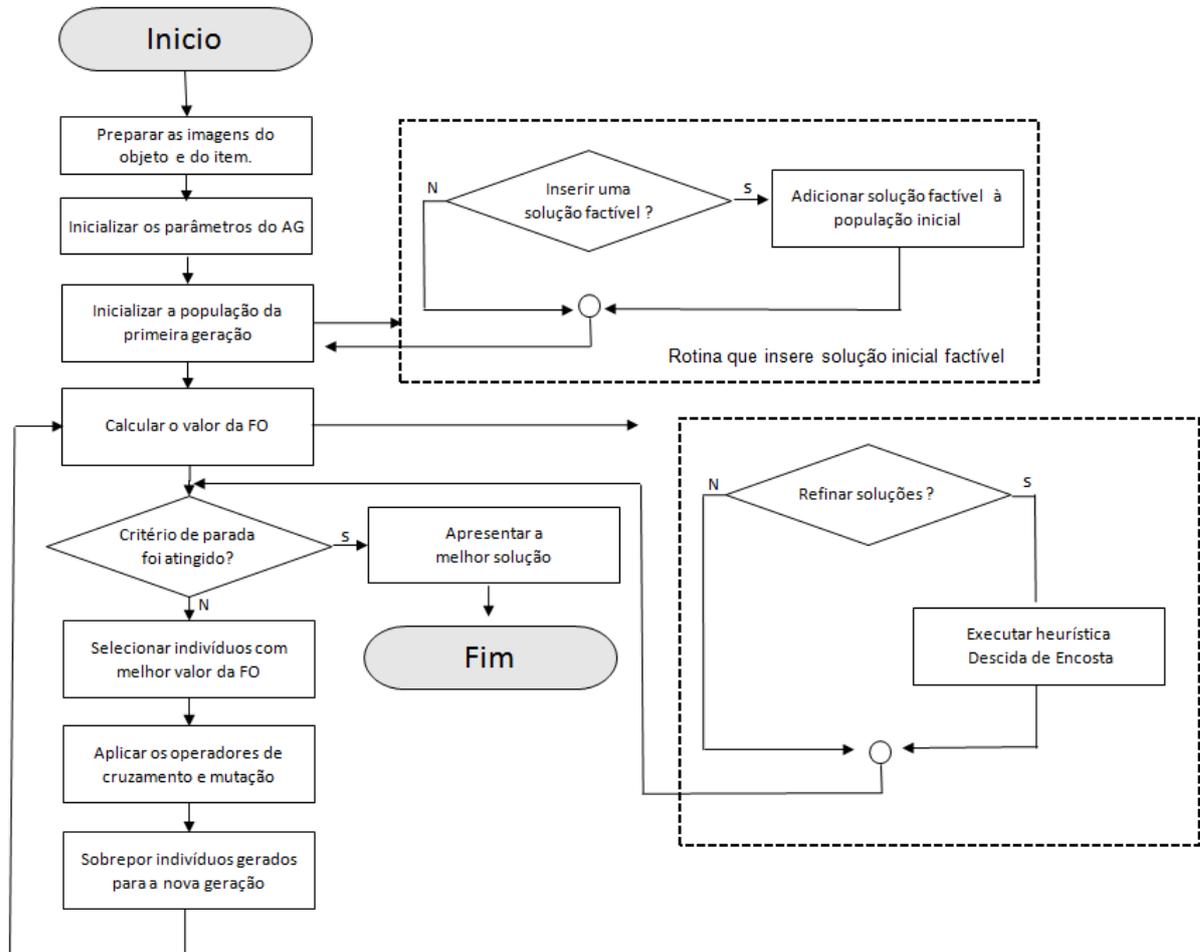
O cálculo do coeficiente c_d , que representa a distância entre os itens, é apresentado na equação 11:

$$c_d = \frac{\left(\frac{\sum_{i=1}^{Q_i} \sum_{j=i+1}^{Q_i} \sqrt{(tx_i - tx_j)^2 + (ty_i - ty_j)^2}}{\beta} \right)}{d_{\max}} \quad (11)$$

na qual: tx_i e ty_i indicam as translações aplicadas ao item i posicionado no objeto. Em outras palavras, representam as coordenadas da imagem Img_P na qual será posicionado o centro do item. Para o cálculo de c_d , primeiro são computadas as distâncias euclidianas entre todos os itens posicionados no objeto (dois a dois), observando que dados dois itens i e j , apenas se totaliza a distância do item i para o item j , já que a distância do item j para o item i é a mesma. Isso é garantido ao fazer $j=i+1$. Em seguida, este valor é dividido por $\beta = Q_i(Q_i - 1)/2$, que é o número de combinações de dois itens sem repetição. Por fim, o valor resultante é normalizado por $d_{\max} = \sqrt{W^2 + H^2}$, sendo que W e H são as dimensões da imagem Img_P .

O fluxograma apresentado na figura 22 exhibe, de forma detalhada, o funcionamento da abordagem proposta indicando o encadeamento lógico dos algoritmos desenvolvidos.

Figura 22 – Fluxograma detalhado do funcionamento da abordagem proposta



Fonte: O autor

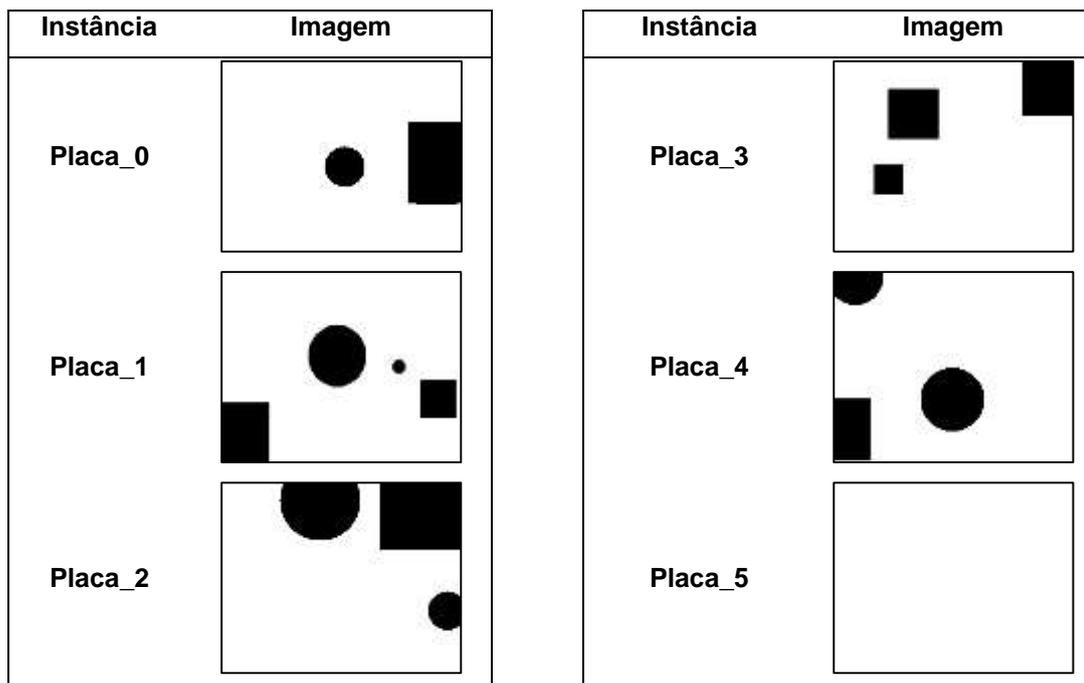
A aplicação das formulações matemáticas apresentadas neste capítulo, bem como outros detalhes da implementação da abordagem proposta pode ser encontrados no Apêndice A.

5 RESULTADOS DOS EXPERIMENTOS COMPUTACIONAIS

Neste capítulo são apresentados e discutidos os resultados obtidos nos experimentos realizados para parametrização do AG, definição dos pesos da função objetivo e avaliação da abordagem proposta. Para tanto, foram realizados os procedimentos constantes nas etapas 4 a 8 da figura 16. A avaliação da abordagem proposta contempla 3 grupos de experimentos com as seguintes características: i) inicialização da população do AG de forma aleatória; ii) inclusão de uma solução inicial factível (SIF) na população inicial do AG e empregando-se a heurística descida de encosta (HDE) e iii) inicialização da população do AG de forma aleatória e empregando-se a HDE. Os resultados desses experimentos são apresentados em forma de imagens e também por meio de dados numéricos.

Inicialmente, a fim de avaliar a abordagem proposta, fez-se necessário criar instâncias ou exemplares do problema investigado, as quais consistem em imagens do objeto e do item. Ao todo são 6 objetos, sendo 5 deles com furos, e 8 itens com geometrias irregulares, como ilustram as figuras 23 e 24. Os itens identificados como “Item_6” e “Item_7” são baseados no trabalho de Pinheiro *et al.* (2015).

Figura 23 - Imagem das instâncias dos objetos



Fonte: O Autor

Figura 24 - Imagem das instâncias dos itens

Instância	Imagem	w	h
Item_0		53	49
Item_1		37	53
Item_2		53	49
Item_3		56	65

Instância	Imagem	W	h
Item_4		31	32
Item_5		37	53
Item_6		38	44
Item_7		53	52

Fonte: O autor

Cabe ressaltar que a criação dos exemplares foi necessária pelo fato de não terem sido encontrados trabalhos com os mesmos objetivos na literatura.

5.1 DEFINIÇÃO DOS PARÂMETROS DE CONFIGURAÇÃO DO AG

Para Chwaab e Pinto (2011) planejamentos experimentais são procedimentos sofisticados no processo de estimação de parâmetros. Segundo Barros Neto *et al.* (2010), quando necessita-se saber a influência de certas variáveis, utiliza-se o planejamento fatorial completo. Sendo assim, esse método foi aplicado a fim de estimar os valores dos principais parâmetros de configuração do AG: tamanho da população (p), taxa de cruzamento (p_{Cross}), taxa de mutação (m) e o número de gerações (n_{Ger}). O Percentual de substituição (p_{Repl}) foi fixado em 0,7 a partir de diversos testes realizados. Este parâmetro é utilizado pela GALib no tipo SST.

No planejamento fatorial completo, fator se refere a cada variável do sistema em estudo e os níveis representam as condições de operação dos fatores de controle investigados que geralmente são: nível baixo (-) e nível alto (+) (CUNICO *et al.*, 2008). As quatro variáveis: p , p_{Cross} , m e n_{Ger} são os fatores e os níveis são -1 e +1.

O delta (Δ) representa a variação estimada do fator utilizado para calcular o valor do nível -1 e +1.

O X_0 corresponde ao ponto central que em conjunto com delta (Δ) fornece os dados necessários para calcular o valor dos níveis -1 e +1.

Os fatores, níveis e delta (Δ) utilizados neste planejamento fatorial são exibidos no quadro 5:

Quadro 5 - Fatores, níveis e delta

	Tamanho População (p)	Número de Gerações ($nGer$)	Taxa de Mutação (m)	Taxa de Cruzamento ($pCross$)
X_0	50	250	0,003	0,7
Δ	24	50	0,001	0,2
-1	26	200	0,002	0,5
0	50	250	0,003	0,7
1	74	300	0,004	0,9
- α	2	150	0,001	0,3
+ α	98	350	0,005	1,1

Fonte: O Autor

Para o cálculo do α (pontos axiais) utilizou-se a fórmula descrita na equação 12, na qual $k=4$ representa o número de fatores:

$$\alpha = \pm(2^k)^{\frac{1}{4}} \quad (12)$$

Os cálculos do limites inferior e superior, são descritos nas equações 13 e 14:

$$\text{nivel_inferior (-1)} = X_0 - \Delta \quad (13)$$

$$\text{nível_superior (+1)} = X_0 + \Delta \quad (14)$$

Os valores de - α e + α são obtidos conforme as equações 15 e 16:

$$-\alpha = X_0 - \alpha * \Delta \quad (15)$$

$$+\alpha = X_0 + \alpha * \Delta \quad (16)$$

Neste caso, foram executados 27 ensaios compondo o planejamento fatorial demonstrado no quadro 6:

