

ALEXANDRE ROMANELLI

**PROPOSTA DE ABORDAGEM HEURÍSTICA PARA
GERAÇÃO DE LEIAUTE DE CORTE IRREGULAR
BIDIMENSIONAL**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA
MINAS GERAIS - BRASIL
2013

**Ficha catalográfica preparada pela Seção de Catalogação e
Classificação da Biblioteca Central da UFV**

T

Romanelli, Alexandre, 1977-
R758p Proposta de abordagem heurística para geração de leiaute de
2013 corte irregular bidimensional / Alexandre Romanelli. – Viçosa,
MG, 2013.
xii, 113 f. : il. ; 29 cm.

Inclui apêndice.

Orientador: Mauro Nacif Rocha.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f.110-113.

1. Métodos orientados à objetos (Computação).
2. Heurística. 3. Geometria. I. Universidade Federal de Viçosa.
Departamento de Informática. Programa de Pós-Graduação em
Ciência da Computação. II. Título.

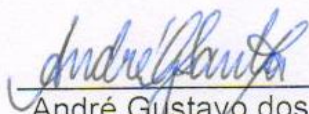
CDD 22. ed. 005.117

ALEXANDRE ROMANELLI

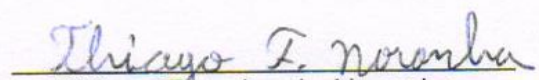
**PROPOSTA DE ABORDAGEM HEURÍSTICA PARA GERAÇÃO DE
LEIAUTE DE CORTE IRREGULAR BIDIMENSIONAL**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

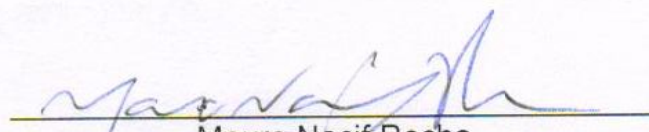
APROVADA: 26 de julho de 2013.



André Gustavo dos Santos
(Coorientador)



Thiago Ferreira de Noronha



Mauro Nacif Rocha
(Orientador)

*Este trabalho é dedicado aos meus pais
Luiz Roberto (em memória) e Cenyra,
por todo amor que recebi deles.*

“A nossa ciência é parcial, a nossa profecia é imperfeita”
(I Coríntios 13, 9)

AGRADECIMENTOS

Muitas pessoas contribuíram para que eu pudesse perseverar nesta jornada de estudos e pesquisa até o final. Agradeço sinceramente a todas estas. Mas infelizmente não sou capaz de citar os nomes de todos neste curto texto. Deixarei então inicialmente meu agradecimento a todos que a minha memória foi incapaz de recordar neste curto momento que posso dedicar à redação desta página.

Tenho muita gratidão pela minha mãe e por meus irmãos, pelo apoio e colaboração durante os anos de envolvimento com este projeto. E a meu pai que me inspirou a estudar. Sem a minha família, seria impossível esta realização. Devo incluir neste conjunto de pessoas especiais a minha filha, que mesmo sem querer forneceu o estímulo mais vigoroso, que me impulsionou quando minha vontade ameaçava faltar.

Também sou grato aos colegas de trabalho, principalmente os professores Ricardo e Jocimar, pelo estímulo e apoio. Agradeço também aos meus alunos. É para melhor servi-los que busco ampliar e aperfeiçoar minhas capacidades.

Agradeço também àqueles que estiveram próximos a mim, ou virtualmente próximos, amparando e estimulando durante este curso. Agradeço ao meu orientador Mauro, pela grande paciência e estímulo nos momentos críticos. Aos professores do DPI que colaboraram para expandir minha visão de mundo, como o José Luis e o André.

Não poderia deixar de agradecer aos amigos de estudo, pessoas extraordinárias, principalmente o Ronney e o Glauber. Nosso trio ficou marcado na minha memória e é parte indissolúvel de quem eu sou.

E, acima de tudo, agradeço a Deus de todo o meu coração. É Nele que eu vivo e é a quem devo todas as minhas experiências na vida.

Sumário

Lista de Figuras	vii
Lista de Tabelas	x
Resumo	xi
Abstract	xii
1 Introdução	1
1.1 Abordagens computacionais ao problema de corte e empacotamento . . .	2
1.2 Definição e tipologia de problemas de corte e empacotamento	3
1.3 Representações dos elementos do problema	6
1.3.1 Aproximação por poligonização	7
1.3.2 Aproximação por discretização	9
1.3.3 Outros métodos para representação	11
1.4 Objetivos	12
1.5 Metodologia	13
1.6 Organização do documento	14
2 Trabalhos relacionados	15
2.1 Abordagens para solução	15
2.1.1 Invólucro de posicionamento e conceitos relacionados	16
2.1.2 Abordagens para determinar solução exata	22
2.1.3 Abordagens heurísticas para determinar solução	29
2.1.4 Compactação de leiautes com programação matemática	36
2.2 Estruturas de testes e de comparativos de desempenho	38
2.3 Melhores resultados conhecidos	41
2.4 Considerações sobre o capítulo	43

3	Proposta de nova abordagem construtiva para <i>C&P</i>	44
3.1	Algoritmo proposto	44
3.1.1	Exemplos de aplicação do algoritmo	48
3.2	Heurísticas sobre sequências	52
3.3	Cálculo do invólucro de posicionamento	55
3.3.1	Exemplos de cálculos de <i>NFP</i>	57
3.4	Determinação da posição do próximo item	62
3.5	Compactação de leiaute	66
3.6	Considerações sobre o capítulo	72
4	Experimentos computacionais e discussão dos resultados	73
4.1	Instâncias selecionadas	73
4.2	Testes realizados	77
4.3	Resultados dos testes	79
4.3.1	Perspectiva de regras de posicionamento	83
4.3.2	Perspectiva de critérios de ordenação	86
4.3.3	Perspectiva de compactação	90
4.4	Discussão de resultados	95
5	Conclusão	97
5.1	Trabalhos futuros	98
	Apêndice A Melhores leiautes construídos	100
	Referências Bibliográficas	110

Lista de Figuras

1.1	Aproximação de forma irregular com polígono envolvente.	8
1.2	Aproximação de forma irregular com discretização.	11
2.1	NFP entre dois polígonos A e B (i); possíveis translações do polígono B guiadas pelo NFP (ii).	17
2.2	Exemplo de caso simples de operação de Minkowski sobre polígonos orientados.	21
2.3	Definições de atributos de polígonos usados nos modelos matemáticos.	24
2.4	Modelo matemático de Luzzi para o problema de $C\&P$. Fonte: adaptado de Luzzi [2002], p. 40.	24
2.5	Restrições de Luzzi para garantir a não-sobreposição de peças. Fonte: adaptado de Luzzi [2002], p. 42.	26
2.6	Modelo matemático de Gomes para o problema de $C\&P$. Fonte: adaptado de Gomes [2005], p. 84.	27
2.7	Algoritmo de busca local adaptado de Gomes [2005].	33
2.8	Algoritmo de busca por recozimento simulado adaptado de Martins [2007].	34
2.9	Algoritmo GRASP adaptado de Gomes [2005].	35
2.10	Modelo matemático de compactação adaptado de Gomes & Oliveira [2006].	37
2.11	Aspectos observados em resultados: comprimento (2.36) e aproveitamento de área (2.37).	40
3.1	Método construtivo proposto.	45
3.2	Progresso da execução do algoritmo construtivo proposto, aplicado à instância Albano.	49
3.3	Progresso da execução do algoritmo construtivo proposto, aplicado à instância Shirts.	51
3.4	Comparação dos resultados obtidos com variações do método construtivo, aplicadas à instância Shirts.	51

3.5	Cálculo da área de um polígono com trapézios.	53
3.6	Um polígono (<i>i</i>), seu invólucro convexo (<i>ii</i>) e seu retângulo envolvente (<i>iii</i>).	54
3.7	Primeiro exemplo: dois polígonos selecionados para calcular o <i>NFP</i> , sendo um convexo (<i>A</i>) e um côncavo (<i>B</i>).	58
3.8	Primeiro exemplo: coleção de arestas de <i>A</i> e $-B$ ordenadas por inclinação.	58
3.9	Primeiro exemplo: polígono complexo gerado pela soma de Minkowski e o seu contorno externo.	59
3.10	Primeiro exemplo: posicionamento do <i>NFP</i> e dos polígonos <i>A</i> e <i>B</i> para aplicação.	59
3.11	Segundo exemplo: dois polígonos côncavos selecionados para calcular o <i>NFP</i>	60
3.12	Segundo exemplo: coleção de arestas de <i>A</i> e $-B$ ordenadas por inclinação.	60
3.13	Segundo exemplo: polígono complexo gerado pela soma de Minkowski e o seu contorno externo.	61
3.14	Segundo exemplo: posicionamento do <i>NFP</i> e dos polígonos <i>A</i> e <i>B</i> para aplicação.	62
3.15	Exemplos da seleção de arestas do <i>NFP</i>	69
3.16	Exemplo da aplicação do procedimento iterativo de compactação.	72
4.1	Combinações de parâmetros dos experimentos realizados.	78
4.2	Resultados em intervalos agrupados por regras para cada instância.	84
4.3	Distribuição de aproveitamento por regra, para a instância <i>marques</i>	85
4.4	Distribuição de aproveitamento por regra, para a instância <i>blaz2</i>	86
4.5	Distribuição de aproveitamento por critério de ordenação, para a instância <i>dagli</i>	88
4.6	Distribuição de aproveitamento por critério de ordenação, para a instância <i>shirts</i>	89
A.1	Melhor leiaute para instância <i>albano</i>	100
A.2	Melhor leiaute para instância <i>blaz2</i>	101
A.3	Melhor leiaute para instância <i>dagli</i>	102
A.4	Melhor leiaute para instância <i>dighe1</i>	103
A.5	Melhor leiaute para instância <i>dighe2</i>	103
A.6	Melhor leiaute para instância <i>fu</i>	104
A.7	Melhor leiaute para instância <i>mao</i>	105
A.8	Melhor leiaute para instância <i>marques</i>	105
A.9	Melhor leiaute para instância <i>shapes0</i>	106
A.10	Melhor leiaute para instância <i>shapes1</i>	107
A.11	Melhor leiaute para instância <i>shapes2</i>	107

A.12 Melhor leiaute para instância shirts.	108
A.13 Melhor leiaute para instância swim.	109
A.14 Melhor leiaute para instância trousers.	109

Lista de Tabelas

2.1	Instâncias de referência para o problema de <i>C&P</i> em estudo.	41
2.2	Melhores resultados encontrados na literatura para as instâncias de referência.	42
4.1	Atributos das instâncias usadas para testes.	75
4.2	Dados complementares das instâncias: área.	76
4.3	Resumo dos resultados dos testes com o método construtivo proposto (tempo e aproveitamento) e comparativo com os melhores resultados da literatura.	80
4.4	Detalhes dos testes com os melhores resultados.	81
4.5	Parâmetros com as melhores médias de aproveitamento.	82
4.6	Dados da eficácia de critérios de ordenação para as instâncias.	87
4.7	Relação entre critérios de ordenação e atributos de instâncias.	90
4.8	Visão geral dos resultados dos testes sem uso da compactação.	91
4.9	Contagem de resultados dos testes com uso da compactação (C) comparados a testes com as mesmas configurações, mas sem uso desta sub-rotina (NC).	92
4.10	Comparação entre as médias de aproveitamentos e consumo de tempo obtidos sem compactação e com a compactação intercalada.	93
4.11	Resultados médios de aproveitamento distribuídos por tipo de agrupamentos entre as compactações.	95

Resumo

ROMANELLI, Alexandre, M.Sc., Universidade Federal de Viçosa, julho de 2013. **Proposta de abordagem heurística para geração de leiaute de corte irregular bidimensional.** Orientador: Mauro Nacif Rocha. Coorientadores: José Elias Claudio Arroyo e André Gustavo dos Santos.

O problema abordado neste trabalho consiste em fornecer um leiaute formado pelo posicionamento de um conjunto de itens sobre a área de um objeto receptor, de modo que todos os itens fiquem completamente posicionados no interior da superfície do recipiente, e que não haja sobreposição entre os itens. É tratado o contexto mais específico em que apenas duas dimensões são consideradas dos objetos relacionados, e o recipiente é retangular, supostamente limitado apenas por sua largura. Neste trabalho, é feita a proposta de uma nova abordagem construtiva para fornecer soluções para o problema de corte e empacotamento, formada pela combinação de elementos de outras técnicas de resolução do mesmo problema. O novo método inclui heurísticas para ordenação da sequência de itens a posicionar, regras de determinação da posição que cada item deve ocupar, e uma técnica de compactação de leiaute baseada em programação linear. Este trabalho contém a descrição detalhada desta abordagem, bem como os experimentos realizados e os resultados obtidos. Os melhores resultados estão comparados a outros métodos construtivos encontrados na literatura, superando-os para 11 das 14 instâncias avaliadas. A contribuição desta pesquisa está na introdução de um novo método para a resolução de problemas de corte e empacotamento, possibilitando a realização de outros trabalhos que explorem a adaptação da abordagem proposta para sua aplicação em procedimentos de busca para obter melhores soluções.

Abstract

ROMANELLI, Alexandre, M.Sc., Universidade Federal de Viçosa, July 2013. **Proposal for a heuristic approach for generation of two-dimensional irregular cutting layout.** Adviser: Mauro Nacif Rocha. Co-Advisers: José Elias Claudio Arroyo and André Gustavo dos Santos.

The problem addressed in this dissertation consists in providing a layout formed by positioning a set of items on the area of a large container object, so that all items are completely positioned in the interior surface of the container, and that there is no overlap between the items. We dealt with a more specific context in which only two dimensions are considered for the related objects, and the container is rectangular, supposedly limited only by its width. In this work, we propose a new constructive approach to provide solutions to the cutting and packing problem, formed by combining elements of other techniques for the same problem. The new method includes heuristics for ordering the sequence of items being placed, rules for determining the position that each item should occupy, and a layout compaction technique based on linear programming. We present a detailed description of this approach, as well as some experiments and results. The best results obtained are compared to other construction methods found in literature, overcoming them in 11 out of 14 instances. The contribution of this research is an introduction of a new method for solving cutting and packing problems, enabling the conduction of other works that can explore and adapt the proposed approach for application procedures aiming for better solutions.

Capítulo 1

Introdução

No cotidiano das atividades humanas de transformação de matérias em produtos, encontram-se tarefas que são feitas sem haver procedimentos formais que sirvam como guias. Investigações científicas podem ser aplicadas para identificar os métodos que as pessoas utilizam mentalmente para resolver problemas relacionados a essas tarefas. Como resultado, são delineados modelos e algoritmos que buscam representar o conhecimento através de alguma linguagem que permita capacitar outras pessoas para realizar as mesmas tarefas, ou explorar possíveis melhorias dos processos, ou até desenvolver mecanismos que os automatizem.

Uma dessas tarefas é a que visa reduzir o desperdício de material em operações de corte para produção de itens, que geralmente são partes usadas como componentes de algum produto. Há muitos segmentos da indústria em que ocorrem casos deste tipo. Alguns exemplos são as indústrias moveleira, têxtil, metal-mecânica e de rochas ornamentais. Nestas, um problema recorrente é planejar como cortar o material para obter os itens necessários, de modo que haja o mínimo de sobras do material que não poderão ser reaproveitadas.

Outra tarefa que se observa é organização de itens sobre a superfície ou no interior de algum recipiente. Isto pode ser observado em atividades de armazenamento e transporte de produtos ou materiais. Nestes casos, o problema é definir a melhor forma de posicionamento dos itens para aproveitar ao máximo o espaço disponível.

Essas duas tarefas, embora tenham características distintas, apresentam questões que se assemelham. Em ambas, o fundamental é definir como dispor itens no interior de algum objeto, seja para planejar o corte de um material, ou para organizar um depósito ou carga. Com isso, os problemas desses tipos são frequentemente denominados de problemas de corte e empacotamento.

1.1 Abordagens computacionais ao problema de corte e empacotamento

Os problemas de corte e empacotamento são estudados desde meados do século XX, e muitos métodos foram propostos para automatizar o processo de disposição de itens sobre a área de um objeto. Algumas características são comuns a esses métodos, como a necessidade de representar os itens e as operações de posicionamento por meio da geometria. Além disso, a solução para o problema é uma combinação de posições para os itens, que recebe o nome de plano de corte ou leiaute.

Computacionalmente, uma grande dificuldade para tratar de problemas deste tipo está relacionada com a análise combinatória. Segundo Hopper [2000], o problema de corte e empacotamento bidimensional irregular pertence à classe de problemas NP-completo. Uma característica comum dos problemas desta classe é a complexidade de tempo em função exponencial ao tamanho da instância. Isto faz com que as abordagens aplicáveis sejam voltadas, principalmente, à determinação de soluções suficientemente boas, mas não necessariamente ótimas.

Outra barreira para a aplicação de abordagens computacionais para os processos de corte e empacotamento é a necessidade de fazer os cálculos geométricos de modo eficiente. Por ser um problema combinatorial, muitas abordagens avaliam um número grande de possíveis soluções até decidir por uma que seja suficientemente boa. Quanto mais simples e rápidos forem os cálculos geométricos, menos tempo será consumido com isto, e o espaço de soluções válidas exploradas possivelmente será maior. Consequentemente, melhores resultados poderão ser obtidos e maior será o potencial de aproveitamento do método na indústria.

As técnicas usadas para resolver este problema podem ser classificadas segundo vários aspectos. Por exemplo, há abordagens que buscam determinar soluções ótimas para o problema, como encontradas em Gomes [2005] e Luzzi [2002], enquanto outras empregam métodos heurísticos para definir soluções aceitáveis. Com relação aos métodos heurísticos, existem os que codificam as soluções em sequências de itens, como os encontrados em Albano & Sapuppo [1980], Bennell & Song [2010], Dowsland et al. [2002], Gomes & Oliveira [2001] e Hopper [2000], e outros que trabalham diretamente sobre as coordenadas do posicionamento de cada item, como os encontrados em Li & Milenkovic [1995a] e Gomes & Oliveira [2006].

1.2 Definição e tipologia de problemas de corte e empacotamento

Antes de fazer um levantamento de trabalhos que possam contribuir com informações aproveitáveis, foi considerado importante conhecer um sistema de classificação que permite identificar os problemas correlatos. Assim, é possível selecionar os trabalhos que abordam um certo subconjunto dos problemas de corte e empacotamento que sejam efetivamente relevantes. Para alguns casos, mesmo que os problemas sejam diferentes, as características em comum que os levam a uma mesma classificação podem indicar possibilidades de aplicação das mesmas abordagens de solução. Esta possibilidade de intercâmbio de avanços técnicos justifica a classificação desses problemas e as tarefas de consultar e atualizar esta classificação, sempre que isto for útil para um novo trabalho ou relevante para os que poderão ser conduzidos no futuro.

Para iniciar, é discorrida uma breve definição do problema em estudo. Historicamente, de acordo com Haessler & Sweeney [1991], a primeira formulação conhecida de um problema de corte de estoque foi feita por Kantorovich no ano de 1939, e os primeiros avanços significativos foram os trabalhos de Gilmore e Gomory, já na década de 1960. Albano & Sapuppo [1980] descreveram o problema de alocação de formas regulares ou irregulares como a tarefa de determinar como um conjunto de peças será ajustado no interior de uma chapa, visando maximizar o aproveitamento dos posicionamentos, ou minimizar o desperdício, que é composto de toda região livre entre as peças. Os autores daquele trabalho enfatizam a analogia das operações de corte de material e de organização de espaço. Dyckhoff [1990] sugere uma estrutura lógica para o problema de forma genérica, ampliando para várias outras aplicações práticas da indústria. A descrição desta estrutura, fornecida por esse autor, é resumida a seguir.

Sejam dois grupos de elementos que definem corpos geométricos de figuras fixas, definidas no espaço \mathbb{R}^n , $n > 0$. Um dos grupos representa um estoque de elementos denominados “objetos (grandes)” e o outro grupo é uma lista de elementos denominados “itens (pequenos)”. O processo de corte ou empacotamento consiste em fazer leiautes que são combinações geométricas de itens pequenos posicionados nos objetos grandes. As figuras do leiaute que não pertencem a itens pequenos são consideradas perdas.

Dyckhoff [1990] descreve que nos problemas de empacotamento e carregamento os objetos grandes são vazios, como espaços úteis de veículos, contêineres ou caixas, entre outros tipos de espaços. Assim, para estes problemas, o posicionamento de um item pequeno nesses espaços, ocupando uma região, é uma ação que pode ser interpretada como um corte do espaço disponível. O autor conclui a analogia em sentido reverso, carac-

terizando o corte de um item a partir de uma chapa como o posicionamento do mesmo no espaço do material para corte. Isto caracteriza uma relação de dualidade entre essas duas tarefas distintas. Na revisão publicada em Dowsland & Dowsland [1995], com relação aos problemas de arranjo de peças sobre uma região ou material, os autores relatam a existência de aplicações diferentes em indústrias diferentes que apresentam características semelhantes, porém com objetivos e restrições distintas. Conseqüentemente, os esforços para aplicação de computadores na automação dessas aplicações são bem específicos. Isto impõe uma barreira para a troca de informações entre os pesquisadores que buscam soluções, já que naturalmente os canais de divulgação de conhecimento adotados podem variar, fluindo para aqueles compartilhados pelas organizações que atuam nos mesmos segmentos produtivos, e provavelmente ignorados por outras, de segmentos variados, que lidam com problemas comparáveis.

Dyckhoff [1990] introduz uma tipologia para os problemas de *C&P*, na qual as descrições de problemas, bem como as abordagens para solução, encontradas em publicações científicas são classificadas de acordo com características que abstraem elementos fundamentais, segundo os quais é possível identificar semelhanças suficientes para suportar o aproveitamento de abordagens de solução em problemas distintos. As características essenciais sugeridas naquele trabalho são dimensionalidade, tipo de atribuição, variedade de objetos grandes, e variedade de itens pequenos.

Ainda segundo aquele autor, a dimensionalidade é definida como o menor número de dimensões de números reais necessárias para descrever a geometria dos leiautes. O problema clássico da mochila, por exemplo, é unidimensional, enquanto o carregamento de um contêiner é comumente tridimensional. Um exemplo de problema com número de dimensões superior a três, de acordo com Lorie and Savage [1955] *apud* Dyckhoff [1990], é o orçamento de capital multi-período. Qualquer problema de carregamento tridimensional em que a sequência de ações no tempo seja relevante para a descrição da solução, pode-se considerar uma quarta dimensão para identificar dados que caracterizem este aspecto.

O tipo de atribuição é uma característica restritiva do problema. Seu significado é indicar a condição relativa ao “uso” dos itens e dos objetos no leiaute produzido. De acordo com a natureza do problema, é possível encontrar variações como ser necessário incluir todos os itens e todos os objetos, ou apenas uma seleção de itens e uma seleção de objetos, ou ainda permutações combinadas dessas variações. Retomando o exemplo do problema clássico da mochila, tem-se a restrição de uso de todas as mochilas disponíveis, mas apenas uma seleção de itens deve ser feita.

As características de variedades de objetos e itens descrevem informações relevantes que podem causar impacto significativo nas estratégias de resolução dos problemas.

São informações sobre a relação entre as geometrias dos objetos disponíveis e dos itens a posicionar. Existem problemas que apresentam apenas um objeto grande, ou vários, de uma única forma ou de formas diversas. Por outro lado, há problemas em que há apenas itens pequenos de formas diferentes, ou muitos itens pequenos de uma mesma forma, ou muitos itens de muitas formas diferentes, ou ainda muitos itens de um pequeno número de formas diferentes. As expressões “formas iguais” e “formas diferentes” serão usadas livremente neste trabalho no sentido de indicar “congruentes” e “não congruentes”, respectivamente.

Segundo a tipologia proposta por Dyckhoff, o problema de posicionamento bidimensional de formas irregulares abordado no presente trabalho é classificado como 2/V/O/R, no formato [dimensionalidade]/[atribuição]/[variação de objetos grandes]/[variação de itens pequenos]. Neste caso, o problema abordado é bidimensional (2), trata de atribuir todos os itens a uma seleção de objetos (“V”¹), possui apenas um objeto grande (“O” do termo em inglês “one”), e muitos itens de um relativamente pequeno número de diferentes formas (“R” do termo em inglês “relatively”).

Entretanto, algumas instâncias do mesmo problema posicionam-se em outra classificação, como 2/V/O/F. Neste caso, “F” é uma referência a poucos itens de formas diferentes (“F” da palavra “few” em inglês). Além disso, não é claramente definido o método de distinção entre as quantificações “muitos” e “poucos”. Essas e outras motivações levaram à revisão desta tipologia, elaborada por Wascher et al. [2007], buscando eliminar ambiguidades e imprecisões da tipologia proposta por Dyckhoff, além de atualizar a base de conhecimento sobre os problemas de *C&P*.

Esta segunda tipologia aproveita aspectos observados na primeira, como o tipo de atribuição, as variedades de itens pequenos e de objetos grandes e a dimensionalidade. Porém, são propostos novos tratamentos para esses aspectos, exceto a dimensionalidade, que permanece como definida anteriormente. Wascher et al. [2007] definem o tipo de atribuição com dois possíveis objetivos, que são a maximização da saída ou a minimização da entrada. No primeiro caso, são classificados os problemas que lidam com a necessidade de selecionar um conjunto de itens que serão posicionados em um determinado número de objetos, de modo que o valor associado aos itens posicionados seja máximo. Por outro lado, no segundo caso são classificados os problemas nos quais deve-se posicionar um determinado número de itens em um ou mais objetos disponíveis, a serem selecionados em número mínimo suficiente.

¹A letra “V” é oriunda da palavra alemã “*verladeproblem*”, que significa “problema de carregamento”. A variante desta característica recebe a letra “B”, da palavra alemã “*beladeproblem*”. O contraste entre essas palavras, “*verlade*” e “*belade*”, é que a primeira remete ao carregamento de uma seleção de objetos, enquanto a segunda indica que os objetos devem ser carregados com uma seleção de itens.

A variedade de itens pequenos é um aspecto sob o qual os problemas de *C&P* são classificados em grupos que apresentam itens pequenos idênticos, ou variedade heterogênea fraca, ou variedade fortemente heterogênea. Com respeito à variedade de objetos grandes, a tipologia de Wascher et al. [2007] apresenta dois agrupamentos, que são o dos problemas em que há apenas um objeto grande, e o dos problemas em que são considerados vários objetos grandes. Quando há apenas um objeto grande, ainda são consideradas duas categorias. Uma é para os problemas em que o objeto grande possui todas as dimensões fixas. Outra é para os problemas em que o objeto grande possui uma ou mais dimensões variadas. Quando há vários objetos grandes, os problemas são classificados em três categorias. Uma delas reúne aqueles em que os objetos são idênticos. Outra categoria é para quando há uma fraca heterogeneidade dos objetos grandes. A terceira categoria é para os problemas em que os objetos grandes são fortemente heterogêneos.

Essas quatro características dos problemas de *C&P* observadas nesta segunda tipologia são revisões dos critérios presentes na tipologia anterior. Um aspecto não considerado na tipologia de Dyckhoff e que foi considerado na de Wascher et al. [2007] é a forma dos itens pequenos, para problemas bidimensionais ou tridimensionais. Segundo este aspecto, os problemas são classificados entre itens pequenos regulares, ou itens pequenos irregulares.

Seguindo as avaliações de critérios proposta por esta tipologia, o problema considerado no desenvolvimento do presente trabalho é bidimensional, como descrito anteriormente, e tem o tipo de associação voltado para a minimização da saída, pois é necessário definir um leiaute que permita posicionar todos os itens pequenos sobre a área do objeto grande. Este objeto, por sua vez, possui uma das dimensões não restrita a um valor fixo predeterminado. Este valor é definido pelo comprimento do leiaute produzido, e precisa ser minimizado. Para problemas com este tipo de objeto grande, a tipologia de Wascher et al. [2007] não considera sub-categorias de acordo com a variedade de itens pequenos. O tipo de problema considerado neste trabalho é classificado, portanto, como um Problema de *C&P* de Dimensão Aberta. A seguir são apresentadas representações encontradas na literatura para este tipo de problema.

1.3 Representações dos elementos do problema

A modelagem dos dados do problema que são significativos para determinar uma solução é uma tarefa que está diretamente associada à abordagem que será adotada. O problema específico em estudo possui apenas um objeto grande para receber os itens pequenos posicionados, e esse objeto possui formato retangular com apenas uma dimensão fixa, a

largura. Isto torna a representação deste objeto relativamente simples. A decisão sobre como representar os dados dos itens pequenos, no entanto, interfere diretamente nos cálculos geométricos que podem ser aplicados e na capacidade de aproveitamento da área disponível. Das fontes consultadas, foram extraídos dois métodos fundamentais distintos para representar os itens que devem ser posicionados: a poligonização e a discretização. Estes dois métodos são baseados na aproximação das formas originais dos itens através de figuras geométricas simples. A seguir, serão relatadas as técnicas observadas na literatura que se enquadram nestas duas abordagens e, por fim, alguns trabalhos que aplicam estruturas diferentes para a modelagem dos dados do problema.

1.3.1 Aproximação por poligonização

A representação de formas irregulares através de polígonos é a técnica mais amplamente usada nas pesquisas relatadas nas publicações que foram consultadas de trabalhos correlacionados. Albano & Sapuppo [1980] descrevem as peças a serem posicionadas como polígonos formados por uma lista de vértices, ordenados por um percurso no sentido anti-horário. Os mesmos autores ainda afirmam que as coordenadas dos vértices são relativas ao ponto de referência do polígono. Eles também acrescentam a simplificação de um agrupamento de peças posicionadas, chamado de envelope, que substitui os dados detalhados dos polígonos que já compõem soluções parciais pelo polígono formado por arestas “externas” das peças, ignorando os espaços vazios entre as peças e as regiões entre estas e os limites do objeto receptor. Burke & Kendall [1999] descrevem abstração análoga, incluindo a informação de que o último ponto da lista deve ser considerado conectado ao primeiro, como uma lista circular.

Vários autores utilizam representações semelhantes, como [Li & Milenkovic, 1995b], que descrevem a relatividade dos dados dos vértices aos pontos de referência dos polígonos através de diferentes sistemas de coordenadas. Um sistema de coordenadas global é definido para indicar os pontos de posicionamento dos itens no objeto grande. E cada item tem um sistema local de coordenadas, cuja origem não necessariamente é o próprio ponto de referência. Dowsland et al. [2002] fazem uso de uma abordagem semelhante, mas com as origens dos sistemas de coordenadas locais sendo os respectivos pontos de referências dos itens. Além disso, cada ponto de referência corresponde, naquele trabalho, ao canto inferior esquerdo do retângulo envolvente de cada item. Este retângulo envolvente é definido pelos vértices extremos à esquerda, à direita, ao topo e à base de um item.

Gomes [2005] utiliza uma ferramenta computacional que aproxima uma forma irregular por meio de polígonos orientados, tratando contornos curvilíneos com segmentos

de reta exteriores que tangenciam a forma irregular. Nestes polígonos, cada aresta é orientada de modo que o lado direito corresponda ao interior do polígono, e o lado esquerdo ao exterior do mesmo. Naquele trabalho, portanto, o autor utiliza polígonos orientados em sentido horário. O mesmo autor também afirma que a orientação das arestas traz como vantagem a possibilidade de utilizar *função-D*, que é a base para métodos eficientes de análise de interseções e sobreposições entre polígonos. O conceito de *função-D* será brevemente discutido na subseção 2.1.1. Adicionalmente, aquela ferramenta computacional adotada por Gomes também é capaz de realizar simplificações de polígono, eliminando reentrâncias consideradas muito estreitas, e reduzindo o número de vértices da representação. O objetivo desta simplificação é diminuir o tempo de processamento demandado pelos métodos de solução propostos, que são fortemente influenciados pelo número de arestas dos polígonos a posicionar. Uma possível consequência dessa simplificação é a ocorrência de regiões não aproveitadas do objeto em decorrência dos excedentes do polígono simplificado em relação à área do item original. De fato, a simplificação apenas acentua esta consequência desvantajosa, que já é característica da representação de objetos com curvaturas através de polígonos.

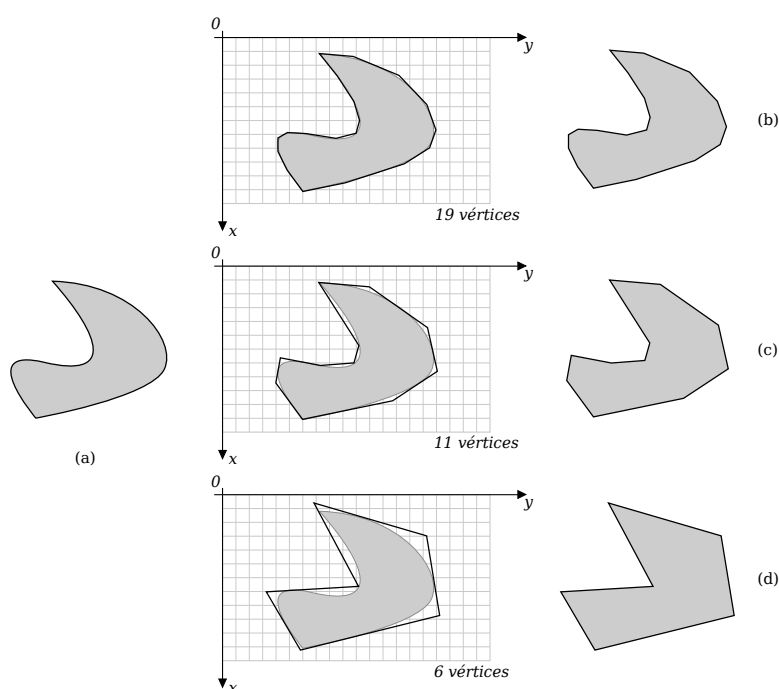


Figura 1.1. Aproximação de forma irregular com polígono envolvente.

A figura 1.1 exibe um exemplo de variações da poligonização de uma forma irregular (a) que possui contorno com curvas. Em (b), a forma irregular é aproximada por um polígono de 19 vértices, que são extremos de segmentos de reta que tangenciam as bordas

da forma original. Os eixos x e y compõem o sistema local de coordenadas usado para descrever os vértices do polígono em \mathbb{R}^2 . Uma aproximação mais simples, com 11 vértices, é apresentada em (c). Nesta, são facilmente percebidas as ocorrências de excedentes em relação à forma original. Em (d), a aproximação é simplificada ainda mais, sendo feita com apenas seis vértices. Com isso, os excedentes passam a corresponder a uma área maior. É importante observar que a expectativa de consumo de tempo para as operações de posicionamento é inversamente proporcional à quantidade de vértices usados para representar a forma irregular. Já a expectativa de aproveitamento do objeto com os posicionamentos dos itens é diretamente proporcional à quantidade desses vértices. Resumidamente, quanto mais detalhada é a representação, mais tempo será necessário para calcular as posições válidas para os itens, porém, também torna-se mais provável que as soluções encontradas sejam de melhor qualidade.

1.3.2 Aproximação por discretização

Whelan & Batchelor [1996] apresentam um mecanismo de empacotamento geométrico que não considera os itens a posicionar como polígonos, como os métodos citados anteriormente. Os autores desse trabalho descrevem o uso de morfologia matemática com imagens binárias para representar os itens e os objetos receptores, respectivamente. A imagem de cada objeto receptor é denominada cena, e acumula a solução parcial que progressivamente recebe cada item a posicionar. Na cena, as posições não ocupadas ou disponíveis ficam em branco, enquanto as regiões ocupadas ou indisponíveis ficam em preto. A definição das regiões válidas para o posicionamento é feita através da combinação da cena com o próximo item a posicionar, usando uma operação de erosão e, no resultado desta operação, é feita uma varredura dos pixels da imagem resultante em busca do primeiro em branco. As coordenadas deste píxel em branco encontrado são usadas para o posicionamento do próximo item.