Quadro 6 - Planejamento fatorial dos parâmetros

Ensaio	p	nGer	m	pCross	FO	Tempo (seg)	Ensaio	p	nGer	m	pCross	FO	Tempo (seg)
1	-1	-1	-1	-1	y1	yy2	1	26	200	0,002	0,5	0,04460	65,96
2	1	-1	-1	-1	y2	yy3	2	74	200	0,002	0,5	0,04813	111,74
3	-1	1	-1	-1	y3	yy4	3	26	300	0,002	0,5	0,04460	66,50
4	1	1	-1	-1	y4	yy5	4	74	300	0,002	0,5	0,04813	111,39
5	-1	-1	1	-1	y5	yy6	5	26	200	0,004	0,5	0,05012	40,81
6	1	-1	1	-1	y6	yy7	6	74	200	0,004	0,5	0,04781	189,17
7	-1	1	1	-1	y7	yy8	7	26	300	0,004	0,5	0,05012	41,24
8	1	1	1	-1	y8	yy8	8	74	300	0,004	0,5	0,04781	189,32
9	-1	-1	-1	1	y9	yy9	9	26	200	0,002	0,9	0,04932	50,73
10	1	-1	-1	1	y10	yy10	10	74	200	0,002	0,9	0,04418	272,75
11	-1	1	-1	1	y11	yy11	11	26	300	0,002	0,9	0,04932	50,96
12	1	1	-1	1	y12	yy12	12	74	300	0,002	0,9	0,04418	273,24
13	-1	-1	1	1	y13	yy13	13	26	200	0,004	0,9	0,04932	50,75
14	1	-1	1	1	y14	yy14	14	74	200	0,004	0,9	0,04418	272,77
15	-1	1	1	1	y15	yy15	15	26	300	0,004	0,9	0,04932	50,43
16	1	1	1	1	y16	yy16	16	74	300	0,004	0,9	0,04418	272,30
17	0	0	0	0	y17	yy17	17	50	250	0,003	0,7	0,04954	112,40
18	0	0	0	0	y18	yy18	18	50	250	0,003	0,7	0,04954	112,40
19	0	0	0	0	y19	yy19	19	50	250	0,003	0,7	0,04954	112,40
20	- α	0	0	0	y20	yy20	20	2	250	0,003	0,7	0,05064	7,61
21	+ α	0	0	0	y21	yy21	21	98	250	0,003	0,7	0,04999	129,56
22	0	- α	0	0	y22	yy22	22	50	150	0,003	0,7	0,04954	112,13
23	0	+ α	0	0	y23	yy23	23	50	350	0,003	0,7	0,04954	111,90
24	0	0	- α	0	y24	yy24	24	50	250	0,001	0,7	0,04352	195,70
25	0	0	+ α	0	y25	yy25	25	50	250	0,005	0,7	0,04771	154,41
26	0	0	0	- α	y26	yy26	26	50	250	0,003	0,3	0,05149	67,92
27	0	0	0	+ α	y27	yy27	27	50	250	0,003	1,1	0,04730	109,30

Fonte: O autor

Os 27 ensaios foram executados somente uma vez utilizando-se os valores contidos no planejamento fatorial. Utilizou-se somente o objeto placa_2 e o item_1, em todos os ensaios, escolhidos arbitrariamente.

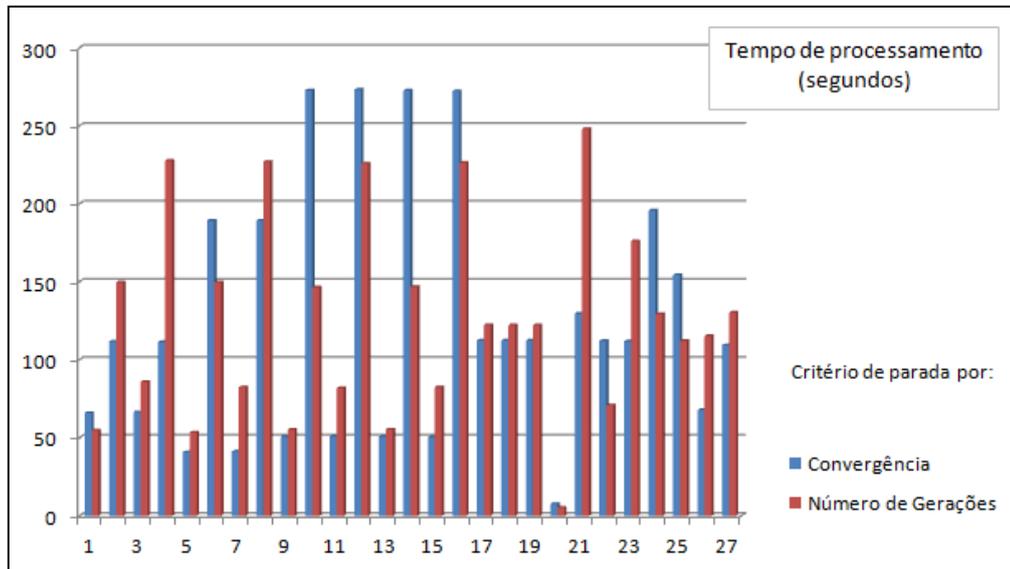
Após a execução dos ensaios foi possível avaliar quantitativamente a influência dos fatores sobre as respostas de interesse. Neste planejamento obteve-se duas respostas: o valor da função objetivo e o tempo de processamento.

O ensaio 24 apresentou o melhor valor para a variável que representa a função objetivo. Os valores desse ensaio são: 50 para p , 250 para $nGer$, 0,001 para m e 0,7 para $pCross$.

O parâmetro $nGer$ é utilizado no critério de parada do AG por número de gerações. Como a biblioteca GALib possui dois critérios de parada: número de gerações e convergência, ainda existia a necessidade de se verificar o comportamento do outro critério de parada. Sendo assim, para determinar qual seria o melhor critério de parada a ser empregado, utilizou-se a matriz de planejamento fatorial do quadro 6 e os 27 ensaios foram executados novamente para o critério de

parada por convergência. As figuras 25 e 26 apresentam os resultados comparativos dos ensaios:

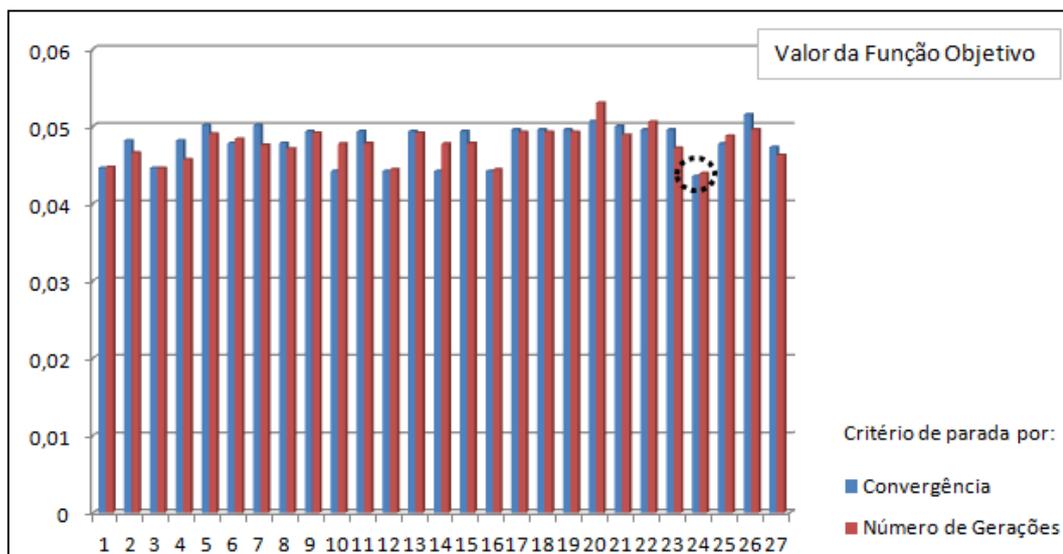
Figura 25 - Gráfico comparativo com o tempo de processamento.



Fonte: O autor

No gráfico da figura 25, observa-se que o tempo de processamento para o critério de parada por número de gerações, em sua maioria, apresenta-se maior que o critério de parada por convergência:

Figura 26 - Gráfico comparativo com o valor da FO.



Fonte: O autor

No gráfico da figura 26, observa-se que ocorreram poucas diferenças no valor da função objetivo. Desta forma, conclui-se que o critério de parada a ser adotado neste trabalho será por convergência, visto que os tempos de processamento foram menores. Os parâmetros adotados foram os obtidos no ensaio 24 e que estão resumidos na tabela 6 a seguir.

Tabela 6 - Parâmetros do AG utilizados no trabalho

Parâmetros	Valores
Tamanho da População (p)	50
Percentual de Mutação (m)	0,001
Percentual de Cruzamento ($pCross$)	0,7
Percentual de Substituição ($pRepl$)	0,7
Critério de parada	Convergência

Fonte: O autor

5.2 DEFINIÇÃO DOS PESOS DOS COEFICIENTES QUE COMPÕEM A FO

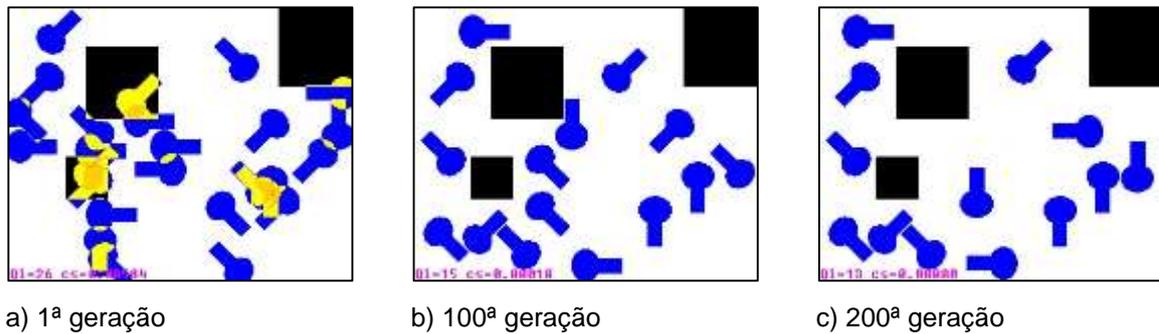
A função objetivo do AG empregado na abordagem proposta utiliza três variáveis (w_s, w_c, w_d) que representam os pesos atribuídos aos respectivos coeficientes (c_s, c_c e c_d). A fim de definir e exemplificar a relevância dos pesos, foram executados experimentos no qual cada variável foi testada isoladamente. Sendo assim, fez-se necessária a participação e observação do pesquisador de forma contínua. A cada geração eram analisados os efeitos dos pesos atribuídos às variáveis por meio da observação dos resultados apresentados na imagem do planejamento de corte (Img_P).

A semente inicial foi fixada com o mesmo valor para todos os experimentos desta seção, assim como o critério de parada ($nGer = 200$). O critério de parada por número de gerações foi utilizado para que todos os experimentos terminassem com a mesma quantidade de gerações. As imagens ilustradas nas figuras 27 a 29 têm a finalidade de demonstrar a evolução do AG destacando a geração inicial (1ª geração), geração central (100ª geração) e geração final (200ª geração). Os demais parâmetros necessários para o funcionamento do AG foram os mesmos elencados na tabela 6.

Neste experimento foram utilizadas as imagens da placa_3 e do item_1, escolhidas arbitrariamente.

Inicialmente a variável w_s , que indica o peso do coeficiente de sobreposição c_s , recebeu o peso 1 e as demais variáveis w_c e w_d foram zeradas. Desta forma, foi possível analisar somente os efeitos da variável w_s sobre o coeficiente c_s . Observa-se os efeitos nas imagens de resultados geradas e ilustradas na figura 27.

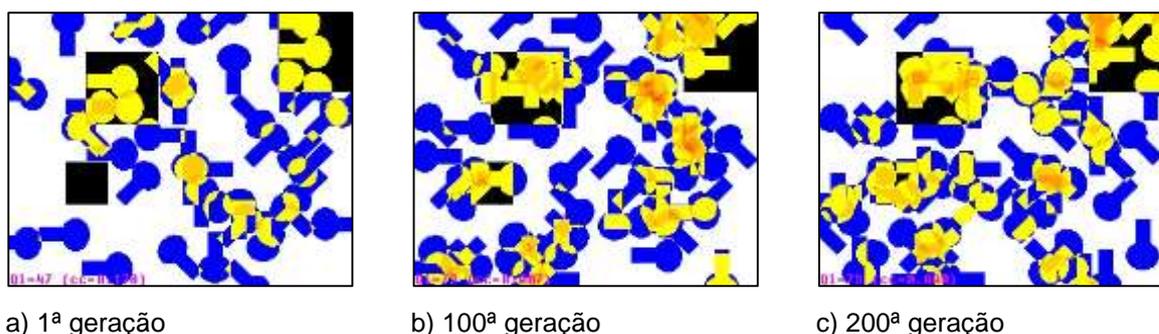
Figura 27 – Análise da influência sobre o coeficiente de sobreposição c_s



Na primeira geração o AG iniciou com 26 itens posicionados sobre o objeto e o $c_s=0,08584$. Na 100ª geração ocorreu um decremento na quantidade de itens para 15 e o $c_s=0,00010$ e, por fim, na última geração existem apenas 13 itens com $c_s=0,00000$. Observa-se que o AG tende a diminuir os pontos com sobreposição reorganizando os itens ou eliminando-os do planejamento de corte na imagem Img_P .