Hopper [2000] indica um tipo de cenário no qual pode ser vantajosa a representação dos itens através de uma grade, como a digitalização de imagens destes itens. Segundo Hopper, a ocorrência de itens com formas altamente irregulares pode levar a situações em que se torna difícil elaborar os parâmetros geométricos que descrevam suficientemente as propriedades geométricas relevantes para o processamento. Assim, a simplicidade da representação em grade, e da manipulação dos itens nesta representação, pode ser uma alternativa considerável para casos de formas complexas.

Em Jain & Gea [1998], é utilizada uma forma de codificação dos itens a posicionar através de aproximação por discretização. Nesta abordagem, as figuras dos itens são representadas por uma coleção de células, cada uma ligada a oito outras células vizinhas,

que são dispostas ao seu redor. A representação dos objetos grandes é feita como um vetor bidimensional de células, sendo cada célula capaz de comportar um número inteiro entre zero e o número de itens a posicionar. O valor zero é reservado para simbolizar uma célula livre do objeto, enquanto qualquer outro número identifica a ocupação da célula por um item específico da instância.

Oliveira & Ferreira [1993] *apud* Bennell & Oliveira [2008] definem um método de discretização que resume os dados geométricos dos itens e do objeto como matrizes bidimensionais, sendo que cada célula de cada matriz armazena um número inteiro. O número zero indica uma posição não ocupada, enquanto o número um simboliza que a célula está ocupada por um item. Números maiores do que zero apontam a ocorrência de sobreposição entre itens. O número indica a quantidade de itens que estão ocupando a célula. Os resultados apresentados em Oliveira & Ferreira [1993] *apud* Dowsland & Dowsland [1995], que fizeram experimentos com discretização e poligonização, indicam claramente que a simplicidade do tratamento feito com a discretização dos dados tem como consequência a redução da qualidade da solução.

Segenreich & Braga [1986] *apud* Bennell & Oliveira [2008] usam uma codificação diferente para a discretização dos itens, na qual o número um é aplicado às células de fronteira da forma, enquanto o número três é aplicado ao interior. Nesta abordagem, quando feito um posicionamento de um novo item à solução em construção, há uma adição entre os números da matriz do item aos valores da matriz que representa o objeto grande. As células resultantes desta adição que tenham valores menores ou iguais a dois são consideradas válidas, enquanto valores maiores indicam inviabilidade do posicionamento. Células com o valor dois indicam ocorrência de contato entre dois itens.

Babu & Babu [2001] *apud* Bennell & Oliveira [2008] usam um método diferenciado para a representação dos objetos receptores. Nesse trabalho, os autores sugerem uma codificação matricial com valores que auxiliam no posicionamento, indicando o número de células que devem ser saltadas à direita, na mesma linha de uma célula consultada, para que se obtenha uma célula livre. Desta forma, o valor zero indica uma posição livre, enquanto valores diferentes de zero, que são posições ocupadas, simplificam o processo de posicionamento, pois indicam um número de células que devem ser ignoradas na busca por uma posição válida. Ao posicionar um novo item no objeto, as células ocupadas por este item devem ser codificadas da mesma forma.

A figura 1.2 ilustra a discretização de uma forma irregular (a) idêntica àquela usada no exemplo da figura 1.1. Em (b), há uma grade com tamanho tal que o retângulo suficiente para comportar completamente a forma possui 21 células do eixo x e 24 células do eixo y . Esta densidade de células da grade fornece uma aproximação relativamente mais precisa do que a feita em (c), na qual a forma é representada em um retângulo com 11

células no eixo x e 13 células no eixo y . Reduzindo ainda mais a resolução de células usadas para aproximar a forma, alguns detalhes desta deixam de ser representadas, como observado em (d). Além disso, a distância entre as formas posicionadas pode ser maior do que o necessário, sendo limitada ao tamanho de uma célula. Quanto menor a célula, menor é a distância máxima entre as formas que é imposta pela representação.

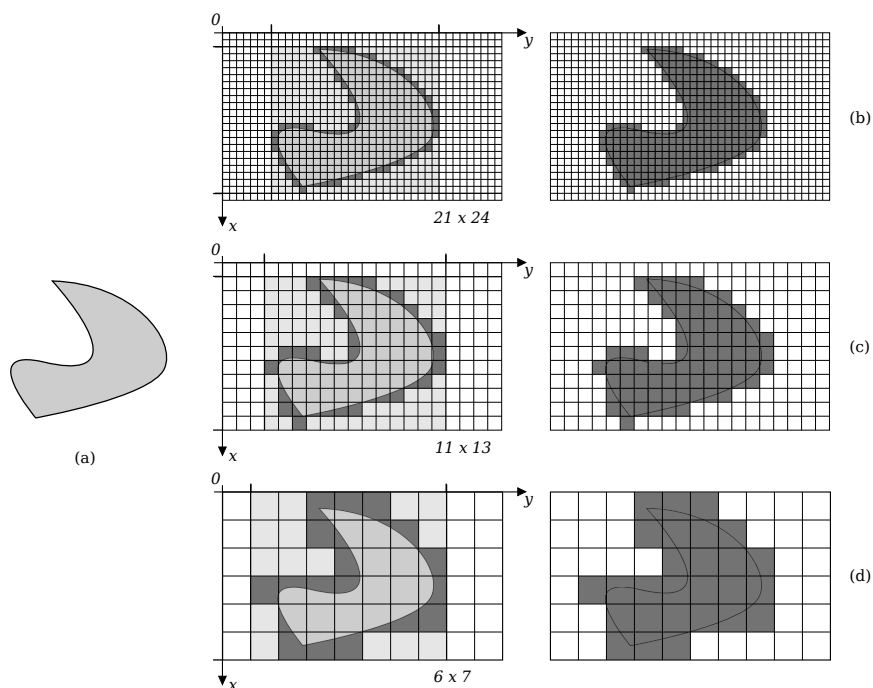


Figura 1.2. Aproximação de forma irregular com discretização.

1.3.3 Outros métodos para representação

Duas das publicações consultadas apresentaram tratamentos diferentes para os dados do problema. Burke et al. [2006] utilizam uma abordagem que representa cada item como um conjunto de listas circulares de elementos primitivos, que podem ser linhas ou arcos. Deste conjunto, uma lista representa o contorno externo da forma irregular, e as demais, caso existam, representam buracos da forma. Em cada lista, os elementos primitivos são orientados em sentido anti-horário, e são conectados, ou seja, o ponto final de um elemento primitivo é igual ao ponto inicial do elemento seguinte. Outra consideração importante sobre esse trabalho é sobre a representação da área interna do objeto receptor. Esta área é codificada como uma lista de retas verticais. Um dos eixos de coordenadas, x , passa a ser tratado como um intervalo discreto finito, de forma semelhante à abordagem com grade. Porém, os autores desse trabalho mantêm as coordenadas do eixo y variando continuamente, preservando infinitas possibilidades de posicionamento. No

mesmo trabalho, são propostos métodos para identificação de situações de sobreposição de itens, bem como para sair dessas situações. Como podem haver arcos de circunferência, esses métodos são consideravelmente diferentes dos métodos mais frequentemente encontrados, quando as formas são aproximadas com polígonos ou com matrizes.

Em Jacquenot et al. [2009], os autores apresentam uma representação das formas irregulares através da aproximação por um conjunto de círculos inscritos na forma irregular. Esta aproximação facilita operações necessárias para o caso específico descrito naquele trabalho, que permite a rotação livre da forma para o posicionamento. Além disso, os cálculos de sobreposição e o método de separação descritos são simplificados, já que podem ser baseados em cálculos de distâncias entre pontos.

A definição do espaço de soluções depende das estruturas dos dados que representam as entidades do problema. Uma vez que seja possível definir este espaço de soluções, podem-se definir métodos que determinem solução ótima global, que são aqui denominados métodos para determinar solução exata, ou que determinem aproximação de solução ótima global, que serão denominados métodos para determinar solução aproximada. São relatados a seguir as abordagens de solução para o problema que foram encontradas na literatura consultada.

1.4 Objetivos

Esta pesquisa tem como objetivo geral propor um método heurístico para a resolução de problemas de corte e empacotamento, que equilibre a qualidade das soluções obtidas e o tempo necessário para obtê-las. Este método combina uma regra de posicionamento gulosa com uma técnica de modificação dos posicionamentos para compactação dos resultados parciais. Juntamente com o estudo de características do método proposto, objetiva-se a realizar um estudo da influência da combinação de atributos das instâncias de testes com parâmetros do método de resolução para a qualidade das soluções obtidas. Sendo assim, as seguintes hipóteses são levantadas:

1. A aplicação da operação de compactação de leiaute com programação linear, entre posicionamentos de subconjuntos de itens, leva a soluções melhores do que a construção de leiautes com a simples aplicação de uma regra de posicionamento gulosa, ou do que esta construção seguida da compactação do leiaute completo;
2. Atributos que caracterizam a instância e parâmetros da execução do método construtivo influenciam na qualidade das soluções obtidas.

Para realizar os testes destas hipóteses, o presente trabalho possui os seguintes objetivos específicos:

1. Pesquisar sobre o cenário atual da pesquisa voltada à solução do problema de *C&P*;
2. Descrever os cálculos geométricos necessários para a construção de leiautes;
3. Descrever o modelo matemático de compactação de leiaute com programação linear;
4. Identificar atributos que possibilitem caracterizar instâncias do problema de *C&P*;
5. Salientar as possíveis parametrizações do método proposto neste trabalho;
6. Obter os resultados da aplicação do método proposto para verificação.

1.5 Metodologia

Este trabalho procura cumprir etapas fundamentais do método científico, como a observação, constatação de fatos, formulação de teoria, experimentos, análise e observação de novos fatos. Para aplicar isto visando alcançar os objetivos específicos, a metodologia empregada compõe-se de:

1. Coletar e organizar informações disponíveis na literatura técnica relacionadas a pesquisas sobre o problema de corte e empacotamento. Isto envolve reunir conhecimento suficiente pelo viés da computação geométrica e sobre maneiras de abordar o problema combinatorial. É importante também consultar o estado da arte deste campo, e a forma como as diferentes abordagens podem ser comparadas.
2. Com base no conhecimento obtido, descrever o método proposto neste trabalho, capaz de solucionar o problema de corte e empacotamento combinando duas diferentes técnicas, sendo uma regra de posicionamento gulosa e uma operação de compactação de leiaute.
3. Verificar empiricamente o desempenho do algoritmo proposto. Para fazer a observação, é necessário implementar o algoritmo para execução por um computador. Os experimentos devem ser feitos seguindo a forma, encontrada na literatura, que é usada para comparar diferentes abordagens.
4. Analisar os resultados obtidos com os experimentos e traçar vertentes para continuidade da pesquisa em trabalhos futuros.

1.6 Organização do documento

O texto deste trabalho está organizado como segue. No capítulo 2 são relatadas as informações coletadas sobre trabalhos relacionados. Essas informações os modelos matemáticos que representam o problema de otimização, as abordagens para obter soluções heurísticas, o estado da arte das pesquisas nesta área e o meio como os experimentos são conduzidos pelos pesquisadores para tornar possível a comparação da eficiência dos métodos.

O capítulo 3 trata da descrição do algoritmo proposto para abordar o problema de corte e empacotamento. O método de resolução é detalhado através da examinação da aplicação em dois casos. As sub-rotinas utilizadas pelo algoritmo são relatadas em seus elementos fundamentais.

No capítulo 4 encontram-se os relatos das experiências computacionais realizadas. Para isto, são apresentadas as instâncias utilizadas para os testes e as variações do algoritmo que foram avaliadas. Também são expostos os resultados obtidos e é feita uma análise dos dados, para possibilitar a avaliação do nível de eficácia do algoritmo proposto.

O capítulo 5 traz algumas conclusões que puderam ser inferidas a partir da pesquisa realizada. Algumas perspectivas sobre trabalhos futuros são incluídas, visando facilitar a continuidade do estudo e a contribuição efetiva do conhecimento obtido para a área envolvida.

Os melhores leiautes que foram obtidos nos experimentos computacionais para as instâncias utilizadas estão no anexo A. Para cada resultado apresentado, são relacionados os parâmetros do algoritmo proposto que levaram a atingir a solução.

Capítulo 2

Trabalhos relacionados

O problema da geração de leiautes de posicionamentos de itens de formas irregulares sobre uma superfície plana foi explorado por vários pesquisadores, com diversas abordagens para atingir soluções suficientemente boas. Das fontes consultadas, foram extraídas algumas informações que serviram de base de conhecimento para a elaboração do trabalho de pesquisa realizado. Houve uma preocupação inicial com a seleção de técnicas para a modelagem computacional do cenário em estudo. A partir do conhecimento de diferentes possibilidades de modelagem conhecidas, a atenção foi deslocada para a reunião de métodos usados para gerar soluções simples, não necessariamente de boa qualidade. Sabendo como solucionar o problema, o foco recaiu sobre o refinamento das soluções, visando atingir resultados aceitáveis para aplicar no setor produtivo real.

As informações coletadas são abreviadas neste capítulo, que inicia com a classificação dos problemas que serão denominados genericamente de corte e empacotamento, simplificado pelo acrônimo *C&P*, da expressão em idioma inglês *Cutting and Packing*. Na sequência, são identificados diferentes mecanismos de representação dos dados relevantes e do problema de uma perspectiva ampla. Posteriormente, apresentam-se abordagens de resolução exata e aproximada, e as consequências da adoção dessas técnicas. São listados, adiante, as instâncias e os formatos de resultados que são comumente utilizados para estabelecer parâmetros de comparação entre os desempenhos das diferentes abordagens de resolução. Por fim, é feita uma observação sobre o estado da arte das pesquisas conduzidas neste segmento.

2.1 Abordagens para solução

São consideradas duas estratégias fundamentais para classificar as abordagens de solução. A primeira delas reúne abordagens que se voltam à determinação da solução ótima

para o problema, denominada solução exata. Diante de restrições de tempo aplicadas em situações reais, a determinação de solução exata pode ser comportada, dada a complexidade computacional do problema. Para as situações impeditivas para a determinação de solução exata, existem abordagens mais flexíveis, que se restringem a avaliar um reduzido subconjunto do espaço de soluções para encontrar uma solução que seja uma boa aproximação da solução ótima. Essas abordagens empregam heurísticas de busca, além de metaheurísticas, para explorar as soluções alternativas.

Nos trabalhos consultados, foi recorrente a referência a um subprocesso comum, usado para definir uma região entre dois itens a serem posicionados, caracterizada por fornecer coordenadas relativas que permitem posicionar um item no leiaute com o mínimo de distância para outro item já posicionado. Este método, denominado invólucro de posicionamento, destina-se a reduzir a sobrecarga imposta pela inerente dependência de subrotinas de geometria computacional para a resolução do problema, e será descrito a seguir. Adiante, serão comentadas abordagens encontradas na literatura para determinar a solução exata, seguidas de abordagens heurísticas para determinar solução. Por fim, será descrito um modelo de programação linear usado para compactação de leiautes, usado por alguns autores para melhorar soluções encontradas.

2.1.1 Invólucro de posicionamento e conceitos relacionados

Uma das restrições do problema de *C&P* é a ausência de qualquer sobreposição entre os itens posicionados no objeto receptor. A análise relativa de dois itens para verificar o respeito a esta restrição pode ser uma tarefa consideravelmente dispendiosa, dependendo da representação adotada para os elementos do problema. Considerando a aproximação das formas irregulares por poligonização, este método reduz a análise de sobreposição entre dois itens ao problema de verificar se um ponto está no interior de um polígono, como apresentado em Gomes [2005]. Além disso, pode ser adotado o invólucro de posicionamento como parte do procedimento de identificação das posições válidas para a inclusão de um novo item no leiaute. Não somente isso, nos trabalhos consultados foi encontrada a utilização do invólucro de posicionamento também para a formulação de modelos de programação linear para a solução exata do problema, bem como para a compactação de leiautes.

Segundo Bennell et al. [2001], o invólucro de posicionamento (*NFP*, do termo em inglês *no-fit polygon*) entre dois polígonos A e B , $NFP_{A,B}$, é o polígono resultante de uma operação na qual A e B têm papéis distintos. O primeiro polígono, A , assume uma posição fixa, enquanto o segundo polígono, B , passa por uma série de movimentos, de modo semelhante a deslizar pelo contorno do polígono A . O resultado da operação

é formado pelo sequenciamento de todo o percurso feito pelo ponto de referência de B , usando coordenadas relativas ao ponto de referência de A . Definido desta forma, o $NFP_{A,B}$ é uma região relativa ao polígono A que traz informações relevantes para o posicionamento do polígono B . De acordo com as coordenadas usadas para transladar B para o objeto receptor, através do seu ponto de referência, é possível afirmar que:

- Se o ponto de referência de B estiver em uma região externa ao $NFP_{A,B}$, não haverá interseção nem toque entre os polígonos A e B ;
- Se o ponto de referência de B estiver posicionado sobre uma aresta do $NFP_{A,B}$, os polígonos A e B estarão tocando um ao outro, mas sem sobreposição;
- Se o ponto de referência de B estiver em uma região interna do $NFP_{A,B}$, então haverá uma interseção entre A e B que caracteriza sobreposição entre os itens.

A figura 2.1 exemplifica uma aplicação de NFP entre dois polígonos A e B . Em (i), são apresentados os dois polígonos e seus pontos de referência, que neste caso são interiores aos respectivos polígonos. O polígono $NFP_{A,B}$ é definido pelo deslizamento da forma do polígono B pelas arestas do polígono A , sem que haja sobreposições. Consequentemente, como pode ser observado em (ii), translações do polígono B , pelo seu ponto de referência, para qualquer ponto pertencente a alguma aresta do $NFP_{A,B}$ resulta em um cenário no qual um ou mais pontos de uma ou mais arestas de B tocam um ou mais pontos pertencentes a uma ou mais arestas do polígono A .

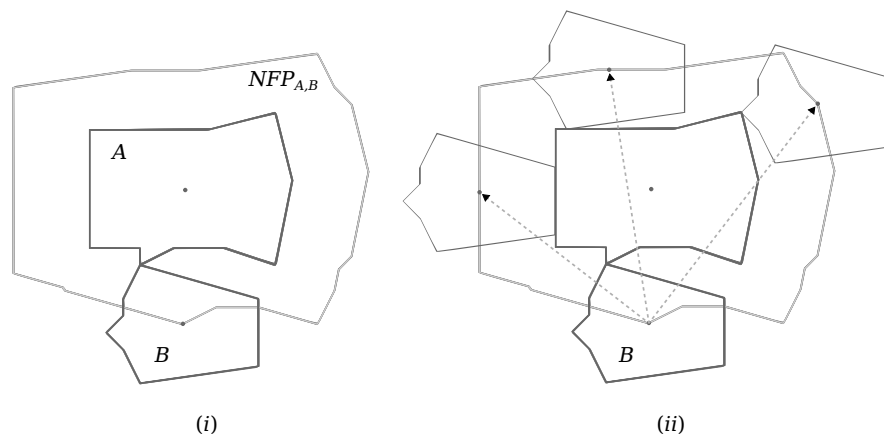


Figura 2.1. NFP entre dois polígonos A e B (i); possíveis translações do polígono B guiadas pelo NFP (ii).

Para o aproveitamento do NFP na inclusão do posicionamento de um item B para a construção de um leiaute, é necessário transladar o polígono $NFP_{A,B}$ para um ponto de origem tal que efetivamente torne prática sua aplicabilidade. Segundo Bennell & Oliveira

[2008], este ponto pode ser definido considerando os pontos de referência e as posições relativas entre A e B . Neste trabalho, será assumido que os vértices do NFP serão definidos em um sistema de coordenadas que tem a mesma origem do sistema de coordenadas usado para definir os vértices de A . Assim, basta transladar $NFP_{A,B}$ para o ponto de referência de A , pelo ponto de referência de $NFP_{A,B}$, que deve ser definido como segue:

- I. Seja \mathcal{R}_A o ponto de referência de A ;
- II. Seja \mathcal{R}_{NFP} o ponto de referência de $NFP_{A,B}$;
- III. Seja \mathcal{O}_A a origem do sistema de coordenadas de A ;
- IV. Seja \mathcal{O}_{NFP} a origem do sistema de coordenadas de $NFP_{A,B}$;
- V. Os segmentos orientados $\overrightarrow{\mathcal{O}_A \mathcal{R}_A}$ e $\overrightarrow{\mathcal{O}_{NFP} \mathcal{R}_{NFP}}$ são equipolentes.

Gomes & Oliveira [2001] definem retângulo de posicionamento interior (acrônimo IFR , da expressão inglesa *inner-fit-rectangle*) entre um retângulo P e um polígono B de maneira análoga ao NFP , porém assumindo que B será deslizado pelo interior de P , se possível, resultando também em um retângulo que demarca a região em que B pode ser posicionado no interior de P , sem que algum vértice de B fique externo a P . A combinação entre os polígonos $NFP_{A_i,B}$, para todos os polígonos A_i que já estão posicionados, e o $IFR_{P,B}$, sendo P o objeto receptor, permite determinar as coordenadas válidas para que o item B seja posicionado. Para os casos específicos nos quais o objeto receptor não é retangular, o $IFR_{P,B}$ deve ser substituído por uma forma semelhante ao NFP , denominada por Bennell & Song [2008] como *Inner Fit Polygon (IFP)*, formado pelo deslizamento de B pelo interior de P . Martins [2007] denomina região livre de colisão aquela que compreende todos os pontos válidos para o posicionamento de B . Ou seja, a região livre de colisão é tal que não está no exterior do $IFP_{P,B}$, e que não está no interior de algum $NFP_{A_i,B}$, para todo item i já posicionado. Art [1966] reúne conceitos de região livre de colisão, NFP , IFR e IFP através da definição de Envelope para uma peça.

A respeito das técnicas para gerar o NFP entre duas formas, duas destacaram-se entre as fontes consultadas para este trabalho. Uma delas trabalha com sucessivos deslocamentos de uma forma pelas arestas da outra, enquanto a outra é baseada em somas de Minkowski. A primeira foi proposta por Burke et al. [2007], e tem como fundamento o deslizamento de um polígono móvel B ao redor do polígono estático A , mantendo B sempre em contato com A . O início do processo requer que B seja transladado para um ponto inicial, de modo que B esteja em contato com A . Para contornar externamente A sem que se incorra em sobreposição, é feita uma seleção de uma aresta de A ou B que

será usado como vetor para o deslizamento. Esta escolha é feita de acordo com características das arestas envolvidas no contato de A e B . Tendo sido selecionado um vetor para o deslizamento, é feita uma análise de dois grupos de projeções. No primeiro grupo, cada vértice de B é usado como origem para a translação do vetor de deslizamento. Se algum ponto deste vetor ficar em uma região interna de A , o vetor tem seu comprimento reduzido até o ponto de toque com A . De modo complementar, é feito o segundo grupo de projeções, que segue a rotina aplicada ao primeiro grupo, mas inverte o sentido do vetor para deslizamento, e usa cada vértice de A como origem para o vetor. Para este grupo de projeções, é feita a verificação de situações nas quais algum ponto deste vetor ficar em uma região interna de B , o vetor é aparado da mesma forma. Ao final, o vetor para deslizamento de B por A será tal que permitirá que, no destino da translação de B , este polígono permaneça em contato com A , sem sobreposição. Quando B atinge a mesma posição que iniciou o processo, a sequência de vetores usados para os deslizamentos consecutivos de B é usada para descrever o contorno externo do $NFP_{A_i, B}$. Esta parte do processo é semelhante ao que foi proposto em Mahadevan [1984] *apud* Burke et al. [2007].

Para detectar pontos internos em que B pode ser transladado, preservando as condições do NFP , Burke et al. [2007] propõem um método para determinar outros possíveis pontos de partida para novos deslizamentos. Para fazer isso, as arestas de A são marcadas inicialmente como não percorridas. A cada vez que um deslizamento é feito, a aresta de A que foi percorrida é marcada como tal. Quando um ciclo de deslizamentos é encerrado, as arestas de A ainda não percorridas são analisadas em busca de uma que possa fornecer um novo ponto de partida para B . Se for encontrado, é feito novamente o processo de deslizamento de B . Se não for encontrado, o processo de geração de $NFP_{A_i, B}$ é encerrado, tendo como resultado um polígono que pode ter buracos. Esses buracos são geralmente associados a algumas regiões específicas de concavidades de algum polígono envolvido, ou mesmo à existência de buracos no polígono estático.

A segunda maneira consultada para gerar o NFP está em Bennell & Song [2008]. Nesse trabalho, os autores descrevem um procedimento que revisa o que foi proposto em Bennell et al. [2001], que por sua vez apresentava um complemento ao trabalho de Ghosh [1990]. Ghosh descreveu um algoritmo para tratar de problemas de organização espacial de polígonos através de somas de Minkowski e do rastreamento de limite da decomposição de Minkowski. Aquele autor trata do problema de interseção entre dois polígonos T e B fazendo uso da operação $T \oplus \check{B}$, sendo que \oplus representa a soma de Minkowski e \check{B} é o conjunto simétrico de B . A figura 2.2 exemplifica a aplicação desta operação para o caso simples de dois polígonos convexos. Nessa figura, em (i) são apresentados dois polígonos com arestas originalmente orientadas em sentido anti-horário, A e B , e o polígono

\check{B} , com os respectivos pontos de referência destacados. Em (ii), as arestas dos polígonos A e \check{B} são dispostas em um diagrama de inclinações. Percorrendo este diagrama no sentido anti-horário, iniciando em 0, obtem-se $A \oplus \check{B}$. Esta soma de Minkowski e sua representação gráfica como um novo polígono são exibidos em (iii). Deve-se observar que o polígono $A \oplus \check{B}$ transladado para o ponto de referência de A serve como $NFP_{A,B}$.

Segundo Bennell & Oliveira [2008], o algoritmo de Ghosh não é suficientemente robusto para instâncias específicas do problema que envolvem polígonos que levam à interação de múltiplas concavidades. Bennell et al. [2001] propuseram uma abordagem para calcular o NFP entre dois polígonos A e B usando o invólucro convexo de B quando houver concavidade neste e, ao término do procedimento para calcular o NFP , as arestas que descrevem concavidades de B são incluídas no polígono resultante, combinadas com arestas de A quando a sequência das inclinações das arestas da concavidade de B passar pela inclinação de alguma aresta de A . Este procedimento foi revisado em Bennell & Song [2008], pois foram detectados casos de falhas para algumas situações específicas. Tomando dois polígonos A e B , estático e móvel, respectivamente, o algoritmo que foi então proposto por aqueles autores separa as arestas do polígono B em grupos formados por sequências que estão dispostas em mesmo sentido anti-horário ou em mesmo sentido horário, o que varia em regiões de concavidades. Esses grupos são combinados com as arestas de A seguindo a ordem de inclinações das arestas. Os grupos combinados com arestas de A são posteriormente ligados entre si, acrescentando arestas de A para manter a precedência de arestas em cada polígono. Por fim, o resultado é um polígono complexo, que precisa ser simplificado para contornos externos e internos, que representam eventuais buracos no NFP .

Segundo Agarwal et al. [2002] *apud* Gomes [2005], o cálculo do $NFP_{A,B}$ é um problema que apresenta consumo de tempo assintótico variando em função das quantidades de arestas de A e de B , e sofrendo influência da presença de concavidades nos polígonos A ou B , como segue. Sejam m e n os números de arestas de A e B , respectivamente. Quando os dois polígonos forem convexos, o problema tem consumo de tempo $O(m+n)$. O exemplo apresentado na figura 2.2 exhibe esta situação. Quando o polígono móvel for convexo e o estático for côncavo, o consumo de tempo assintótico é $O(mn \log mn)$. Já para os casos quando A e B forem côncavos, um limite superior apertado para o consumo de tempo é $O(m^2n^2)$.

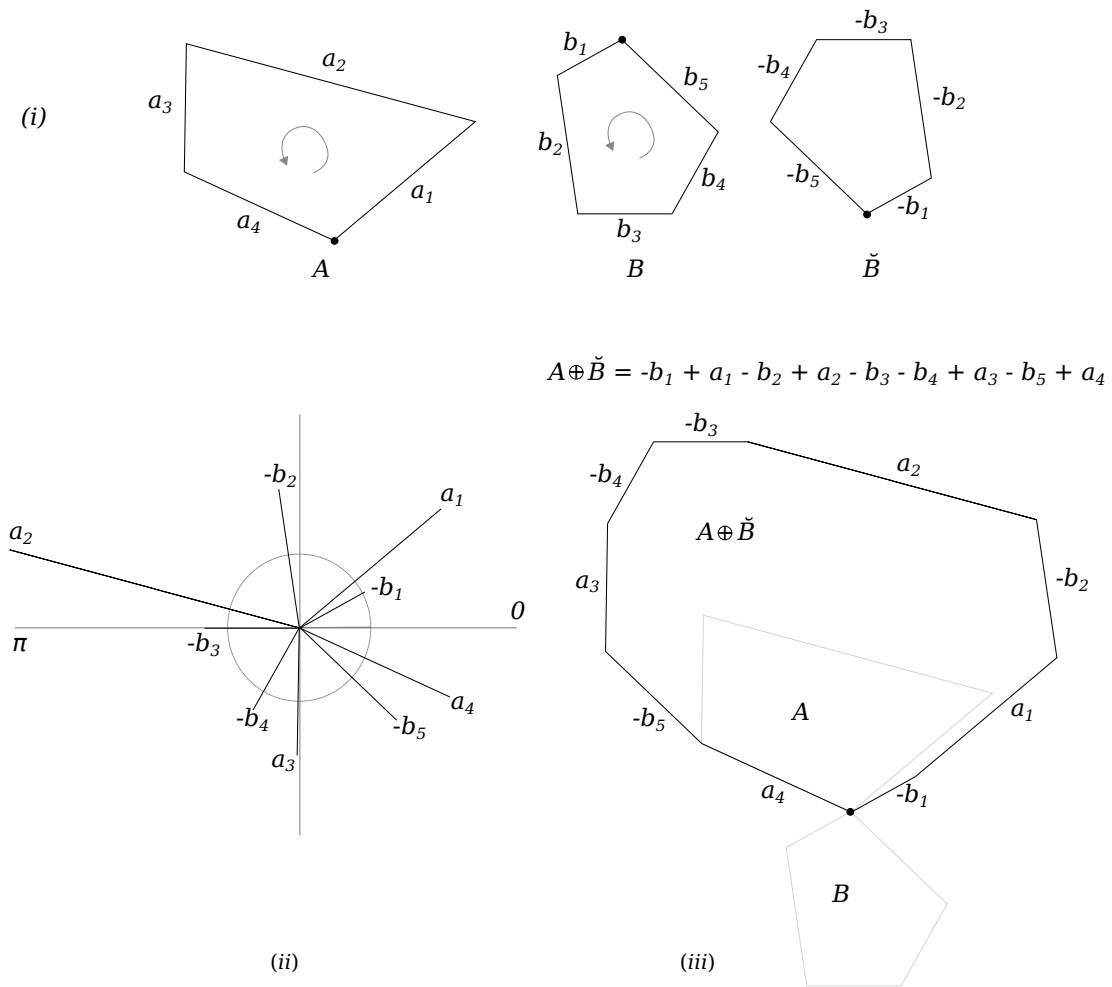


Figura 2.2. Exemplo de caso simples de operação de Minkowski sobre polígonos orientados.

Relembrando, o invólucro de posicionamento tem como atratividade principal simplificar o cálculo de interseções entre polígonos para verificar a ocorrência de sobreposições entre os itens posicionados. Segundo Bennell & Song [2008], o consumo de tempo para calcular sobreposições usando trigonometria diretamente é $O(nm + n + m)$, enquanto calcular esta sobreposição com o invólucro de posicionamento tem consumo de tempo $O(k)$, onde k é o número de arestas do *NFP*. Embora isto pareça ser favorável ao uso de trigonometria direta, já que o uso do *NFP* implica na necessidade de gerar os invólucros de posicionamento, deve-se acrescentar ao comparativo a capacidade de reaproveitamento do *NFP* calculado. É possível reter os dados de *NFP* em memória para sua possível reutilização posterior, como indicado por Burke & Kendall [1999]. Bennell & Song [2010] afirmam que a maioria das aplicações que usam *NFP* fazem esses cálculos para todos os pares de peças em uma fase de pré-processamento. Em consequência da

possivelmente grande quantidade de soluções que serão avaliadas, dependendo da abordagem usada para determinar a solução para o problema de *C&P*, é comum que um mesmo *NFP* seja requerido várias vezes. Ao reaproveitar o *NFP*, o consumo de tempo para calcular sobreposições cai sensivelmente, já que basta realizar a operação de translação do polígono *NFP* para reusá-lo. O cálculo de sobreposição com uso de trigonometria direta, porém, dependente das posição relativas dos polígonos envolvidos, não pode ser reaproveitado, mantendo seu consumo de tempo assintoticamente superior.

Há ainda nos trabalhos consultados uma recorrente preocupação em determinar a relação entre um ponto p_1 e um segmento de reta orientado $\overrightarrow{p_2p_3}$. Entre as possibilidades, p_1 pode ser colinear, estar à direita ou estar à esquerda de $\overrightarrow{p_2p_3}$. Para lidar com este requisito, Konopasek [1981] *apud* Bennell & Oliveira [2008] introduz o conceito de *função-D*. Seja $p_1 = (x_1, y_1)$, $p_2 = (x_2, y_2)$ e $p_3 = (x_3, y_3)$. A *função-D* que fornece a relação entre o ponto p_1 e o segmento orientado $\overrightarrow{p_2p_3}$ é dado pela equação 2.1.

$$D_{p_2,p_3,p_1} = (x_2 - x_3)(y_2 - y_1) - (y_2 - y_3)(x_2 - x_1) \quad (2.1)$$

A relação será estabelecida de acordo com o resultado fornecido pela *função-D*. Se $D_{p_2,p_3,p_1} = 0$, p_1 é colinear à reta suporte do segmento orientado $\overrightarrow{p_2p_3}$. Se $D_{p_2,p_3,p_1} > 0$, p_1 está à esquerda da reta suporte do segmento orientado $\overrightarrow{p_2p_3}$. E se $D_{p_2,p_3,p_1} < 0$, p_1 está à direita da reta suporte do segmento orientado $\overrightarrow{p_2p_3}$.

Os conceitos de *função-D*, *NFP*, *IFP* e *IFR* fornecem tratamentos para os subproblemas básicos de *C&P* que envolvem a geometria computacional. A seguir, serão comentadas abordagens para determinar soluções para o problema, que exploram esses conceitos fundamentais e acrescentam interessantes técnicas para definir leiautes através da combinação de itens visando o aproveitamento máximo da área do objeto receptor.

2.1.2 Abordagens para determinar solução exata

De acordo com a representação usada, é possível formular modelos de programação matemática que permitam usar técnicas de resolução de problemas de programação linear para determinar a solução ótima. Foram consultados na literatura trabalhos que descrevem modelos de programação linear que representam a minimização do comprimento do leiaute, em casos de aplicação da poligonização como maneira para representar os itens. Entre as variáveis de decisão desses modelos estão as que indicam os pares de coordenadas, referentes à posição no objeto receptor, que deverão ser usadas para transladar cada um dos itens por seus pontos de referência. Alguns detalhes distinguem os modelos consultados, como a representação da função objetivo e o tratamento da condição de não-sobreposição entre os itens posicionados.

Para formalizar os modelos de programação matemática apresentados a seguir, é necessário definir alguns atributos que serão usados para lidar com as coordenadas dos itens. Sejam $\mathcal{I} = \{1, \dots, n\}$ o conjunto de indexadores para os itens que devem ser posicionados, havendo um índice para cada item, e $\mathcal{P} = \{p_i : \forall i \in \mathcal{I}\}$ o conjunto de todos os n itens a posicionar. E sejam $\mathcal{L}_i = \{1, \dots, m_i\}$ o conjunto de indexadores para os vértices de um polígono que representa o item p_i , e $\mathcal{V}_i = \{v_{i,k} = (x_{i,k}, y_{i,k}) : \forall k \in \mathcal{L}_i\}$ o conjunto de todos os vértices do mesmo polígono. Seja também $v_i = (x_i, y_i)$ o ponto de referência para o mesmo item. As definições das equações 2.2 a 2.9 são feitas sobre o conjunto \mathcal{V}_i e o ponto v_i .

$$x_i^{\max} = x_{i,j} \quad |x_{i,j} \geq x_{i,k} \quad , \forall k, j \in \mathcal{L}_i : k \neq j, x_{i,j}, x_{i,k} \in \mathcal{V}_i \quad (2.2)$$

$$y_i^{\max} = y_{i,j} \quad |y_{i,j} \geq y_{i,k} \quad , \forall k, j \in \mathcal{L}_i : k \neq j, x_{i,j}, x_{i,k} \in \mathcal{V}_i \quad (2.3)$$

$$x_i^{\min} = x_{i,j} \quad |x_{i,j} \leq x_{i,k} \quad , \forall k, j \in \mathcal{L}_i : k \neq j, x_{i,j}, x_{i,k} \in \mathcal{V}_i \quad (2.4)$$

$$y_i^{\min} = y_{i,j} \quad |y_{i,j} \leq y_{i,k} \quad , \forall k, j \in \mathcal{L}_i : k \neq j, x_{i,j}, x_{i,k} \in \mathcal{V}_i \quad (2.5)$$

$$right_i = x_i^{\max} - x_i \quad (2.6)$$

$$left_i = x_i - x_i^{\min} \quad (2.7)$$

$$top_i = y_i^{\max} - y_i \quad (2.8)$$

$$bottom_i = y_i - y_i^{\min} \quad (2.9)$$

A equação 2.2 define o maior valor de componente x encontrado na lista de coordenadas dos vértices que descrevem o i -ésimo polígono. A equação 2.3 define o maior valor de componente y presente nas coordenadas dos vértices do i -ésimo polígono. As equações 2.4 e 2.5 definem os menores valores de componentes x e y , respectivamente, dos vértices do i -ésimo polígono. As equações 2.6, 2.7, 2.8 e 2.9 definem as distâncias perpendiculares do ponto de referência do polígono p_i até as laterais do seu retângulo envolvente. A figura 2.3 apresenta os atributos de polígonos que foram definidos acima.

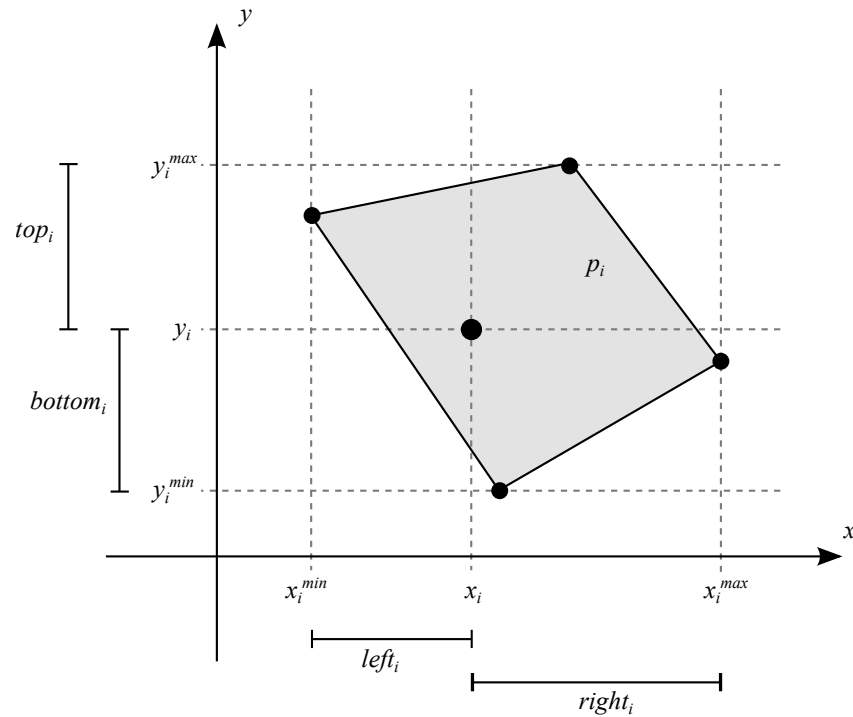


Figura 2.3. Definições de atributos de polígonos usados nos modelos matemáticos.

O modelo proposto em Luzzi [2002] é reproduzido na figura 2.4. Nesse modelo, o autor define o objetivo (2.10) como a minimização do comprimento da solução somado a um valor que representa a intenção de tornar o leiaute compacto, ou seja, de modo que os itens sejam posicionados com seus pontos de referência mais próximos da origem do sistema de coordenadas do objeto receptor. Esta função usa uma constante ϵ , que deve ser um coeficiente positivo suficientemente pequeno para não desfavorecer soluções que tenham comprimentos menores.

$$\min length + \epsilon \sum_{i \in \mathcal{P}} (x_i + y_i) \quad (2.10)$$

$$\text{s.a } x_i + right_i \leq length, \quad \forall i \in \mathcal{P} \quad (2.11)$$

$$left_i \leq x_i, \quad \forall i \in \mathcal{P} \quad (2.12)$$

$$bottom_i \leq y_i \leq y_i^{max} - top_i, \quad \forall i \in \mathcal{P} \quad (2.13)$$

$$\text{Peças } i \text{ e } j \text{ não se sobrepõem} \quad \forall i, j \in \mathcal{P} : i < j \quad (2.14)$$

Figura 2.4. Modelo matemático de Luzzi para o problema de C&P. Fonte: adaptado de Luzzi [2002], p. 40.

As restrições desta formulação são resumidas aos conjuntos de inequações 2.11 a

2.14. As inequações 2.11 são responsáveis por provocar o ajuste do valor da variável *length*, que precisa ser maior ou igual ao valor da maior coordenada entre as coordenadas mais distantes da base do objeto receptor, para todos os itens. Por outro lado, cada uma das inequações 2.12 são usadas para garantir que nenhum vértice $v_i \in \mathcal{V}_i$ fique abaixo da base do objeto receptor. O conjunto de inequações 2.13, por sua vez, garante que, para cada item $p_i \in \mathcal{P}$, nenhum vértice $v_i \in \mathcal{V}_i$ esteja fora dos limites laterais do retângulo que descreve o objeto receptor. A restrição 2.14 sugere que não há sobreposição entre pares de peças (itens).

Aquele mesmo autor descreve uma representação para a garantia de não sobreposição entre qualquer par de itens p_i e p_j usando o invólucro de posicionamento NFP_{p_i,p_j} . Sendo este polígono disposto em termos de posições relativas a p_i , como descrito na seção 2.1.1, a avaliação de posicionamento relativo de p_j ao NFP_{p_i,p_j} pode ser feita pelo vetor $v_j - v_i$. Como o invólucro de posicionamento define um polígono tal que, se algum ponto do interior deste for usado para o posicionamento de p_j , então p_j estará sobrepondo p_i , então uma posição válida para $v_j - v_i$ será qualquer uma localizada sobre uma aresta de NFP_{p_i,p_j} ou em uma região externa a este polígono.

Define-se $\overline{NFP_{p_i,p_j}}$ como o conjunto de pontos do interior do objeto receptor que não estejam no interior do invólucro de posicionamento entre p_i e p_j . Logo, qualquer ponto válido para o posicionamento de p_j será parte de $\overline{NFP_{p_i,p_j}}$. Para formular as restrições que garantam que p_j seja um desses pontos válidos, é feita naquele trabalho uma abordagem que sugere o uso de poliedros para a divisão da região $\overline{NFP_{p_i,p_j}}$ em partes, sendo que cada parte tem como base uma aresta convexa do NFP_{p_i,p_j} , ou um conjunto de arestas côncavas sequenciais deste mesmo polígono. Luzzi também apresenta, em nota, uma breve mas precisa definição para arestas convexas e arestas côncavas, como segue. Uma aresta de um polígono é chamada convexa se sua reta suporte não intercepta o interior do polígono, ao passo que uma aresta côncava é tal que a sua reta suporte intercepta o interior do polígono.

Qualquer ponto do interior de qualquer uma dessas partes pode ser definido limitando as coordenadas à equação da reta da aresta base. Representando os coeficientes angulares da equação de cada reta da aresta base como A_{ij}^k , e o coeficiente linear como b_{ij}^k , chega-se às inequações 2.15, como mostradas na figura 2.5. Há do lado direito de cada restrição deste conjunto uma adição do coeficiente linear da equação da reta da aresta base a uma parcela $M(1 - z_{ij}^k)$. Este valor somado representa um mecanismo de ativação ou desativação da restrição, usando uma constante suficientemente grande M e uma variável binária z_{ij}^k para cada restrição. É informado no conjunto de restrições 2.16 que apenas uma dessas variáveis binárias deve ser igual a 1. Portanto, o ponto para o posicionamento de p_j deve atender a uma das restrições do conjunto 2.15. As restrições de integralidade

do conjunto 2.17 indicam a natureza binária das variáveis z_{ij}^k .

$$A_{ij}^k(v_j - v_i) \leq b_{ij}^k + M(1 - z_{ij}^k) \quad , \forall i, j \in \mathcal{P} : i < j, k = 1, \dots, m_{ij} \quad (2.15)$$

$$\sum_{k=1}^{m_{ij}} z_{ij}^k = 1 \quad , \forall i, j \in \mathcal{P} : i < j \quad (2.16)$$

$$z_{ij}^k \in \{0, 1\} \quad , \forall i, j \in \mathcal{P} : i < j, k = 1, \dots, m_{ij} \quad (2.17)$$

Figura 2.5. Restrições de Luzzi para garantir a não-sobreposição de peças. Fonte: adaptado de Luzzi [2002], p. 42.

O valor m_{ij} representa a quantidade de partes em que a região $\overline{NFP}_{p_i, p_j}$ foi dividida. Como indicado anteriormente, esta quantidade será o número de arestas convexas somado ao número de conjuntos de arestas côncavas sequenciais. É perceptível que o número de restrições deste modelo cresce rapidamente com o número de itens a posicionar. A quantidade de pares de polígonos tem uma relação quadrática com o número de itens, e o número m_{ij} , que define a quantidade de restrições do tipo 2.15 e de variáveis binárias z_{ij}^k para cada par de polígonos, é um valor que depende de características individuais e combinadas dos polígonos de cada par.

Os experimentos conduzidos em Luzzi [2002] indicam que este modelo é bem restritivo quanto à sua aplicabilidade em situações reais. As instâncias produzidas para testes possuíam entre cinco e nove itens, e a solução ótima para cada instância era conhecida, pois se baseavam em figuras semelhantes a “vidros partidos”. Na resolução dessas instâncias, o algoritmo de *branch-and-bound* teve o tamanho da árvore gerada limitado a 100000 nós. Este limite afetou 50% dos testes realizados, para os quais a solução ótima não pode ser encontrada. Conclusivamente, o modelo mostrou-se funcional para as instâncias experimentadas, mas o consumo de tempo tende a ser consideravelmente insatisfatório para instâncias mais complexas.

Outro trabalho que foi consultado e que apresenta um estudo da formulação de modelos de programação matemática para a resolução do problema de *C&P* é o de Gomes [2005]. Seu modelo matemático global é reproduzido na figura 2.6. Naquele trabalho, o autor também usa o *NFP* como referência para manter a condição de não-sobreposição. Para garantir a inclusão total dos itens no interior do objeto receptor é usado o *IFR* de cada item. A condição de inclusão dos itens no interior do objeto receptor usada por Luzzi atua de igual maneira sobre os pontos de referência dos itens. Gomes, no entanto, estende a discussão do modelo para detalhes do tratamento de situações em que há itens que apre-

sentam arestas côncavas. O modelo de programação matemática de Gomes, apresentado na figura 2.6, é comentado a seguir.

$$\min z \quad (2.18)$$

$$\text{s.a } z \geq x_i + L_i \quad , i = 1, \dots, N \quad (2.19)$$

$$F(x_i - X_0, y_i - Y_0) \geq 0 \quad , i = 1, \dots, N \quad (2.20)$$

$$G(x_i - x_j, y_i - y_j) \geq 0 \quad , i = 1, \dots, N, j = i + 1, \dots, N \quad (2.21)$$

$$x_i, y_i, z \geq 0 \quad i = 1, \dots, N \quad (2.22)$$

Figura 2.6. Modelo matemático de Gomes para o problema de *C&P*. Fonte: adaptado de Gomes [2005], p. 84.

O objetivo deste modelo, apresentado na fórmula 2.18, é minimizar o valor da variável z . Esta variável, devido ao objetivo de minimização e ao conjunto de restrições 2.19, terá seu valor ajustado para o mínimo não inferior aos valores $x_i + L_i$, para todos os polígonos p_i , onde $L_i = \text{right}_i$, sendo, portanto, o comprimento do leiaute. O conjunto de inequações 2.20 cuida da manutenção de cada item no interior do objeto receptor. Nestas restrições, os valores X_0 e Y_0 são as coordenadas do ponto de referência do objeto receptor. Já a função F indica a inclusão do item no interior do seu respectivo *IFR*. Como já afirmado, observa-se nessas restrições uma equivalência com o tratamento proposto por Luzzi para o mesmo fim. O conjunto de inequações 2.21 visa garantir a não sobreposição entre os itens no leiaute. Para cada item, deve haver um sistema de inequações que garanta que o ponto de referência do item estará na região externa do *NFP*. Gomes aborda essa questão verificando se o ponto de referência de cada polígono p_j está localizado em algum semiplano do lado externo do *NFP*, formado por uma reta suporte de uma aresta do *NFP* p_i, p_j , para cada p_i tal que $i < j$. Isto é representado no modelo através da função G , que indica o afastamento do ponto de referência de um item p_i em relação ao *NFP* p_j, p_i .

A verificação da localização do ponto de referência de um polígono no semiplano do lado externo do *NFP* é feita usando a função que define a reta suporte da aresta do invólucro de posicionamento, sendo esta a reta que divide o plano. É acrescentado ao modelo uma disjunção de inequações para cada par de polígonos p_i e p_j , o que reflete na presença de variáveis binárias. Isto é semelhante ao que foi observado no trabalho de Luzzi. A diferença conceitual entre essas propostas reside na possibilidade que é assumida por Gomes, e vetada por Luzzi, de que as regiões externas do *NFP* associadas a duas diferentes arestas possam ter interseções.

Outra diferença já comentada da descrição do modelo feita por Gomes, em relação

ao de Luzzi, está no detalhamento feito ao traçar o procedimento para formação das restrições de não-sobreposição para os casos de existência de arestas côncavas no *NFP*, bem como concavidades dentro de concavidades. Para cada sequência de arestas côncavas do *NFP* p_i, p_j que pertençam a uma mesma concavidade, garantir que o ponto de referência r_j do item p_j esteja no exterior desta concavidade torna necessário que r_j esteja na região formada pela interseção entre os semiplanos gerados pelas retas suporte de todas as arestas côncavas que constituem a concavidade. Isto guia à possível existência de conjunções de inequações no sistema de disjunção de inequações citado anteriormente. Em termos práticos, cada conjunção deste tipo pode ser feita usando a mesma variável binária do sistema de disjunção de inequações para todas que representam uma mesma concavidade.

Quando há concavidades não convexas, Gomes indica a necessidade de dividir as concavidades em zonas convexas. Para cada zona convexa, uma variável binária deve ser usada para garantir a inclusão da interseção de todos os semiplanos formados pelas arestas desta zona no sistema de disjunção de inequações. Para cada concavidade que apresenta zonas convexas, é preciso incluir uma disjunção de sistemas de inequações formados por cada zona convexa. Consequentemente, será necessário incluir um novo grupo de variáveis binárias no modelo.

Os testes realizados por Gomes para este modelo salientam a incapacidade de sua aplicação em situações com instâncias que tenham um número de itens ainda consideravelmente baixos relativamente às condições de necessidades reais. Para instâncias simples com apenas dez a doze itens, a configuração do equipamento usado no experimento feito por aquele autor foi insuficiente para permitir a conclusão da execução do processo de resolução do modelo. Ainda é importante destacar que esses modelos de programação linear não lidam com a possibilidade de rotações de itens. Incluir esta possibilidade em uma abordagem que visa determinar a solução exata acrescenta à abordagem a necessidade de resolução de um diferente modelo de programação linear para cada diferente combinação de rotação de itens. Embora seja evidente a inviabilidade de aplicação em situações típicas dos processos de trabalho que requerem a resolução de problemas de *C&P*, há um grande potencial de aplicação dos conceitos usados na formulação destes modelos para a tarefa de melhoria de leiautes previamente construídos, como será comentado mais adiante. Por outro lado, a necessidade de obter bons resultados em tempos suficientemente curtos para garantir a aplicabilidade do método em situações reais leva às abordagens que não visam a obtenção da solução ótima para cada instância do problema. Algumas dessas são descritas na subseção seguinte.

2.1.3 Abordagens heurísticas para determinar solução

A aplicação prática de soluções computacionais para o problema de *C&P* ocorre em circunstâncias que não permitem o uso de algoritmos de resolução que apresentem a quantidade de instruções necessárias, ou tempo de execução, em função exponencial ou fatorial do número de itens a posicionar. A preocupação em maximizar o aproveitamento do espaço ocupado pelos itens e minimizar o tempo consumido para determinar a solução é citada por Burke et al. [2006]. É interessante, portanto, usar métodos que determinem uma solução satisfatória, mesmo que não seja ótima, em tempo suficientemente pequeno. Para isto, é comum utilizar métodos de busca para explorar o espaço de possíveis soluções. Os métodos de busca que usam estratégias que podem decidir se uma solução não ótima é mais promissora do que outras são chamados de métodos de busca heurística, conforme Russell et al. [2005]. Para esta pesquisa, foram consultados trabalhos relacionados que continham métodos heurísticos em três níveis diferentes do processo de busca, que serão denominados de heurísticas construtivas, métodos de busca local e meta-heurísticas. Estes três níveis de busca, juntamente com métodos de posicionamento, constituem a estrutura básica encontrada na literatura como uma linha geral para compor abordagens de resolução heurística para este problema.

2.1.3.1 Regras de posicionamento e heurísticas construtivas

De modo geral, as abordagens caracterizam-se por um estágio em que os dados relevantes são geométricos, visando garantir posicionamentos válidos para os itens. Neste nível estão envolvidos os processos de geometria computacional, como os já comentados *NFP*, *IFP* e *IFR*, por exemplo. Esses processos são usados em um nível acima, denominado método ou regra de posicionamento. A regra de posicionamento é responsável por determinar o conjunto de pontos válidos para cada item e decidir em qual ponto cada item será posicionado, além de também poder considerar rotações dos itens, como informado por Gomes [2005]. As heurísticas construtivas consultadas para este trabalho baseiam-se na seleção da sequência ordenada de itens aos quais será aplicada uma regra de posicionamento. Uma característica intrínseca dessas heurísticas construtivas é que um item posicionado não será removido ou reposicionado, ou seja, as soluções parciais são definitivas para estas heurísticas. Isto as define como algoritmos gulosos. Serão comentados duas regras de posicionamento, sendo uma a mais amplamente encontrada na literatura, denominada *bottom-left*, e outra denominada *TOPOS*.

A regra de posicionamento *bottom-left* (termo inglês que significa inferior-esquerda) é proposta com o objetivo de reduzir o comprimento do leiaute. Este comprimento é a maior das distâncias entre cada vértice de cada item posicionado e a reta

suporte da aresta inferior do objeto receptor. Esta reta suporte será referenciada como base do objeto receptor. Posicionando os itens com os seus respectivos pontos de referência o mais próximo possível da aresta inferior do objeto receptor evidencia a tentativa de minimização do comprimento do leiaute. Em caso de haver mais de uma posição com a mesma distância para a base do objeto receptor, a regra indica que o desempate deve favorecer sempre a escolha do ponto que tenha a menor distância para a reta suporte da aresta esquerda do objeto receptor. Deve-se lembrar que são consideradas neste trabalho as situações em que o objeto receptor é retangular, de largura fixa e comprimento suficiente para comportar qualquer leiaute.

Hopper [2000] descreve um algoritmo de posicionamento *bottom-left* que usa uma estratégia inicialmente semelhante à adotada para o problema de posicionamento de itens retangulares, usando retângulos envolventes para determinar a coordenada inicial. Em seguida, é feita uma sucessão de movimentos horizontais e verticais do polígono por distâncias livres, até que não haja mais movimentos possíveis que levem o polígono a alguma posição mais abaixo ou mais à esquerda. Esta posição final é denominada posição estável. Em Burke et al. [2006], os autores descrevem um algoritmo para o posicionamento dos itens, denominado *Bottom-Left-Fill*, que para cada item repete um procedimento de posicionar no ponto de origem do objeto receptor e, em seguida, resolve sobreposições até que seja encontrada uma posição válida. Neste algoritmo, a região interna do objeto receptor é dividida em segmentos de retas paralelas e verticais, separados por uma distância que é denominada resolução.

Gomes [2005] também utiliza a regra de posicionamento *bottom-left*, e apresenta um método para restringir a um conjunto discreto e finito os pontos válidos para o posicionamento de um item, com base no *NFP* e no *IFR*. Seja p_k um item a posicionar, e os itens p_i , $i = 1, \dots, k - 1$, já posicionados. Segundo aquele autor, cada vértice da região de posicionamento admissível é um vértice do NFP_{p_i, p_k} , para algum $i \in \{1, \dots, k - 1\}$, ou um vértice de IFR_{p_k} , ou um ponto de interseção entre quaisquer arestas $a_1 \in NFP_{p_j, p_k}$ e $a_2 \in NFP_{p_l, p_k}$, sendo que $j, l \in \{1, \dots, k - 1\}$, ou uma interseção entre duas arestas $a_1 \in IFR_{p_k}$ e $a_2 \in NFP_{p_l, p_k}$, sendo $l \in \{1, \dots, k - 1\}$. Além disso, os pontos admissíveis devem estar no interior de IFR_{p_k} e não estar no interior de qualquer NFP_{p_j, p_k} , $j \in \{1, \dots, k - 1\}$. Basta ordenar a lista formada com os pontos válidos pela proximidade com a base do objeto receptor, resolvendo empates com o critério de proximidade com a reta suporte da lateral esquerda do objeto receptor. O primeiro ponto desta lista ordenada é a posição que deverá ser ocupada pelo item p_k . Este processo é repetido para cada item, até que todos estejam posicionados.

Oliveira et al. [2000] apresentam um algoritmo construtivo, denominado TOPOS, que permite o uso de uma variedade de heurísticas para definir um leiaute. Em linhas

gerais, o algoritmo TOPOS permite o uso de duas variantes de execução básicas, sendo que uma se vale da ordenação inicial dos itens, e a outra é baseada no que foi descrito pelos autores como uma busca local. Esta segunda variante verifica, a cada vez que seleciona um item para posicionar, qual é a melhor alternativa entre os itens ainda não posicionados, observando os itens e as orientações disponíveis, segundo um critério de avaliação que pode ser um dos sugeridos pelos autores, que são o desperdício gerado, a sobreposição de retângulos envolventes, a distância para o ponto central do leiaute parcial, ou alguma combinação destes. A primeira variante deste algoritmo seleciona um item de acordo com uma ordenação feita previamente ao início dos posicionamentos, e usa os mesmos critérios citados anteriormente para decidir entre as orientações possíveis para o item selecionado.

Após selecionar um item e sua orientação, o algoritmo procede a seleção do ponto de posicionamento, que é avaliado segundo outro critério, como um dos que foram sugeridos pelos autores, que são a minimização da área do retângulo envolvente da solução parcial, ou a minimização do comprimento do retângulo envolvente da solução parcial, ou a maximização da sobreposição entre os retângulos envolventes dos itens posicionados. A cada item posicionado, a solução parcial é agrupada em um só polígono, denominado perfil, formado pelo contorno externo da união dos polígonos já posicionados, o que despreza as possíveis regiões não ocupadas entre os itens.

Bennell & Song [2010] fazem uma revisão do algoritmo TOPOS, e empregam algumas modificações. Uma destas é a inclusão da possibilidade de aproveitamento dos espaços entre os itens posicionados, deixando de usar o perfil da solução parcial como proposto na formulação original do algoritmo. Além disso, este algoritmo revisado também inclui modificações nas funções dos critérios de avaliações usados. Aqueles autores apresentam ainda um comparativo entre o TOPOS revisado e o original, aplicando o algoritmo revisado a cinco instâncias do problema também testadas em Oliveira et al. [2000], sendo que o algoritmo revisado superou o original em três destas.

2.1.3.2 Buscas locais

Vários trabalhos consultados apresentam algoritmos que fazem tentativas de melhorar a qualidade da solução fornecida pela heurística construtiva. Esses algoritmos de busca por melhorias podem ser classificados em duas categorias, ou níveis, que compreendem a busca local e a busca meta-heurística. A busca local é definida por um processo que avalia possíveis pequenas modificações na solução original, que é fornecida pela heurística construtiva. Por outro lado, a busca meta-heurística é geralmente definida por um processo que visa ser mais abrangente na observação do espaço de soluções possíveis.

A busca local, segundo Kendall [2000], baseia-se no conceito de vizinhança para explorar o espaço de soluções. A maneira usada para definir uma vizinhança é denominada estrutura de vizinhança. Nos trabalhos consultados foram encontradas várias diferentes estruturas de vizinhanças. Gomes [2005] classifica a busca local para o problema de *C&P* em duas categorias, sendo uma a pesquisa sobre sequências, e outra a pesquisa sobre padrões de corte. A pesquisa sobre sequências é feita através de duas etapas, sendo que a primeira consiste em fazer pequenas modificações na sequência da solução original, e a segunda é a execução da regra de posicionamento para transformar a sequência em um leiaute. A pesquisa sobre padrões de corte é feita através de modificações nas posições dos itens em um leiaute pronto, possivelmente recorrendo a alguma subrotina para eliminar possíveis situações inadmissíveis, como sobreposições entre itens ou posicionamento de parte de algum item em região externa ao objeto receptor.

Hopper [2000] reúne em sua pesquisa informações sobre estruturas de vizinhanças definidas através de movimentos de itens, como translação, rotação, troca de posições entre dois itens adjacentes ou aleatórios, e reordenando os itens de acordo com os tipos. Além disso, aquela autora também usa uma estrutura de vizinhança formada com a combinação de partes de duas sequências de itens de duas soluções diferentes. Gomes & Oliveira [2001] definem a vizinhança através de um procedimento com nome *2-exchange*. Esta vizinhança é estabelecida pela troca de posições de dois itens na sequência original. Um parâmetro Δ indica a distância, na sequência, entre os itens trocados. Burke et al. [2006] definem operadores para estabelecer a estrutura de vizinhança, sendo estes operadores capazes de realizar a seleção aleatória de itens e trocar suas posições. A quantidade de itens selecionados para movimentação pode variar, possibilitando variar o impacto da perturbação provocada.

Gomes & Oliveira [2006] definem a vizinhança usando uma combinação de movimentos, sendo que o primeiro é uma pequena perturbação na solução, feita com a mudança da posição de um item ou pela troca de posições de dois itens, seguida de um processo de separação, para eliminar possíveis sobreposições entre itens, e terminando com um processo de compactação, que visa reduzir os espaços não aproveitados entre os itens. Ao processo de busca local com esta estrutura, aqueles autores referem-se como *LocalCompact*. Egeblad et al. [2007] definem a vizinhança usando alguns movimentos sobre os itens posicionados, como a translação horizontal, translação vertical, rotação e inversão, que podem incorrer em situações de sobreposição entre itens.

Um procedimento comum de busca local consiste em avaliar a vizinhança à procura de uma solução melhor do que a atual, como representado na figura 2.7. Ao encontrar solução melhor na avaliação da vizinhança, adota esta como a solução atual e repete o processo. Quando não há solução melhor na vizinhança, a busca local é encerrada. A este

procedimento dá-se o nome de *Hill climbing* ou pesquisa local pura. Algumas variações deste procedimento devem ser consideradas, com relação à observação da vizinhança e a decisão de assumir uma solução melhor. Uma possibilidade é avaliar todas as soluções da vizinhança, de forma exaustiva, e selecionar a melhor delas para assumir como solução corrente. Em virtude do potencialmente grande número de soluções da vizinhança, outras alternativas são usadas, como a substituição da solução atual assim que uma solução melhor é encontrada na vizinhança. Como indicado em Hopper [2000] e Gomes [2005], o problema dos métodos de busca local está na incapacidade de escapar de soluções ótimas locais, que não representam ótimos globais.

```

PesquisaLocalPura() {
  s = GeraSoluçãoInicial()
  condição = 0
  repita {
    s' = PesquisaVizinhança(s)
    se (s' == s)
      condição = 1
    s = s'
  } até (condição == 1)
  retorna solução s
}

```

Figura 2.7. Algoritmo de busca local adaptado de Gomes [2005].

2.1.3.3 Meta-heurísticas

Na tentativa de realizar pesquisas com a capacidade de escapar de ótimos locais insatisfatórios, vários autores experimentam mecanismos de busca meta-heurística para obter boas soluções para o problema de *C&P*. Em linhas gerais, as buscas meta-heurísticas atuam expandindo a avaliação de soluções de modo mais distribuído pelo espaço de soluções. Entre esses mecanismos estão a pesquisa por recozimento simulado, *GRASP* e algoritmos genéticos.

Martins [2007] emprega a meta-heurística de recozimento simulado para explorar o espaço de soluções. Segundo aquele autor, este é um método de busca iterativo que avalia, a cada iteração, apenas uma solução x_i^* do problema, obtida aleatoriamente na vizinhança da solução atual x_i . Esta vizinhança é definida com uma operação escolhida aleatoriamente entre três alternativas, sendo uma troca de itens da sequência de construção, ou uma rotação do item por um ângulo permitido, ou uma translação de um item para uma nova posição aceitável. Uma função de custo f é usada para avaliar a solução da vizinhança e compará-la à atual. Se $f(x_i^*) \leq f(x_i)$, a solução avaliada tem custo menor ou igual à solução atual e, neste caso, há uma imediata substituição da solução atual

pela solução avaliada. Por outro lado, se $f(x_i^*) > f(x_i)$, o custo da solução avaliada é maior do que o custo da solução atual. Para este caso, é considerada uma probabilidade de substituição da solução atual pela solução avaliada, sendo esta probabilidade determinada pela diferença entre os custos das soluções e pela variável que armazena a “temperatura” do processo. Quanto mais alta é a temperatura, maior é a probabilidade do processo optar por soluções com maior custo. Esta característica permite escapar de ótimos locais. Durante a busca, a temperatura é reduzida gradualmente, até que seja fixada em uma solução final. A figura 2.8 apresenta o algoritmo que foi descrito. Gomes & Oliveira [2006] também usam o método de busca por recozimento simulado aplicado ao problema de *C&P*, mas de modo diferenciado por adotarem a estrutura de vizinhança *LocalCompact* já mencionada.

```

RecozimentoSimulado() {
  x = solução inicial aleatória
  i = 0
  enquanto (critério de parada global não satisfeito) {
    t = Resfriamento(i)
    i = i + 1
    enquanto (critério de parada local não satisfeito) {
      x* = ModificaçãoAleatória(x, t)
      ΔE = f(x*) - f(x)
      se (ΔE < 0)
        x = x*
      senão {
        a = número aleatório entre 0 e 1
        se (a < limite de aceitação)
          x = x*
      }
    }
  }
  retorna solução x
}

```

Figura 2.8. Algoritmo de busca por recozimento simulado adaptado de Martins [2007].

Gomes & Oliveira [2001] e Gomes [2005] fazem uso do método de busca meta-heurística *GRASP* (*Greedy Randomized Adaptive Search Procedure*) para pesquisar o espaço de soluções. Neste método, um número determinado de iterações é executado. A cada iteração, três etapas são realizadas. Primeiro, uma nova solução é obtida pela heurística construtiva modificada pela inclusão de um fator de aleatoriedade. Este fator possibilita obter soluções variadas com a aplicação da mesma heurística construtiva. Segundo, a solução obtida é submetida ao processo de busca local pura, que determina um ótimo local a partir dela. Por fim, é feita uma comparação da solução ótima local com

a melhor solução já avaliada. Se a solução ótima local for melhor, esta substitui a melhor solução avaliada. Ao término do processo iterativo, a melhor solução encontrada é retornada como resultado do processo. A figura 2.9 exhibe o algoritmo comentado. Naqueles trabalhos, os autores usam para esta busca a estrutura de vizinhança *2-exchange* já comentada.

```

GRASP(númeroIterações,  $\alpha$ ) {
   $s^* = +\infty$ 
  para ( $i = 0$ ;  $i < \text{númeroIterações}$ ;  $i = i + 1$ ) {
     $s' = \text{HeurísticaConstrutiva}(\alpha)$ 
     $s = \text{PesquisaLocalPura}(s')$ 
    se ( $s < s^*$ )
       $s^* = s$ 
  }
  retorna solução  $s^*$ 
}

```

Figura 2.9. Algoritmo GRASP adaptado de Gomes [2005].

Hopper [2000] conduz um estudo que envolve a aplicação de algoritmos genéticos ao problema de *C&P*. Naquele trabalho, o algoritmo genético é implementado sobre a representação de sequências de itens, sendo que permutações de itens na sequência formam soluções alternativas. A função objetivo é a minimização do comprimento do leiaute. Na abordagem com algoritmos genéticos, cada solução é tratada como um indivíduo, e o progresso da execução do algoritmo simula cruzamentos entre os indivíduos da população, com eventuais mutações interferindo em características de alguns indivíduos. A sequência de itens da solução representa os genes do indivíduo. Os indivíduos que apresentem melhores valores para a função objetivo são favorecidos, tendo maior probabilidade de reproduzirem. Como afirma aquela autora, isto simula simplificadaamente um processo de evolução da espécie inspirado na teoria da evolução de Darwin. Alguns operadores são aplicados ao processo de evolução, que são o próprio cruzamento e mutações. O cruzamento consiste na combinação de parte da sequência de itens de um indivíduo, completada com parte da sequência de itens de outro indivíduo. Isto permite gerar novas soluções, como se fosse a geração de novos indivíduos de uma população. Naquele trabalho, as mutações são implementadas com a troca de itens da sequência e mudando a orientação de itens. Além disso, a população inicial é gerada aleatoriamente com 50 indivíduos, e o processo segue por 1000 gerações. Ao término, a melhor solução encontrada é retornada como resultado da busca.

Alguns trabalhos consultados empregam em suas abordagens combinações de técnicas de busca com a melhoria de soluções com uso de modelos de programação linear. A

seguir, é comentado um desses modelos, usado para compactar leiautes através de ajustes coordenados nas posições dos itens.

2.1.4 Compactação de leiautes com programação matemática

No trabalho publicado em Li & Milenkovic [1995b], são apresentados dois modelos de compactação de leiautes. Um deles é baseado na simulação da aplicação de forças sobre os itens, havendo deslocamentos dos mesmos em intervalos de tempo. Neste processo, o modelo de simulação de movimentos é construído de modo a identificar um conjunto de velocidades, uma para cada item, que podem ser aplicadas aos itens em um intervalo de tempo, sem que haja ocorrência de sobreposição. O processo é repetido até que não seja possível aplicar novos deslocamentos que reduzam o comprimento do leiaute. Como limitação, o trabalho daqueles autores tem sua aplicação restrita a polígonos em forma de estrela. Esses polígonos caracterizam-se por possuir pelo menos um ponto em seu interior, tal que deste ponto haja um segmento de reta para qualquer outro ponto do interior do polígono, de modo que este segmento de reta esteja completamente no interior do polígono. Outros modelos de compactação são apresentados por outros autores, com modificações que possibilitam evitar a restrição de polígonos em forma de estrela.