Outro coeficiente analisado foi o c_c que representa a quantidade de itens posicionados sobre o objeto para corte. Neste experimento somente a variável w_c que representa o peso desse coeficiente recebeu o peso 1 e as demais w_s e w_d receberam o peso 0. Na figura 28 pode-se observar a evolução do AG em três momentos ao longo das 200 gerações.

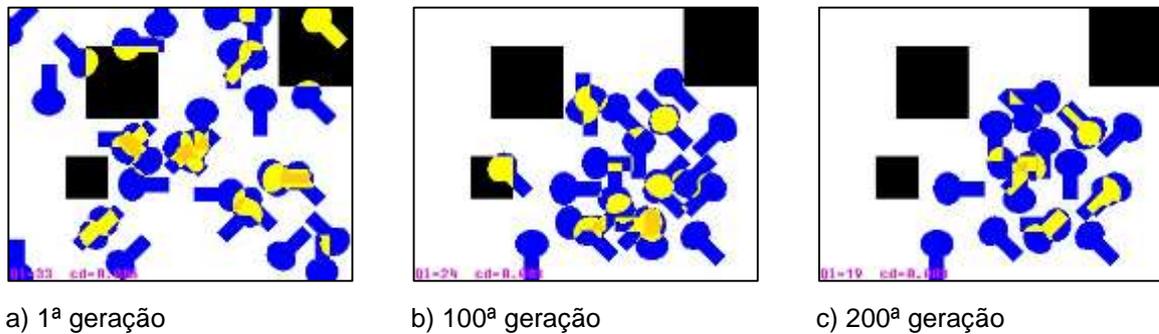
Figura 28 - Análise da influência sobre o coeficiente de quantidade de itens c_c



O c_c necessita da variável T_i para ser calculado. Neste caso, $T_i=73$, indicando que o número máximo de itens que podem ser posicionados sobre o objeto é 73. Analisa-se que na 1ª geração existiam 47 itens posicionados e 72 na 100ª geração. A última geração finalizou com 73 itens. Observou-se que sempre que ocorre a inserção de um novo item sobre o objeto, o valor de c_c diminui em $\frac{1}{T_i}$, que neste caso corresponde a 0,014, representando um bônus. Percebe-se ainda que o AG insere novos itens independente da ocorrência de sobreposições. A tendência é posicionar o máximo de itens possível até o limite T_i .

O último coeficiente analisado foi o c_d , que representa a distância euclidiana entre todos os itens posicionados sobre o objeto. Ele é importante porque mantém os itens mais próximos uns dos outros. Neste experimento somente a variável w_d que indica o peso do coeficiente c_d , recebeu o peso 1 e as demais w_c e w_s receberam o peso 0. Na figura 29 observa-se a influência deste peso:

Figura 29 - Análise da influência sobre o coeficiente de distância c_d



Pode-se observar na figura 29 que na primeira geração os itens estavam fora das bordas e muito espalhados sobre o objeto com $c_d=0,006$ e $Qi=33$. Na 100ª e 200ª geração os itens apresentam-se mais aglutinados no centro e os valores de Qi foram respectivamente 24 e 19 apresentando o coeficiente $c_d=0,003$ para ambos. Este coeficiente sozinho, tende a diminuir a quantidade de itens e também não verifica as sobreposições. A quantidade de itens tende a diminuir porque ao eliminar um item do planejamento de corte, o valor do coeficiente c_d também tende a diminuir.

Os três experimentos realizados nesta seção demonstraram a relevância de cada coeficiente na composição da FO e que isoladamente eles produzem resultados insatisfatórios.

A partir desses e outros experimentos de refinamento que foram realizados, definiu-se que os pesos w_c e w_d dos coeficientes c_c e c_d receberiam o valor 1 e o peso w_s do coeficiente c_s receberia o valor 5. A sobreposição representa uma violação às restrições específicas deste PC, sendo assim sua ocorrência deve ser penalizada com um peso maior na composição da FO.

5.3 AVALIAÇÃO DA ABORDAGEM PROPOSTA

Este conjunto de experimentos (divididos em três subconjuntos) teve como finalidade avaliar a abordagem proposta para solução do 2D-I-IIPP, considerando diferentes tipos de objetos e de itens. Para tanto, foram utilizadas as imagens dos 6 objetos e dos 8 itens ilustrados nas figuras 23 e 24. Em adição, pode-se observar o comportamento da abordagem com e sem solução inicial factível injetada na população inicial do AG e com e sem a heurística descida de encosta acoplada ao AG.

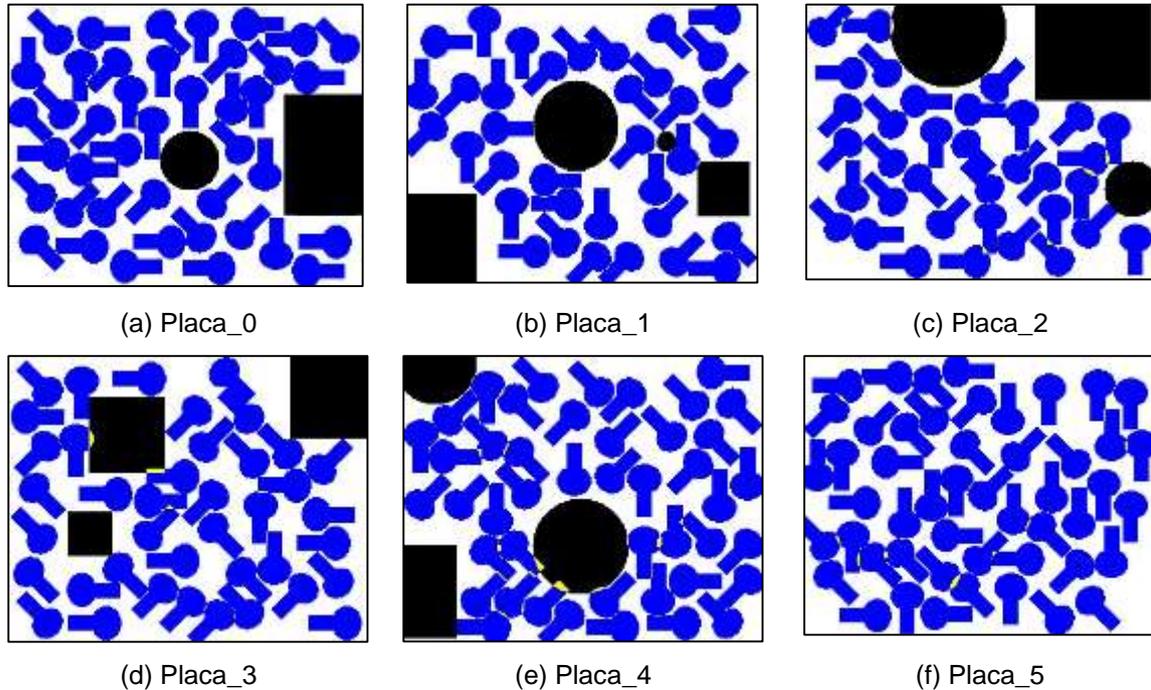
No primeiro subconjunto de experimentos (seção 5.3.1) utilizou-se o AG com a rotina de Processamento de Imagens Digitais (RPID), inicializando-se a população do AG de forma aleatória. No segundo subconjunto (seção 5.3.2), empregou-se o AG com a RPID, inserindo-se uma solução factível (SIF) na população inicial do AG, além da heurística descida de encosta (HDE) acoplada ao AG para fazer um refinamento de suas soluções após um certo número de gerações sem melhorias. Por fim, no terceiro subconjunto de experimentos (seção 5.3.3), utilizou-se o AG com a RPID com a HDE e a população inicial do AG inicializada de forma aleatória.

5.3.1 Experimentos com AG + RPID

Neste subconjunto de experimentos buscou-se identificar como o AG + RPID funcionaria para diversos tipos de objetos. A população inicial foi gerada de forma aleatória adotando-se a mesma semente para todos os objetos. Utilizou-se apenas

um formato de item com 6 diferentes formatos de objeto. A figura 30 apresenta os resultados obtidos neste experimento.

Figura 30 - Resultados do experimento AG + RPID



Os resultados obtidos demonstram que a abordagem proposta funciona para qualquer tipo de objeto, porém ainda acontecem sobreposições e invasão da área que representa o furo. Apesar de apresentar poucos pontos não factíveis, estes acabam por comprometer um ou mais itens. Analisando-se a tabela 7, pode-se verificar numericamente esses mesmos resultados. O percentual de ocupação apresentado na 4ª coluna é calculado somando-se todo pixel na cor azul (0,0,255) posicionado sobre a imagem Imp_P e posteriormente dividindo-se esse valor pela área útil disponível do objeto A_o . Também é possível obter esse valor dividindo-se a variável Q_i pela variável T_i , mas, neste caso, ocorre um arredondamento, sendo assim não foi empregado.

Tabela 7 - Resultados numéricos do experimento AG+RPID

Objeto	Máximo itens (T_i)	Quant. de itens posicionados (Q_i)	% de ocupação	Coefficiente de Sobreposição (c_s) (em pixel)	Tempo de processamento (segundos)
Placa_0	75	42	55,89 %	0,00159	1374,10
Placa_1	72	36	49,81%	0,00358	893,59
Placa_2	66	35	52,68 %	0,00330	557,80
Placa_3	73	40	54,06 %	0,00303	828,61
Placa_4	73	45	61,53 %	0,00818	1077,74
Placa_5	85	50	58,29 %	0,06417	1522,82

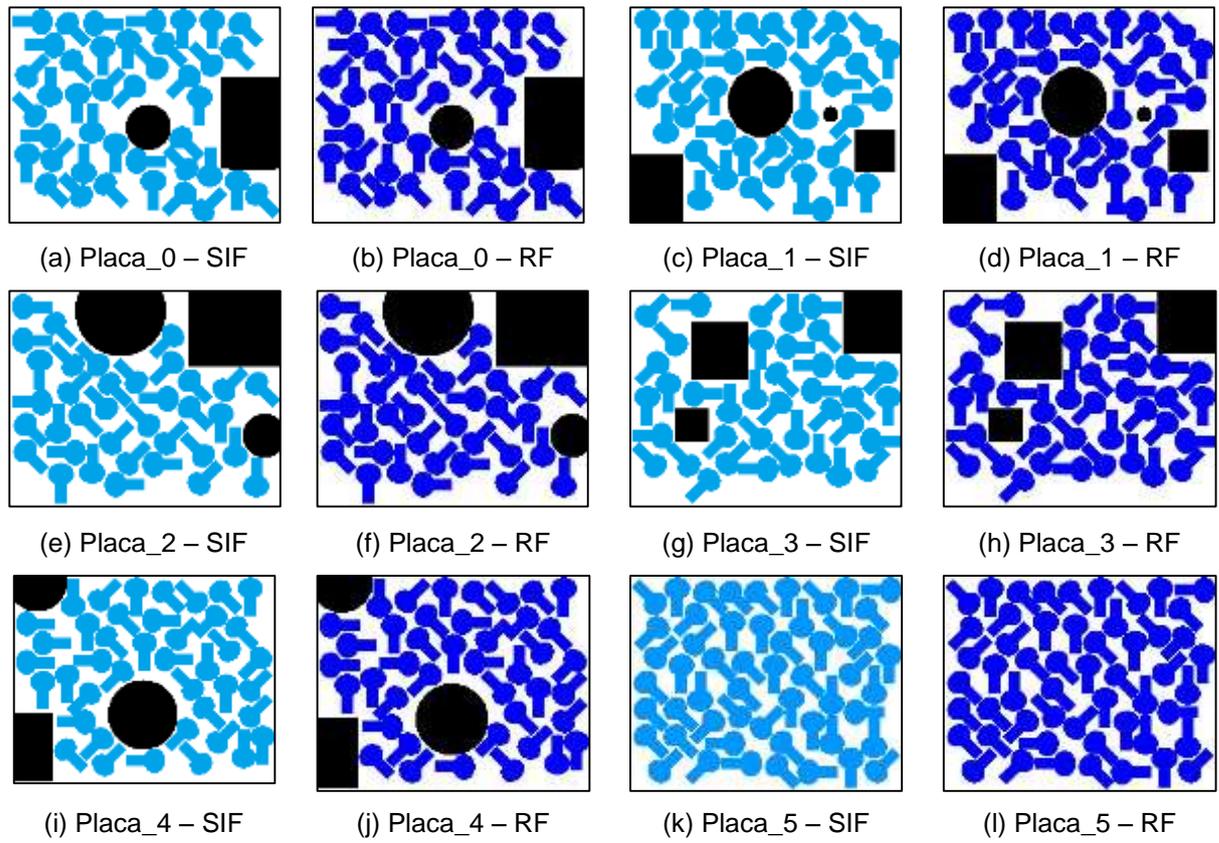
Percebe-se que os resultados apresentados neste experimento estão próximos de soluções factíveis, porém nota-se que a abordagem usando AG + RPID inicializando-se a população de forma aleatória, ainda necessita ser melhorada devido ao baixo percentual de ocupação, sobreposições e alto custo computacional.