Um desses modelos é descrito em Gomes & Oliveira [2006]. Neste modelo, o objetivo é reduzir o comprimento do leiaute para elevar a eficiência do posicionamento. Usando as definições apresentadas nas equações 2.2 a 2.9, além da adequação ao sistema de coordenadas adotado no presente trabalho, que usa um eixo de valores x colinear à base do objeto receptor e um eixo y colinear à lateral esquerda do objeto receptor, o modelo de compactação apresentado naquele trabalho pode ser adaptado como descrito na figura 2.10. Para este modelo, é necessário definir novas constantes. O par de constantes X_i^* e Y_i^* é responsável por reter as coordenadas da posição inicial do item p_i . A localização do ponto de referência r_i do item p_i é usada como a posição inicial do item. É definido um par de constantes (X_i^*, Y_i^*) para cada $p_i \in \mathcal{P}$. As constantes W_0 e H_0 são definidas, respectivamente, como a largura do objeto receptor e o comprimento, ou altura, do leiaute inicial. Um outro grupo de constantes, $DISTANCE_i$, sendo uma para cada $p_i \in \mathcal{P}$, destina-se a limitar o deslocamento dos itens. Segundo aqueles autores, a função deste limite de deslocamentos é tornar a compactação mais suave. Vale frisar aqui a inclusão da resolução deste modelo de compactação em um procedimento iterativo. Ou seja, a solução deste modelo indica novas posições para os itens, e a diferença entre as posições anteriores e as novas constituem os deslocamentos dos itens. Ao término de um reposicionamento dos itens, um novo modelo de compactação é construído e novas posições são definidas.

Este processo é encerrado quando a resolução do modelo de compactação não fornece modificações nas posições dos itens. Segundo Gomes [2005], que apresenta um modelo de compactação no qual usa uma constante para cada componente x e outra para cada componente y de cada item, a restrição da amplitude dos deslocamentos e o processo iterativo tornam o método mais eficaz.

$$\begin{aligned} \min z & & (2.23) \\ \text{sujeito a:} & & (2.24) \\ z \geq y_i + L_i & & i = 1, \dots, N & (2.25) \\ x_i - X_i^* \leq DISTANCE_i & & i = 1, \dots, N & (2.26) \\ y_i - Y_i^* \leq DISTANCE_i & & i = 1, \dots, N & (2.27) \\ -x_i + X_i^* \leq DISTANCE_i & & i = 1, \dots, N & (2.28) \\ -y_i + Y_i^* \leq DISTANCE_i & & i = 1, \dots, N & (2.29) \\ x_i \geq 0 + right_i & & i = 1, \dots, N & (2.30) \\ y_i \geq 0 + bottom_i & & i = 1, \dots, N & (2.31) \\ x_i \geq W_0 - left_i & & i = 1, \dots, N & (2.32) \\ y_i \geq H_0 - top_i & & i = 1, \dots, N & (2.33) \\ f(x_j - x_i, y_j - y_i) \leq 0 & & i, j = 1, \dots, N; i \neq j & (2.34) \\ x_i, y_i, z \geq 0 & & i = 1, \dots, N & (2.35) \end{aligned}$$

Figura 2.10. Modelo matemático de compactação adaptado de Gomes & Oliveira [2006].

O conjunto de inequações 2.25 define a variável z como o comprimento do leiaute. Os conjuntos de inequações de 2.26 a 2.29 fazem a limitação da amplitude dos deslocamentos dos itens. Os conjuntos de inequações de 2.30 a 2.33 garantem que todo item p_i será posicionado no interior do respectivo IFR_i . O conjunto de inequações 2.34, sendo uma inequação para cada par de itens p_i e p_j , são responsáveis por garantir a não sobreposição dos itens. A função $f(x_j - x_i, y_j - y_i)$ refere-se à penetração do item p_j no interior do $NFP_{i,j}$, sendo caracterizada uma sobreposição na ocorrência de valor positivo. As inequações do conjunto 2.35 formam as restrições de integralidade das variáveis do modelo.

É importante destacar a composição da função f . Para cada par de itens p_i e p_j , será selecionada uma aresta do $NFP_{i,j}$ que tenha o ponto de referência r_j à sua esquerda, se os vértices do $NFP_{i,j}$ estiverem em sentido horário, ou à sua direita, caso contrário. Se mais de uma aresta do invólucro de posicionamento satisfizer esta condição, a que tiver

sua reta suporte mais distante do ponto r_j será selecionada. A função f deve garantir que os deslocamentos dos itens p_i e p_j preservem r_j do mesmo lado da reta suporte da aresta selecionada do $NFP_{i,j}$. Isto pode ser feito usando uma inequação que defina a distância de r_j à reta suporte da aresta selecionada do invólucro de posicionamento, em função do posicionamento do item p_i .

Segundo Gomes & Oliveira [2006], três situações distintas são derivadas desta decisão de seleção de aresta. Uma delas é a situação trivial, na qual uma aresta pertence a uma sequência convexa do invólucro de posicionamento e não há empates com outras arestas na distância para r_j . As outras duas situações são casos que merecem atenção especial. Uma ocorre quando r_j coincide com um vértice do $NFP_{i,j}$. Neste caso, duas arestas do invólucro de posicionamento estarão a distância zero de r_j . Gomes [2005] determina que a escolha favoreça a aresta que seja menos restritiva à compactação, ou seja, deve-se optar pela reta cuja inclinação seja mais próxima da inclinação da lateral do objeto receptor. A outra situação especial ocorre quando a aresta selecionada do $NFP_{i,j}$ pertence a uma sequência de arestas côncavas. Neste caso, deve ser incluído no modelo uma inequação para cada aresta da sequência que forma a concavidade.

Este modelo de compactação permite reorganizar os itens, possibilitando a redução do comprimento do leiaute. Deve-se observar ainda que, a cada passo do processo iterativo descrito, as posições relativas entre cada par de itens não são alteradas. Isto ocorre em decorrência da ação das restrições de não-sobreposição. As reconstruções do modelo, a cada iteração, permitem que essas posições relativas sejam reavaliadas, o que favorece a obtenção de novas melhorias a cada passo. Por ser um modelo de programação linear sem variáveis binárias ou inteiras, sua resolução é simples. Isto torna este método atrativo para aplicação em ferramentas de resolução do problema de $C\&P$, podendo levar as abordagens a níveis mais competitivos.

2.2 Estruturas de testes e de comparativos de desempenho

Para afirmar que uma abordagem é competitiva, é necessário haver conhecimento sobre os aspectos que caracterizam tangivelmente cada abordagem, possibilitando estabelecer algum critério de classificação. Neste caso, os objetos a classificar são abordagens para obter solução para o problema de $C\&P$, bidimensional e possivelmente irregular, ou seja, novos algoritmos. Segundo Hopper [2000], instâncias de referência são extremamente importantes para avaliar o desempenho de novos algoritmos, facilitando a formação de disputas entre diferentes abordagens.

Hopper [2000] também faz uma organização de instâncias obtidas na literatura especializada, aplicadas para avaliar a eficiência das abordagens publicadas. Como, com raras exceções, as instâncias apresentadas nas publicações científicas não estavam devidamente detalhadas, aquela autora fez um trabalho de digitalização, voltado à identificação de dados que permitissem reproduzir aproximadamente aquelas instâncias. Como resultado, foram fornecidas naquele trabalho uma coleção com 17 instâncias de referência. Estas são disponibilizadas na *World Wide Web* pelo grupo europeu de interesse em problemas de corte e empacotamento (*ESICUP - EURO Special Interest Group on Cutting and Packing*), e são amplamente adotadas na literatura para testar e comparar abordagens de solução para este tipo de problema.

Uma estrutura geral é usada para descrever cada instância dessa coleção. Esta estrutura é composta pelos seguintes atributos:

- Um nome que identifica a instância;
- O número de itens a posicionar;
- A largura do objeto receptor;
- Uma coleção de itens, sendo cada item caracterizado pela seguinte relação de atributos:
 - A quantidade de ocorrências do item que devem ser posicionadas;
 - Os ângulos admissíveis para efetuar rotações no item;
 - Uma lista com coordenadas em \mathbb{R}^2 de cada vértice do polígono que representa o item, sendo esta lista ordenada em sentido anti-horário.

A tabela 2.1 exibe dados das instâncias mencionadas, com exceção da instância denominada “Han”, pois sua aplicação em experimentos computacionais não foi encontrada em outros trabalhos consultados nesta pesquisa. Na primeira coluna está o nome identificador da instância. A segunda coluna exibe a largura w do objeto receptor. A terceira coluna apresenta o número total de itens a posicionar. Na quarta coluna está o número médio de vértices por item da instância. E por fim, na quinta coluna estão os ângulos que são aceitos para que possam ser aplicadas rotações nos itens.

A resolução das instâncias é descrita por uma relação de pares de valores, um par para cada item, indicando a posição em que os itens devem ser posicionados, através de translações guiadas pelos pontos de referência dos itens. Para comparativos, é assumido o valor do comprimento do leiaute. Este comprimento \mathcal{C} equivale ao maior valor de coordenada y encontrado entre os itens, como apresentado na equação 2.36 da figura 2.11.

Outra forma usual de apresentar os resultados, e que será usado neste trabalho, é através do aproveitamento proporcionado por um leiaute, que é equivalente ao comprimento do leiaute. Este aproveitamento é representado através do percentual de ocupação da área retangular do objeto receptor, pelos itens posicionados. Sendo s_i o valor da área do interior do item p_i , w o valor da largura do objeto receptor e n o número de itens posicionados, a equação 2.37 da figura 2.11 descreve o cálculo do aproveitamento \mathcal{A} . Outro dado interessante para comparar as abordagens é o tempo necessário para atingir as soluções. Isto pode interferir na aplicabilidade prática de uma abordagem em alguma atividade relacionada ao problema de *C&P*.

$$\mathcal{C} = \max\{y_i^{\max}, \forall p_i \in \mathcal{P}\} \quad (2.36)$$

$$\mathcal{A} = 100 \times \frac{\sum_{i=1}^n s_i}{w \times \mathcal{C}} \quad (2.37)$$

Figura 2.11. Aspectos observados em resultados: comprimento (2.36) e aproveitamento de área (2.37).

É comum que os algoritmos avaliados tenham parâmetros que interferem no seu comportamento e que, se variados, levam a soluções alternativas para uma mesma instância. Portanto, nos casos de abordagens que usem algoritmos parametrizados, os testes devem explorar suficientemente as combinações válidas de diferentes valores para os parâmetros. Se possível, é válido buscar associações entre combinações específicas de parâmetros e características das instâncias. Isto pode enriquecer a análise da abordagem em estudo. A seguir, são apresentados os melhores resultados conhecidos para as instâncias de referência apresentadas na tabela 2.1.

Tabela 2.1. Instâncias de referência para o problema de *C&P* em estudo.

Instância	w	n	Média vértices/item	Ângulos permitidos (graus)
albano	4900	24	7.25	0, 180
blaz2	15	20	7.50	0, 180
dagli	60	30	6.30	0, 180
dighe1	100	16	3.87	0
dighe2	100	10	4.70	0
fu	38	12	3.58	0, 90, 180
jakobs1	40	25	5.60	0, 90, 180
jakobs2	70	25	5.36	0, 90, 180
mao	2550	20	9.22	0, 90, 180
marques	104	24	7.37	0, 90, 180
shapes0	40	43	8.75	0
shapes1	40	43	8.75	0, 180
shapes2	15	28	6.29	0, 180
shirts	40	99	6.63	0, 180
swim	5752	48	21.90	0, 180
trousers	79	64	5.06	0, 180

2.3 Melhores resultados conhecidos

Sendo a minimização do comprimento do leiaute o objetivo fundamental do problema de *C&P* bidimensional irregular, o valor associado, correspondente ao aproveitamento \mathcal{A} , pode ser usado para classificar os resultados obtidos pelas novas abordagens, que são propostas por vários pesquisadores em seus trabalhos. É comum que os resultados publicados sobre as abordagens conhecidas sejam mais eficazes para algumas instâncias e menos para outras. Para fornecer dados mais precisos sobre o desempenho das abordagens nos experimentos realizados, alguns autores incluem em seus relatos os registros de médias de resultados por instância, além de registros de tempos de execução. Exemplos de trabalhos que apresentam tais dados são Bennell & Oliveira [2008], Burke et al. [2006], Egeblad et al. [2007], Gomes [2005], Imamichi et al. [2009] e Nielsen [2007].

Os melhores resultados obtidos de publicações de trabalhos científicos para as instâncias de referência relacionadas são resumidos na tabela 2.2. Na segunda e na terceira colunas dessa tabela, estão incluídas na classificação as abordagens baseadas em técnicas de busca local ou meta-heurísticas, ou combinações destas com programação linear

Tabela 2.2. Melhores resultados encontrados na literatura para as instâncias de referência.

Instância	Todas as abordagens		Heurísticas construtivas	
	Aproveitamento (%)	Fonte	Aproveitamento (%)	Fonte
albano	88,39	B	80,90	H
blaz2	79,90	G	-	-
dagli	87,97	A	78,83	H
dighe1	100,00	C	68,78	H
dighe2	100,00	C	71,84	H
fu	92,03	E	79,17	H
jakobs1	89,07	E	75,39	H
jakobs2	84,83	B	68,93	H
mao	85,15	E	72,82	H
marques	90,01	B	80,59	H
shapes0	68,44	D	61,39	H
shapes1	73,84	E	67,60	J
shapes2	84,25	D	74,74	I
shirts	89,69	A	84,10	J
swim	75,60	F	66,82	H
trousers	90,60	F	86,00	J

ou não-linear. Na quarta e na quinta colunas da mesma tabela estão os melhores resultados obtidos apenas por algoritmos que representam heurísticas construtivas. Apenas para a instância identificada pelo nome “blaz2” não foram encontrados resultados obtidos apenas com heurísticas construtivas. Os resultados de heurísticas construtivas são usados para comparativos de um método proposto no presente trabalho. Nos dois casos, as fontes estão identificadas por símbolos que se referem às seguintes abordagens:

- A. *Beam-Search*, Bennell & Song [2010];
- B. *ResolveDual*, Sato [2011];
- C. *SAHA*, Gomes [2005];
- D. *ILSQN*, Imamichi et al. [2009];
- E. *2DNest*, Egeblad et al. [2007];
- F. *AutoNester-T*, Nielsen & Odgaard [2003];

G. *Local-Search*, Burke et al. [2006];

H. *Bottom-Left*, Gomes [2005];

I. TOPOS, Oliveira et al. [2000];

J. TOPOS revisado, Bennell & Song [2010].

2.4 Considerações sobre o capítulo

Neste capítulo foi apresentada uma visão geral sobre os trabalhos da literatura, relacionados ao problema de *C&P*, que foram consultados. A partir do estudo dessas publicações, foi possível estabelecer uma definição clara para o problema. Várias abordagens para solução foram brevemente descritas. Essas abordagens incluem, de um modo geral, tratamentos para os subproblemas de geometria computacional e de otimização combinatória. Para a otimização combinatória, foram observados métodos que visam obter soluções ótimas ou aproximadas, estas últimas por meio de técnicas de buscas heurísticas. Este estudo permitiu reunir informações consideradas valiosas para o desenvolvimento da pesquisa relatada nas seções seguintes.

Capítulo 3

Proposta de nova abordagem construtiva para *C&P*

Esta pesquisa propõe o estudo de um novo método construtivo para compor abordagens de resolução para o problema de corte e empacotamento, com formas bidimensionais e irregulares. Este método baseia-se na intercalação de operações de posicionamento e operações de compactação, estas aplicadas ao leiaute parcialmente construído. Neste capítulo, será apresentada a estrutura principal do algoritmo proposto, seguida de explicações sobre as sub-rotinas que são invocadas pelo algoritmo. Estão entre estas as heurísticas para obter o próximo item a ser posicionado, o cálculo do invólucro de posicionamento entre dois itens, a regra para determinação do local para o posicionamento do próximo item, e a operação de compactação empregada.

3.1 Algoritmo proposto

O algoritmo proposto neste trabalho tem por objetivo atuar no posicionamento de polígonos no interior de um objeto retangular de largura fixa e comprimento virtualmente ilimitado. A abordagem desenvolvida oferece soluções que podem ser aproveitadas diretamente ou como entradas para um processo de alocação de itens através de algum tipo de busca. Como está comentado na seção 2.1.3.1, neste nível de operação é formado um elo entre três elementos, sendo estes uma coleção de funções de geometria computacional, uma regra de posicionamento e uma heurística para composição de uma sequência dos itens a posicionar.

O método construtivo proposto neste trabalho mantém componentes da estrutura comum presente nas abordagens consultadas na literatura. Ou seja, este método consiste em um laço repetitivo de dois passos fundamentais, aplicados uma vez para cada item da

coleção de itens que devem ser posicionados. Um dos passos é o emprego de uma heurística gulosa para selecionar o próximo item a posicionar. O outro passo é a determinação da posição que este item deve ocupar no objeto receptor. A inovação deste método está na inclusão de um terceiro passo, de execução intermitente, que efetua a compactação do leiaute parcialmente construído. O algoritmo está apresentado na figura 3.1.

```

01 Constrói_Compactando (P, w, ordemPrimária, ordemSecundária,
02                       regra, opçãoAgrupamento,
03                       intervaloEntreCompactações)
04 Início
05   L := 0;
06   contaAgrupamento := 0;
07   para i de 1 a n faça
08     item := Selecciona_Próximo_Item(ordemPrimária, ordemSecundária);
09     pos := Selecciona_Posicionamento_Válido(regra, item, L, w);
10     L := L + Aplica_Posicionamento(item, pos);
11     se (opçãoAgrupamento = porItens) ou
12         (opçãoAgrupamento = porFormas e
13          Quantidade_Itens_Restantes_Forma(item) = 0) então
14       contaAgrupamento := contaAgrupamento + 1;
15     fim_se
16     se contaAgrupamento > intervaloEntreCompactações então
17       contaAgrupamento := 0;
18       Compacta_Leiaute_Parcial(L, w);
19     fim_se
20   fim_para
21   se contaAgrupamento > 0 então
22     Compacta_Leiaute_Parcial(L, w);
23   fim_se
24   retorne L;
25 Fim.

```

Figura 3.1. Método construtivo proposto.

O método *Constrói_Compactando* possui sete parâmetros. Os dois primeiros são, respectivamente, uma coleção de polígonos que representam os itens a posicionar, **P**, e o valor da largura fixa do objeto receptor, **w**. A variação dos outros parâmetros permite controlar o comportamento do algoritmo. A descrição destes é feita na relação abaixo.

ordemPrimária Este valor identifica a heurística gulosa que será aplicada para selecionar o próximo item a ser posicionado.

ordemSecundária Este valor identifica a heurística gulosa que será usada para resolver ocorrências de empates na aplicação da heurística primária.

regra É a identificação da regra de posicionamento que determinará as posições no sistema de coordenadas do objeto receptor, para que os itens sejam transladados até elas.

opçãoAgrupamento É um valor que identifica o tipo de agrupamento de itens feito entre as aplicações da rotina de compactação. Pode assumir um dos valores *porItens* ou *porFormas*. O agrupamento por itens é definido pela contagem de cada item posicionado como um elemento adicional. No agrupamento por formas, a contagem de elemento do agrupamento é incrementada apenas após o posicionamento de todos os itens que sejam representados por polígonos congruentes ao que representa o item selecionado pela heurística.

intervaloEntreCompactações Este parâmetro define o tamanho do agrupamento feito entre as aplicações da rotina de compactação, ou seja, representa o número de itens que deverão ser posicionados até que a compactação seja novamente efetuada.

Internamente, são usadas quatro variáveis para a construção do leiaute, e o valor n , que representa o número de itens da coleção P . Essas variáveis são descritas abaixo.

\mathcal{L} É uma estrutura que permite manter uma coleção de associações de itens a posições. Isto será usado para representar o leiaute em construção.

contaAgrupamento Este valor retém o número de elementos do agrupamento formado desde a última aplicação da rotina de compactação, ou desde o início do processo, caso ainda não tenha sido feita alguma compactação.

item Armazena temporariamente o item que foi selecionado pela heurística responsável por fornecer a sequência para o posicionamento.

pos Armazena temporariamente uma posição, formada pelas coordenadas bidimensionais no sistema de coordenadas do objeto receptor, que foi indicada pela regra de posicionamento, e também um ângulo para que seja aplicada uma rotação ao item que está sendo posicionado, se necessário.

O algoritmo também emprega cinco sub-rotinas para auxiliar na construção do leiaute. Essas sub-rotinas são descritas na relação abaixo, e serão analisadas mais detalhadamente nas seções adiante.

Seleciona_Próximo_Item Executa a heurística de seleção do próximo item a posicionar. Usa dois critérios para ordenação dos itens, sendo um aplicado primariamente e outro aplicado secundariamente, para resolver casos de empates.

Seleciona_Posicionamento_Válido Este método aplica a regra de posicionamento para determinar o ângulo de rotação do item e as coordenadas da posição que deve ser ocupada pelo item selecionado no leiaute em construção.

Aplica_Posicionamento Faz a rotação do item selecionado, com o ângulo indicado, e faz a translação do mesmo item para a posição indicada pela regra.

Quantidade_Itens_Restantes_Forma Informa quantos itens congruentes ao item selecionado ainda não foram posicionados.

Compacta_Leiaute_Parcial Este método transforma o leiaute parcial, usando modelo de programação linear baseado no trabalho de Gomes [2005], que foi comentado na seção 2.1.4, para obter novas coordenadas para os itens posicionados.

A execução do algoritmo proposto inicia, nas linhas 5 e 6, com a definição dos valores iniciais para o leiaute construído, que é vazio, e da contagem do número de elementos do agrupamento, que é zero, pois considera-se que nenhum item foi posicionado antes do início da execução. Em seguida, na linha 7, parte-se para um bloco de operações que serão executadas n vezes, sendo n o número de itens de P . Este bloco de operações, que segue até a linha 20, consiste em selecionar o próximo item a posicionar, usando a combinação de critérios primário e secundário de ordenação da coleção P , na linha 8, aplicar a regra de posicionamento para determinar o local para onde o item selecionado deve ser transladado, na linha 9, e incluir este item posicionado no leiaute \mathcal{L} , na linha 10. Em seguida, nas linhas 11, 12 e 13, é verificado se o posicionamento feito na iteração atual deve incrementar a contagem de agrupamentos. Isto será satisfeito se a opção de agrupamento for por itens, pois neste caso cada item é considerado um elemento adicional do agrupamento, ou se a opção de agrupamento for por formas e não houver mais itens representados por polígonos congruentes ao que acabou de ser posicionado. O incremento da contagem de agrupamentos é feita pelo comando da linha 14. Na continuação do bloco iterativo de operações, entre as linhas 16 e 19, é feito o passo de compactação, que está condicionado à contagem de elementos do agrupamento ter superado o limite do intervalo de posicionamentos entre as compactações. Se esta condição for satisfeita, então a contagem de agrupamentos é reiniciada em zero e é efetuada uma compactação do leiaute parcial. Ao término das iterações para todos os itens de P , entre as linhas 21 e 23, é feita uma verificação pela necessidade de realizar uma nova compactação, através da análise do valor da contagem de elementos do agrupamento atual. Este passo está incluído para garantir que todo leiaute final entregue pelo método *Constrói_Compactando* seja resultante imediato da otimização feita pela rotina de compactação.

Este processo explora a possibilidade de que a maximização do aproveitamento durante o posicionamento dos itens mais “difíceis” seja mais vantajosa do que manter a alocação gulosa original e deixar que os itens posicionados por último, supostamente mais “fáceis”, ocupem os espaços vazios deixados pelos itens que já compõem o leiaute.

Entende-se que as heurísticas de ordenação dos itens favoreça a seleção inicial dos itens que tendem a contribuir mais significativamente para o comprimento do leiaute, deixando para o final os itens menores ou que poderão ser incluídos no leiaute sem grandes alterações no comprimento deste. A incerteza da eficiência da simples aplicação da regra de posicionamento e da heurística de ordenação dos itens é tratada pelo algoritmo proposto através da aplicação frequente da compactação dos leiautes parciais. Esta frequência de aplicação permite que os espaços não preenchidos entre os itens sejam acumulados de maneira suficiente para permitir que a rotina de compactação possa fazer os ajustes dos posicionamentos dos itens. E como esta rotina não altera a posição relativa entre os polígonos do leiaute, é parcialmente preservada a ocorrência de espaços não ocupados, que poderão ser alocados para os itens deixados para o final da sequência.

3.1.1 Exemplos de aplicação do algoritmo

A aplicação desta rotina de construção de leiaute é exibida na figura 3.2. Nesta figura, é possível observar o progresso da execução do algoritmo construtivo proposto aplicado à instância *Albano*, usando agrupamento por itens, havendo um intervalo de cinco itens entre as compactações.

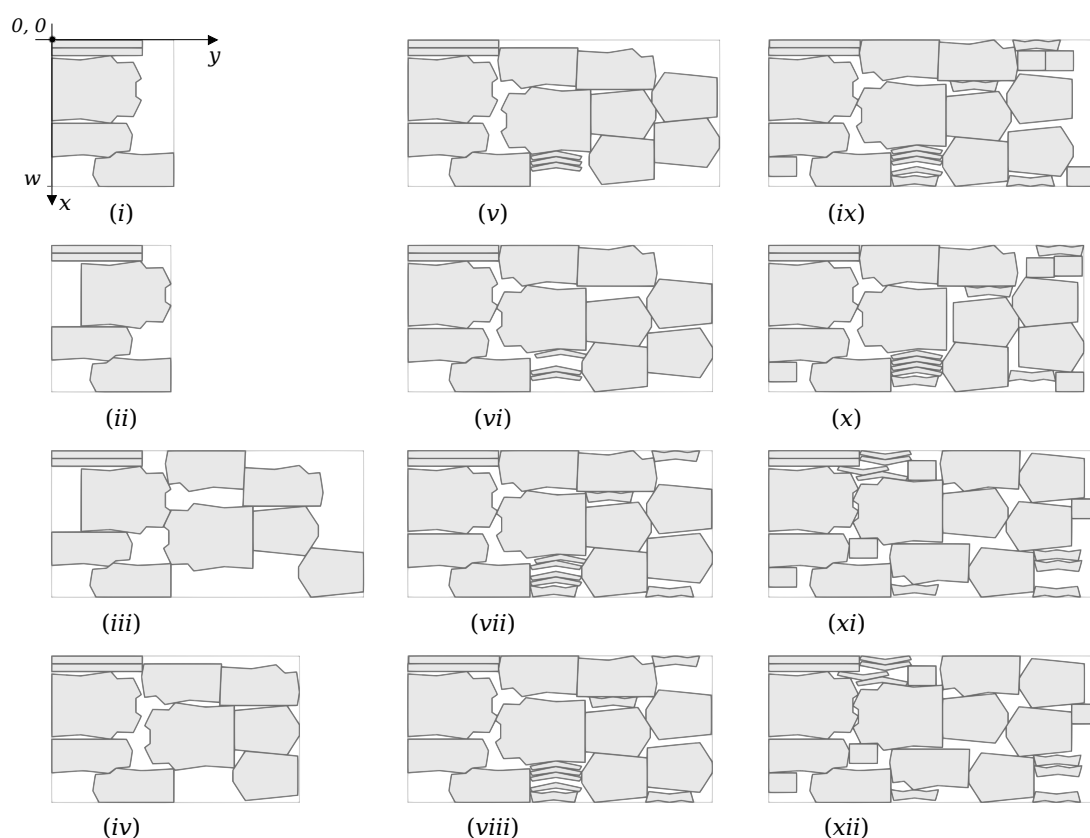


Figura 3.2. Progresso da execução do algoritmo construtivo proposto, aplicado à instância Albano.

Uma análise da figura 3.2 é feita a seguir. Em (i) está o resultado do posicionamento dos cinco primeiros itens selecionados pela heurística de ordenação. Este leiaute parcial é então compactado, (ii), mas sem conseguir reduzir o comprimento. Em (iii), outros cinco itens são posicionados, atingindo o comprimento do leiaute de 10435,9. No próximo passo, (iv), novamente é feita a compactação sobre o leiaute parcial. Desta vez, há uma redução do comprimento para 8292,5, ou 20,5%. Em seguida, (v), outros cinco itens são posicionados, elevando o comprimento do leiaute para 10420,3. Deve-se notar que, neste passo, alguns itens relativamente pequenos ocupam espaços vazios entre itens já posicionados. No passo seguinte, (vi), a compactação resulta numa redução de apenas 2,2% no comprimento do leiaute. Em (vii), outros cinco itens são posicionados, devido às suas dimensões, em espaços vazios deixados pelos posicionamentos anteriores, não havendo alteração no comprimento do leiaute. Conseqüentemente, em (viii), a compactação não consegue ajustar o leiaute de modo que reduza seu comprimento. Em (ix), os últimos quatro itens são posicionados, levando o leiaute ao comprimento de 10910,9. Neste ponto, a fase iterativa do algoritmo é encerrada, restando aplicar novamente a rotina de compactação, já que houve inclusão de itens no leiaute desde a última aplicação

desta rotina. Como resultado, em (x), os ajustes feitos pela última compactação reduzem o comprimento do leiaute em 3.4%. Com isso, o comprimento final do leiaute construído foi de 10541.4. Este resultado é melhor do que o obtido com as mesmas opções de regra e heurísticas, mas sem a compactação, (xi), e com a compactação aplicada apenas uma vez, após posicionar todos os itens, (xii). Nestes casos, os comprimentos dos leiautes foram iguais, com valor 11068.1, pois a compactação não conseguiu fazer ajustes que reduzissem o comprimento, mantendo as posições relativas entre os itens.

A figura 3.3 exibe outro exemplo da aplicação do algoritmo construtivo proposto, desta vez com a instância *Shirts*, usando agrupamento por formas, com intervalo de três formas posicionadas entre as compactações. Seja o conjunto $\mathcal{P}_k \subseteq \mathcal{P}$, tal que $\forall p_i, p_j \in \mathcal{P}_k, p_i \cong p_j$ (p_i é congruente a p_j). Considera-se aqui que posicionar uma forma significa posicionar todos os itens $p_i \in \mathcal{P}_k$, para algum k . A figura está dividida em seis partes, cada uma correspondente a um passo de agrupamento ou a um passo de compactação. Em (i), são posicionados itens de três formas diferentes, que foram as três primeiras selecionadas pela combinação de heurísticas, totalizando 31 itens, e o leiaute parcial tem comprimento 468.8. No passo seguinte, (ii), é feita a compactação, que reduz o comprimento para 466.9, ou apenas 0.4%. Em (iii), outras três formas são posicionadas, totalizando 69 itens no leiaute parcial, que atinge comprimento 632.8. No próximo passo, (iv), a operação de compactação reduz o comprimento em 2.16%, com valor de 619.1. Em (v), as últimas formas são posicionadas, concluindo o posicionamento dos 99 itens, e atingindo o comprimento do leiaute de 632.2. A execução final da operação de compactação ajusta as posições dos itens, fazendo o comprimento ficar em 620.6, caracterizando assim uma redução de 1.8%.

A figura 3.4 exibe o comparativo entre os resultados da aplicação do método construtivo sobre a instância *Shirts*, com as mesmas opções de regra e heurísticas do exemplo que acabou de ser apresentado, mas sem o uso da compactação (i), com uso de compactação apenas após concluir o posicionamento de todos os itens da instância (ii), e o mesmo leiaute produzido no exemplo da figura 3.3 (iii). Os comprimentos desses leiautes são, respectivamente, 661.5, 655.7 e 620.6. Observa-se que o leiaute (ii) tem comprimento 0.9% menor do que o leiaute (i). Já o leiaute (iii) tem comprimento 5.35% menor do que o leiaute (ii), e 6.1% menor do que o leiaute (i).

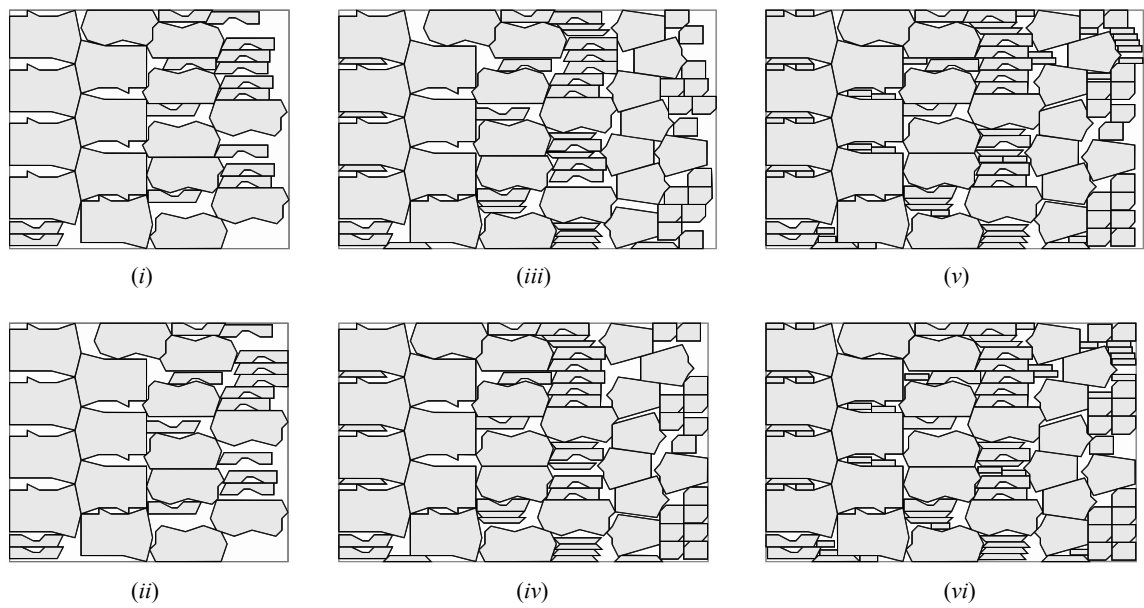


Figura 3.3. Progresso da execução do algoritmo construtivo proposto, aplicado à instância Shirts.

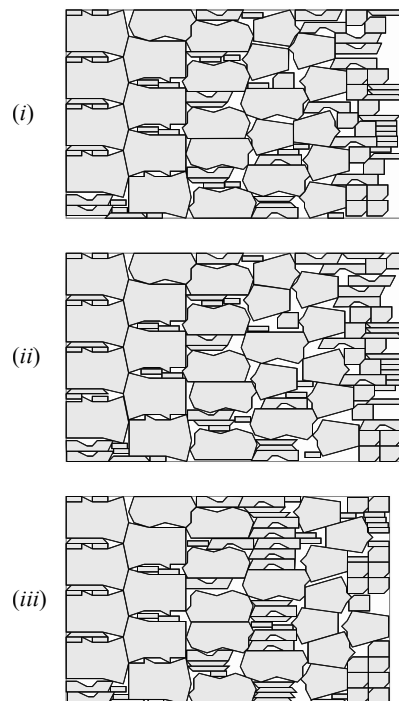


Figura 3.4. Comparação dos resultados obtidos com variações do método construtivo, aplicadas à instância Shirts.

Uma investigação detalhada dos resultados obtidos com o método construtivo pro-

posto é feita nos capítulos seguintes. A seguir, são apresentadas considerações sobre os métodos usados pelo algoritmo proposto para a construção dos leiautes.

3.2 Heurísticas sobre sequências

A qualidade da solução fornecida pelo método construtivo proposto é fortemente afetada pela sequência na qual os itens são posicionados. Na tentativa de que o resultado alcançado pelos posicionamentos leve a um leiaute de comprimento aceitável, duas diferentes perspectivas compõem os critérios de seleção do próximo item a posicionar que foram considerados neste trabalho, e foram baseados naqueles encontrados em Oliveira et al. [2000] e Gomes [2005].

Em uma das perspectivas, a decisão sobre qual item selecionar é feita observando unicamente o conjunto de itens. Isto leva à possibilidade de preparar toda a sequência de itens antes de iniciar os posicionamentos. Aos critérios fundamentados nesta perspectiva dá-se o nome de estáticos. Cada critério experimentado nesta pesquisa observa um atributo específico do polígono que representa um item, ou uma combinação desses atributos. Os atributos observados foram área do polígono, perímetro externo, comprimento, largura, concavidade e não-retangularidade, ordenados com sentido do maior valor para o menor. Os métodos para calcular esses atributos são comentados abaixo. Um critério estático adicional, denominado valor agregado, reproduzido de Gomes [2005], é calculado pela soma das posições obtidas por cada item nas ordenações feitas com todos os atributos observados. A ordenação dos itens pelo valor agregado é feito no sentido do menor valor para o maior.

O cálculo da área de um polígono foi feito através da divisão do mesmo em trapézios, e efetuando o somatório das áreas dos trapézios obtidos. A divisão de um polígono em trapézios é feita pela sequência dos vértices do polígono. Seja p_i um polígono descrito por uma sequência de m vértices $\mathcal{V}_i = (v_{i,0}, v_{i,1}, \dots, v_{i,m})$, tal que $v_{i,j} = (x_{i,j}, y_{i,j})$, $0 \leq j \leq m$. A ordem da sequência \mathcal{V}_i representa um percurso pelos vértices de p_i no sentido anti-horário. Cada par consecutivo de vértices $(v_{i,j-1}, v_{i,j})$ de \mathcal{V}_i gera um trapézio $\mathcal{T}_{i,j} = \{\vec{ab}, \vec{bc}, \vec{cd}, \vec{da}\}$, sendo $a = v_{i,j}$, $b = v_{i,j-1}$, $c = (x_{i,j-1}, 0)$ e $d = (x_{i,j}, 0)$. Sabe-se que a área de um trapézio é dada pela equação 3.1, na qual B e b são as medidas dos lados paralelos, e h é a distância entre estes. Para calcular a área do polígono, deve-se proceder a soma das áreas dos trapézios gerados. Porém, é necessário considerar que a área de um trapézio será positiva, se o mesmo estiver em uma região à esquerda da aresta do polígono que foi usada para o gerar, e negativa, caso contrário. A figura 3.5 ilustra a obtenção dos trapézios e o sinal que a área destes receberá para o somatório. Isto faz com

que as áreas de regiões externas ao polígono, que tenham sido somadas em um trapézio, sejam subtraídas em outros. Sendo assim, a área de cada $\mathcal{T}_{i,j}$ é determinada pela equação 3.2, que usa as coordenadas dos vértices que delimitam uma aresta de p_i , já atribuindo o sinal apropriado para o valor calculado, através do cálculo da distância h entre os lados paralelos do trapézio. Já a área do polígono p_i é o resultado da soma exibida na equação 3.3, onde m é o número de vértices do polígono p_i , e a referência ao vértice $v_{i,m+1}$ deve levar ao primeiro vértice $v_{i,1}$ de \mathcal{V}_i . É possível observar que este método para o cálculo da área de um polígono requer um número de operações $\Theta(m)$, mas pode ser usado em etapa de pré-processamento, pois é desnecessário recalculá-la para um mesmo polígono.

$$A_{\mathcal{T}} = \frac{B + b}{2} \times h \quad (3.1)$$

$$A_{\mathcal{T}_{i,j}} = \frac{y_{i,j-1} + y_{i,j}}{2} \times (x_{i,j-1} - x_{i,j}) \quad (3.2)$$

$$A_i = \sum_{j=2}^{m+1} A_{\mathcal{T}_{i,j}} \quad (3.3)$$

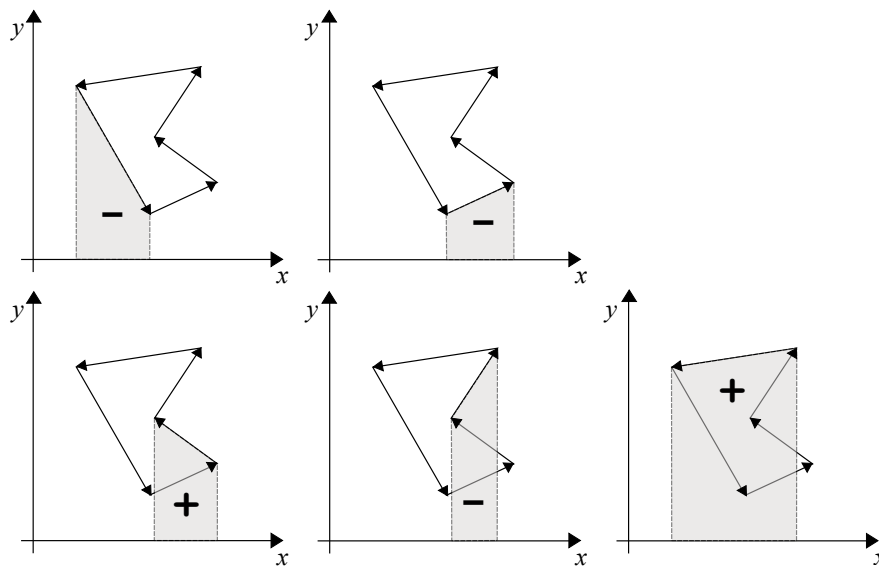


Figura 3.5. Cálculo da área de um polígono com trapézios.

Os três atributos seguintes, perímetro externo, comprimento e largura, são relativamente mais simples de calcular. O perímetro $2P_i$ de um polígono p_i , assim como a área, não é alterado por operações de rotação ou translação do polígono e, portanto, pode ser calculado em etapa de pré-processamento. O cálculo do perímetro é feito como mostra a equação 3.4, através da soma dos comprimentos das arestas de p_i . O comprimento H_i do

polígono p_i , como mostra a equação 3.5, é a distância perpendicular entre o vértice que tenha a maior componente y e o vértice que tenha a menor componente y . A largura W_i do polígono p_i é calculada de maneira análoga ao cálculo do comprimento, como apresentado na equação 3.6, apenas observando a distância sobre o eixo x de coordenadas.

$$2P_i = \left(\sum_{j=2}^m \overline{v_{i,j-1}v_{i,j}} \right) + \overline{v_{i,m}v_{i,1}} \quad (3.4)$$

$$H_i = y_{i,j} - y_{i,k}, \quad y_{i,j} = \max\{y_{i,l} \in \mathcal{V}_i\}, \quad y_{i,k} = \min\{y_{i,l} \in \mathcal{V}_i\} \quad (3.5)$$

$$W_i = x_{i,j} - x_{i,k}, \quad x_{i,j} = \max\{x_{i,l} \in \mathcal{V}_i\}, \quad x_{i,k} = \min\{x_{i,l} \in \mathcal{V}_i\} \quad (3.6)$$

O atributo de concavidade é a diferença de medidas da área do polígono e da área do invólucro convexo deste. O método para a obtenção do invólucro convexo foi definido como uma busca por concavidades em uma cópia do polígono original, e substituindo-as por arestas envoltórias. Para tratar casos de concavidades no interior de concavidades, foi usado um subprocesso iterativo que repete a busca por concavidades, até que o polígono modificado, que representa o invólucro, seja convexo. A opção por este método pouco eficiente para obter o invólucro convexo deve-se à necessidade de identificar as arestas que pertencem a cada concavidade, para usar esta informação durante a formulação do modelo de programação linear de compactação. Já o atributo de não-retangularidade é a diferença entre a área do polígono e a área do seu retângulo envolvente. A área do retângulo envolvente de um polígono p_i pode ser calculada pelo produto $H_i \times W_i$. A figura 3.6 ilustra os atributos de concavidade em (i) e não-retangularidade em (ii), com destaque escurecido para as áreas que definem esses atributos.

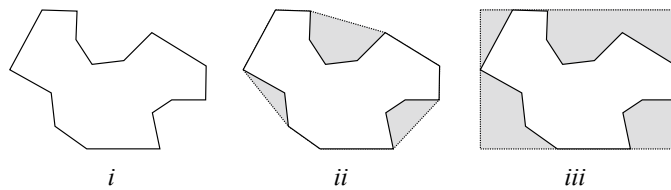


Figura 3.6. Um polígono (i), seu invólucro convexo (ii) e seu retângulo envolvente (iii).

Os atributos que foram descritos permitem definir a sequência de itens a posicionar observando apenas os polígonos que descrevem os itens. Por outro lado, há uma outra perspectiva na qual a decisão sobre a seleção do próximo item é feita observando também o leiaute até então produzido. Ou seja, a cada vez que uma seleção deve ser feita, é realizada uma averiguação para identificar qual é o item ainda não posicionado

que, se for selecionado, levará algum determinado atributo do leiaute, ou combinação de atributos, a um valor considerado melhor do que seria com o posicionamento de outro item naquele momento. Os critérios formados por esta perspectiva recebem o nome de dinâmicos. Neste trabalho, apenas um critério dinâmico foi experimentado. Este avalia, a cada seleção de item, a contribuição para o acréscimo do comprimento do leiaute pelo posicionamento de cada item, em relação ao comprimento do respectivo item. Após avaliar todos os itens, aquele que oferecer a menor contribuição relativa é selecionado.

Se for considerado que, para cada item avaliado, é necessário aplicar a regra de posicionamento para determinar as coordenadas da possível alocação do item, o que seria uma parcela significativa do esforço necessário para posicionar um item, a busca pela melhor opção em um critério dinâmico representa um subprocesso que pode ser caracterizado como não-guloso. Entretanto, se a avaliação do local para receber um item candidato à seleção não for considerada como uma tarefa de posicionamento, então o subprocesso da heurística de seleção do próximo item pode ter preservada a sua caracterização como guloso. A seguir, será analisado o método empregado para o cálculo do invólucro de posicionamento.

3.3 Cálculo do invólucro de posicionamento

O método construtivo proposto neste trabalho faz uso indireto de um procedimento para o cálculo do invólucro de posicionamento entre dois polígonos que explora a aplicação de operações de Minkowski. Para dois polígonos A e B , o polígono $NFP_{A,B}$ pode ser obtido com uso da operação $A \oplus (-B)$, como descrito na seção 2.1.1. Se A e B forem convexos, $NFP_{A,B} \cong A \oplus (-B)$, e o invólucro de posicionamento pode ser obtido com o método exemplificado na figura 2.2. Para os casos de haver concavidade em A ou B , a operação de soma de Minkowski torna-se mais complexa. A implementação desta operação foi baseada no método definido em Bennell & Oliveira [2008], como descrito a seguir.

Primeiramente, é necessário definir o conceito de sentido horário e sentido anti-horário em um percurso pelas arestas de um polígono. Será considerado neste trabalho que todas as arestas que delimitam um polígono são orientadas, e que o interior do polígono está à esquerda das arestas. Sejam duas arestas consecutivas e_{j-1} e e_j , e seja o ângulo $\theta = \sphericalangle e_{j-1}e_j$ formado pela abertura à esquerda das arestas. A aresta e_j está em sentido anti-horário se $\theta < 180^\circ$, e está em sentido horário se $\theta > 180^\circ$.

A rotina para o cálculo do $NFP_{A,B}$, para o caso de haver concavidade em A ou em B , é dividida em quatro etapas. A primeira etapa consiste na análise dos polígonos e organização das arestas. O polígono estacionário A é analisado para identificar inversões

no sentido, de anti-horário para horário ou vice-versa, formado pelo percurso através da sequência de arestas. Isto pode ser feito observando cada três arestas consecutivas, e_{j-1} , e_j e e_{j+1} , na sequência de arestas que forma A , e os respectivos ângulos $\theta_{j-1} = \sphericalangle e_{j-1}e_j$ e $\theta_j = \sphericalangle e_je_{j+1}$, sendo θ_{j-1} e θ_j formados pela abertura à esquerda das arestas. A aresta e_j é marcada como local de inversão se a proposição $(\theta_{j-1} < 180^\circ \wedge \theta_j > 180^\circ) \vee (\theta_{j-1} > 180^\circ \wedge \theta_j < 180^\circ)$ for verdadeira. Por sua vez, o polígono orbital B é analisado para a formação de grupos de arestas consecutivas que compartilham do mesmo sentido de percurso. Isto é feito em dois passos, sendo um para a identificação dos grupos formados por sequências de arestas em sentido anti-horário, denominados grupos convexos, e outro para a identificação dos grupos formados por sequências de arestas em sentido horário, denominados grupos côncavos. O polígono $-B$ será definido pela mesma sequência de arestas de B , porém com a inversão de sentido de cada aresta. As arestas de A e $-B$ são então organizadas em uma coleção \mathcal{E} ordenada pelos ângulos de inclinação. Esta primeira etapa é preparatória para a operação seguinte, que calcula a soma de Minkowski $A \oplus (-B)$.

A operação $A \oplus (-B)$ é feita percorrendo a coleção \mathcal{E} uma vez para cada grupo formado com as arestas de B , para obter sequências parciais. Tais sequências são então unidas pela adição de arestas de A , visando manter a ordem de precedência de A , para produzir o resultado da operação. O percurso realizado sobre \mathcal{E} segue a ordem de precedência de A , o que significa que as arestas de A serão percorridas na sequência que descreve este polígono, e o sentido do caminho em \mathcal{E} será anti-horário, se a próxima aresta de A a ser percorrida tiver este mesmo sentido, ou o sentido do caminho em \mathcal{E} será horário, se a próxima aresta de A tiver este sentido. Esta ordem será invertida, bem como os sentidos das arestas de A , se o grupo de B que estiver sendo processado for côncavo.

Para cada grupo de arestas de B , o procedimento efetuado percorre cada aresta do grupo, e parte para um processo que será repetido uma vez para cada aresta restante do grupo, como descrito a seguir. Partindo da aresta atual do grupo, as arestas de \mathcal{E} são percorridas, e a cada aresta de A que for atravessada, respeitando a precedência deste polígono, será acrescentada à soma em construção. Quando uma aresta do mesmo grupo de B é encontrada, se for a aresta procurada, esta é acrescentada à soma e parte-se para a busca da próxima aresta do grupo. Se não for a aresta procurada, a aresta será adicionada à soma com a orientação atual, se o sentido da última aresta de A que foi atravessada for anti-horário, ou, caso contrário, invertendo a orientação atual da aresta de B .

Após terem sido geradas sequências combinadas de arestas de A e B para cada grupo de B , é necessário reuni-las em uma sequência só. Para isso, são acrescentadas arestas de A visando assegurar a precedência deste polígono desde a última aresta de A de uma sequência, até a primeira aresta de A da sequência seguinte. Se uma sequência

for formada por um grupo côncavo de B , as arestas acrescentadas formam um percurso invertido pelas arestas de A . Quando todas as sequências estiverem reunidas em uma, esta representa a soma de Minkowski entre A e $-B$.

Esta soma é interpretada como um novo polígono, que representa a interação das arestas de A e B . Este novo polígono, entretanto, é um polígono complexo. Ou seja, possivelmente há arestas que participam de ocorrências de ciclos internos. Este polígono complexo deve ser tratado para extrair deste os contornos do $NFP_{A,B}$. Diferentemente do método de Bennell & Oliveira [2008], a rotina empregada neste trabalho não visa encontrar possíveis furos do NFP , por questão de simplicidade e de restrição de tempo para a pesquisa. Sendo assim, basta obter o contorno externo do polígono complexo. Isto pode ser feito pelo seguinte procedimento. Iniciando na aresta que possui a extremidade inicial com a menor coordenada y , é feita uma análise de interseções de outras arestas do polígono complexo com a aresta atual, sendo que essas arestas formam abertura no sentido horário com a aresta atual. Se as interseções se derem apenas na extremidade final da aresta atual, o contorno segue para a aresta que formar a maior abertura no sentido horário com a aresta atual. Se as interseções forem observadas antes da extremidade final da aresta atual, o contorno segue para a aresta que intercepta a atual em um ponto que tenha a menor distância euclidiana para a extremidade inicial da aresta atual, se comparada aos outros pontos de interseção encontrados. Havendo duas ou mais arestas empatadas, o contorno segue para a que forma a maior abertura no sentido horário com a aresta atual. Persistindo empate, o contorno segue para a aresta que tenha a maior distância euclidiana do ponto de interseção até sua extremidade final. O procedimento termina quando o contorno atingir o mesmo vértice que foi usado para selecionar a primeira aresta do contorno.

3.3.1 Exemplos de cálculos de NFP

Para exemplificar, serão considerados dois polígonos exibidos na figura 3.7. Como o polígono A não tem concavidades, não há arestas com inversão de sentido. O polígono B possui duas concavidades, e a análise de suas arestas leva à formação dos seguintes grupos:

1. b_8, b_1 ;
2. $b_2, b_3, b_4, b_5, b_6, b_7$.

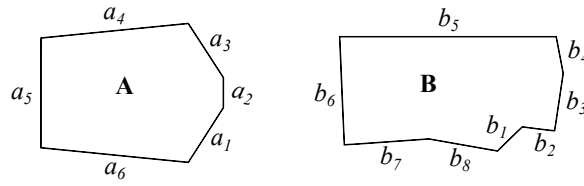


Figura 3.7. Primeiro exemplo: dois polígonos selecionados para calcular o *NFP*, sendo um convexo (*A*) e um côncavo (*B*).

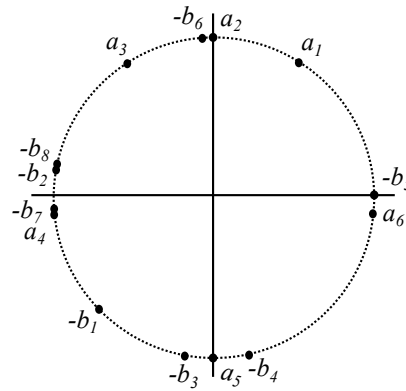


Figura 3.8. Primeiro exemplo: coleção de arestas de *A* e $-B$ ordenadas por inclinação.

A ordenação das arestas de *A* e $-B$ pode ser representada pelo diagrama de inclinações que está na figura 3.8. A geração das sequências de arestas combinadas de *A* e $-B$ para os grupos formados, seguindo a precedência de arestas de *A* tem o resultado apresentado abaixo:

$$1: -b_8, a_4, -b_1;$$

$$2: -b_2, a_4, -b_3, a_5, -b_4, a_6, -b_5, a_1, a_2, -b_6, a_3, -b_7;$$

Para unir as duas sequências em uma só, são acrescentadas arestas de *A*. O resultado desta união é a soma de Minkowski de *A* e $-B$, como apresentado na equação 3.7. Esta soma forma o polígono complexo exibido na figura 3.9. Na mesma figura, à direita do polígono complexo está seu contorno externo, que é o próprio $NFP_{A,B}$.

$$\begin{aligned} A \oplus (-B) = & -b_8 + a_4 - b_1 + a_5 + a_6 + a_1 + a_2 + a_3 - b_2 \\ & + a_4 - b_3 + a_5 - b_4 + a_6 - b_5 + a_1 + a_2 - b_6 \\ & + a_3 - b_7 + a_4 + a_5 + a_6 + a_1 + a_2 + a_3 \end{aligned} \quad (3.7)$$

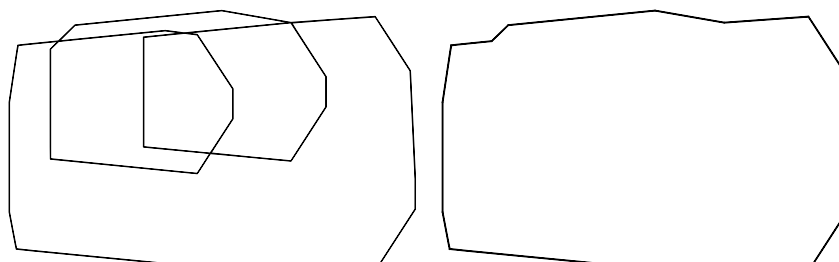


Figura 3.9. Primeiro exemplo: polígono complexo gerado pela soma de Minkowski e o seu contorno externo.

A aplicação do *NFP* pode ser feita como mostra a figura 3.10. Neste exemplo, o *NFP* é posicionado de modo que o seu ponto de referência coincida com o ponto de referência de *A*, sendo ambos os vértices mais abaixo e mais à esquerda dos respectivos polígonos. Para aplicação do *NFP* nesta posição, o ponto de referência de *B* deve ser definido como o vértice mais acima e mais à direita deste polígono. Na mesma ilustração, *B* está posicionado com o seu ponto de referência coincidindo com o ponto de referência de *A*. Todos os pontos pertencentes às arestas do *NFP* podem ser usados para transladar *B*, de forma que este ficará tocando *A*, sem haver sobreposição.

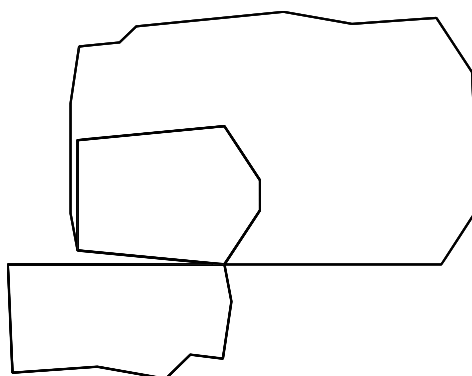


Figura 3.10. Primeiro exemplo: posicionamento do *NFP* e dos polígonos *A* e *B* para aplicação.

Para outro exemplo, serão considerados os polígonos apresentados na figura 3.11. Observa-se que se tratam de dois polígonos côncavos. O polígono *A* possui inversão de sentido nas arestas $a_1, a_2, a_4, a_6, a_8, a_9, a_{10}, a_{11}, a_{13}$ e a_{14} . A formação de grupos de arestas de mesmo sentido em *B* resulta em apenas um grupo convexo, com a sequência de arestas b_6, b_1, b_2, b_3, b_4 e b_5 .

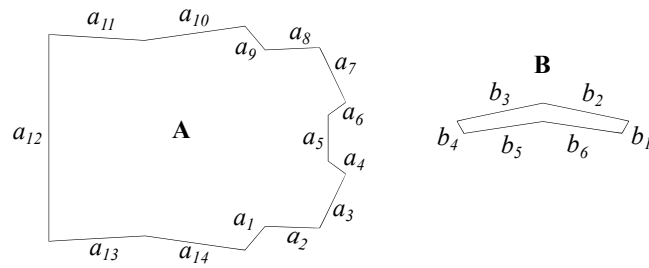


Figura 3.11. Segundo exemplo: dois polígonos côncavos selecionados para calcular o *NFP*.

A disposição das arestas de A e $-B$ no diagrama de inclinações é apresentada na figura 3.12. O percurso pelo diagrama é feito apenas uma vez, já que há somente um grupo formado em B . O número de inversões de sentido em A , porém, contribui para prolongar esse percurso. É perceptível neste exemplo a inclusão da última aresta procurada de B na sequência a cada vez que o percurso passa por ela. Quando a aresta de B é atravessada no sentido anti-horário pelo diagrama de inclinações, o sinal da aresta é negativo, o que preserva a inversão deste polígono. O sinal a aresta de B é positivo quando esta é atravessada no sentido horário. A sequência de arestas combinadas de A e $-B$ para o percurso é a seguinte:

- 1: $-b_6, a_8, b_6, a_9, -b_6, a_{10}, a_{11}, -b_1, a_{12}, -b_2, a_{13}, a_{14}, -b_3, a_1, b_3, a_2, -b_3, a_3, -b_4, a_4, b_4, a_5, a_6, a_7, -b_4, a_8, a_9, a_{10}, a_{11}, -b_5$.

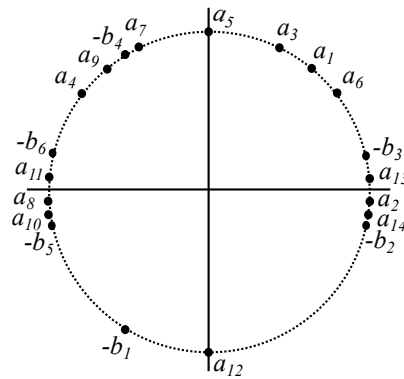


Figura 3.12. Segundo exemplo: coleção de arestas de A e $-B$ ordenadas por inclinação.

Havendo apenas uma sequência de arestas combinadas de A e $-B$, é necessário verificar a necessidade de acrescentar arestas de A nesta sequência para preservar a precedência das arestas deste polígono. Como a última aresta de A da sequência é a_{11} e a

primeira é a_8 , é preciso incluir arestas para concluir o caminho entre as duas. A operação $A \oplus (-B)$ para este exemplo fica como mostra a equação 3.8. A figura 3.13 exibe o polígono complexo formado pela sequência de arestas da soma de Minkowski efetuada. Na mesma figura, à direita deste polígono está o seu contorno externo, que representa o $NFP_{A,B}$.

$$\begin{aligned}
 A \oplus (-B) = & -b_6 + a_8 + b_6 + a_9 - b_6 + a_{10} + a_{11} - b_1 + a_{12} - b_2 & (3.8) \\
 & + a_{13} + a_{14} - b_3 + a_1 + b_3 + a_2 - b_3 + a_3 - b_4 + a_4 \\
 & + b_4 + a_5 + a_6 + a_7 - b_4 + a_8 + a_9 + a_{10} + a_{11} - b_5 \\
 & + a_{12} + a_{13} + a_{14} + a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7
 \end{aligned}$$

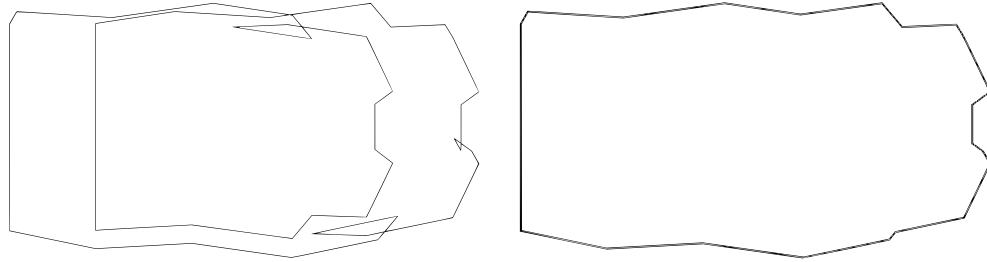


Figura 3.13. Segundo exemplo: polígono complexo gerado pela soma de Minkowski e o seu contorno externo.

A aplicação do polígono $NFP_{A,B}$ requer que este seja posicionado considerando os pontos de referência de A e de B . A figura 3.14 apresenta uma possível ocorrência de posicionamentos relativos desses três polígonos, considerando que o ponto de referência de A seja o seu vértice mais abaixo e mais à esquerda, assim como o ponto de referência do $NFP_{A,B}$ em relação ao próprio. E considera-se que o ponto de referência de B seja o seu vértice mais acima e mais à direita. Neste caso, o invólucro de posicionamento deve ser posicionado com seu ponto de referência coincidindo com o ponto de referência de A . E B deve ser posicionado com seu ponto de referência também coincidindo com o ponto de referência de A .

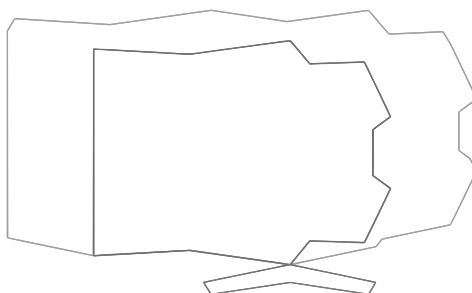


Figura 3.14. Segundo exemplo: posicionamento do *NFP* e dos polígonos *A* e *B* para aplicação.

Resta analisar a subrotina usada para definir o ponto de alocação para um item no leiaute, e a subrotina para compactar os leiautes parciais. Isto é feito nas seções seguintes.

3.4 Determinação da posição do próximo item

Duas heurísticas foram consideradas para o método construtivo proposto, no que tange a determinação da posição de um item no leiaute. Uma delas é denominada *bottom-left*, e foi reproduzida do procedimento descrito em Gomes [2005]. A segunda foi desenvolvida com base na primeira, mas acrescentando características inspiradas no método TOPOS, apresentado em Oliveira et al. [2000], e foi denominada *middle-left*. Ambas utilizam a mesma combinação dos elementos *NFP* e *IFR*.

A abordagem *bottom-left* está descrita na seção 2.1.3.1. A ideia simples que caracteriza este procedimento é a identificação do ponto mais abaixo e mais à esquerda que possa receber o item que está sendo posicionado, sem que se incorra em sobreposições de itens. Computacionalmente, a preocupação relacionada com esta rotina é reduzir o número de pontos candidatos, inicialmente em quantidade infinita, para um valor que permita a aplicabilidade em situações reais. Como informado anteriormente, a redução do número de pontos candidatos a um conjunto discreto e finito formado pelos vértices dos *NFPs* e do *IFR*, e por qualquer interseção entre arestas dos *NFPs* entre si ou entre essas e as arestas do *IFR*. Adicionalmente, deve-se considerar possíveis rotações do polígono que representa o item a posicionar, o que gera mais pontos candidatos, e agrega aos pontos o ângulo usado para rotação. Uma simples ordenação deste conjunto de pontos pela distância para a base do objeto receptor, e resolvendo empates em favor do ponto mais próximo da lateral esquerda do mesmo objeto, fornece o ponto para o posicionamento, e o ângulo associado, como sendo o primeiro do conjunto de pontos candidatos.

A variante *middle-left* proposta é semelhante à abordagem *bottom-left*, mas com duas diferenças fundamentais. Primeiro, são aceitos posicionamentos com coordenadas y

negativas, possibilitando que a acomodação de itens proporcione o crescimento do leiaute em dois sentidos. Outra diferença é o critério primário de seleção do ponto para posicionamento, que visa alocar os itens em posições mais próximas do centro do leiaute, que é definido neste trabalho como uma reta paralela à base do objeto receptor, na coordenada $y = 0$. Ao terminar de posicionar todos os itens, é feito um deslocamento vertical do leiaute completo para levá-lo ao interior do objeto receptor, que está definido no quadrante positivo do seu sistema de coordenadas.

Para a abordagem *middle-left*, foram ainda consideradas três diferentes possibilidades de referências para avaliar os pontos candidatos para o posicionamento, que podem ser úteis quando são permitidas rotações dos itens. A primeira é a mesma já aplicada à abordagem *bottom-left*, e foi denominado melhor ponto de referência, identificada pelo termo *BEST_REF*. Esta primeira opção consiste simplesmente em avaliar cada ponto candidato, independente de características do polígono. A segunda opção é observar o ponto central do polígono após o posicionamento, sendo este ponto definido através do retângulo envolvente do item. Tomando este retângulo envolvente formado pelos pontos $ABCD$, o ponto central é a interseção entre os segmentos de retas \overline{AC} e \overline{BD} . Nos casos em que são permitidas algumas rotações nos itens, é possível que o ponto central fique mais abaixo para uma dada rotação. Isto pode levar ao maior aproveitamento do espaço entre os itens posicionados. Portanto, esta segunda opção, identificada pelo termo *BEST_CENT*, consiste em avaliar cada ponto candidato, buscando aquele que leve ao ponto central do item mais próximo da base do objeto receptor. A terceira opção considerada trata de observar o valor da maior distância entre um vértice do item posicionado à base do objeto receptor. Esta opção é identificada pelo termo *MIN_LEN*, e consiste em comparar os pontos candidatos para selecionar o que leva o item posicionado a ter o valor mínimo para a maior distância de um vértice deste item até a base do objeto receptor.

Para organizar os dados necessários para esta rotina, foram usadas estruturas matriciais indexadoras de *NFPs* e de combinações destes em um leiaute. Para definir essas matrizes, é necessário estabelecer elementos do problema que são referidos. Sejam os seguintes elementos:

1. \mathcal{F} é o conjunto dos diferentes polígonos que representam as formas dos itens a posicionar;
2. $|\mathcal{F}| = n_f$, ou seja, n_f é o número de elementos de \mathcal{F} ;
3. $f_i \in \mathcal{F}$, é um polígono que representa a forma de um ou mais itens a posicionar;
4. \mathcal{P} é o conjunto de itens a posicionar;

5. $|\mathcal{P}| = n_p$, ou seja, n_p é o número de elementos de \mathcal{P} ;
6. $p_j \in \mathcal{P}$ é um item a ser posicionado, descrito por um polígono, sendo $p_j \cong f_i$ (relação de congruência), $f_i \in \mathcal{F}$, $1 \leq j \leq n_p$;
7. $R_i = (\theta_{i,1}, \dots, \theta_{i,|R_i|})$, $1 \leq i \leq n$, é a sequência de ângulos admissíveis para serem usados em eventuais rotações de itens representados pela forma poligonal f_i ;
8. $r = \max\{|R_h|, \forall h \in [1, n]\}$ é a maior quantidade de ângulos admissíveis para uma forma, na instância atual;
9. f_i^k , $1 \leq i \leq n$, $1 \leq k \leq |R_i|$, é o resultado da aplicação de uma rotação em f_i usando o k -ésimo ângulo de R_i ;
10. $NFP_{i,j}$ é o NFP calculado para o polígono estacionário f_i e o polígono orbital f_j ;
11. $P[a]$, representa o resultado da translação do polígono P , pelo seu ponto de referência, para o ponto a do sistema de coordenadas do objeto receptor;
12. \mathcal{L} é o conjunto formado pelos itens já posicionados no leiaute em construção;
13. $s_j \in \mathcal{L}$ representa um item posicionado no leiaute em construção, $s_j = p_j^l[v]$, sendo $p_j \in \mathcal{P}$, $p_j \cong f_i$, $1 \leq l \leq |R_i|$, $v = (x_j, y_j)$, onde (x_j, y_j) são as coordenadas e l é a posição do ângulo de rotação em R_j , definidos pela regra de posicionamento para alocação de p_j .

Com base nesses elementos, pode-se definir a matriz \mathcal{A} de invólucros de posicionamentos calculados, conforme equação 3.9. Cada elemento de \mathcal{A} é definido pela equação 3.10. Já a matriz \mathcal{B} de combinações de invólucros de posicionamentos é definida pela equação 3.11. E cada item da matriz \mathcal{B} está definido pela equação 3.12.

$$\mathcal{A}_{n,r,n,r} = (t_{i,k,j,l}), \quad 1 \leq i \leq n, 1 \leq j \leq n \quad (3.9)$$

$$, 1 \leq k \leq |R_i|, 1 \leq l \leq |R_j|$$

$$t_{i,k,j,l} = NFP_{A,B}, \quad A = f_i^k, B = f_j^l \quad (3.10)$$

$$\mathcal{B}_{n,r} = (u_{i,k}), \quad r = |R_i| \quad (3.11)$$

$$u_{i,k} = \{NFP_{A,B} | A = s_j, B = f_i^k, \forall s_j \in \mathcal{L}\} \quad (3.12)$$

Para identificar os pontos candidatos ao posicionamento de um item p_i , o método da regra de posicionamento consulta, na matriz \mathcal{B} , os $NFPs$ armazenados no item $u_{i,k}$,

para cada $k \in [1, |R_i|]$. Também precisa ser consultado o IFR_i , que pode ser descrito sem informações de itens já posicionados no leiaute em construção.

Para a montagem do IFR_i , há diferenças entre o método *bottom-left* e *middle-left*, pois este último permite posicionamentos em pontos com coordenada $y < 0$. Assim, serão definidas duas estruturas para IFR , sendo o IFR_i^{BL} , para aplicação da abordagem *bottom-left*, formado por arestas identificadas por apenas duas extremidades conhecidas, que estabelecem as posições mínimas no canto inferior esquerdo e no canto inferior direito do objeto receptor, de modo que o item p_i tenha todos os seus vértices no interior deste objeto. Esta definição de IFR_i^{BL} está representada pela equação 3.13, sendo τ um valor suficientemente grande. Já para *middle-left*, o IFR_i^{ML} é definido com duas arestas paralelas formadas por coordenadas que representam o menor afastamento aceito da extremidade esquerda do objeto receptor, e o menor afastamento aceito da extremidade direita do mesmo objeto, de modo que o item p_i tenha todos os seus vértices no interior deste objeto. A equação 3.14 apresenta esta definição do IFR_i^{ML} , sendo os valores suficientemente grandes τ , positivo, e β , negativo.

$$\begin{aligned}
 IFR_i^{BL} &= (\overrightarrow{AB}, \overrightarrow{BC}, \overrightarrow{CD}, \overrightarrow{DA}), & A &= (left_i, bottom_i), & (3.13) \\
 & & B &= (right_i, bottom_i), \\
 & & C &= (right_i, \tau), \\
 & & D &= (left_i, \tau) \\
 IFR_i^{ML} &= (\overrightarrow{FG}, \overrightarrow{GH}, \overrightarrow{HI}, \overrightarrow{IF}), & F &= (left_i, \beta), & (3.14) \\
 & & G &= (right_i, \beta), \\
 & & H &= (right_i, \tau), \\
 & & I &= (left_i, \tau)
 \end{aligned}$$

Assim, estão definidos os elementos que permitem obter candidatos para a determinação de local e inclinação para cada item. Após a rotação do item pelo ângulo indicado e sua translação para a posição determinada, a estrutura que armazena as combinações de *NFPs* deve ser atualizada. Sendo o item recém posicionado $s_j \in \mathcal{L}$, l a posição do ângulo usado para sua rotação em R_j , v_j o ponto determinado pela regra para sua translação, então $s_j = p_j^l[v_j]$, e a atualização é feita executando a operação descrita pela equação 3.15, para cada elemento $u_{i,k}$ da matriz \mathcal{B} .

$$u_{i,k} = u_{i,k} \cup \{NFP_{A,B}[v_j] \mid A = s_j, B = f_i^k\} \quad (3.15)$$

A regra será aplicada para cada item a posicionar, até que todos os itens estejam alocados no leiaute. Resta, portanto, analisar o subprocesso de compactação de leiaute que foi empregado no método proposto, o que é feito a seguir.