5.3.2 Experimentos com AG + RPID + HDE + SIF

Neste subconjunto de experimentos utilizou-se AG + RPID com a heurística descida de encosta (HDE) acoplada ao AG. Além disso, uma solução inicial factível (SIF) foi introduzida na população inicial do AG. Essa solução factível foi gerada por um algoritmo semi-exaustivo que considera todas as possíveis translações para os itens, porém com rotações aleatórias.

Pretendeu-se verificar se o AG + RPID + HDE conseguiria melhorar a solução inicial que foi inserida na população do AG. A figura 31 apresenta, em azul claro, as soluções iniciais factíveis injetadas no AG, enquanto os resultados finais (RF) obtidos pelo AG + RPID +HDE são mostrados em azul marinho.

Figura 31 - Resultados do experimento AG + RPID + HDE + SIF



Na tabela 8 pode-se observar os resultados numéricos obtidos neste experimento.

Tabela 8 - Resultados numéricos do experimento AG+RPID+HDE+SIF

Objeto	Máximo itens (T_i)	Quant. de itens posicionados (Q_i)	% de ocupação	Coefficiente de Sobreposição (c_s) (em pixel)	Tempo de processamento (segundos)
Placa_0	75	41	54,56 %	0,00000	50,87
Placa_1	72	40	55,34 %	0,00000	38,63
Placa_2	66	36	54,18 %	0,00000	39,87
Placa_3	73	39	52,71 %	0,00000	24,15
Placa_4	73	42	57,42 %	0,00000	65,97
Placa_5	85	54	62,95 %	0,00000	104,37

Na figura 31, pode-se observar que o algoritmo efetuou pequenas alterações em alguns itens da solução inicial factível, porém o AG + RPID + HDE convergiram rapidamente e não alcançaram resultados muito significativos.

5.3.3 Experimentos com AG + RPID + HDE

Neste subconjunto de experimentos acoplou-se a heurística descida de encosta (HDE) ao AG + RPID e a população inicial foi gerada pelo AG, de forma aleatória. Foram executados experimentos com 8 itens e 6 objetos distintos. Neste experimento, cujos resultados podem ser vistos nas figuras 32 a 39 e nas tabelas 9 a 16, pretendia-se verificar se a abordagem proposta conseguiria eliminar totalmente as sobreposições, além de verificar como ela se comportaria com diferentes formatos de itens e objetos.

Figura 32 - Resultados do experimento AG + RPID + HDE com o item_0

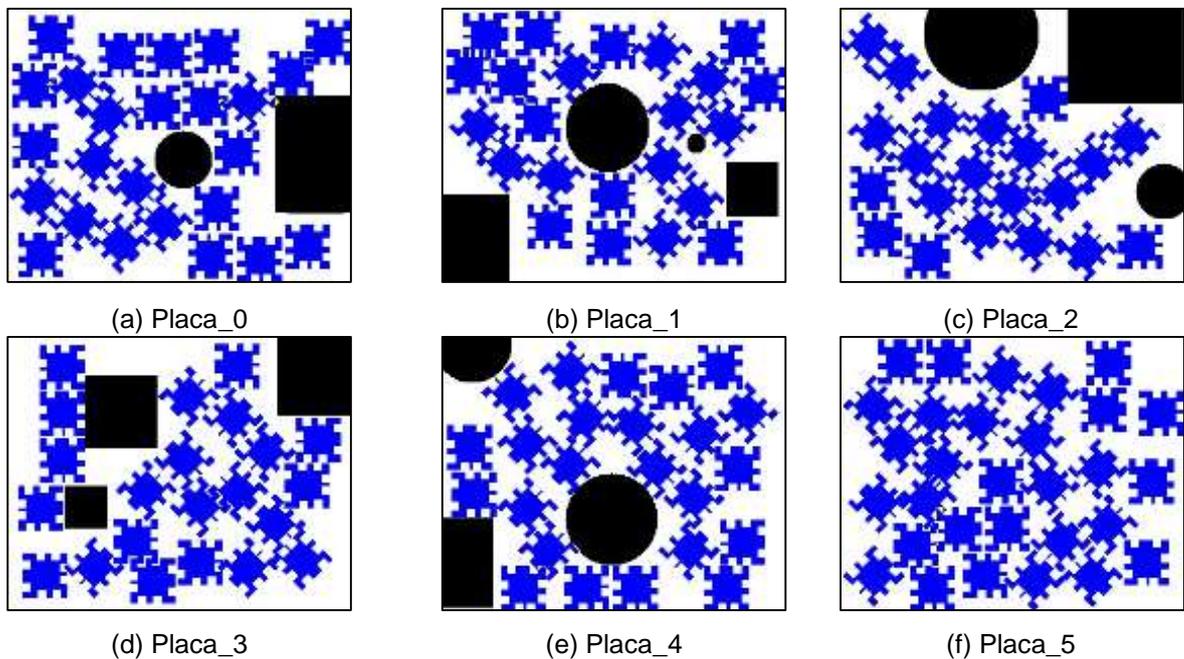


Tabela 9 - Resultados numéricos do experimento AG+RPID+HDE com o item_0

Objeto	Máximo itens (T_i)	Quant. de itens posicionados (Q_i)	% de ocupação	Coefficiente de Sobreposição (c_s) (em pixel)	Tempo de processamento (segundos)
Placa_0	54	25	45,56%	0,00074	859,46
Placa_1	52	23	43,58%	0,00075	622,37
Placa_2	48	21	43,29%	0,00025	433,80
Placa_3	54	22	40,73%	0,00000	826,85
Placa_4	53	24	44,94%	0,00032	684,82
Placa_5	62	30	47,90%	0,00058	749,64

A figura 32 e a tabela 9, apresentam os resultados do experimento com o item_0. Pode-se observar que os itens estão mais próximos do centro, não há invasão das bordas e dos furos. Porém, ocorrem pequenas sobreposições de itens e percebe-se a ocorrência de áreas brancas no objeto, que poderiam ser preenchidas.

Figura 33 - Resultados do experimento AG + RPID + HDE com o item_1

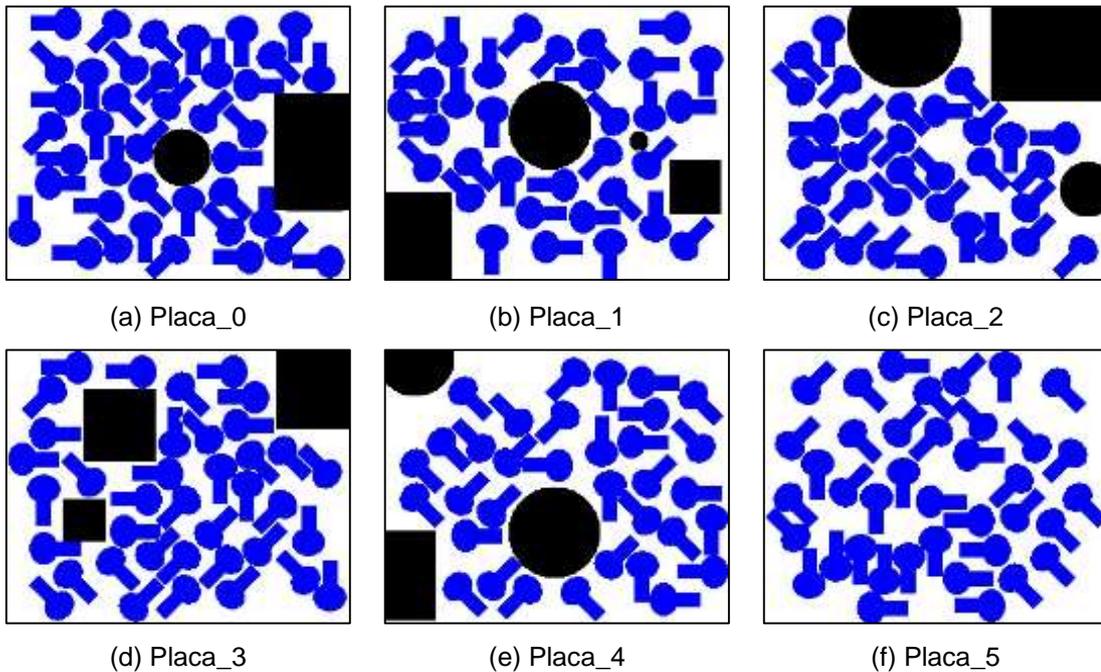


Tabela 10 - Resultados numéricos do experimento AG+RPID+HDE com o item_1

Objeto	Máximo itens (T_i)	Quant. de itens posicionados (Q_i)	% de ocupação	Coefficiente de Sobreposição (c_s) (em pixel)	Tempo de processamento (segundos)
Placa_0	75	40	53,23%	0,00000	2773,22
Placa_1	72	33	45,66%	0,00017	722,94
Placa_2	66	33	49,67%	0,00011	1544,12
Placa_3	73	38	51,36%	0,00032	1857,20
Placa_4	73	34	46,49%	0,00006	622,15
Placa_5	85	39	45,47%	0,00034	692,76

A figura 33 e a tabela 10 apresentam os resultados do experimento utilizando o item_1. Pode-se observar que os itens estão mais próximos do centro, não há invasão das bordas e dos furos. Porém algumas placas apresentam poucos pixels com sobreposição de itens e percebe-se a ocorrência de áreas brancas no objeto, que

poderiam ser preenchidas. A exceção foi a placa_0 que teve o coeficiente de sobreposição zerado e apresentou um percentual de ocupação de 53,23%.

Figura 34 - Resultados do experimento AG + RPID + HDE com o item_2

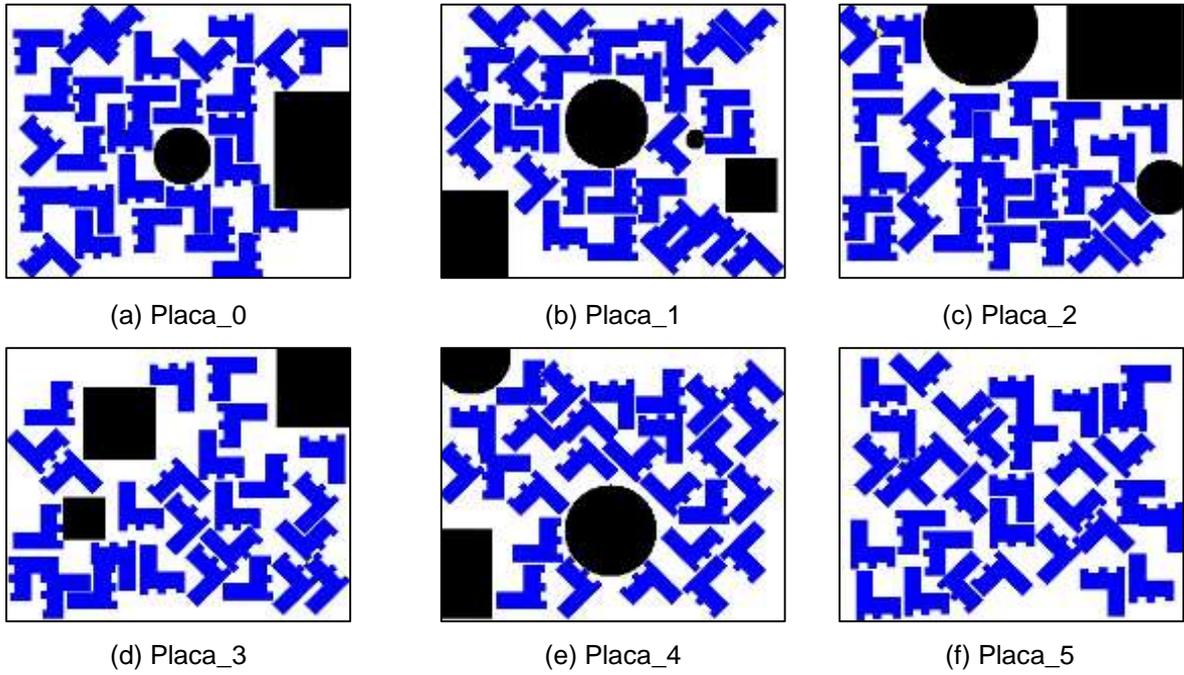


Tabela 11 - Resultados numéricos do experimento AG+RPID+HDE com o item_2.