3.5 Compactação de leiaute

A rotina de compactação de leiaute utilizada no método construtivo proposto é baseada nos modelos matemáticos de compactação que podem ser encontrados em Gomes [2005] e Gomes & Oliveira [2006]. Um formato geral destes modelos matemáticos está na figura 2.10. A aplicação deste processo de compactação requer a formulação do modelo matemático de programação linear, nos moldes do formato geral apresentado, de acordo com o estado atual do leiaute em construção. Isto envolve as coordenadas de posicionamentos de cada item $p_i \in \mathcal{L}$, seus *IFRs*, e as combinações de *NFPs* retidas na matriz \mathcal{B} definida em 3.11. A solução para o modelo é uma combinação de variáveis que representam coordenadas para os itens posicionados, de modo que, se aplicadas para o reposicionamento destes, levariam o comprimento do leiaute ao valor indicado na variável z , que é a própria função objetivo. Se este comprimento z for menor do que o comprimento atual do leiaute, são feitos os reposicionamentos dos itens de acordo com as coordenadas obtidas, e, em seguida, o processo de compactação é repetido do início. Se z não for menor do que o comprimento atual, a compactação é encerrada.

Para solucionar os modelos de programação linear formulados durante a execução do método, foi utilizado o programa *LPSolve*. Este programa possui código aberto, que pode ser obtido em Berkelaar et al. [2010]. Também está disponível uma coleção de interfaces para programação de aplicações, ou APIs (do termo em inglês *Application Programming Interface*), que permitem programar o uso do *LPSolve* através de comandos incluídos no código-fonte de outros programas, como o que foi implementado para realização dos experimentos com o método proposto.

A formulação do modelo divide-se em seis passos, que são descritos a seguir:

- 1. Declaração de variáveis** Deve ser declarado um par de variáveis contínuas reais (x_i, y_i) para cada item $p_i \in \mathcal{L}$ e uma variável auxiliar contínua $z \in \mathbb{R}$. Os valores das variáveis x_i e y_i representam as coordenadas do local para onde o item p_i deve ser transladado para reduzir o comprimento do leiaute. A variável z é usada para representar o próprio valor do comprimento do leiaute. Se a regra de posicionamento for *middle-left*, deve ser acrescentada uma segunda variável auxiliar contínua, m , sendo que neste caso z passa a representar o maior afastamento da linha central para os vértice de todos os itens posicionados acima da linha central,

e $m \in \mathbb{R}$ é análogo a z , porém observando os itens posicionados abaixo da linha central.

2. Função objetivo Como o objetivo do modelo é minimizar o comprimento do leiaute, a função é minimizar z . Quando se estiver usando a regra de posicionamento *middle-left*, a função objetivo deve ser minimizar o valor de $z - m$.

3. Restrições para minimização do comprimento do leiaute Uma vez que o comprimento do leiaute deve estar representado pela variável z , é necessário acrescentar restrições que associem o valor desta ao comprimento. Para isso, deve-se incluir uma restrição como a da inequação 3.16 para cada item p_i posicionado. Se a regra de posicionamento for *middle-left*, deve ser acrescentada ao modelo uma restrição como a da inequação 3.17 para cada item p_i posicionado. Os valores top_i e $bottom_i$ estão definidos nas equações 2.8 e 2.9, e ilustrados na figura 2.3.

$$-y_i + z \geq top_i \quad (3.16)$$

$$-y_i + m \leq bottom_i \quad (3.17)$$

4. Restrições para respeitar os limites do objeto receptor É preciso manter todos os vértices de cada item no interior do objeto receptor. Portanto, são incluídas restrições para as variáveis x_i e y_i , para cada $p_i \in \mathcal{L}$, visando assegurar que, ao transladar p_i , movendo o seu ponto de referência para o ponto (x_i, y_i) , nenhum vértice deixe de respeitar essa condição. Devem ser incluídas no modelo de compactação restrições formadas pelas inequações 3.18, 3.19 e 3.20 para cada item posicionado, quando a regra de posicionamento for *bottom-left*. Quando esta regra for *middle-left*, é necessário apenas incluir as inequações 3.18 e 3.19 para cada item posicionado. Neste caso, não há necessidade de restringir o valor de y_i , que pode assumir também valores negativos. O valor w representa a largura do objeto receptor, e $left_i$, $right_i$ e $bottom_i$ estão definidos nas equações 2.7, 2.6 e 2.9, respectivamente, e ilustrados na figura 2.3.

$$x_i \geq left_i \quad (3.18)$$

$$x_i \leq w - right_i \quad (3.19)$$

$$y_i \geq bottom_i \quad (3.20)$$

5. Restrições para limitar a distância dos movimentos A limitação imposta à distância dos movimentos visa melhorar a eficiência da compactação, que deve também ser caracterizada por um processo iterativo, segundo Gomes & Oliveira [2006] e comentado na seção 2.1.4. Para impor esta limitação, são acrescentadas ao modelo restrições formadas pelas inequações 3.21, 3.22, 3.23 e 3.24, sendo um conjunto destas para cada item p_i posicionado. A decomposição de cada constante $DISTANCE_i$ em duas constantes, DX_i e DY_i , segue o modelo descrito em Gomes [2005]. As constantes DX_i e DY_i são definidas pelas equações 3.25 e 3.26, sendo W_i a largura do item p_i e H_i seu comprimento. As constantes X_i^* e Y_i^* retêm, respectivamente, as componentes x e y da posição inicial do ponto de referência do item p_i .

$$x_i \geq X_i^* - DX_i \quad (3.21)$$

$$x_i \leq X_i^* + DX_i \quad (3.22)$$

$$y_i \geq Y_i^* - DY_i \quad (3.23)$$

$$y_i \leq Y_i^* + DY_i \quad (3.24)$$

$$DX_i = W_i/2 \quad (3.25)$$

$$DY_i = H_i/2 \quad (3.26)$$

6. Restrições para garantir a não-sobreposição A condição de não-sobreposição está associada, nesta abordagem, ao invólucro de posicionamento entre dois itens. Se todas as arestas do invólucro de posicionamento fossem usadas para garantir a não sobreposição entre os itens, seria necessário incluir no modelo várias disjunções de inequações, que seriam tratadas com variáveis binárias. Para evitar isso, optando pela simplicidade, o modelo de compactação adotado envolve o uso de uma heurística de localidade para seleção de uma aresta do *NFP* entre cada par de itens, ou uma seleção de arestas pertencentes à mesma concavidade. Na implementação feita para os experimentos do método proposto, a heurística de seleção da aresta do *NFP* $p_i, p_j [(x_i, y_i)]$ busca uma aresta deste que intercepte o segmento de reta formado entre os pontos de referência de p_i e p_j . Caso seja encontrado um número de arestas maior do que um, é selecionada a que interceptar o segmento de reta em um ponto que esteja mais próximo do ponto de referência de p_j , em comparação com as demais.

A figura 3.15 apresenta três exemplos de seleção de aresta do *NFP* para inclusão de

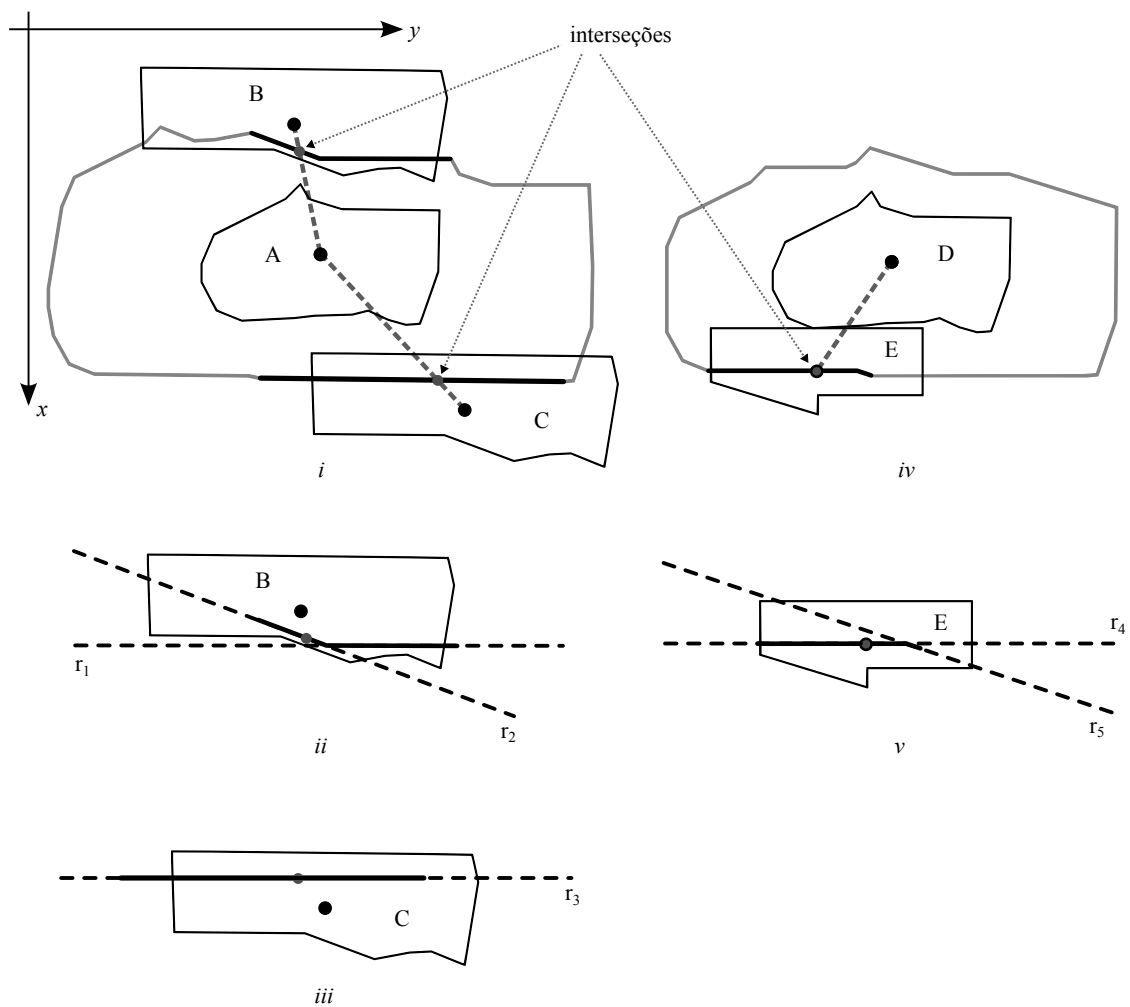


Figura 3.15. Exemplos da seleção de arestas do *NFP*.

restrições de não-sobreposição entre itens no modelo de compactação. Nesta figura, os pontos centrais dos retângulos envolventes das formas são usados como referências dos itens. Em (i), é aplicada a heurística de localidade para definir a aresta do $NFP_{A,B}$ que será usada para garantir que B não sobreponha A , e também é aplicada esta heurística para definir a aresta do $NFP_{A,C}$, para que C não sobreponha A . As arestas selecionadas são as que interceptam o segmento de reta entre os pontos de referência de B e de A , para B não sobrepor A , e que interceptam o segmento de reta entre os pontos de referência de C e de A , para C não sobrepor A . Em (ii), são destacadas as arestas selecionadas do $NFP_{A,B}$ e suas respectivas retas suportes. Deve-se observar que a aresta selecionada deste invólucro de posicionamento pertence a uma concavidade, e a outra aresta que compõe esta concavidade também precisa ser respeitada. Em (iv), é aplicada a mesma heurística de localidade para selecionar a aresta do $NFP_{D,E}$ que será usada para a condição de não-sobreposição

de E a D . Como a aresta selecionada está em uma concavidade, a outra aresta que compõe esta concavidade também é selecionada. Em (ii), (iii) e (v), são exibidas as retas suporte de cada aresta de NFP selecionada para garantir a não-sobreposição de B em A , de C em A e de E em D . As equações de 3.27 a 3.29 descrevem as seleções comentadas, e os conjuntos definidos nas equações de 3.30 a 3.32 representam os conjuntos de arestas selecionadas para cada caso, incluindo as arestas adicionais de casos de concavidades.

$$\overline{BA} \cap NFP_{A,B} = I_1 \quad , I_1 \in \overline{v_7v_8} \in NFP_{A,B} \quad (3.27)$$

$$\overline{CA} \cap NFP_{A,C} = I_2 \quad , I_2 \in \overline{v_{21}v_{22}} \in NFP_{A,C} \quad (3.28)$$

$$\overline{ED} \cap NFP_{D,E} = I_3 \quad , I_3 \in \overline{v_{15}v_{16}} \in NFP_{D,E} \quad (3.29)$$

$$E_{B,A} = \{\overline{v_6v_7}, \overline{v_7v_8}\} \quad (3.30)$$

$$E_{C,A} = \{\overline{v_{21}v_{22}}\} \quad (3.31)$$

$$E_{E,D} = \{\overline{v_{15}v_{16}}, \overline{v_{16}v_{17}}\} \quad (3.32)$$

As equações de 3.33 a 3.41 apresentam a obtenção dos coeficientes angulares e lineares para cada reta suporte do exemplo da figura 3.15. Nota-se que, como as retas r_1 , r_3 e r_4 são paralelas ao eixo y , não existem tais coeficientes para estas retas.

$$r_1 : \quad a_1 = \frac{y_7 - y_6}{x_7 - x_6} \quad (3.33)$$

$$x_6 = x_7 \Rightarrow \nexists a_1, \nexists b_1 \quad (3.34)$$

$$r_2 : \quad a_2 = \frac{y_8 - y_7}{x_8 - x_7} \quad (3.35)$$

$$b_2 = y_8 - a_2x_8 \quad (3.36)$$

$$r_3 : \quad a_3 = \frac{y_{22} - y_{21}}{x_{22} - x_{21}} \quad (3.37)$$

$$x_{21} = x_{22} \Rightarrow \nexists a_3, \nexists b_3 \quad (3.38)$$

$$r_4 : \quad a_4 = \frac{y_{16} - y_{15}}{x_{16} - x_{15}} \quad (3.39)$$

$$x_{15} = x_{16} \Rightarrow \nexists a_4, \nexists b_4 \quad (3.40)$$

$$r_5 : \quad a_5 = \frac{y_{17} - y_{16}}{x_{17} - x_{16}} \quad (3.41)$$

$$b_5 = y_{17} - a_5 x_{17} \quad (3.42)$$

Deve ser incluída no modelo de compactação uma restrição para cada reta suporte de aresta selecionada. Cada restrição deve assegurar que o ponto de referência de um item F não seja levado a uma região interna do $NFP_{G,F}$, para algum item G posicionado. As inequações de 3.43 a 3.47 representam as restrições que devem ser incluídas no modelo de compactação para a situação exibida no exemplo da figura 3.15.

$$R_{B,A}(r_1) : \quad x_B \leq x_7 \quad (3.43)$$

$$R_{B,A}(r_2) : \quad -a_2 x_B + a_2 x_A + y_B - y_A \geq b_2 \quad (3.44)$$

$$R_{C,A}(r_3) : \quad x_C \geq x_{21} \quad (3.45)$$

$$R_{E,D}(r_4) : \quad x_E \leq x_{16} \quad (3.46)$$

$$R_{E,D}(r_5) : \quad -a_5 x_E + a_5 x_D + y_E - y_D \leq b_5 \quad (3.47)$$

Estando concluída a formulação do modelo de programação linear de compactação, parte-se para a resolução do modelo com o programa de computador especializado. De posse da solução para o modelo formulado, deve-se proceder ao ajuste das posições dos itens do leiaute em construção e, posteriormente, reiniciar o processo de formulação do modelo em outra iteração.

Para exemplificar a aplicação da rotina de compactação, a figura 3.16 apresenta a sequência de estados de um leiaute durante as iterações do processo. Em (i) está o resultado da aplicação da regra de posicionamento para incluir sete itens no leiaute. O resultado do modelo de compactação indica novas posições para alguns itens, como exibido em (ii). O estado mostrado em (iii) é o resultado da primeira iteração de compactação. Uma nova formulação do modelo leva a novas indicações de reposicionamentos para alguns itens, como indicado em (iv). Isto leva ao estado exibido em (v). Mais uma formulação do modelo de compactação permite determinar novos reposicionamentos, como pode ser observado em (vi). Com isto, o leiaute chega ao estado mostrado em (vii), cuja aplicação

da compactação não leva a novas mudanças.

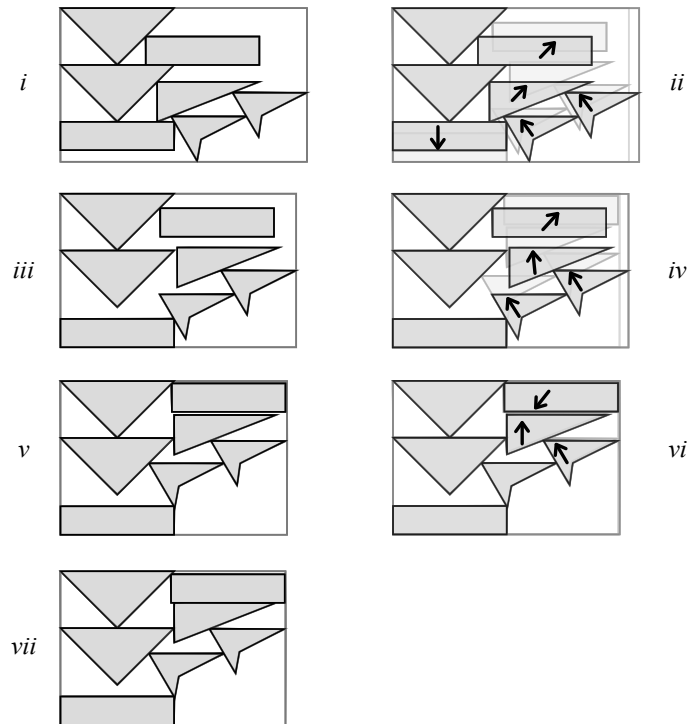


Figura 3.16. Exemplo da aplicação do procedimento iterativo de compactação.

3.6 Considerações sobre o capítulo

Neste capítulo foi apresentado o método construtivo para o problema de *C&P* que é proposto neste trabalho. As sub-rotinas que são usadas por este método foram detalhadamente explicadas. É importante ressaltar que este método pode ser facilmente modificado, com a adoção de alguma regra de posicionamento diferente das que foram expostas, ou com a aplicação de novas heurísticas para ordenação da sequência de itens a posicionar, ou mesmo com a formação de grupos com algum critério diferente. Além disso, também é possível incorporar este método em algoritmos de busca, como uma subrotina responsável por montar os leiautes. Os resultados dos experimentos que foram realizados para observar a aplicação do método construtivo proposto são apresentados no próximo capítulo.

Capítulo 4

Experimentos computacionais e discussão dos resultados

Os experimentos computacionais realizados com o algoritmo construtivo proposto para o problema de *C&P* são relatados neste capítulo. A intenção desses experimentos foi verificar a capacidade que o algoritmo proposto tem de oferecer soluções de qualidade aceitável para o problema abordado. Para fazer esta verificação, foram realizados testes de execução que exploraram várias combinações de parâmetros de configuração do comportamento do método construtivo.

Este capítulo inclui o detalhamento dos experimentos. Inicialmente, são caracterizadas as instâncias usadas para os testes. Em seguida, as combinações de parâmetros são expostas, revelando a cobertura dos testes. Após isto, estão os dados dos resultados obtidos, com a análise da influência dos parâmetros sobre a qualidade das soluções obtidas. Ao final, são apresentados exames da interrelação dos parâmetros e da influência das combinações feitas sobre as instâncias utilizadas.

4.1 Instâncias selecionadas

Neste trabalho, optou-se por adotar o conjunto de instâncias de referência reunido por Hopper [2000]. Seguindo o argumento dessa autora e tendo observado a recorrente adoção dessas instâncias por outros pesquisadores em seus trabalhos, acrescentando-se a recomendação destas por grupos de interesse como o já citado ESICUP, o uso dessas instâncias fornece um indicador de desempenho que possibilita fazer comparações entre diferentes abordagens de resolução para o problema em estudo. Porém, a limitação imposta pela ausência do tratamento de ciclos internos no polígono complexo gerado pela soma de Minkowski, parte do método de geração do *NFP* comentado na seção 3.3, causou

falhas durante a resolução das instâncias *jakobs1* e *jakobs2*. Nestas, ocorre a possibilidade de encaixes de arestas côncavas de um item com arestas convexas de outro item. Porém, esses encaixes ficam em ciclos internos que são desprezados pela implementação do cálculo do NFP em uso neste trabalho. Conseqüentemente, os resultados obtidos para essas instâncias foi evidentemente prejudicado. Como era um problema pontual esperado, já que o método da fonte consultada não foi completamente reproduzido, optou-se por desconsiderar essas instâncias. Assim, foram usadas as outras 14 instâncias apresentadas na tabela 2.1.

Foram identificados atributos que caracterizam as instâncias de teste, com objetivo de fazer possíveis relações entre estes, os parâmetros do algoritmo e os resultados. Uma parte desses atributos estão na tabela 4.1. Em cada linha desta tabela estão dados de uma instância, que são, da esquerda para a direita, o nome da instância, a largura do objeto receptor, o número de itens a posicionar, o número de diferentes formas dos itens, o total de vértices somados de todas as diferentes formas, o número médio de vértices por forma, o total de vértices somados de todos os itens, o número médio de vértices por item, os ângulos admissíveis para rotacionar os itens, a taxa média de concavidade dos itens e a taxa média de não-retangularidade dos itens. É possível observar que há diferenças significativas entre as instâncias com relação aos atributos coletados, o que é favorável à cobertura dos experimentos.

Tabela 4.1. Atributos das instâncias usadas para testes.

Instância	w	P	F	Número de vértices				Ângulos admissíveis (graus)	Outros atributos das formas (médias %)	
				Formas		Itens			Concav	N-ret
				Σv	\bar{v}	Σv	\bar{v}			
albano	4.900	24	8	58	7,25	164	6,83	0;180	5,16	13,95
blaz2	15	16	4	30	7,50	120	7,50	0;180	16,30	25,66
dagli	60	30	10	62	6,20	186	6,20	0;180	2,03	26,61
dighe1	100	16	16	62	3,88	62	3,88	0	3,38	49,07
dighe2	100	10	10	47	4,70	47	4,70	0	1,65	32,97
fu	38	12	12	43	3,58	43	3,58	0;90;180	0,00	25,26
mao	2.550	20	9	83	9,22	174	8,70	0;90;180	9,14	26,56
marques	104	24	8	57	7,13	162	6,75	0;90;180	6,36	12,99
shapes0	40	43	4	35	8,75	391	9,09	0	27,93	49,37
shapes1	40	43	4	35	8,75	391	9,09	0;180	27,93	49,37
shapes2	15	28	7	44	6,29	176	6,29	0;180	14,29	30,77
shirts	40	99	8	53	6,63	599	6,05	0;180	7,30	20,07
swim	5.752	48	10	219	21,90	960	20,00	0;180	24,32	45,05
trousers	79	64	17	86	5,06	388	6,06	0;180	3,78	15,56

O atributo de área dos itens das instâncias foi destacado por revelar dificuldades específicas de posicionamento. A tabela 4.2 exibe dados sobre as áreas dos polígonos que representam os itens, sendo uma linha para cada instância. As colunas apresentam, da esquerda para a direita, o nome da instância, a soma das áreas das diferentes formas da instância, a área média das diferentes formas, a soma das áreas de todos os itens da instância, a área média dos itens, e a amplitude de áreas dos itens. Este último valor refere-se à diferença entre a maior e a menor áreas de itens da instância, sendo isto apresentado em relação ao valor da maior área.

Tabela 4.2. Dados complementares das instâncias: área.

Instância	Formas		Itens		
	Σ área	$\bar{área}$	Σ área	$\bar{área}$	Ampl %
albano	16.954.015,50	2.119.251,94	42.656.785,00	1.777.366,04	95,30
blaz2	5.650,00	1.412,50	22.600,00	1.412,50	31,25
dagli	101.178,00	10.117,80	303.534,00	10.117,80	92,92
dighe1	10.000,00	625,00	10.000,00	625,00	89,23
dighe2	10.000,00	1.000,00	10.000,00	1.000,00	94,25
fu	1.083,00	90,25	1.083,00	90,25	83,67
mao	1.860.430,00	206.714,44	3.758.617,00	187.930,85	94,77
marques	282.700,00	35.337,50	719.400,00	29.975,00	95,51
shapes0	16.000,00	4.000,00	159.600,00	3.711,63	72,22
shapes1	16.000,00	4.000,00	159.600,00	3.711,63	72,22
shapes2	8.100,00	1.157,14	32.400,00	1.157,14	75,00
shirts	23.500,00	2.937,50	216.000,00	2.181,82	96,51
swim	6.294.595,33	629.459,53	25.447.218,15	530.150,38	87,21
trousers	330.600,00	19.447,06	1.720.650,00	26.885,16	96,69

Pode-se identificar situações distintas nas instâncias. As instâncias *albano*, *mao*, *marques*, *shapes0*, *shapes1*, *shirts* e *swim* possuem médias de áreas dos itens menor do que a média das áreas das diferentes formas. Isso indica que as áreas dos itens de formas menores são dominantes sobre o conjunto dos itens. Em um processo iterativo de posicionamento, esses itens podem vir a ocupar espaços internos deixados entre os itens maiores já posicionados. Nas instâncias *blaz2*, *dagli*, *dighe1*, *dighe2*, *fu* e *shapes2* esta diferença de áreas médias de formas e de itens não ocorre, pois as quantidades de itens são idênticas para todas as formas.

A instância *trousers* mostra uma situação inversa, onde a média de áreas dos itens é maior do que a área média das formas. Isto indica que as áreas dos itens maiores dominam sobre o conjunto de itens. Consequentemente, a menos que as formas grandes permitam encaixes entre si, os itens de formas menores podem não ocupar satisfatoriamente os espaços vagos deixados durante o posicionamento dos maiores. Entretanto, a tabela 4.1 mostra um valor médio de não-retangularidade de 15,56%, que é o terceiro menor valor médio deste atributo entre as instâncias utilizadas, e um baixo índice médio de

concavidade. Estas características apontam para condições em que se espera haver menos dificuldades para o aproveitamento do espaço do objeto receptor.

4.2 Testes realizados

Os experimentos consistiram na realização de lotes de testes para cada instância selecionada. Em cada lote, foram executadas várias vezes o método construtivo proposto, fazendo variação de parâmetros de configuração do algoritmo a cada teste. Para os parâmetros que não dependem de características da instância para serem definidos, os valores possíveis foram os relacionados a seguir.

- Posicionamento:
 - Regras:
 - * *Bottom-left* (BL);
 - * *Middle-left* (ML);
 - Opções:
 - * Melhor ponto de referência (BEST_REF);
 - * Melhor ponto central (BEST_CENT);
 - * Menor distância máxima (MIN_LEN);
- Heurísticas de seleção:
 - Estáticas:
 - * Maior área primeiro (G_S_AREA);
 - * Maior comprimento primeiro (G_S_LEN);
 - * Maior largura primeiro (G_S_WID);
 - * Maior perímetro primeiro (G_S_PERIM);
 - * Maior concavidade primeiro (G_S_IRREG);
 - * Maior não-retangularidade primeiro (G_S_NONRECT);
 - * Menor valor agregado primeiro (G_S_AGGR);
 - Dinâmica:
 - * Menor contribuição relativa para o aumento do comprimento do leiaute (G_D_L_LEN_REL);
- Tipo de agrupamento entre compactações:

- Por unidade (un);
- Por conjunto de itens de mesma forma (sh).

O parâmetro que configura o tamanho do intervalo entre as compactações, medido em termos do tipo de agrupamento definido, depende do número de itens ou do número de diferentes formas de cada instância. O tamanho do intervalo pode assumir qualquer valor da sequência $\{1, \dots, k\}$, sendo que k é o número de itens da instância, se o tipo de agrupamento for por unidade, ou k é o número de diferentes formas da instância, se o tipo de agrupamento for por conjunto de itens de mesma forma.

A figura 4.1 ilustra o esquema que foi adotado para fazer a combinação dos parâmetros. Nesta figura estão cinco conjuntos de valores, separados por símbolos “X” que representam as combinações. Os conjuntos referem-se a, da esquerda para a direita, regra e opção de posicionamento, critério primário da heurística de seleção do próximo item, critério secundário desta heurística, intervalo entre compactações, e tipo de agrupamento feito entre compactações. Devido a restrições de tempo para fazer os testes, foi necessário limitar as combinações. Uma limitação foi aplicar somente a opção BEST_REF para a regra de posicionamento BL. Outra limitação, menos severa, está no uso de apenas quatro critérios secundários para a heurística de seleção do próximo item. Uma terceira limitação foi a exploração de agrupamentos por conjuntos de formas congruentes em intervalos de até oito conjuntos entre cada compactação, o que corresponde ao piso da metade do maior número de formas diferentes encontradas nas instâncias de teste.

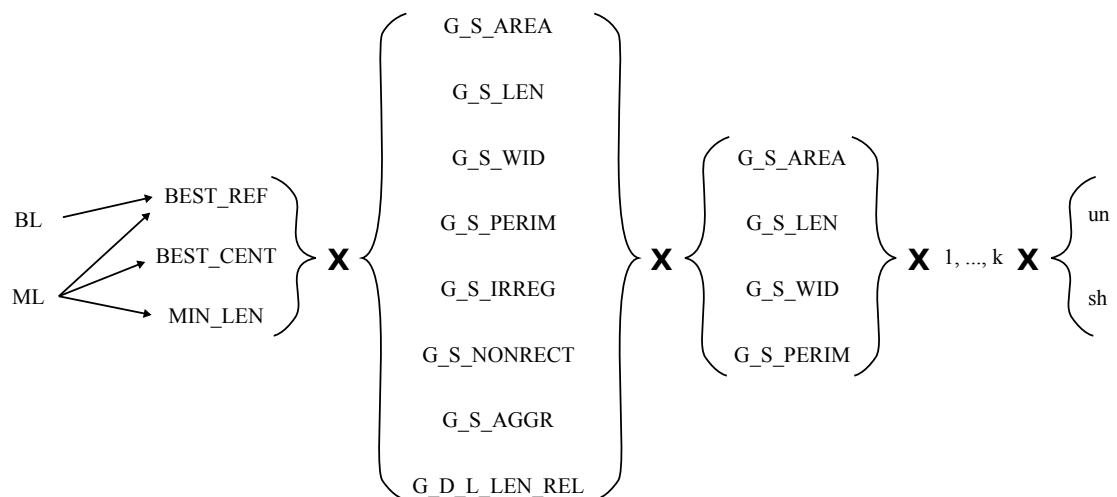


Figura 4.1. Combinações de parâmetros dos experimentos realizados.

Para cada instância, foram feitas 128 combinações dos parâmetros de configuração da regra e opção de posicionamento, critério primário e critério secundário da heurística

de seleção. Essas combinações foram repetidas para cada variação do intervalo de compactações e do tipo de agrupamento. Como o critério secundário da heurística de seleção pode não ser aplicado em alguns casos, a exploração das combinações foi interrompida quando identificadas situações deste tipo, e retomadas da próxima combinação de critérios. Assim, foram programados 75.445 testes com uso da compactação intercalada com posicionamentos, mas realizados 25.143, havendo um grande número de descartes por repetição de sequências já testadas de itens da instância. Também foram executados 128 testes para cada instância, sem o uso da compactação intercalada, a fim de confrontar os resultados obtidos com e sem a aplicação desta sub-rotina. Somam-se, portanto, o total de 26.935 testes realizados.

Todos os testes foram executados em computador com CPU *Phenom 9550* 2,2 GHz, 4 GB de memória RAM, executando sistema operacional *Linux*. Os modelos de programação linear foram solucionados com a biblioteca *lp_solve* na versão 5.5.2.0. Informações dos resultados dos testes são apresentadas a seguir.

4.3 Resultados dos testes

Como o método em análise tem o objetivo de fornecer soluções de boa qualidade para a etapa de construção de leiautes válidos, não era esperado atingir resultados que competissem com as principais abordagens conhecidas para este problema, que usam técnicas de busca como as que estão comentadas nas seções 2.1.3.2 e 2.1.3.3. Os leiautes com melhores aproveitamentos obtidos para cada instância ficaram 8,4%, em média, abaixo dos melhores resultados encontrados na literatura para as mesmas instâncias, que estão na tabela 2.2.

Outra observação sobre os experimentos realizados foi relativa aos tempos de execução registrados. Sabe-se que a montagem do modelo de PL para compactação é computacionalmente dispendiosa, destacadamente a formulação dos critérios para assegurar a não-sobreposição dos itens. E a própria resolução dos problemas de compactação contribui para elevar o consumo de tempo. O impacto da adoção da compactação intercalada aos posicionamentos é analisado na seção 4.3.3.

A tabela 4.3 mostra o resumo dos resultados obtidos. Esta tabela exibe uma linha para os dados de cada instância, e as colunas apresentam, da esquerda para a direita, o nome da instância, o número de testes realizados, a média de tempo por teste, a soma dos tempos dos testes, a média de aproveitamento das soluções encontradas, o maior aproveitamento de solução para a instância nos testes realizados, e nas duas últimas colunas estão os melhores resultados da literatura para cada instância usando apenas métodos

construtivos¹, e usando métodos que exploram o espaço de soluções viáveis com técnicas de busca. Os números sobrescritos ao lado dos resultados encontrados na literatura indicam os trabalhos em que foram publicados, conforme informado na seção 2.3. Os leiautes com melhores aproveitamentos superam os resultados dos métodos construtivos que foram encontrados na literatura em 11 das 14 instâncias usadas para os testes. Os melhores resultados estão assinalados com asterisco ao lado do valor do aproveitamento. Nos testes realizados, apenas para as instâncias *shapes0* e *shapes1* o método construtivo proposto não conseguiu superar os resultados de métodos construtivos encontrados na literatura. O método *bottom-left* apresentado em Gomes [2005] encontrou solução melhor para a instância *shapes0* usando um critério aleatório de ordenação dos itens. A melhor solução de método construtivo para a instância *shapes1* foi obtida pelo método TOPOS revisado, encontrado em Bennell & Song [2010].

Tabela 4.3. Resumo dos resultados dos testes com o método construtivo proposto (tempo e aproveitamento) e comparativo com os melhores resultados da literatura.

instância	unid.	tempo		resultados obtidos		melhores da literatura	
		\bar{t}	Σt	média (\bar{A})	melhor (A)	construtivo	geral
albano	1260	1,20	1510,88	74,92	82,58 *	80,90 ⁰	88,39 ⁴
blaz2	1142	0,68	771,28	66,30	71,92 *	-	68,60 ⁹
dagli	1686	1,40	2352,82	71,56	80,63 *	78,83 ⁰	87,97 ³
dighe1	1254	0,48	595,80	63,65	77,97 *	68,78 ⁰	100,00 ⁵
dighe2	943	0,25	239,36	64,92	77,11 *	71,84 ⁰	100,00 ⁵
fu	1455	0,35	504,31	71,52	83,82 *	79,17 ⁰	92,03 ⁷
mao	1186	2,11	2498,24	70,50	80,06 *	72,82 ⁰	85,15 ⁷
marques	1252	1,77	2218,19	76,55	84,73 *	80,59 ⁰	90,01 ⁴
shapes0	1632	1,94	3171,76	55,16	60,45	61,39 * ⁰	68,44 ⁶
shapes1	1632	2,50	4076,05	57,38	64,88	67,60 * ²	73,84 ⁷
shapes2	2036	1,52	3085,69	72,22	79,41 *	74,74 ¹	84,25 ⁶
shirts	4286	10,01	42906,10	82,33	87,01 *	84,10 ²	89,69 ³
swim	1795	23,19	41621,69	62,76	69,36 *	66,82 ⁰	75,60 ⁸
trousers	3584	7,88	28230,37	80,02	87,72 *	86,00 ²	90,60 ⁸

Os testes que obtiveram os melhores resultados para cada instância estão detalhados na tabela 4.4. Nesta tabela, há uma linha para cada instância, e as colunas representam, da esquerda para a direita, o nome da instância, a regra de posicionamento usada, a opção da regra de posicionamento, o critério primário da heurística de seleção do próximo item, o

¹Para a instância *blaz2*, não foi encontrado, na literatura, relato de solução obtida apenas com método construtivo.

critério secundário da mesma heurística, a quantidade de grupos e o tipo de agrupamento feito entre cada compactação, o tempo decorrido na execução do teste, em segundos, e o número total de compactações efetuadas para construir o leiaute. Os leiautes gerados que obtiveram os melhores aproveitamentos estão expostos no apêndice A.

Tabela 4.4. Detalhes dos testes com os melhores resultados.

Instância	Regra	Op-regra	Ordem-prim	Ordem-sec	Agrup	t	# Comp
albano	BL	BEST_REF	G_D_L_LEN_REL	G_S_PERIM	5 un	1,36	34
blaz2	ML	MIN_LEN	G_D_L_LEN_REL	G_S_AREA	13 un	0,63	13
dagli	ML	BEST_REF	G_D_L_LEN_REL	G_S_PERIM	14 un	1,46	20
dighe1	BL	BEST_REF	G_S_AREA	G_S_LEN	5 sh	0,37	15
dighe2	ML	BEST_CENT	G_S_AGGR	G_S_WID	1 sh	0,31	26
fu	BL	BEST_REF	G_D_L_LEN_REL	G_S_AREA	1 un	0,66	50
mao	ML	BEST_CENT	G_S_PERIM	G_S_AREA	3 un	2,73	36
marques	BL	BEST_REF	G_D_L_LEN_REL	G_S_WID	10 un	1,59	16
shapes0	ML	BEST_CENT	G_S_NONRECT	G_S_AREA	2 un	6,24	89
shapes1	ML	BEST_REF	G_S_AREA	G_S_PERIM	5 un	4,04	39
shapes2	BL	BEST_REF	G_S_LEN	G_S_WID	11 un	1,46	19
shirts	BL	BEST_REF	G_S_IRREG	G_S_LEN	3 sh	5,77	19
swim	BL	BEST_REF	G_S_PERIM	G_S_LEN	10 un	22,29	23
trousers	BL	BEST_REF	G_S_PERIM	G_S_AREA	2 un	27,61	151

As configurações dos testes que obtiveram os melhores resultados revelam que o uso da regra *middle-left* levou ao melhor resultado dos testes para seis das 14 instâncias avaliadas, sendo superado pela regra *bottom-left* para as demais. Com respeito ao critério primário da heurística de seleção do próximo item, predominou a avaliação dinâmica da contribuição para o aumento do comprimento do leiaute, com melhores resultados para cinco instâncias. Os agrupamentos dos testes com melhores aproveitamentos representam intervalo médio de 25,8% dos itens sendo posicionados entre cada compactação. E o número de compactações reflete a pequena necessidade de repetições do processo iterativo da sub-rotina de compactação.

No entanto, quando se avalia os parâmetros do algoritmo com as melhores médias de aproveitamento, a diferença da eficiência da regra *bottom-left* para a da *middle-left* fica acentuada. Os testes que usaram a regra *bottom-left* atingiram aproveitamentos médios melhores para 11 das 14 instâncias. A tabela 4.5 exhibe esses dados, com uma linha para

cada instância, e as colunas mostram, da esquerda para a direita, o nome da instância, a combinação de regra e opção que levou à melhor média de aproveitamento, o valor desta média, o critério primário da heurística de seleção que levou à melhor média de aproveitamento, o valor desta média, o agrupamento entre compactações que levou à melhor média de aproveitamento, e o valor desta média.

Tabela 4.5. Parâmetros com as melhores médias de aproveitamento.

Instância	Regra e opção		Ordem-prim		Agrupamento	
	Combinação	\bar{A}	Parâmetro	\bar{A}	Parâmetro	\bar{A}
albano	BL-BEST_REF	75,76	G_S_AGGR	77,67	4 un	75,96
blaz2	BL-BEST_REF	66,95	G_S_AREA	67,16	13 un	67,15
dagli	ML-BEST_CENT	72,28	G_D_L_LEN_REL	76,48	4 un	72,65
dighe1	BL-BEST_REF	65,49	G_S_AREA	68,01	1 un	65,73
dighe2	ML-BEST_CENT	66,92	G_S_PERIM	68,09	8 un	66,02
fu	BL-BEST_REF	73,07	G_D_L_LEN_REL	74,63	9 un	72,72
mao	ML-MIN_LEN	72,11	G_S_AREA	72,42	1 un	71,99
marques	BL-BEST_REF	79,39	G_S_AREA	80,56	1 un	78,19
shapes0	BL-BEST_REF	56,70	G_D_L_LEN_REL	57,17	5 un	55,78
shapes1	BL-BEST_REF	59,33	G_S_AREA	58,99	7 un	58,13
shapes2	BL-BEST_REF	74,26	G_D_L_LEN_REL	72,51	1 un	73,38
shirts	BL-BEST_REF	84,10	G_S_NONRECT	83,02	4 un	83,27
swim	BL-BEST_REF	64,08	G_S_LEN	63,51	5 un	64,04
trousers	BL-BEST_REF	83,29	G_S_AREA	82,87	1 un	82,07

Ainda sobre as melhores médias de aproveitamento, quanto ao critério primário da heurística de seleção do próximo item, houve destaque para a seleção da maior área primeiro. Os testes com este critério primário de seleção alcançaram médias de aproveitamentos melhores para seis instâncias, enquanto o uso do critério dinâmico de seleção pela menor contribuição para o aumento do comprimento do leiaute levou às melhores médias para quatro instâncias. Já os agrupamentos que levaram às melhores médias de resultados não foram expressivamente diferentes dos testes que atingiram os melhores resultados gerais. Dos agrupamentos que participaram das melhores médias de aproveitamento, a média do número de posicionamentos entre cada compactação ficou em 23,5% do número de itens das instâncias.

Como os experimentos foram realizados para obter várias diferentes combinações de parâmetros do método proposto, para cada instância selecionada, buscou-se conduzir a análise dos resultados para verificação das hipóteses levantadas na seção 1.4, de modo que se pudesse identificar relações entre as características das instâncias e os parâmetros de execução do método. Ao mesmo passo que também se buscou identificar as contribuições da compactação de leiautes parciais para a qualidade das soluções construídas. Sendo assim, são apresentadas, a seguir, análises dos resultados a partir de diferentes visões. Primeiro, são observadas as regras de posicionamento e a influência destas para a qualidade dos leiautes produzidos. Na sequência, está a verificação do impacto das diferentes formações da heurística de seleção do próximo item nos resultados. Depois, está um exame da aplicação da compactação e como isso afetou a construção dos leiautes. Por fim, é feita uma discussão de todos os resultados obtidos, tratando da relação entre os diferentes parâmetros e as características das instâncias de teste.

4.3.1 Perspectiva de regras de posicionamento

As regras de posicionamento aplicadas nos testes foram observadas juntamente com as opções usadas por estas. Como houve uma variação significativa de índices de aproveitamento nos experimentos para cada instância, com cada regra, ao variar os demais parâmetros do método proposto, optou-se por analisar os resultados por meio de medidas de dispersão. Portanto, foram coletados dos resultados, para cada combinação de regra e opção, com relação ao aproveitamento de solução para cada instância, o valor máximo A_{max} , o mínimo A_{min} , a média \bar{A} e o desvio padrão σ . Estes dados estão apresentados no gráfico da figura 4.2. Nesta, para cada instância usada nos testes há quatro intervalos, sendo um para cada uma das combinações BL-BEST_REF, ML-BEST_REF, ML-BEST_CENT, ML-MIN_LEN. Cada intervalo está dividido em três subintervalos, sendo $[\bar{A} - \sigma, \bar{A} + \sigma]$ o central, destacado em tonalidade mais escura, $[A_{min}, \bar{A} - \sigma]$ à esquerda e $[\bar{A} + \sigma, A_{max}]$ à direita, em tonalidade mais clara.

O subintervalo central é aquele com a maior incidência dos resultados obtidos nos experimentos para cada combinação de regra e opção, para cada instância. Ao observar a posição deste subintervalo, relativa aos valores A_{min} e A_{max} , é possível identificar a eficácia do uso da regra para a instância. De fato, quanto mais elevados são os aproveitamentos deste subintervalo, mais aplicável é a referida regra para a referida instância. Mas, além disso, quanto menor for o desvio padrão, maior é a estabilidade da aplicação da referida regra para a referida instância. E, quanto maior é o subintervalo da direita em relação ao da esquerda, maior é a expectativa de que o uso da referida regra para a referida instância obtenha resultados mais próximos do valor máximo fornecido por esta combinação.

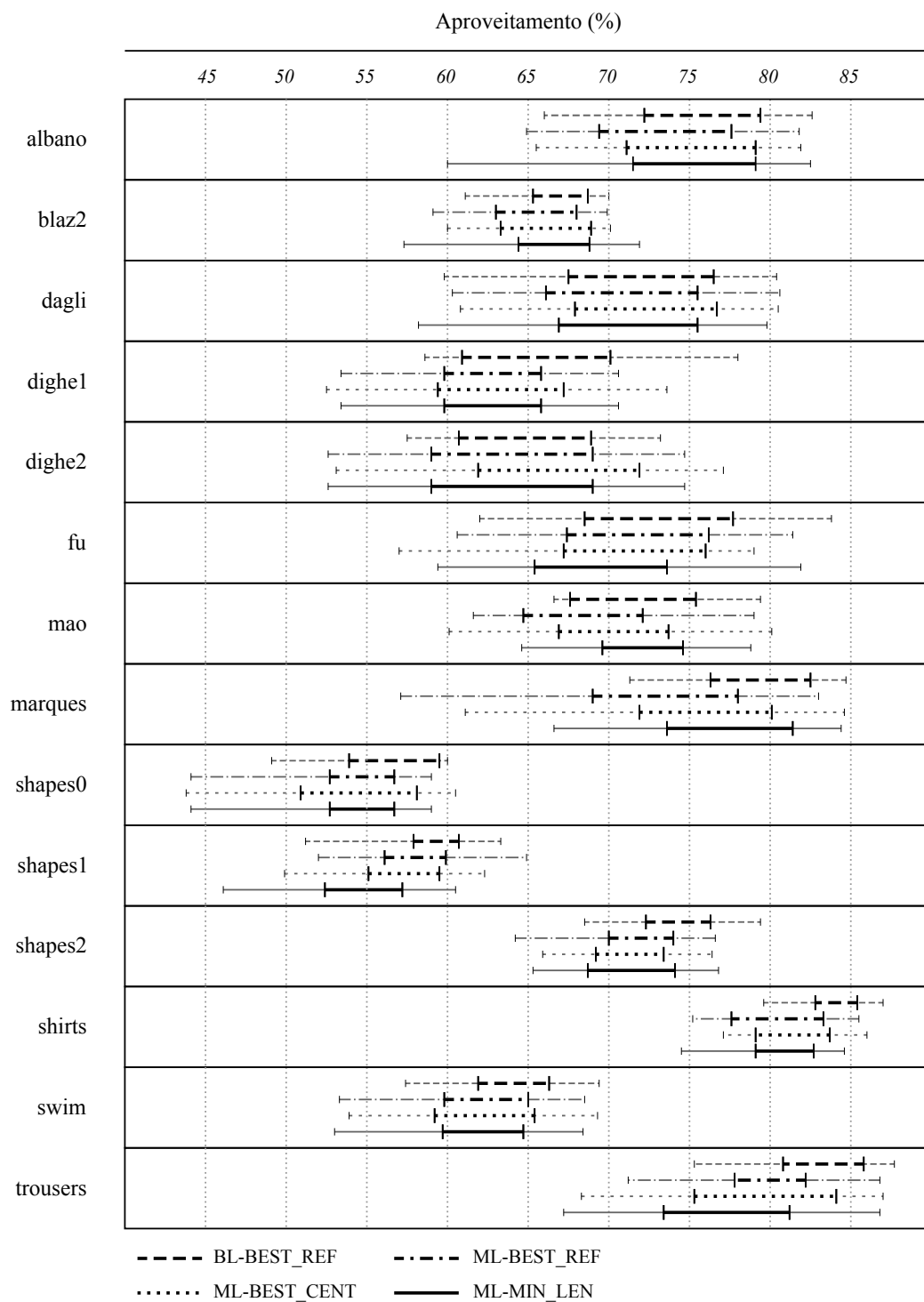


Figura 4.2. Resultados em intervalos agrupados por regras para cada instância.

Percebe-se uma ampla superioridade da regra BL-BEST_REF sobre as outras, sendo esta mais eficaz para 11 das 14 instâncias avaliadas. Houve um equilíbrio entre as variações de opções para a regra ML. É importante salientar a questão de estabilidade

das regras. Os valores de desvios-padrões ficaram, em média, entre 3% e 3,5%, havendo discrepâncias que atingem até 5%, no máximo, e 1,3%, no mínimo. Isto indica que as maiores concentrações de resultados ficaram relativamente próximas da média. Esta característica é interessante dada a dificuldade associada com a tarefa de designar a melhor regra a ser usada para uma dada instância qualquer.

Para algumas instâncias, a influência da regra de posicionamento sobre os resultados é mais fortemente sugerida pela distribuição dos resultados de aproveitamentos de leiautes. A figura 4.3 ilustra uma situação em que há indícios da participação da regra como fator que interfere na qualidade da solução fornecida pelo método construtivo. O gráfico mostrado nesta figura representa a distribuição dos resultados por faixas de aproveitamentos para a instância *marques*, sendo cada faixa um intervalo de comprimento fixado em 5%. Neste gráfico, é possível observar a nítida diferença entre as regras, havendo maior concentração de aproveitamentos elevados para a regra BL-BEST_REF, e a maior concentração de aproveitamentos baixos para a regra ML-BEST_REF.

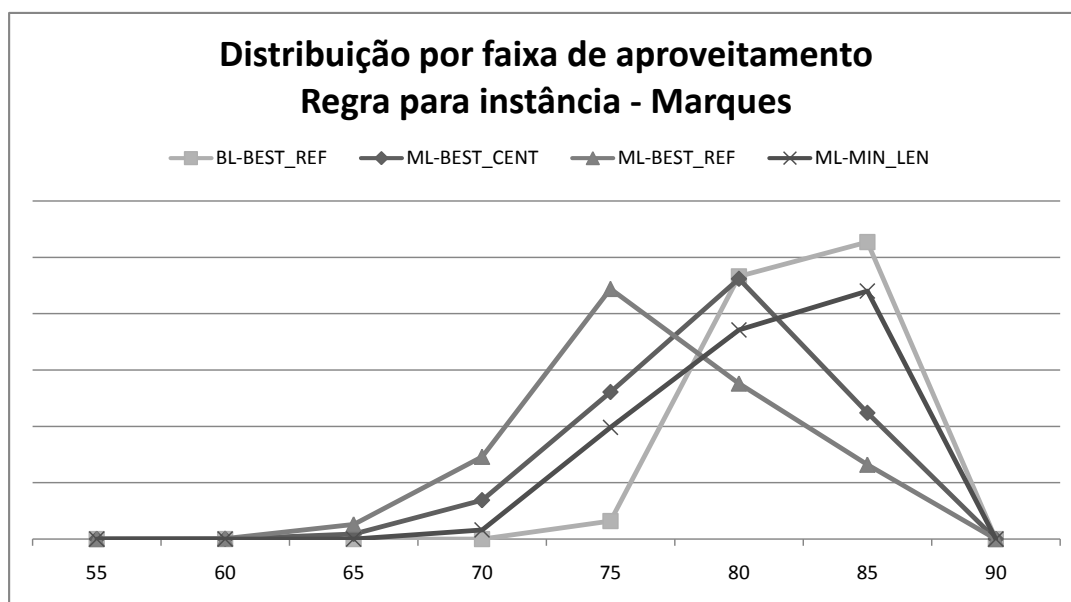


Figura 4.3. Distribuição de aproveitamento por regra, para a instância *marques*.

O mesmo não ocorre para outras instâncias, como é o caso de *blaz2*. O gráfico da figura 4.4 mostra a distribuição de resultados da aplicação das regras de posicionamento para essa instância. Observando este gráfico, não é tão nítida a contribuição desses parâmetros para a qualidade do leiaute construído.

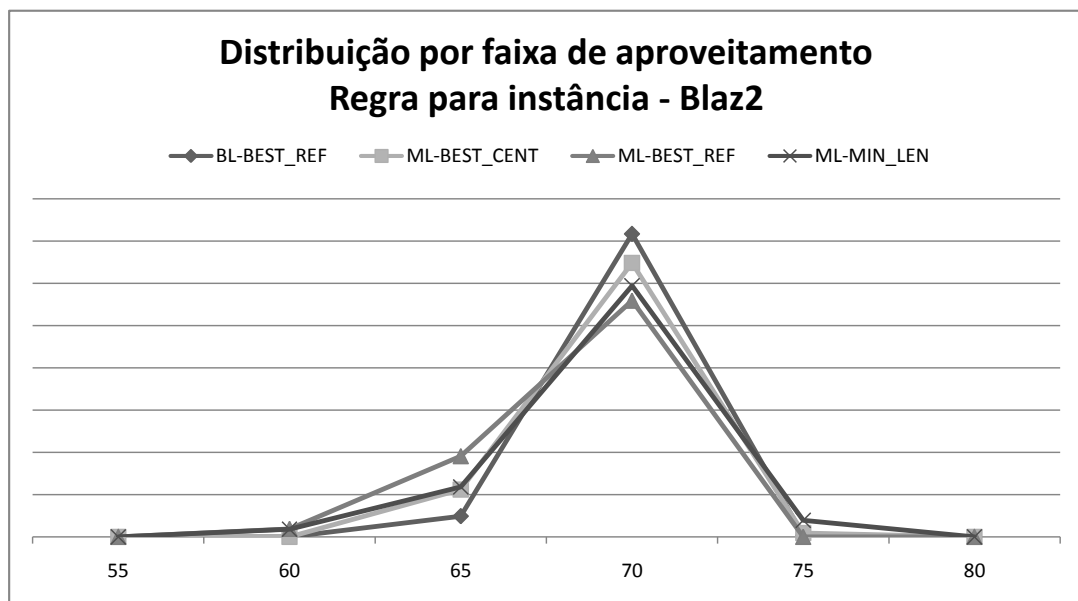


Figura 4.4. Distribuição de aproveitamento por regra, para a instância *blaz2*.

4.3.2 Perspectiva de critérios de ordenação

Para observar a influência, nos resultados, do uso dos diferentes critérios para a heurística de seleção do próximo item a posicionar, foi feito o agrupamento dos testes pelo parâmetro de ordenação primária. Assim, foram obtidas as médias de aproveitamento para cada combinação de instância e critério primário de seleção. Os valores estão dispostos na tabela 4.6. As melhores médias para cada instância estão destacados com fundo cinza. A linha abaixo dos dados dos valores de aproveitamento informa a soma das posições ocupadas pelo critério de seleção, ao classificar todos os critérios para cada instância do melhor para o pior resultado médio. A última linha apresenta a classificação para a soma das posições.

Tabela 4.6. Dados da eficácia de critérios de ordenação para as instâncias.

Instância	G_S_AREA	G_S_LEN	G_S_WID	G_S_PERIM	G_S_IRREG	G_S_NONRECT	G_S_AGGR	G_D_L_LEN_REL
albano	73,71	76,85	69,95	77,67	74,06	74,59	77,67	76,93
blaz2	67,16	66,56	66,23	65,78	65,78	65,78	65,78	66,74
dagli	74,23	75,01	67,47	73,56	68,89	69,90	75,07	76,48
dighe1	68,01	64,91	59,17	62,93	62,43	64,67	64,01	64,81
dighe2	63,83	66,21	63,69	68,09	63,13	65,23	66,40	65,18
fu	71,69	73,46	69,38	72,29	71,59	68,32	71,87	74,63
mao	72,42	71,30	69,38	72,20	69,63	69,51	69,55	70,46
marques	80,56	79,50	77,14	76,44	73,89	74,67	76,44	77,94
shapes0	56,13	57,10	50,13	55,10	53,58	55,28	57,10	57,17
shapes1	58,99	57,62	56,71	57,17	54,59	58,25	57,62	58,24
shapes2	72,35	72,46	72,43	71,75	71,75	71,55	71,66	72,51
shirts	81,66	82,79	81,39	82,92	82,17	83,02	82,29	82,67
swim	63,22	63,51	60,79	62,80	63,19	62,52	62,80	63,36
trousers	82,87	78,80	78,05	82,85	79,82	76,54	82,18	79,12
Soma das posições	46	40	95	54	86	77	56	38
Classificação	3°	2°	8°	4°	7°	6°	5°	1°

Com esses dados, observa-se que o critério de seleção mais regularmente eficaz para todas as instâncias foi o dinâmico, que escolhe o item que leva ao menor aumento relativo do comprimento do leiaute. O segundo critério mais regularmente eficaz foi a ordenação decrescente pelo comprimento. O terceiro foi a ordenação decrescente pela área. As ordenações decrescentes por largura e por irregularidade foram as menos eficazes, segundo esta classificação.

Assim como ocorre com as regras de posicionamento testadas, os resultados sugerem contribuição mais incisiva dos critérios de seleção para algumas instâncias. Um exemplo é a instância *dagli*, cujos resultados trazem concentrações bem distintas de apro-

veitamentos de acordo com o critério usado para seleção dos itens. A figura 4.5 mostra este cenário ao apresentar a distribuição de resultados para as ordenações primárias, agrupados em faixas formadas por intervalos de comprimento fixado em 5%. Nota-se claramente picos de concentração de aproveitamentos em três faixas distintas. Na faixa de 70%, a maior concentração é de G_S_WID e, um pouco menor, de G_S_IRREG, revelando o pouco benefício destes critérios para a instância *dagli*. Na faixa de 75% está a maior concentração de G_S_PERIM, com G_S_NONRECT e G_S_LEN mais abaixo. Já na faixa de 80% está a maior concentração de G_S_AGGR, com G_D_L_LEN_REL um pouco abaixo, indicando o bom rendimento desses critérios para essa instância.

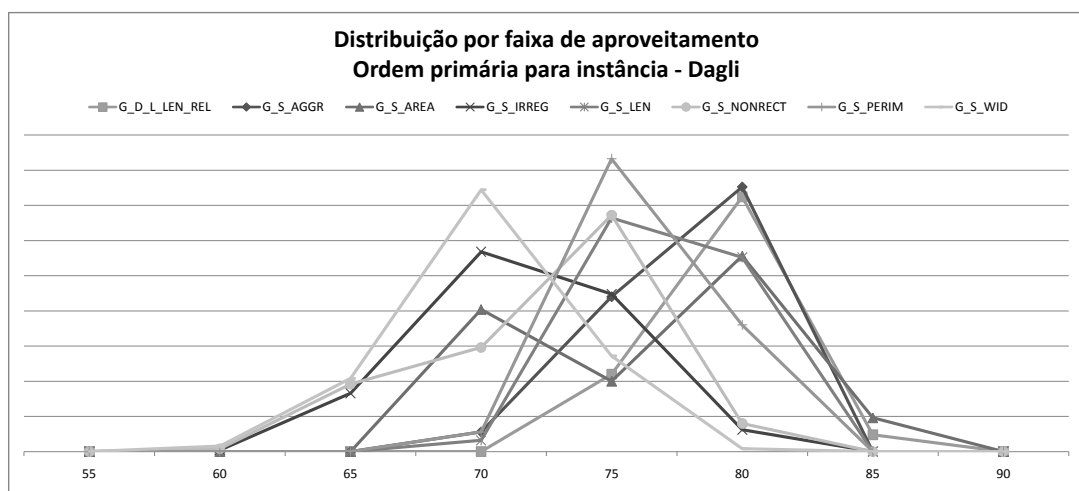


Figura 4.5. Distribuição de aproveitamento por critério de ordenação, para a instância *dagli*.

Para outras instâncias, como *shirts*, em contrapartida, não foi possível identificar indícios de relações entre critérios de seleção e a qualidade dos leiautes construídos. Como é possível observar na figura 4.6, houve uma regularidade muito forte entre os critérios usados, exceto G_S_WID, que teve distribuição de aproveitamento um pouco mais concentrada em valores mais baixos.

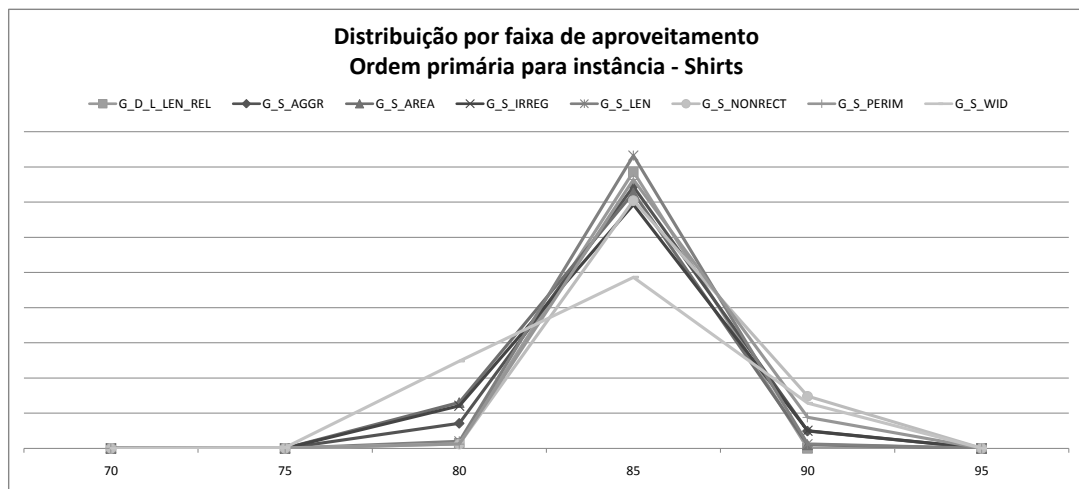


Figura 4.6. Distribuição de aproveitamento por critério de ordenação, para a instância *shirts*.

Outros tipos de observações podem ser feitas sobre os dados obtidos para adquirir informações adicionais. Por exemplo, pode-se agrupar as instâncias em categorias de acordo com os valores de algum atributo, e calcular os dados de aproveitamentos médios com os critérios de ordenação para cada categoria. A tabela 4.7 apresenta os resultados de critérios primários de seleção agrupados em categorias de acordo com três atributos. O primeiro é a relação entre a máxima diferença de áreas dos itens e a média das áreas dos itens. Isto permite identificar que, para as instâncias testadas com pequenas variações entre as áreas dos itens maiores e as dos menores, com valores menores do que 1,5, o uso do critério de seleção *G_S_LEN* levou aos melhores resultados, em média. Para as instâncias testadas que apresentavam variações um pouco maiores entre as áreas dos itens, com relação oscilando entre 1,5 e 2,75, o uso do critério dinâmico *G_D_L_LEN_REL* gerou os melhores resultados, em média. E para as instâncias cujas relações de variação de áreas entre os itens e a área média dos itens obtiveram valores maiores do que 2,75, o uso do critério de seleção *G_S_AGGR* foi o mais eficaz, em média. A última linha desta tabela mostra o número de testes agrupados em cada categoria.

Tabela 4.7. Relação entre critérios de ordenação e atributos de instâncias.

Ordem Primária	AmplitudeÁrea / MédiaÁrea			Irregularidade / Árealtens			Não-retangularidade / Árealtens		
	X < 1,5	1,5 ≤ X ≤ 2,75	X > 2,75	X < 2	2 ≤ X ≤ 4	X > 4	X < 8	8 ≤ X ≤ 20	X > 20
G_S_AREA	62,39	68,48	80,81	78,69	65,60	67,97	79,51	64,93	67,97
G_S_LEN	65,52	69,45	80,26	77,84	65,33	69,07	77,26	66,91	69,55
G_S_WID	64,85	64,98	78,30	75,60	60,97	68,69	76,27	64,67	65,16
G_S_PERIM	61,07	68,04	81,33	79,55	64,09	67,38	79,90	64,04	67,46
G_S_IRREG	59,79	66,59	79,00	75,84	65,49	67,18	78,50	64,43	65,53
G_S_NONRECT	61,42	66,64	78,59	75,73	63,83	66,70	77,54	63,37	66,88
G_S_AGGR	61,82	67,75	81,45	79,18	64,79	67,29	80,60	64,94	66,98
G_D_L_LEN_REL	64,47	69,68	80,21	78,91	65,35	68,48	78,52	66,19	70,39
# Testes	6442	8319	10382	11954	7030	6159	11524	9967	3652

Na mesma tabela, o segundo atributo usado para organizar as instâncias foi a irregularidade dos itens das instâncias. Para formar as categorias, foi feita a relação entre a área total de concavidades e a área total dos itens de cada instância, expressada em valores percentuais. Assim, foram formadas três categorias de instâncias, sendo uma para aquelas que apresentam esta relação com valor menor do que 2%, outra para as que têm valores entre 2% e 4% para esta relação, e a terceira para as instâncias com esta mesma relação tendo valores maiores do que 4%. Para a primeira categoria, com baixo índice de concavidade, o uso do critério de ordenação G_S_PERIM foi o mais eficaz. Para a segunda categoria, o critério de ordenação que levou à melhor média de aproveitamentos foi G_S_AREA. E para a terceira categoria, que apresenta taxas maiores de concavidade, o critério G_S_LEN obteve os melhores resultados, em média.

O terceiro atributo apresentado na mesma tabela refere-se à relação entre a não-retangularidade dos itens e a área total destes, expressada em valores percentuais. Para as instâncias com esta relação de não-retangularidade menor do que 8%, o critério de seleção que obteve os melhores resultados, em média, foi G_S_AGGR. Para instâncias com índice de não-retangularidade entre 8% e 20%, o uso do critério G_S_LEN levou à melhor média de aproveitamento. Já o critério dinâmico G_D_L_LEN_REL foi o que obteve a melhor média de aproveitamento com instâncias que apresentam relação de não-retangularidade acima de 20%.

4.3.3 Perspectiva de compactação

Provavelmente, a observação mais elementar sobre a aplicação da sub-rotina de compactação intercalada com posicionamentos é verificar se isto faz com que os resultados sejam

melhores, comparados à execução do método construtivo sem usar esta sub-rotina. Para buscar informações sobre esta questão, foram executados testes sem uso da compactação, de modo que as mesmas configurações de parâmetros fossem experimentadas.

Tabela 4.8. Visão geral dos resultados dos testes sem uso da compactação.

Instância	# Testes	Σt	\bar{t}	\bar{A}	max A
albano	128	35,64	0,28	73,14	78,65
blaz2	128	17,68	0,14	63,45	68,24
dagli	128	38,00	0,30	69,15	78,15
dighe1	128	18,80	0,15	60,56	70,67
dighe2	128	13,53	0,11	62,97	72,35
fu	128	17,34	0,14	67,92	79,17
mao	128	87,08	0,68	65,99	78,93
marques	128	61,16	0,48	74,42	84,19
shapes0	128	24,86	0,19	54,98	59,55
shapes1	128	41,88	0,33	57,08	61,38
shapes2	128	29,94	0,23	69,77	76,73
shirts	128	98,47	0,77	80,69	85,43
swim	128	451,98	3,53	60,29	66,49
trousers	128	130,80	1,02	76,91	85,16

Os resultados desses testes estão sintetizados na tabela 4.8. Esta tabela apresenta dados equivalentes aos que estão na tabela 4.3, mas com os resultados sem uso de compactação. Observa-se que para todas as instâncias o aproveitamento máximo alcançado com uso da compactação superou o aproveitamento máximo sem uso desta sub-rotina. Esses resultados têm o custo do aumento do tempo necessário para gerar os leiautes. Em média, foi registrado elevação do tempo de execução, com a compactação, em 5,73 vezes o tempo sem compactar.

Tabela 4.9. Contagem de resultados dos testes com uso da compactação (C) comparados a testes com as mesmas configurações, mas sem uso desta sub-rotina (NC).

Instância	C > NC		C = NC		C < NC	
	#	%	#	%	#	%
albano	1002	79,5	9	0,7	249	19,8
blaz2	998	87,4	41	3,6	103	9,0
dagli	1511	89,6	13	0,8	162	9,6
dighe1	1081	86,2	48	3,8	125	10,0
dighe2	758	80,4	33	3,5	152	16,1
fu	1131	77,7	179	12,3	145	10,0
mao	1087	91,7	0	0,0	99	8,3
marques	925	73,9	95	7,6	232	18,5
shapes0	397	24,3	1016	62,3	219	13,4
shapes1	546	33,5	899	55,1	187	11,5
shapes2	1680	82,5	5	0,2	352	17,3
shirts	3783	88,3	0	0,0	503	11,7
swim	1402	78,1	162	9,0	231	12,9
trousers	3257	90,9	19	0,5	308	8,6

Para explorar mais detalhadamente esta comparação, os resultados dos testes sem uso da compactação foram comparados aos testes feitos com uso da compactação, tendo os mesmos parâmetros de regra e opção de posicionamento, e critérios primário e secundário da heurística de seleção. Foram feitas as contagens de ocorrências de resultados iguais, de casos em que os resultados com uso da compactação foram melhores do que sem esta sub-rotina, e de testes em que o uso da compactação levou a aproveitamentos menores do que sem esta sub-rotina. Esses dados estão na tabela 4.9, que apresenta essas contagens em valores diretos (#) e em valores percentuais (%), com destaque em cinza para as maiores contagens de cada instância. Pode-se notar que apenas para duas instâncias o número de empates entre o uso ou não da compactação foi predominante nos testes. Para todas as outras instâncias avaliadas, o uso da compactação teve resultados superiores em número significativamente maior do que quando não foi aplicada esta sub-rotina.

Tabela 4.10. Comparação entre as médias de aproveitamentos e consumo de tempo obtidos sem compactação e com a compactação intercalada.

Instância	Média de Aproveitamento (%)			Média de consumo de tempo (s)		
	NC	C - NC	C	NC	C / NC	C
albano	73,14	+1,78	74,92	0,28	+4,29	1,20
blaz2	63,45	+2,85	66,30	0,14	+4,86	0,68
dagli	69,15	+2,41	71,56	0,30	+4,67	1,40
dighe1	60,56	+3,09	63,65	0,15	+3,20	0,48
dighe2	62,97	+1,95	64,92	0,11	+2,27	0,25
fu	67,92	+3,60	71,52	0,14	+2,50	0,35
mao	65,99	+4,51	70,50	0,68	+3,10	2,11
marques	74,42	+2,13	76,55	0,48	+3,69	1,77
shapes0	54,98	+0,18	55,16	0,19	+10,21	1,94
shapes1	57,08	+0,30	57,38	0,33	+7,58	2,50
shapes2	69,77	+2,45	72,22	0,23	+6,61	1,52
shirts	80,69	+1,64	82,33	0,77	+13,00	10,01
swim	60,29	+2,47	62,76	3,53	+6,57	23,19
trousers	76,91	+3,11	80,02	1,02	+7,73	7,88
	Elevação média de aproveitamento		2,32	Média de relação de tempo C/NC		5,73

NC: sem compactação; C: compactação intercalada.