Objeto	Máximo itens (T_i)	Quant. de itens posicionados (Q_i)	% de ocupação	Coefficiente de Sobreposição (c_s) (em pixel)	Tempo de processamento (segundos)
Placa_0	60	28	46,50%	0,00029	1125,70
Placa_1	57	28	48,35%	0,00057	1133,20
Placa_2	53	27	50,72%	0,00112	967,55
Placa_3	59	26	43,86%	0,00030	509,54
Placa_4	58	27	46,08%	0,00040	994,68
Placa_5	68	32	46,56%	0,00000	747,65

A figura 34 e a tabela 11 apresentam os resultados do experimento utilizando o item_2. Este experimento apresentou uma solução factível para a placa_5. É interessante observar como o algoritmo encaixou um item no outro. Ainda ocorreram sobreposições, mas os itens tendem a se agrupar ao centro do objeto.

Figura 35 - Resultados do experimento AG + RPID + HDE com o item_3

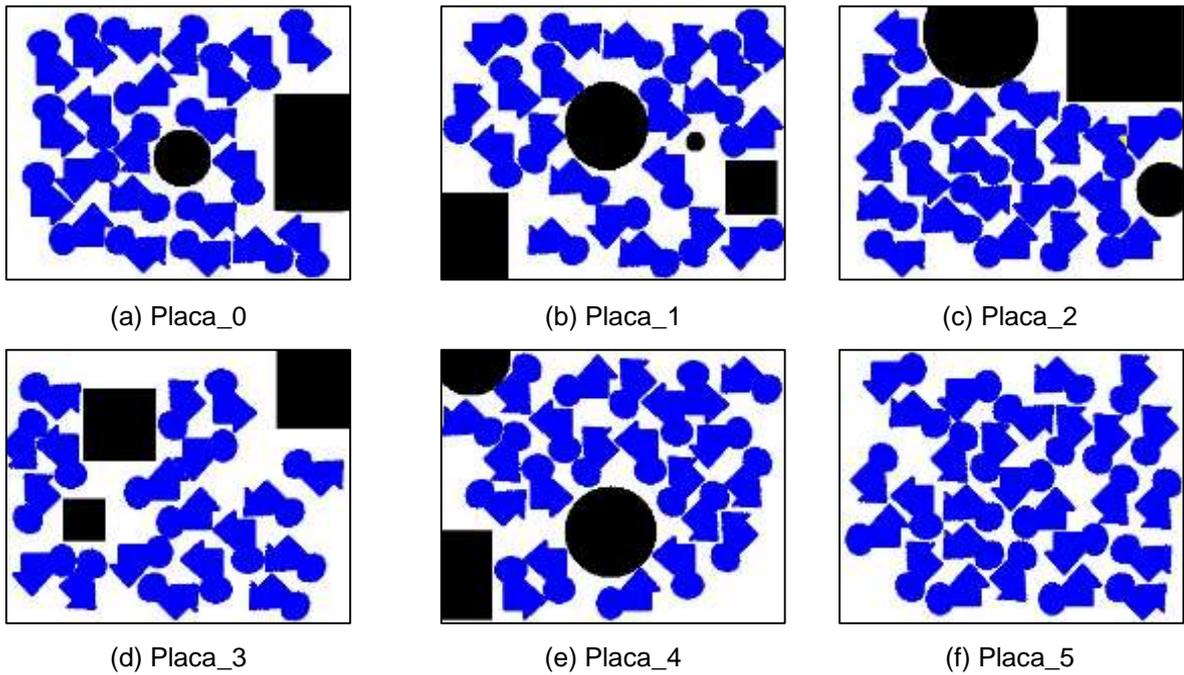


Tabela 12 - Resultados numéricos do experimento AG+RPID+HDE com o item_3

Objeto	Máximo itens (T_i)	Quant. de itens posicionados (Q_i)	% de ocupação	Coefficiente de Sobreposição (c_s) (em pixel)	Tempo de processamento (segundos)
Placa_0	44	22	50,00%	0,00000	1087,62
Placa_1	42	21	49,62%	0,00057	674,41
Placa_2	38	21	53,98%	0,00082	1226,37
Placa_3	43	19	43,86%	0,00083	458,85
Placa_4	42	22	51,37%	0,00013	1226,82
Placa_5	50	27	53,76%	0,00002	1462,37

A figura 35 e a tabela 12 apresentam os resultados do experimento utilizando o item_3. Como nos experimentos anteriores, pode-se observar que os itens estão mais agrupados ao centro e não há invasão dos furos. Porém ocorrem sobreposições de itens, invasão das bordas e percebe-se a ocorrência de áreas brancas no objeto, que poderiam ser preenchidas.

Novamente a placa_0 é uma exceção, não ocorreram sobreposições e o percentual de ocupação ficou em 50%.

Figura 36 - Resultados do experimento AG + RPID + HDE com o item_4

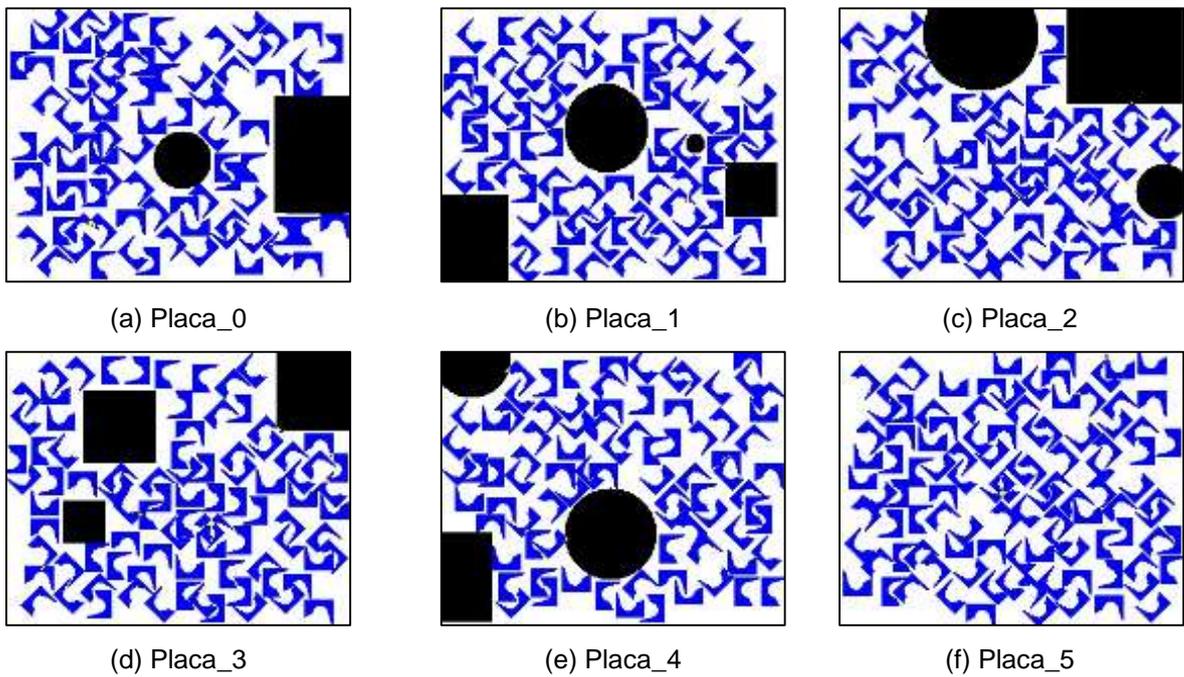


Tabela 13 - Resultados numéricos do experimento AG+RPID+HDE com o item_4

Objeto	Máximo itens (T_i)	Quant. de itens posicionados (Q_i)	% de ocupação	Coefficiente de Sobreposição (c_s) (em pixel)	Tempo de processamento (segundos)
Placa_0	218	82	37,59%	0,00130	1467,72
Placa_1	209	82	39,08%	0,00060	2074,13
Placa_2	192	77	39,93%	0,00063	1675,96
Placa_3	214	87	40,51%	0,00163	2253,62
Placa_4	212	83	39,09%	0,00110	1348,59
Placa_5	248	104	41,77%	0,00135	3077,55

A figura 36 e a tabela 13 apresentam os resultados do experimento utilizando o item_4. Este é o menor item utilizado nos testes. O que chama a atenção é o tempo de processamento, percebe-se que é muito superior quando comparado aos demais itens. O coeficiente de sobreposição também é em sua maioria maior apresentando valores na terceira casa decimal. Neste experimento todas as soluções apresentadas não são factíveis.

Figura 37 - Resultados do experimento AG + RPID + HDE com o item_5

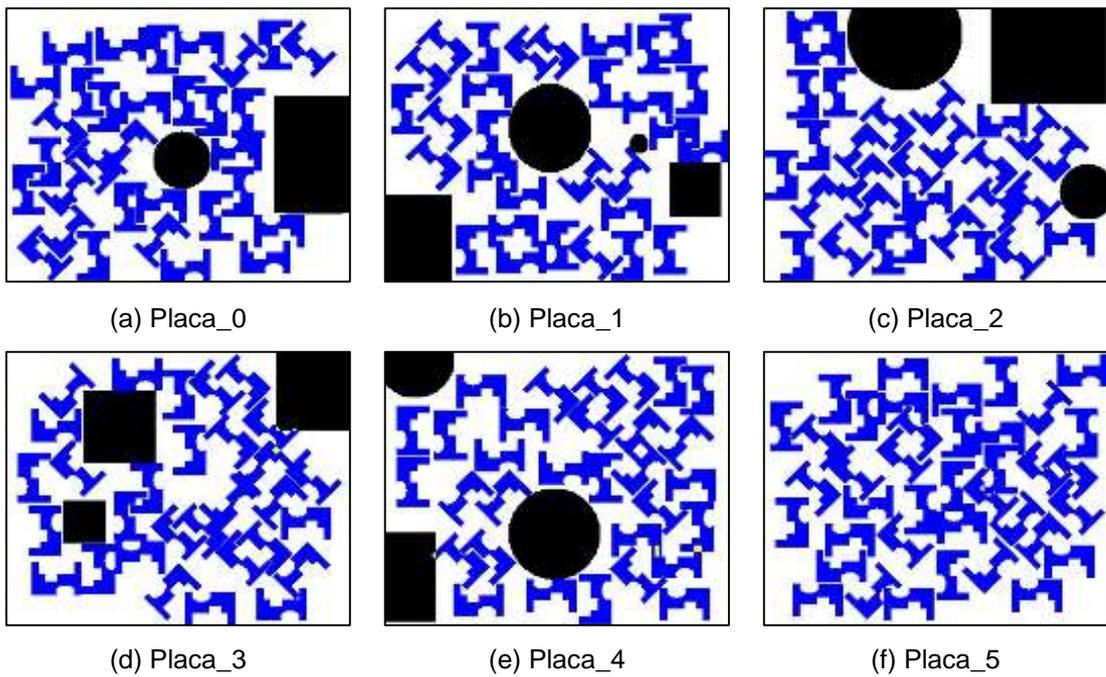


Tabela 14 - Resultados numéricos do experimento AG+RPID+HDE com o item_5

Objeto	Máximo itens (T_i)	Quant. de itens posicionados (Q_i)	% de ocupação	Coefficiente de Sobreposição (c_s) (em pixel)	Tempo de processamento (segundos)
Placa_0	81	32	39,19%	0,00029	810,67
Placa_1	78	30	38,20%	0,00022	652,38
Placa_2	72	29	40,17%	0,00000	689,64
Placa_3	80	30	37,32%	0,00074	464,70
Placa_4	79	31	39,01%	0,00157	341,70
Placa_5	93	39	41,85%	0,00043	829,39

A figura 37 e a tabela 14 apresentam os resultados do experimento utilizando o item_5. Observa-se que o item apresenta concavidade em duas extremidades. Percebe-se que a Placa_2 apresentou uma solução factível com 0 de sobreposições. As demais apresentaram pixels com sobreposição e o percentual de ocupação atingiu no máximo 41,85%.