Com relação aos resultados médios de aproveitamento de leiaute, o uso da compactação atingiu valores melhores para todas as instâncias, em comparação aos testes sem compactar. A tabela 4.10 apresenta este comparativo, tendo uma linha para cada instância, e as colunas representam o nome da instância, o aproveitamento médio dos resultados dos testes sem uso da compactação, a diferença entre esta média e os resultados compactando, e na última coluna está a média dos aproveitamentos aplicando a compactação intercalada com posicionamentos. Observa-se que para a instância *mao* os resultados com uso da compactação foram 4,57% melhores, em média, do que os testes sem compactar. Para outras instâncias, como *shapes0*, a melhoria não foi tão significativa. Em média, a elevação do aproveitamento para as instâncias avaliadas foi de 2,45%.

Para verificar o impacto dos diferentes tipos de agrupamentos de itens para posicionamentos feitos entre as operações de compactação, os resultados obtidos foram organizados de acordo com este parâmetro. Foram feitas cinco categorias de agrupamentos

por itens, sendo a primeira categoria formada pela reunião dos testes nos quais o número de itens agrupamentos entre compactações ficou no intervalo entre 0% e 20% do total de itens da instância. As demais categorias modificam este intervalo para entre 20% e 40%, entre 40% e 60%, entre 60% e 80%, e a quinta categoria com intervalo de 80% a 100%. Quanto aos agrupamentos por formas congruentes, foram feitas quatro categorias, sendo a primeira formada pela reunião de testes em que o número de diferentes formas posicionadas entre as compactações foi de 0% a 25% do número total de diferentes formas na instância. As outras três categorias variam este intervalo para entre 25% e 50%, entre 50% e 75%, e entre 75% e 100%.

Os dados acumulados desta maneira estão apresentados na tabela 4.11. A categoria de agrupamentos que obteve a melhor média de aproveitamento está destacada em cinza. Pode-se observar que para nove das 14 instâncias avaliadas, o tipo de agrupamento que gerou os melhores resultados médios usou intervalos de itens iguais ou inferiores a 20% do número de itens das instâncias. Para as instâncias *dighe2* e *fu*, os tipos de agrupamentos com melhores resultados médios foram com intervalos de itens entre 60% e 80% do número de itens das instâncias, e para a instância *shapes0*, entre 80% e 100%. Apenas para as instâncias *blaz2* e *mao* os tipos de agrupamentos por formas congruentes foram os que obtiveram as melhores médias de resultados. Para *blaz2*, com agrupamentos entre 50% e 75% de conjuntos de formas congruentes em relação ao número total de diferentes formas na instância, enquanto para *mao*, este intervalo foi entre 0% e 25%. As células marcadas com “-” representam a ausência de testes com este tipo de agrupamento para a instância.

Tabela 4.11. Resultados médios de aproveitamento distribuídos por tipo de agrupamentos entre as compactações.

Instância	Por itens					Por formas			
	0 a 20%	20 a 40%	40 a 60%	60 a 80%	80 a 100%	0 a 25%	25 a 50%	50 a 75%	75 a 100%
albano	75,62	75,27	74,85	74,58	74,79	75,02	74,75	74,03	74,54
blaz2	66,21	66,47	65,98	66,47	66,31	66,13	65,25	66,64	66,57
dagli	72,25	71,22	71,39	71,88	71,28	72,03	70,63	71,32	71,77
dighe1	64,45	63,96	63,79	63,66	62,69	63,86	63,17	-	-
dighe2	64,53	64,41	64,87	65,68	65,17	64,65	65,26	64,99	64,57
fu	71,86	71,63	71,13	72,22	71,83	71,19	71,24	71,01	-
mao	71,22	70,50	70,47	70,15	69,75	71,54	70,76	70,27	70,19
marques	77,34	76,51	76,35	76,41	76,33	76,94	75,83	76,59	76,30
shapes0	55,21	55,21	55,11	55,22	55,25	55,13	55,02	54,77	54,87
shapes1	57,56	57,39	57,35	57,40	57,29	57,54	57,54	57,22	57,18
shapes2	72,53	72,14	72,06	71,91	72,47	72,27	72,40	71,80	72,37
shirts	82,64	82,29	82,04	82,53	82,23	82,14	82,27	82,22	82,12
swim	63,49	62,88	62,38	62,59	62,61	62,63	62,59	62,66	62,03
trousers	81,04	80,30	79,83	79,43	79,44	80,39	79,67	-	-

4.4 Discussão de resultados

O problema abordado neste trabalho é constituído de dois sub-problemas distintos, porém fortemente correlacionados, que lidam com os cálculos geométricos e com a natureza combinatorial da geração automatizada de leiautes para *C&P*. Os experimentos realizados demonstraram a complexidade da identificação de parâmetros que possam ser mais efetivos para instâncias específicas do problema. A caracterização das instâncias através de atributos que sejam significativos é, propriamente, uma tarefa difícil. Isto é reflexo da possibilidade de haver qualquer forma no conjunto de entrada. Mesmo havendo a simplificação das formas por poligonização, a ausência do conhecimento de padrões de encaixe de itens, ou a inexistência de tais padrões, pode levar as abordagens computacionais para este problema a tomar caminhos não-ótimos, e atingir resultados que, mesmo sendo de qualidade aceitável, poderiam ser melhores. O método construtivo proposto compartilha dessas barreiras.

Como foi observado, o método proposto atinge resultados superiores aos métodos construtivos encontrados na literatura, para 11 das 14 instâncias avaliadas nos testes. Isto

revela o potencial de aplicação deste algoritmo construtivo em abordagens mais elaboradas, que poderiam incluir algumas técnicas de busca para melhorar os resultados iniciais.

O aproveitamento médio dos leiautes construídos nos testes realizados com o método que intercala posicionamentos e compactações, se comparado ao aproveitamento médio dos leiautes gerados sem uso de compactação, destacou o grande valor desta técnica para melhorar a qualidade das soluções. Mesmo que a redução dos espaços internos entre os itens pareça, em primeiro momento, contribuir para a incapacidade de posicionar itens menores no interior do leiaute, a frequência de compactações mostrou ser capaz de fazer ajustes efetivos nos leiautes parciais, melhorando a distribuição dos itens pela área do objeto receptor e, conseqüentemente, possibilitando a redução do comprimento necessário para acomodar os itens. Porém, a decisão de que tipo de agrupamento de itens fazer na seqüência de posicionamentos entre as compactações é uma questão que requer uma exploração mais ampla. Os experimentos realizados, mesmo que numerosos, foram insuficientemente variados para permitir a formação de associações seguras entre atributos das instâncias e os parâmetros da compactação intercalada.

Com base nos experimentos realizados, a heurística de seleção do próximo item mostrou ser fundamental no método proposto. Para um subconjunto significativo das instâncias avaliadas, as variações de configuração desta heurística afetaram diretamente a qualidade das soluções. Isto deve-se principalmente ao fato de que, em um método construtivo para este problema, a ordenação dos itens da entrada é parte notória do subproblema combinatorial associado. Novamente, embora seja possível apontar algumas relações entre atributos das instâncias e critérios de ordenação aplicados para alcançar boas soluções, a amostragem de resultados não respalda um sistema de recomendação de configuração desta heurística.

Também foi observado que a regra de posicionamento *bottom-left* apresentou resultados predominantemente melhores do que os obtidos com a regra *middle-left*. Porém, deve-se ressaltar a carência que esta segunda apresenta devido ao seu desenvolvimento rápido e com baixa maturidade. É possível que revisões desta regra permitam identificar melhorias nesta sub-rotina.

Os tempos para construção dos leiautes com o método proposto foram sensivelmente elevados se comparados aos experimentos realizados sem o uso da compactação. Isto é consequência da complexidade do método de definição das restrições para garantir a não sobreposição dos itens dos leiautes parciais. Também contribui para este aumento do tempo de execução a própria resolução dos modelos de programação matemática. Como a média dos experimentos foi de 25,3 compactações por teste, é compreensível a expansão do intervalo de tempo necessário para fornecer soluções.

Capítulo 5

Conclusão

O estudo feito para este trabalho abordou a aplicação de métodos computacionais para fornecer soluções para o problema de posicionamento de formas irregulares bidimensionais, denominado problema de corte e empacotamento. Como é um problema combinatório pertencente à classe NP, as abordagens heurísticas são geralmente preferíveis, já que soluções ótimas podem demandar tempo acima do aceitável para situações reais. Neste contexto, foi apresentado um novo método construtivo fundamentado no conhecimento de vários estudos coletados na literatura especializada nesse campo.

O método construtivo apresentado foi desenvolvido com a combinação de duas abordagens distintas. Uma delas é a aplicação de regra de posicionamento de sequências de itens, ordenados por critérios que observam atributos da geometria das formas dos itens, ou observam atributos do leiaute parcial em construção. A outra abordagem consiste na elaboração de um modelo de programação linear capaz de indicar ajustes nos posicionamentos dos itens do leiaute, de modo que o comprimento deste seja reduzido. A proposta foi efetuar a técnica de compactação depois de posicionar um subconjunto dos itens da instância que está sendo tratada. Depois, posicionar outro subconjunto e efetuar nova compactação, e repetir esses dois passos até que todos os itens tenham sido posicionados e compactados no leiaute.

O sub-problema de posicionamento foi tratado com uso do invólucro de posicionamento, *NFP*, entre pares de polígonos, e de uma heurística que permite reduzir o conjunto de todas as posições válidas para receber um item a um conjunto finito e discreto. O conceito de *NFP* também foi aproveitado para a elaboração dos modelos de programação linear de compactação, sendo usado para a elaboração de restrições que impedem a ocorrência de sobreposição entre os itens.

Os experimentos computacionais que foram realizados mostraram que o método construtivo proposto neste trabalho tem um considerável potencial de aplicação para a re-

solução de problemas de corte e empacotamento. É importante destacar que os resultados obtidos apresentam leiautes com aproveitamentos inferiores aos dos métodos que constituem o estado da arte neste campo de pesquisa. Porém, se comparados aos resultados de métodos construtivos conhecidos, o novo método proposto obteve aproveitamentos superiores para 11 das 14 instâncias avaliadas, demonstrando o poder do modelo de compactação para elevar a qualidade das soluções.

Uma observação necessária da análise dos resultados é feita sobre o grande número de combinações de parâmetros exploradas nos experimentos. Embora os resultados médios tenham sido satisfatórios, principalmente se comparados aos de outros métodos construtivos, as associações entre atributos das instâncias e parâmetros específicos não foram suficientemente conclusivas. Isto reduz a expectativa de aplicação deste método isoladamente para resolver problemas reais, mas mantém a de incorporação em abordagens baseadas em processos de busca no espaço de soluções viáveis.

5.1 Trabalhos futuros

Este trabalho deixou várias questões em aberto, que requerem estudos específicos e mais aprofundados. Um importante exemplo é a identificação de associações de atributos das instâncias, parâmetros do algoritmo, e aproveitamentos de leiautes. Isto requer um número maior de instâncias para testes, o que provavelmente necessitaria do refinamento do método de cálculo de *NFP* que foi implementado neste trabalho. Para fazer esta identificação, um caminho possível seria selecionar uma grande quantidade de instâncias, caracterizá-las e resolvê-las com todas as combinações de parâmetros possíveis. Em seguida, buscar correlações entre as instâncias e os parâmetros que levaram aos melhores resultados, identificando padrões de associações favoráveis para a boa qualidade dos leiautes. Por fim, as associações identificadas poderiam ser validadas em novas instâncias, confrontando os resultados obtidos pelas combinações sugeridas pelas associações feitas, com os resultados de outras combinações possíveis de parâmetros.

Além disso, o método proposto pode ser usado em abordagens baseadas em processos de busca. Uma possibilidade de nova pesquisa seria desenvolver algoritmos de busca que usem este método construtivo para fornecer soluções iniciais. Alternativamente, o próprio método construtivo poderia ser modificado de modo a explorar retrocessos na sequência de posicionamentos, passando a incluir uma rotina interna de busca.

Finalmente, existe a possibilidade de incluir novos tipos de agrupamentos entre as compactações. Neste trabalho foram consideradas apenas dois tipos de agrupamentos, por unidades na sequência da heurística de seleção, ou por formas congruentes. A divisão

do conjunto de itens da instância pode ser feita a partir de qualquer atributo que caracterize os itens. Isto poderia revelar novos potenciais da abordagem por intercalação de posicionamentos e compactações, e, conseqüentemente, facilitar o aproveitamento desses estudos para os casos reais da indústria.

Apêndice A

Melhores leiautes construídos

- **Instância:** albano
 - **Comprimento:** 10.541,4
 - **Aproveitamento:** 82,6%
 - **Regra:** BL
 - **Opção-regra:** BEST_REF
 - **Ordem-primária:** G_D_L_LEN_REL
 - **Ordem-secundária:** G_S_PERIM
 - **Agrupamento:** 5 un
 - **Número de compactações:** 34
 - **Tempo de execução:** 1,36 s

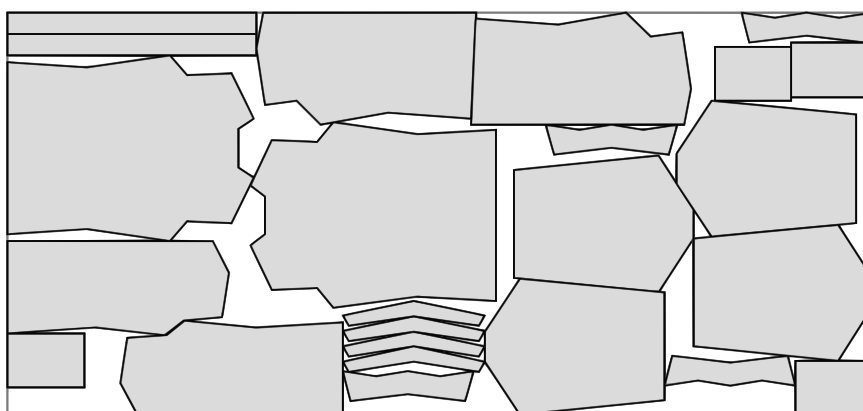


Figura A.1. Melhor leiaute para instância albano.

- **Instância:** blaz2

- **Comprimento:** 209,5
- **Aproveitamento:** 71,9%
- **Regra:** ML
- **Opção-regra:** MIN_LEN
- **Ordem-primária:** G_D_L_LEN_REL
- **Ordem-secundária:** G_S_AREA
- **Agrupamento:** 13 un
- **Número de compactações:** 13
- **Tempo de execução:** 0,63 s

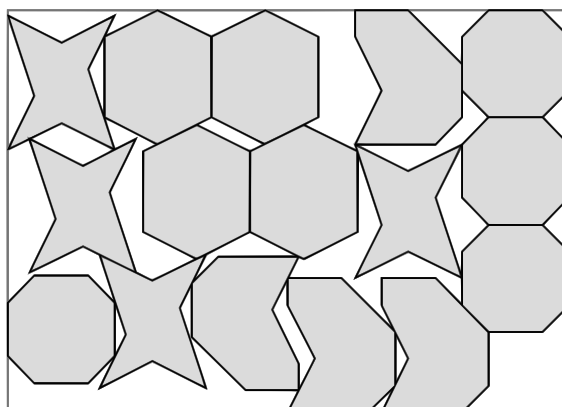


Figura A.2. Melhor leiaute para instância blaz2.

- **Instância:** dagli

- **Comprimento:** 627,4
- **Aproveitamento:** 80,6%
- **Regra:** ML
- **Opção-regra:** BEST_REF
- **Ordem-primária:** G_D_L_LEN_REL
- **Ordem-secundária:** G_S_PERIM
- **Agrupamento:** 14 un
- **Número de compactações:** 20

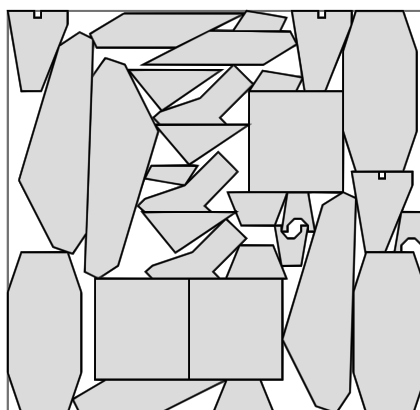


Figura A.3. Melhor leiaute para instância dagli.

– **Tempo de execução:** 1,46 s

• **Instância:** dighe1

- **Comprimento:** 128,3
- **Aproveitamento:** 78,0%
- **Regra:** BL
- **Opção-regra:** BEST_REF
- **Ordem-primária:** G_S_AREA
- **Ordem-secundária:** G_S_LEN
- **Agrupamento:** 5 sh
- **Número de compactações:** 15
- **Tempo de execução:** 0,37 s

• **Instância:** dighe2

- **Comprimento:** 129,7
- **Aproveitamento:** 77,1%
- **Regra:** ML
- **Opção-regra:** BEST_CENT
- **Ordem-primária:** G_S_AGGR
- **Ordem-secundária:** G_S_WID

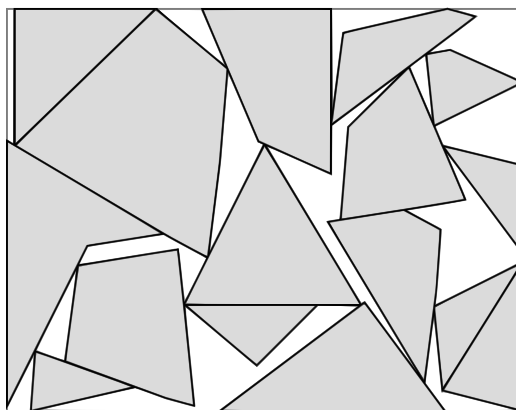


Figura A.4. Melhor leiaute para instância dighe1.

- **Agrupamento:** 1 sh
- **Número de compactações:** 26
- **Tempo de execução:** 0,31 s

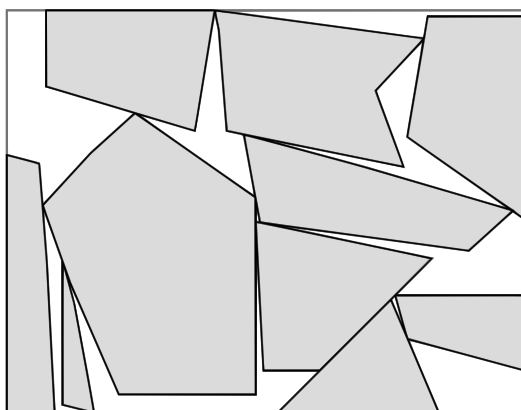


Figura A.5. Melhor leiaute para instância dighe2.

- **Instância:** fu

- **Comprimento:** 34,0
- **Aproveitamento:** 83,8%
- **Regra:** BL
- **Opção-regra:** BEST_REF
- **Ordem-primária:** G_D_L_LEN_REL
- **Ordem-secundária:** G_S_AREA

- **Agrupamento:** 1 un
- **Número de compactações:** 50
- **Tempo de execução:** 0,66 s

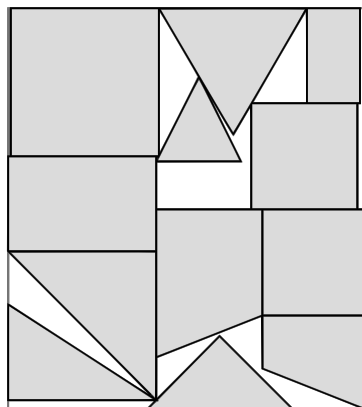


Figura A.6. Melhor leiaute para instância fu.

- **Instância:** mao
 - **Comprimento:** 1.841,0
 - **Aproveitamento:** 80,1%
 - **Regra:** ML
 - **Opção-regra:** BEST_CENT
 - **Ordem-primária:** G_S_PERIM
 - **Ordem-secundária:** G_S_AREA
 - **Agrupamento:** 3 un
 - **Número de compactações:** 36
 - **Tempo de execução:** 2,73 s

- **Instância:** marques
 - **Comprimento:** 816,4
 - **Aproveitamento:** 84,7%
 - **Regra:** BL
 - **Opção-regra:** BEST_REF

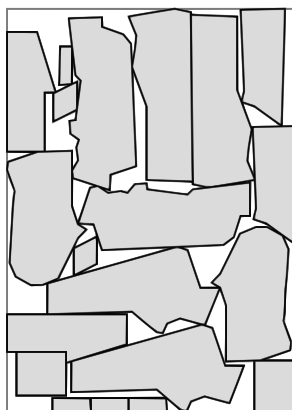


Figura A.7. Melhor leiaute para instância maos.

- **Ordem-primária:** G_D_L_LEN_REL
- **Ordem-secundária:** G_S_WID
- **Agrupamento:** 10 un
- **Número de compactações:** 16
- **Tempo de execução:** 1,59 s

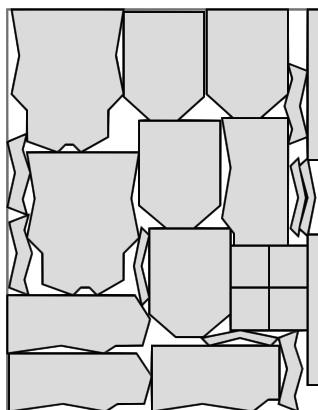


Figura A.8. Melhor leiaute para instância marques.

- **Instância:** shapes0
 - **Comprimento:** 660,0
 - **Aproveitamento:** 60,5%
 - **Regra:** ML
 - **Opção-regra:** BEST_CENT

- **Ordem-primária:** G_S_NONRECT
- **Ordem-secundária:** G_S_AREA
- **Agrupamento:** 2 un
- **Número de compactações:** 89
- **Tempo de execução:** 6,24 s

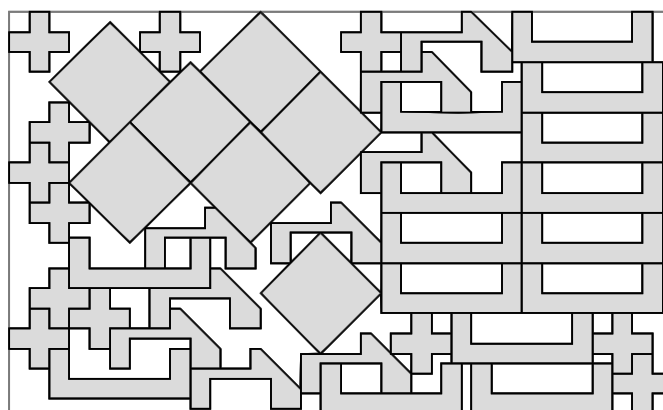


Figura A.9. Melhor leiaute para instância shapes0.

- **Instância:** shapes1
 - **Comprimento:** 615,0
 - **Aproveitamento:** 64,9%
 - **Regra:** ML
 - **Opção-regra:** BEST_REF
 - **Ordem-primária:** G_S_AREA
 - **Ordem-secundária:** G_S_PERIM
 - **Agrupamento:** 5 un
 - **Número de compactações:** 39
 - **Tempo de execução:** 4,04 s

- **Instância:** shapes2
 - **Comprimento:** 272,0
 - **Aproveitamento:** 79,4%

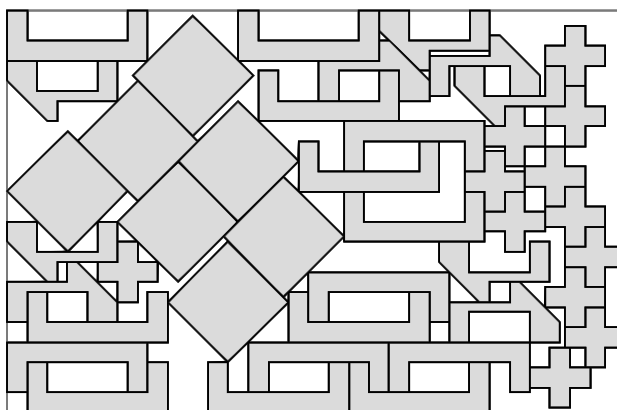


Figura A.10. Melhor leiaute para instância shapes1.

- **Regra:** BL
- **Opção-regra:** BEST_REF
- **Ordem-primária:** G_S_LEN
- **Ordem-secundária:** G_S_WID
- **Agrupamento:** 11 un
- **Número de compactações:** 19
- **Tempo de execução:** 5,77 s

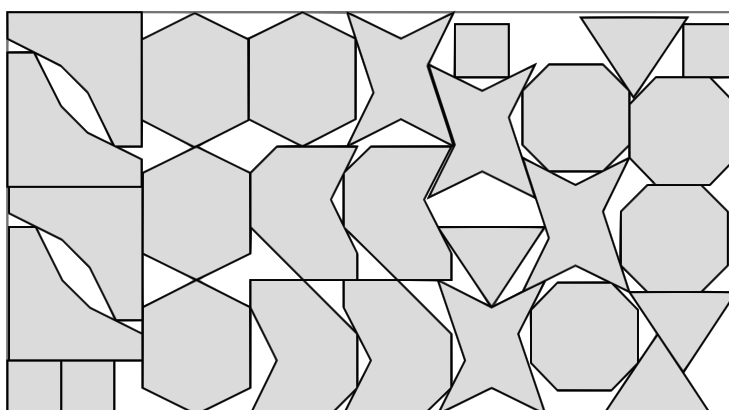


Figura A.11. Melhor leiaute para instância shapes2.

- **Instância:** shirts
 - **Comprimento:** 620,6
 - **Aproveitamento:** 87,0%

- **Regra:** BL
- **Opção-regra:** BEST_REF
- **Ordem-primária:** G_S_IRREG
- **Ordem-secundária:** G_S_LEN
- **Agrupamento:** 3 sh
- **Número de compactações:** 19
- **Tempo de execução:** 5,77 s

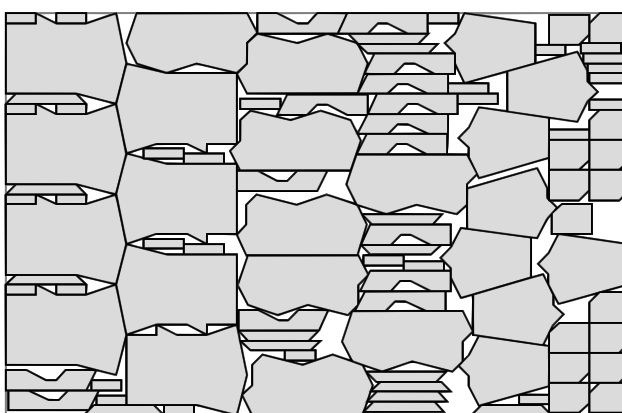


Figura A.12. Melhor leiaute para instância shirts.

- **Instância:** swim
 - **Comprimento:** 6.378,4
 - **Aproveitamento:** 69,4%
 - **Regra:** BL
 - **Opção-regra:** BEST_REF
 - **Ordem-primária:** G_S_PERIM
 - **Ordem-secundária:** G_S_LEN
 - **Agrupamento:** 10 un
 - **Número de compactações:** 23
 - **Tempo de execução:** 22,29 s

- **Instância:** trousers

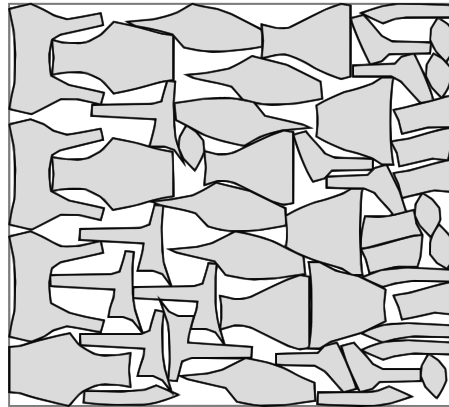


Figura A.13. Melhor leiaute para instância swim.

- **Comprimento:** 2.483,0
- **Aproveitamento:** 87,7%
- **Regra:** BL
- **Opção-regra:** BEST_REF
- **Ordem-primária:** G_S_PERIM
- **Ordem-secundária:** G_S_AREA
- **Agrupamento:** 2 un
- **Número de compactações:** 151
- **Tempo de execução:** 27,61 s

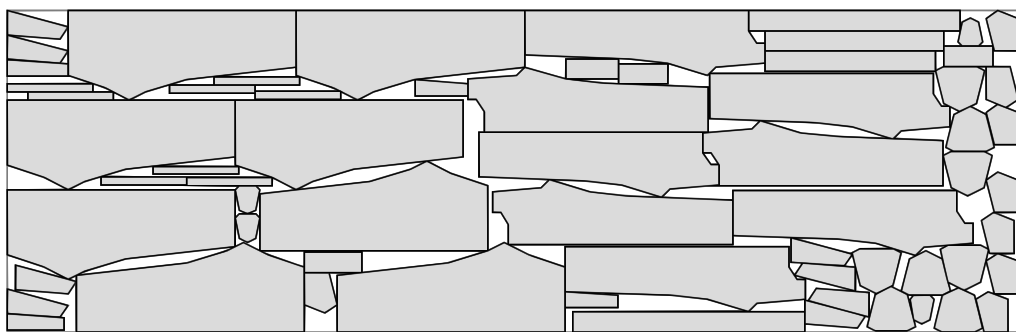


Figura A.14. Melhor leiaute para instância trousers.

Referências Bibliográficas

- Agarwal, P. K.; Flato, E. & Halperin, D. (2002). Polygon decomposition for efficient construction of minkowski sums. *Computational Geometry*, 21(1-2):39 – 61. Sixteenth European Workshop on Computational Geometry - EUROCG-2000.
- Albano, A. & Sapuppo, G. (1980). Optimal allocation of two-dimensional irregular shapes using heuristic search methods. *Systems, Man and Cybernetics, IEEE Transactions on*, 10(5):242–248.
- Art, R. (1966). An approach to the two dimensional irregular cutting stock problem. Technical Report 320-2006, IBM Cambridge Scientific Center, Cambridge, MA.
- Babu, A. & Babu, N. (2001). A generic approach for nesting of 2-d parts in 2-d sheets using genetic and heuristic algorithms. *Computer-Aided Design*, (33):879–891.
- Bennell, J. & Song, X. (2008). A comprehensive and robust procedure for obtaining the nofit polygon using Minkowski sums. *Computers & Operations Research*, (June):1–37.
- Bennell, J. & Song, X. (2010). A beam search implementation for the irregular shape packing problem. *Journal of Heuristics*, 16:167–188.
- Bennell, J. A.; Dowsland, K. A. & Dowsland, W. B. (2001). The irregular cutting-stock problem - a new procedure for deriving the no-fit polygon. *Comput. Oper. Res.*, 28(3):271–287.
- Bennell, J. A. & Oliveira, J. F. (2008). The geometry of nesting problems: A tutorial. *European Journal of Operational Research*, 184(2):397–415.
- Berkelaar, M.; Eikland, K. & Notebaert, P. (2010). Ipsolve v5. 5.2.0: Open source (mixed-integer) linear programming system.
- Burke, E.; Hellier, R.; Kendall, G. & Whitwell, G. (2006). A New Bottom-Left-Fill Heuristic Algorithm for the Two-Dimensional Irregular Packing Problem. *Operations Research*, 54(3):587–601.

- Burke, E.; Hellier, R.; Kendall, G. & Whitwell, G. (2007). Complete and robust no-fit polygon generation for the irregular stock cutting problem. *European Journal of Operational Research*, 179(1):27–49.
- Burke, E. K. & Kendall, G. (1999). Implementation and performance improvement of the evaluation of a two dimensional bin packing problem using the no fit polygon. (ASAP99001).
- Dowsland, K. & Dowsland, W. (1995). Solution approaches to irregular nesting problems. *European Journal of Operational Research*, (84):506–521.
- Dowsland, K.; Vaid, S. & Dowsland, W. (2002). An algorithm for polygon placement using a bottom-left strategy. *European Journal of Operational Research*, 141(2):371–381.
- Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2):145–159.
- Egeblad, J.; Nielsen, B. K. & Odgaard, A. (2007). Fast neighborhood search for two- and three-dimensional nesting problems. *European Journal of Operational Research*, 183(3):1249–1266.
- Ghosh, P. (1990). A Solution of Polygon Containment, Spatial Planning and Other Related Problems Using Minkowski Operations. *Computer Vision, Graphics, and Image Processing*, 49:1–35.
- Gomes, A. (2005). *Abordagens Heurísticas ao Posicionamento de Formas Irregulares*. PhD thesis, Universidade do Porto.
- Gomes, A. & Oliveira, J. (2001). A GRASP approach to the nesting problem. *Proceedings of MIC*, pp. 47–52.
- Gomes, A. & Oliveira, J. (2006). Solving Irregular Strip Packing problems by hybridising simulated annealing and linear programming. *European Journal of Operational Research*, 171(3):811–829.
- Haessler, R. & Sweeney, P. (1991). Cutting stock problems and solution procedures. *European Journal of Operational Research*, 54(2):141–150.
- Hopper, E. (2000). *Two-dimensional packing utilising evolutionary algorithms and other meta-heuristic methods*. PhD thesis, University of Wales, Cardiff.

- Imamichi, T.; Yagiura, M. & Nagamochi, H. (2009). An iterated local search algorithm based on nonlinear programming for the irregular strip packing problem. *Discrete Optimization*, 6(4):345 – 361.
- Jacquenot, G.; Bennis, F.; Maisonneuve, J. & Wenger, P. (2009). 2D multi-objective placement algorithm for free-form components. *Arxiv preprint arXiv:0911.5657*.
- Jain, S. & Gea, H. C. (1998). Two-dimensional packing problems using genetic algorithms. *Engineering with Computers*.
- Kendall, G. (2000). *Applying meta-heuristic algorithms to the nesting problem utilising the no fit polygon*. Number October.
- Konopasek, M. (1981). Mathematical treatments of some apparel marking and cutting problems. *U.S. Department of Commerce Report*, (99-26-90857-10).
- Li, Z. & Milenkovic, V. (1995a). Compaction and separation algorithms for non-convex polygons and their applications. *European Journal of Operational Research*, pp. 1–20.
- Li, Z. & Milenkovic, V. (1995b). Compaction and separation algorithms for non-convex polygons and their applications. *European Journal of Operational Research*, pp. 539–561.
- Luzzi, I. (2002). *Exact and heuristic methods for nesting problems*. PhD thesis, University of Padova.
- Mahadevan, A. (1984). *Optimisation in computer aided pattern packing*. PhD thesis, North Carolina State University.
- Martins, T. (2007). *Estudo do Recozimento Simulado e do Polígono de Obstrução Aplicados ao Problema de Empacotamento Rotacional de Polígonos Irregulares Não-Convexos em Recipientes Fechados*. PhD thesis, Escola Politécnica da Universidade de São Paulo.
- Nielsen, B. K. (2007). An efficient solution method for relaxed variants of the nesting problem. Em *Proceedings of the thirteenth Australasian symposium on Theory of computing - Volume 65, CATS '07*, pp. 123–130, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.
- Nielsen, B. K. & Odgaard, A. (2003). Fast neighborhood search for the nesting problem. *Sci. Report, University of Copenhagen*.

- Oliveira, J. & Ferreira, J. (1993). Algorithms for nesting problems, applied simulated annealing. *Vidal, R.V.V. (Ed.), Lecture Notes in Econ. and Maths Systems*, (396):255–274.
- Oliveira, J.; Gomes, A. & Ferreira, J. (2000). TOPOS-A new constructive algorithm for nesting problems. *OR Spectrum*, 22(2):263–284.
- Russell, S.; Norvig, P.; Canny, J. & Bratko, I. (2005). *Artificial Intelligence: A Modern Approach*. Pearson Education, Limited.
- Sato, K. (2011). Proposta de Algoritmo para a Determinação da Região Livre de Colisão e sua Aplicação na Solução de Leiautes Bidimensionais Irregulares com Recozimento Simulado. Master's thesis, Universidade de São Paulo.
- Segenreich, S. & Braga, L. (1986). Optimal nesting of general plane figures: a monte carlo heuristical approach. *Computers & Graphics*, (10):229–237.
- Wascher, G.; Hausner, H. & Schumann, H. (2007). An improved typology of cutting and packing problems. *European Journal of Operational Research*, 183(3):1109–1130.
- Whelan, P. & Batchelor, B. (1996). Automated packing systems-a systems engineering approach. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 26(5):533–544.