Figura 38 - Resultados do experimento AG + RPID + HDE com o item_6

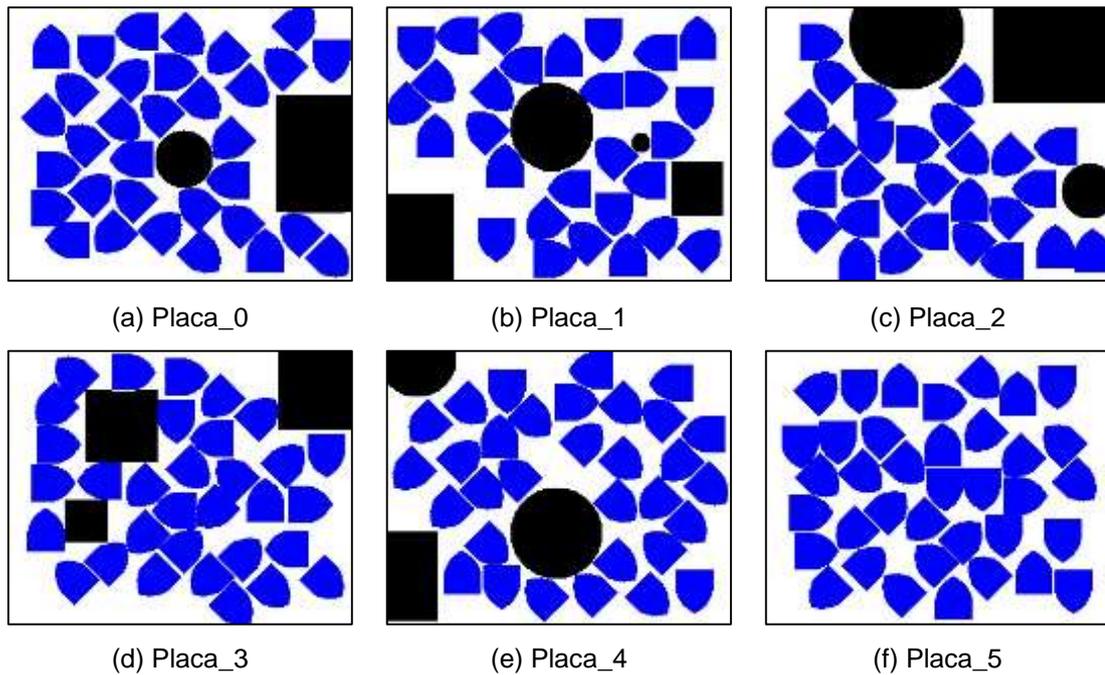


Tabela 15 - Resultados numéricos do experimento AG+RPID+HDE com o item_6

Objeto	Máximo itens (T_i)	Quant. de itens posicionados (Q_i)	% de ocupação	Coefficiente de Sobreposição (c_s) (em pixel)	Tempo de processamento (segundos)
Placa_0	60	33	54,40%	0,00010	1095,00
Placa_1	58	29	49,70%	0,00003	360,73
Placa_2	53	29	54,07%	0,00078	1118,20
Placa_3	59	32	53,58%	0,00040	765,40
Placa_4	59	32	54,20%	0,00000	764,18
Placa_5	69	37	53,43%	0,00000	1152,76

A figura 38 e a tabela 15 apresentam os resultados do experimento utilizando o item_6. Este experimento com o item convexo, apresentou percentuais de ocupação, em sua maioria, acima de 53% e duas placas apresentaram soluções factíveis.

Figura 39 - Resultados do experimento AG + RPID + HDE com o item_7

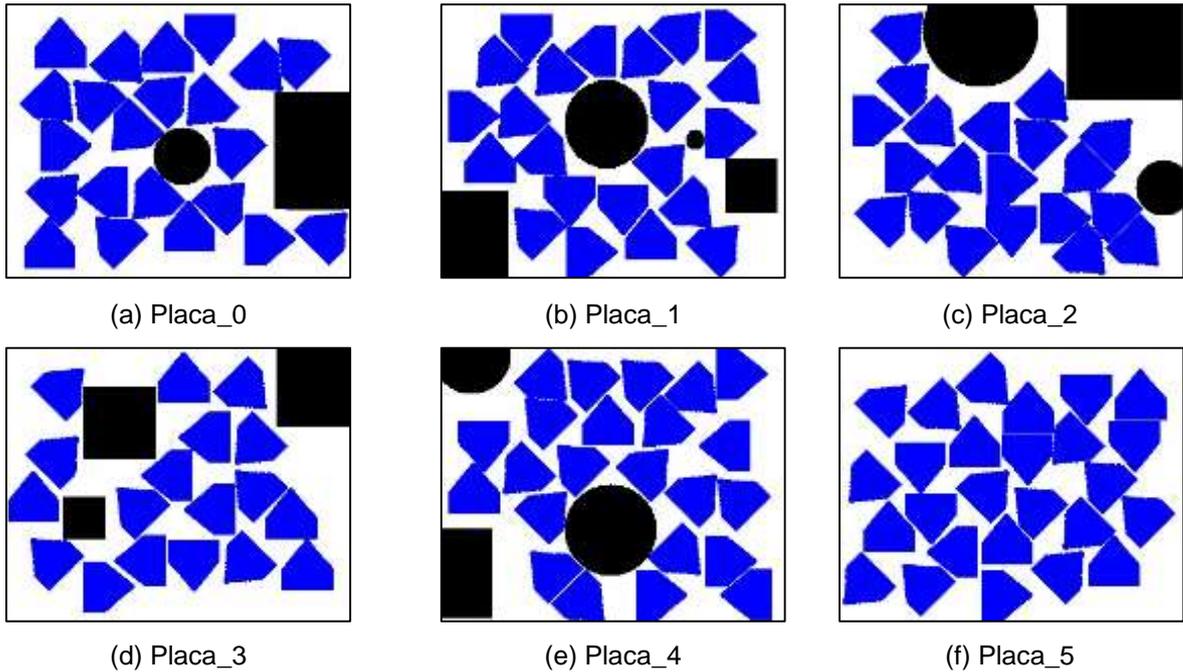


Tabela 16 - Resultados numéricos do experimento AG+RPID+HDE com o item_7

Objeto	Máximo itens (T_i)	Quant. de itens posicionados (Q_i)	% de ocupação	Coefficiente de Sobreposição (c_s) (em pixel)	Tempo de processamento (segundos)
Placa_0	42	22	51,19%	0,00000	707,35
Placa_1	41	22	53,23%	0,00007	573,14
Placa_2	37	20	52,64%	0,00019	817,12
Placa_3	42	18	42,54%	0,00000	279,22
Placa_4	41	23	54,99%	0,00009	684,61
Placa_5	49	26	53,00%	0,00000	947,24

A figura 39 e a tabela 16 apresentam os resultados do experimento utilizando o item_7. Este experimento apresentou 3 soluções factíveis para as placas 0, 3 e 5. Observa-se que os itens apresentam-se agrupados ao centro do objeto e o percentual de ocupação, em sua maioria, acima de 51%.

É importante ressaltar que as sobreposições, em boa parte dos experimentos mostrados nas figuras 32 a 39 e quantificadas nas tabelas 9 a 16, ocorrem em poucos pixels das bordas dos itens e provavelmente são decorrentes de ruídos introduzidos no processamento de imagens, principalmente na rotação dos itens.

De um modo geral também observa-se que os objetos apresentaram muitos espaços em branco que poderiam ser preenchidos, principalmente próximo às bordas do objeto. É possível verificar esse problema, analisando-se o percentual de ocupação nas tabelas 9 a 16.

Na seção seguinte é feita uma discussão a cerca dos resultados encontrados.

5.4 DISCUSSÃO DOS RESULTADOS

O primeiro experimento (seção 5.1) com 27 ensaios que determinou os valores para os parâmetros de configuração do AG foi essencial para avaliar a abordagem proposta, pois identificar os melhores valores utilizando-se um método confiável como o planejamento fatorial, é uma maneira de assegurar que o algoritmo está devidamente calibrado, dirimindo possíveis dúvidas a respeito dos resultados apresentados.

Com relação ao experimento para determinação dos pesos que foram atribuídos aos coeficientes c_s , c_c e c_d , seção 5.2, foi possível demonstrar isoladamente o comportamento de cada variável, por meio das imagens do planejamento de corte (img_P), em 3 gerações fixas. Constatou-se que cada coeficiente possui uma característica específica e necessária para a composição da FO. Enquanto o coeficiente c_s procura eliminar as sobreposições, o coeficiente c_c agrega novos itens ao planejamento de corte e o coeficiente c_d destina-se a manter os itens mais próximos. Os reflexos no ajuste desses pesos foram percebidos nos experimentos de avaliação da abordagem proposta, apresentados na seção 5.3.

Com respeito aos experimentos conduzidos para avaliar a estratégia proposta, pode-se verificar que a utilização do AG+RPID com uma população inicial aleatória gerada pelo AG, funciona para diversos tipos de objetos, mas apresentou resultados numéricos que precisam ser melhorados, como o alto custo computacional, algumas sobreposições e baixo percentual de ocupação. Com a introdução da heurística descida de encosta (HDE) e a inserção de uma solução inicial factível (SIF) esperava-se que a abordagem conseguisse melhorar a solução inicial, porém o algoritmo finalizou rapidamente sem melhorias representativas.

Nos experimentos que empregaram o AG + RPID + HDE com uma população aleatória gerada pelo AG, observou-se que houve uma melhora nos resultados

apresentados. Em especial os itens com o formato convexo apresentaram melhor resultado que os itens irregulares e com formato côncavo. Percebeu-se que o formato do item afeta os resultados computacionais conforme constataram Pinheiro *et al.* (2015). Além disso, constatou-se que o algoritmo procura agrupar os itens ao centro, mas por outro lado, ainda existem muitos espaços em branco que poderiam ser preenchidos. Constatou-se que em alguns casos, a heurística aplicada diminuiu o valor do coeficiente de sobreposição chegando a soluções factíveis. É importante ressaltar que essas sobreposições ocorrem em poucos pixels das bordas dos itens e provavelmente são decorrentes de ruídos introduzidos no processamento de imagens, principalmente na rotação dos itens. Para solução desse problema, sugere-se que ao iniciar a rotina de Processamento de Imagens Digitais, os itens sejam dilatados (expandindo-se um pixel em todas as direções) e, antes do computo da FO, esses mesmos elementos sejam erodidos (contraíndo-se um pixel em todas as direções), levando-os aos seus tamanhos originais

6 CONCLUSÕES E SUGESTÕES PARA A CONTINUIDADE DO TRABALHO

Neste trabalho foi proposta uma abordagem baseada em AG e RPID para lidar com o problema de cortar itens com formas irregulares idênticas em objetos, categorizado na literatura como 2D-I-IIPP, cujo objetivo principal é maximizar o número de itens a serem cortados na área disponível do objeto visando, entre outras coisas, reduzir desperdícios de matérias-primas e agregar ganhos econômicos ao processo de corte.

De um modo geral, os resultados apresentados mostraram que a abordagem proposta é viável, mas ainda necessita de ajustes principalmente no que tange os baixos índices de ocupação, o alto custo de processamento e a existência, mesmo que pequena, de pixels com sobreposição.

Contudo, se do ponto de vista prático a abordagem ainda necessita de ajustes para que possa ser usada, por exemplo, em uma indústria metalúrgica, do ponto de vista acadêmico, essa pesquisa traz como importantes contribuições: uma abordagem para solução do 2D-I-IIPP, que é pouco explorado na literatura, além de um conjunto de instâncias desse problema que podem ser disponibilizadas para que outros pesquisadores possam testar seus métodos. Em adição, os algoritmos desenvolvidos estão disponíveis na forma de pseudocódigos para facilitar novas implementações.

Com o propósito de melhorar a abordagem proposta e dar continuidade ao trabalho sugere-se: i) testar outros operadores de seleção disponíveis na biblioteca GALib, em especial o operador por ranking; ii) analisar o funcionamento do *software* Irace³ e verificar se é possível substituir o planejamento fatorial na determinação dos parâmetros de configuração do AG; iii) utilizar uma biblioteca específica para a manipulação de polígonos como a CGAL⁴ visando a redução do tempo de processamento despendido no cômputo das sobreposições, uma vez que é possível considerar apenas as bordas (ou apenas os vértices) do polígono; iv) inserir um novo critério de parada que seja dinâmico, analisando-se o valor do coeficiente de sobreposição e percentual de ocupação, evitando-se paradas prematuras; v) aplicar o método de dilatação e erosão nos itens, a fim de evitar os problemas com sobreposição de pixels nas bordas dos itens em decorrência das rotações. vi) explorar

³ <http://iridia.ulb.ac.be/irace/>

⁴ <http://www.cgal.org/>

o uso de outros métodos de busca local como o VNS (*Variable Neighborhood Search*) em substituição ao método de descida de encosta empregado. vii) disponibilizar na literatura as instâncias do problema (imagens dos objetos e itens) criadas neste trabalho para que outros autores possam comparar seus métodos; viii) por fim, estender a abordagem proposta para lidar com itens irregulares heterogêneos e comparar os resultados obtidos com os resultados apresentados na literatura;

REFERÊNCIAS

- ALVAREZ-VALDEZ, R.; MARTINEZ, A.; TAMARIT, J. A branch & bound algorithm for cutting and packing irregularly shaped pieces. **International Journal of Production Economics**. v. 45, n. 2, p. 463-477, 2013.
- BALDACCI, R.; BOSCHETTI, M. A.; GANOVELLI, M.; MANIEZZO, V. Algorithms for nesting with defects. **Discrete Applied Mathematics**. v. 163, p. 17-33, 2014.
- BARBOSA, E. **Dicionário: a origem das palavras**. 1 ed., São Paulo: RG Editores, 2011.
- BARROS NETO, B.; SCARMINIO, I. S, ROY, E. B. **Como fazer experimentos: Aplicações na Ciência e na Indústria**. 4 ed., Porto Alegre: Bookman, 2010.
- BEASLEY, D.; BULL, D. R.; MARTIN, R. R. An Overview of Genetic Algorithms; Part 1, Fundamentals. **University Computing. Inter-University Committee on Computing**. v. 15, n. 2, p. 58-69, 1993.
- BELFIORE, P.; FÁVERO, L. P. **Pesquisa Operacional para cursos de Engenharia**. Rio de Janeiro: Elsevier, 2013.
- BENNEL, J. A.; OLIVEIRA, J. F., The geometry of nesting problems: A tutorial. . **European Journal of Operational Research**. v. 184, p. 397-415, 2008.
- BERTRAND, J. W. M.; FRANSOO, J. C. Modelling and Simulation - operations management research methodologies using quantitative modeling. **International Journal of Operations & Production Management**, v. 22, n. 3, p. 241-264, 2002.
- BIRGIN, E. G.; LOBATO, R. D. Orthogonal packing of identical rectangles within isotropic convex regions. **Computers and Industrial Engineering**, v. 59, p. 595-602, 2010.
- BURKE, E. K.; HYDE, M.; KENDALL, G. A Genetic Hiper-Heuristic Approach for Evolving 2-D Strip Packing Heuristics. **Transactions on Evolutionary Computation**. v. 14, n. 6, 2010.
- CAUDILL, M.; BUTLER, C. **Naturally Intelligent Systems**. 5 ed., London: MIT Press, 2000.
- CHWAAB, M.; PINTO, J. C. **Análise de Dados Experimentais II : Planejamento de Experimentos**. Rio de Janeiro : E-Papers. v. 2, 2011.
- COELHO, L. S. Notas em Matemática Aplicada. Fundamentos, Potencialidades e Aplicações de Algoritmos Evolutivos. **Sociedade Brasileira de Matemática Aplicada e Computacional**. São Carlos, BR., 2003.

COELHO, W. B.; MATIAS, I. O.; SHIMODA, E. Aplicação Da Regressão Estatística no Ajuste dos Parâmetros do Algoritmo Genético. **Revista Eletrônica Pesquisa Operacional para o Desenvolvimento**. Rio de Janeiro: SOBRAPO. v. 7, n. 1, p. 1-18, 2015.

CUI, Y.; CUI, Y.; YANG, L. Heuristic for the two-dimensional arbitrary stock-size cutting stock problem. **Computers & Industrial Engineering**. v. 78, p. 195-204, 2014.

CUNICO, M. W. M.; CUNICO, M. M.; MIGUEL, O. G., ZAWADZKI, S. F.; PERALTA-ZAMORA, P. ; VOLPATO, N. Planejamento fatorial: uma ferramenta estatística valiosa para a definição de parâmetros experimentais empregados na pesquisa científica. **Visão Acadêmica**, Curitiba, v. 9, n. 1, p. 23-32, 2008.

DEL VALLE, A. M.; QUEIROZ, T. A.; MIYAZAWA, F. K.; XAVIER, E. C. Heuristics for two-dimensional knapsack and cutting stock problems with items of irregular shape. **Expert Systems with Applications**. v. 39, p. 12589-12598, 2012.

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

EGEBLAD, J. Heuristics for Multidimensional Packing Problems. 2008. 236f. PhD Thesis - Department of Computer Science - University of Copenhagen, 2008.

FURINI, F.; MALAGUTI, E. Models for the two-dimensional two-stage cutting stock problem with multiple stock size. **Computers & Operation Research**. v. 40, n. 8, p. 1953-1962, 2013.

GAREY, M. R.; JOHNSON, D. S. Computers and intractability: a guide to the Theory of NP-Completeness. **W.H. Freeman and Company**. São Francisco, CA, USA, 1979.

GIL, A. C. **Como elaborar projetos de pesquisa**. São Paulo: Atlas, 2010.

GOLDBERG. D. E. Genetic Algorithms in Search, Optimization, and Machine Learning. **Addison-Wesley**. Longman Publishing Co. Boston, MA, USA, 1989.

GOLDBERG, D. E.; DEB, K. **A comparative Analysis of Selection Schemes Used in Genetic Algorithms**. Morgan Kaufmann Publishers: San Mateo, CA. 1991.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 2 ed., New Jersey: Prentice-Hall, 2002.

HADJICONSTANTINO, E.; IORI, M. A hybrid genetic algorithm for the two-dimensional single. **European Journal of Operational Research**. v. 183, p.1150-1166, 2007.

HAN, W.; BENNELL, J. A.; ZHAO, X.; SONG, X. Construction heuristics for two-dimensional irregular shape bin packing with guillotine constraints. **European Journal of Operational Research**. v. 230, p. 495-504, 2013.

JAYARAMAN, S.; ESAKKIRAJAN, S.; VEERAKUMAR, T. **Digital Image Processing**. New Delhi : Tata McGraw Hill, 2009.

JUNQUEIRA, L.; MORABITO, R.; YAMASHITA, D. S.; Three-dimensional container loading models with cargo stability and load bearing constraints. **Computers and Operations Research**. v. 39, n. 1, p. 74-85, 2012.

KALLRATH, J.; REBENNACK, S.; KALLRATH, J.; KUSCHE, R. Solving real-world cutting stock-problems in the paper industry: Mathematical approaches, experience and challenges. **European Journal of Operational Research**. v. 238, p. 374-389, 2014.

KATAOKA, S.; YAMADA, T. Upper and lower bounding procedures for the multiple knapsack assignment problem. **European Journal of Operational Research**. v. 237, p. 440-447, 2014.

KHAMASSI, I.; HAMMAMI M.; GHÉDIRA K. Ant-Q Hyper-Heuristic Approach for solving 2-Dimensional Cutting Stock Problem. **Swarm Intelligence (SIS)**. p. 1-7, 2011.

KIM, H. Y. **ProEikon - Rotinas e programas em C++ para processamento de imagens e visão computacional**. São Paulo. Disponível em: <<http://www.lps.usp.br/~hae/software>>. Acesso em: fev. 2012.

KOZA, J. R. Genetic Programming. **Encyclopedia of Computer Science and Technology**. p. 1-26, 1997.

LAGHA, G. B.; DAHMANI, N.; KRICHEN, S.; Particle Swarm Optimization Approach For Resolving The Cutting Stock Problem. **International Conference on Advanced Logistics and Transport**. Tunisia. p. 259-263, 2014.

LIBRANTZ, A. F. H.; COPPINI, N. L.; BAPTISTA, E. A. ; ARAÚJO, S. A.; ROSA, A. F. C. Genetic Algorithm Applied to Investigate Cutting Process Parameters Influence on Workpiece Price Formation. **Materials and Manufacturing Processes**. v. 26, n. 3, p. 550 - 557, 2011.

LINDEN, R. **Algoritmos Genéticos**. 2 ed., Rio de Janeiro: Brasport, 2008.

LU, H.; HUANG, Y. An efficient genetic algorithm with a corner space algorithm for a cutting stock problem in the TFT-LCD industry. **European Journal of Operational Research**. v. 246, n.1, p. 51-65, 2015.

MALAGUTI, E.; MEDINA DURÁN, E.; TOTH, P. Approaches to real world two-dimensional cutting problems. **Omega**. United Kingdom. v. 47, p. 99-115, 2013

MARCONI, M. A.; LAKATOS, E. M. **Fundamentos de metodologia científica**. 7 ed., São Paulo: Atlas, 2010.

MELIÁN, B.; PÉREZ, J. A. M.; VEGA, J. M. M. Meta-Heurísticas: Una Visión Global. **Revista Iberoamericana de Inteligência Artificial**. v. 7, n.19, p. 7-28, 2003.

MELLO, C. A; **Aplicações**. Disponível em: <<http://www.cin.ufpe.br/~cabm/visao/>>

PV_Aula07_Applications.pdf>. Acesso em: Dez. 2014.

MITCHELL, M. **An Introduction to Genetic Algorithms**. 5 ed., London, England: MIT Press., 1996.

MITCHELL, T. M. **Machine Learning**. McGraw-Hill Science/Engineering/Math. 1997.

OMAN, S.; CUNNINGHAM, P. Using Case Retrieval to Seed Genetic Algorithms. **International Journal of Computational Intelligence and Applications**. v. 1, n. 1, p. 71-82, 2001.

PEDRINI, H.; SCHWARTZ, W. R., **Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações**. São Paulo: Thomson Learning, p. 508, 2008.

PENG, J.; CHU, Z.S. A Hybrid multi-chromosome Genetic Algorithm for the Cutting Stock Problem. **3 rd International Conference on Information Management, Innovation Management and Industrial Engineering**. p. 508-511, 2010.

PINHEIRO, P. R.; JÚNIOR, B. A.; SARAIVA, R. D. A random-key genetic algorithm for solving the nesting problem. **International Journal of Computer Integrated Manufacturing**, p. 1-7, 2015.

PINHO, A. F.; MONTEVECHI, J. A.B .; MARINS, F. A. S.; Análise da aplicação de projeto de experimentos nos parâmetros dos algoritmos genéticos. **Sistema e Gestão**. v. 2, n. 3, p. 319-331. 2007.

POLI, G. I., PUREZA, V. Um algoritmo de busca tabu para o carregamento de contêineres com caixas idênticas. **Gestão Produção**. São Carlos. v.19, n.2, p. 323-336, 2012.

PUREZA, V.; MORABITO, R. Some experiments with a simple tabu search algorithm for the manufacturer's pallet loading problem. **Computers & Operations Research**. v. 33, n. 3, p. 804-819, 2006.

RASKA, P.; ULRYCH, Z. Comparison of Modified Downhill Simplex and Differential Evolution with other Selected Optimization Methods used for Discrete Event Simulation Models. **Procedia Engineering**. v. 100, p. 807-815, 2014.

RICH, E.; KNIGHT, K. **Inteligência Artificial**. 2 ed., Makron Books, 1994.

ROSA, A. F. C. **Um Estudo Comparativo das Técnicas Meta-heurísticas Algoritmo Genético e Simulated Annealing Aplicadas a Sistemas de Apoio à Decisão para Otimização de Parâmetros em Processos de Usinagem**. Dissertação (Mestrado em Engenharia de Produção) - Universidade Nove de Julho, São Paulo, 196f., 2011.

RUSSELL, S. J.; NORVIG, P. **Inteligência Artificial**. 2 ed., Rio de Janeiro: Campus, 2004.

SEDGEWICK, R. **Algorithms in C++**. 3 ed., New Delhi: Pearson Education, 2002.

SHEN, G.; ZHANG, Y. Solving the Stock Reduction Problem with the Genetic Linear Programming Algorithm. **International Conference on Computational and Information Sciences**. p. 561-564, 2010.

SILVA, E.; OLIVEIRA, F.J.; WÄSCHER.,G. 2DCPackGen: A problem generator for two-dimensional rectangular cutting and packing problems. **European Journal of Operational Research**. v. 237, p. 846-856, 2014.

SONG, C.; BAI, S.;JIANG, J.; BAO, L.; An Improved Artificial Fish Swarm Algorithm for Cutting Stock Problem. **9th International Conference on Natural Computation (ICNC)**. China, p. 501-505, 2013.

SOTO. D.; SOTO. W.; PINZÓN. Y. A Parallel Nash Genetic Algorithm for the 3D Orthogonal Knapsack Problem. **International Journal of Combinatorial Optimization Problems and Informatics**. v. 4, n.3, p. 2-10, 2013.

VIANNA, A. C. G.; ARENALES, M. N. O problema de corte de placas defeituosas. **Pesquisa Operacional**. v. 26, n. 2, p. 185-202, 2006.

WALL, M. **GAlib: A C++ Library of Genetic Algorithm Components. Version 2.4**. 1996. Disponível em: <<http://lancet.mit.edu/ga/dist/galibdoc.pdf>>. Acesso em: jul. 2014

WÄSCHER, G., HAUßNER, H.; SCHUMANN, H.; An Improved typology of cutting and packing problems. **European Journal of Operational Research**. v. 183, p. 1109-1130, 2007.

WEISE, T. **Global Optimization Algorithm: Theory and Application**. 2 ed., 2008. Disponível em: < <http://www.it-weise.de/projects/book.pdf> >. Acesso em: Abr. 2014.

YADAV, A.; YADAV, P. **Digital Image Processing**. New Delhi : University Science Press, 2009.

YANG, Q. No Fit Polygon for Nesting Problem Solving with Hybridizing Ant Algorithms. **Journal of Software Engineering and Applications**, v. 7, p. 433-439, 2014.

ZHANG, K.; WEISE, T.; JINLONG, L. Fitness Level based Adaptive Operator Selection for Cutting Stock Problems with Contiguity. **Congress on Evolutionary Computation (CEC)**. China, p. 2539-2546, 2014.

APÊNDICES

APÊNDICE A - Detalhamento dos algoritmos desenvolvidos na abordagem proposta.

Algoritmo 1 - AG para o problema 2D-I-IIPP

```

Algoritmo Corte_2D_I_IIPP (pRepl, p, m, pCross, ImgObj, ImgItem)
Início
  Preparar_imagem_Objeto()
  Preparar_imagem_Item()
  Calcular_Qtd_Max_Itens()
  Preparar_mapa_Phenotype()
  Gerar_semente_inicial()
  Atribui_parâmetros_AG()

  Inicializa_população()
  Insere_solução_inicial_factive()
  Calcula_Valor_FO()

  pConvergence ← 1.0
  nConvergence ← 100

  Enquanto ( pConvergence_calc < pConvergence ) faça
    Selecionar_individuos()
    Cruzar_individuos()
    Mutar_individuos()
    Sobrepor_população_anterior()
    Calcula_Valor_FO()
    Se ( nGerações > 300 e sem_melhoria > 5 ) então
      Descida_Encosta(Melhor_indv)
    Fim-Se
  Fim-Enquanto
  Exibir_melhor_individuo()
Fim

```

Fonte: O autor

As funções *Preparar_imagem_Objeto()* e *Preparar_imagem_Item()*, são responsáveis por preparar as imagens em um formato que utiliza a estrutura de dados do tipo matriz além de inicializar parâmetros específicos.

Para iniciar a execução do AG é necessário determinar qual a quantidade máxima de itens que serão manipulados. A função *Calcular_Qtd_Max_Itens()* é responsável por verificar o limite máximo de itens que poderão ser posicionados sobre o objeto e está representado pela equação 5.

Na função *Preparar_mapa_Phenotype()* são atribuídos os limites inferiores e superiores para os genes tx , ty , θ e st . Neste trabalho os limites de tx e ty são determinados pelas dimensões fixas da imagem do objeto, desta forma obtém-se os valores de tx e ty entre 1 e 63. O limite inferior da rotação (θ) é de 0° e o superior 315° , observando-se um intervalo permitido de 45° . Quanto ao status (st) os valores possíveis são 0 ou 1, sendo 1 para posicionar o item na imagem e 0 não posicionar item na imagem. As equações 1, 2 e 3 demonstram o cálculo desses valores.

A função *Gerar_semente_inicial()* inicializa uma variável *seed* com o propósito de indicar um ponto inicial para a geração dos números aleatórios. Para a execução dos experimentos é necessário fixar o valor da semente, a fim de verificar como o AG evolui para os diferentes objetos e itens, considerando-se o mesmo ponto de partida.

Conforme apresentado anteriormente, a definição dos parâmetros para o AG, constitui um passo muito importante, visto que pode comprometer os resultados dos experimentos. A função *Atribui_parâmetros_AG()* atribui os parâmetros iniciais do AG utilizando os valores apresentados na tabela 6.

Inicializa_população() é uma função do AG que gera uma população inicial aleatória de acordo com o valor contido no parâmetro p . Neste caso, serão gerados 50 indivíduos, ou seja, 50 possíveis soluções para o problema de corte.

A função responsável por inserir uma solução inicial factível (SIF) é a *Inserir_solução_inicial_factivel()*. A inserção ocorre por meio de um arquivo texto contendo um cromossomo com uma solução factível.

Para verificar a qualidade das soluções geradas pelo AG executa-se a função *Calcula_valor_FA()*. Esta função é responsável por informar ao AG o valor da FA de todos os indivíduos *indv* de uma população p .

Com os valores da FA da primeira geração, inicia-se a aplicação dos operadores do AG. Este processo ocorrerá até que um ponto de parada seja atingido. Neste algoritmo, o critério de parada é a convergência. Na biblioteca GAlib, as medidas de convergência utilizam a pontuação do melhor indivíduo. Neste caso, a convergência ocorre quando o melhor indivíduo, das 100 gerações anteriores, possuir o mesmo valor do melhor indivíduo atual.

O operador de seleção do AG descrito como *Selecionar_individuos()* fará a seleção de $(1 - pRepl) \cdot p$ indivíduos. Neste caso como o parâmetro $pRepl$ é 0,7 e p é 50, serão selecionados 14 indivíduos que farão parte da próxima geração e 36 indivíduos serão substituídos. A seleção é proporcional, ou seja, o indivíduo que

possui maior valor de FA possui maior probabilidade de ser escolhido para compor a próxima geração.

O próximo operador é o de cruzamento, descrito no algoritmo como *Cruzar_individuos()*. Este operador seleciona $(pCross * p)/2$ pares de indivíduos da população atual. Neste caso seleciona-se 18 pares de indivíduos utilizando-se a mesma probabilidade da seleção. Para cada par serão gerados dois novos indivíduos, totalizando 36 novos indivíduos para a nova geração.

O último operador a ser executado é o de mutação descrito como *Mutar_individuos()*. Este operador seleciona m indivíduos da nova geração e inverte um bit do indivíduo selecionado.

Por fim, o AG substituirá toda a população atual pela nova geração, descrita como a função *Sobrepor_população_anterior()*. Esta nova geração será avaliada pela função *Calcula_valor_FO()*. Caso se atinjam 300 gerações, uma heurística será executada. Este laço se repetirá até que a convergência seja alcançada.

Quando o AG converge, terminando a sua execução, obtêm-se os valores de tx , ty , θ e st de todos os itens, que correspondem à melhor solução encontrada.

Algoritmo 2 - Calcula Valor da FO.

```

Função Calcula_valor_FO()
Início
  Var indv : inteiro
   $i \leftarrow 0$ 
  Enquanto ( indv <  $p$  ) faça
    Selecionar_itens_status_1indv()
    Valor_FO  $\leftarrow$  Avaliar_itens_selecionadosindv(Qtd_itens_solucao)
  Fim-Enquanto
Fim

```

Fonte: O autor

No algoritmo 2, observa-se o pseudocódigo da função *Calcula_valor_FO* no qual os indivíduos *indv* de uma população p são separados para serem avaliados. Além de separar os itens, esta função também é responsável por informar ao AG o valor da FO de todos os indivíduos *indv* de uma população p .

Cada indivíduo *indv* possui T_i itens e cada item i possui um conjunto $(tx, ty, \theta$ e $st)$ gerados pelo AG, porém somente os itens com $st=1$ serão analisados. Esta análise ocorre na função *Avaliar_itens_selecionados* que além de efetuar o cálculo da FO,

descrito na equação 8, também é a responsável por posicionar o item i no objeto Img_P .

Algoritmo 3 - Avaliar itens

```

Função Avaliar_itens_selecionados(Qtd_itens_solucao)
Início
Var  $i$  : inteiro

 $i \leftarrow 0$ 

Enquanto ( $i < Qtd\_itens\_solucao$ ) faça
    Prepara_rotação_item $i$ 
    Posiciona_item $i$ _objeto()
    Verifica_Calcula_item $i$ _fora_objeto()
    Verifica_Calcula_item $i$ _sobreposicao()
    Verifica_Calcula_item $i$ _factivei()
    Totaliza_valores_solucao()
Fim-Enquanto
Calcular_sobreposicao_normalizada()
Calcular_fora_normalizado()
Calcular_Qtd_itens_normalizado()
Calcular_Distancia_normalizada()
Calcular_FO()
Retornar FO
Fim

```

Fonte: O autor

A função representada no algoritmo 3, é o cerne do algoritmo proposto, pois é neste local que todo o processamento de imagem e da FO ocorrem.

Cada item i é posicionado no objeto Img_P conforme o valores tx , ty , θ e st , gerados pelo AG. O posicionamento do item i e o cálculo dos coeficientes c_s e c_c acontecem simultaneamente. As equações 9 e 10 descrevem este cálculo.

O cálculo do coeficiente de distância c_d , expresso na equação 11, ocorrerá após o posicionamento de todos os itens i no objeto Img_P .

O cálculo da FO descrito na equação 8 corresponde à média ponderada de todos os coeficientes c_s , c_c e c_d .

Ao final deste processo o valor da FO de cada indivíduo é enviado ao AG finalizando a função `Avaliar_itens_selecionados()`.

Algoritmo 4 - Descida de Encosta

```

Função Descida_Encosta (Indv)
Início
Var FO_0, FO_1, i, Qtd, acao_tx, acao_ty, acao_rot : inteiro

    Selecionar_itens_status_1indv()
    FO_0 ← Calcula_Valor_FO()
    i ← 0
    Qtd ←  $Q_i / 3$ 
    Selecionar_Qtd_itens_randomicamente()
    Enquanto (i < Qtd) faça
        Acao_tx ← Selecciona_movimento()
        Acao_ty ← Selecciona_movimento()
        Acao_rot ← Selecciona_movimento()
        Movimentar_item()
        FO_1 ← Calcula_Valor_FO()
        Enquanto (FO_1 < FO_0) faça
            FO_0 ← FO_1
            Alterar_posição_item()
            Movimentar_item()
            FO_1 ← Calcula_Valor_FO()
        Fim-Enquanto
    i ← i + 1
Fim-Enquanto
Fim

```

Fonte: O autor

A função *Descida_Encosta()* é uma heurística desenvolvida e aplicada com o objetivo de encontrar uma solução melhor e mais rapidamente na vizinhança. Neste caso, o refinamento da solução ocorre somente sobre o melhor indivíduo da população atual. Deste indivíduo apenas 1/3 dos itens serão selecionados randomicamente. Esta restrição é devido ao alto custo computacional empregado.

A busca na vizinhança ocorre alterando-se randomicamente os valores de tx , ty e θ . As alterações, no indivíduo atual, somente serão efetivadas se os novos valores atribuídos às variáveis apresentarem um decremento no valor da FO.

Observa-se que essa heurística será aplicada somente após um período de 5 gerações sem melhorias no valor da FO e quando o número de gerações for superior a 300. Esses valores são utilizados porque observou-se nos experimentos realizados, que o AG evolui pouco após atingir 300 gerações e além disso aplicar a cada 5 gerações sem melhorias minimiza os impactos do processamento.

APÊNDICE B - Publicações resultantes das pesquisas realizadas durante o mestrado:

1.0 TRABALHO COMPLETO PUBLICADO EM ANAIS DE EVENTOS

1.1 GAVA, M. C. V.; FILHO, A. G.; ROSARIO, C. D. P.; ARAUJO, S. a. An approach based on genetic algorithm for solving 2d cutting problem with items of irregular shapes. In: **International Conference on Management of Computational and Collective Intelligence in Digital Ecosystems (MEDES 2015)**. Caraguatatuba/SP, 2015.

1.2 GAVA, M. C. V.; LIBRANTZ, A. F. H.; ARAUJO, S. A. Corte em chapas metálicas com itens irregulares usando algoritmos genéticos e Processamento de imagens digitais In: **CILAMCE 2014, XXXV Iberian Latin American Congress on Computational Methods in Engineering**. Fortaleza/CE, v. 01, p.1-160, 2014.

2.0 RESUMO PUBLICADO EM ANAIS DE EVENTOS

2.1 GAVA, M. C. V.; ROSARIO, C. D. P.; DOM PEDRO, T. P.; FILHO, A. G.; ARAUJO, S. A. Aplicação de algoritmos genéticos na resolução do problema de corte bidimensional com itens irregulares idênticos. In: **XII Encontro de Iniciação Científica**. São Paulo, 2015.

2.2 GAVA, M. C. V.; NEVES, E. M. C.; DOM PEDRO, T. P.; FERREIRA, G. P.; ANDRIGHETTI, E.; ARAUJO, S. A. Resolução do problema de corte em chapas metálicas com itens irregulares Usando algoritmos genéticos e processamento de imagens digitais. In: **XI Encontro de Iniciação Científica**, Pesquisa Científica como Formação Profissional e Social. São Paulo, v.1, p. 257, 2014.