

Thiago de Castro Martins

Estudo do Recozimento Simulado e do
Polígono de Obstrução Aplicados ao
Problema de Empacotamento Rotacional
de Polígonos Irregulares Não-Convexos
em Recipientes Fechados

Tese apresentada à Escola Politécnica
da Universidade de São Paulo como
requisito para obtenção do Título de
Doutor em Engenharia.

Thiago de Castro Martins

Estudo do Recozimento Simulado e do
Polígono de Obstrução Aplicados ao
Problema de Empacotamento Rotacional
de Polígonos Irregulares Não-Convexos
em Recipientes Fechados

Tese apresentada à Escola Politécnica
da Universidade de São Paulo como
requisito para obtenção do Título de
Doutor em Engenharia.

Área de concentração:
Engenharia Mecânica

Orientador:
Prof. Dr. Marcos de Sales Guerra
Tsuzuki

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com anuência de seu orientador. São Paulo, 2 de Maio de 2007.

Assinatura do autor.

Assinatura do orientador.

Ficha Catalográfica

Martins, Thiago de Castro

Estudo do Recozimento Simulado e do Polígono de Obstrução Aplicados ao Problema de Empacotamento Rotacional de Polígonos Irregulares Não-Convexos em Recipientes Fechados. São Paulo, 2007. 103 p.

Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Mecatrônica e de Sistemas Mecânicos.

1. Otimização. 2. Empacotamento. 3. Recozimento Simulado. I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Mecatrônica e de Sistemas Mecânicos. II.

a Mel e Luana
espíritos livres

Agradecimentos

Agradeço

a minha família, pelo suporte e inspiração,

a Marcos de Sales Guerra Tsuzuki, por toda a dedicação

a Fabiano Silva e a Braspelco pelo interesse e abertura

a Marco Antonio Poli Jr., pela sugestão do tema

a Vinicius Rodrigues de Moraes, por ter voltado à guerra, a dois minutos para a meia-noite

Resumo

Este trabalho trata da proposta de um processo de otimização para o problema do posicionamento rotacional e translacional de formas irregulares em recipientes de dimensões fixas baseado em heurísticas probabilísticas sem o uso de penalização externa. Para tanto, é empregado o *polígono de obstrução*, acoplado a uma heurística baseada no *Recozimento Simulado*. O comportamento discreto da função custo em problemas com recipientes de dimensões limitadas foi mitigado através de uma heurística de “desempate”, que busca diferenciar soluções com valores idênticos através de uma estimativa de quão próxima está uma determinada solução de conseguir encaixar uma forma não-encaixada em seu leiaute. A comparação de resultados deste trabalho com resultados publicados na literatura comprova a validade da abordagem aqui adotada.

Abstract

This work deals with the proposal of an optimization process for the packing problem with free translations and rotations of irregular shapes on containers with limited dimensions based on probabilistic heuristics without use of external penalty techniques. For such, the *no-fit polygon* is used, coupled with an heuristic based on *Simulated Annealing*. The discrete behavior of the objective function in problems with limited containers is mitigated by a “tie breaker” heuristic that sorts solutions with identical values by estimating how close a given solution is of fitting an unplaced shape on its layout. The comparison of these work’s results with results published on the literature validates the approach here adopted.

Conteúdo

Lista de Figuras

1	Introdução aos problemas de posicionamento	16
1.1	Introdução	16
1.2	Revisão de trabalhos com problemas de posicionamento	18
1.3	Conclusões	23
2	O problema de posicionamento translacional e rotacional em recipientes com dimensões fixas	24
2.1	Introdução	24
2.2	Motivação do estudo	25
2.3	Definição do problema	26
2.4	Complexidade	27
2.5	Características da função objetivo e dualidade	27
2.6	A penalização externa	28
2.7	Conclusões	30
3	O Polígono de obstrução e a decomposição do problema de posicionamento	32
3.1	Introdução	32
3.2	Polígonos de obstrução e regiões livres de colisão	32
3.3	Decomposição do macro-problema de posicionamento	35
3.3.1	Heurística para a ordem de posicionamento	37
3.3.2	Heurística para a rotação das formas	39
3.3.3	Heurística para a translação das formas	40

3.4	Conclusões	40
4	Aplicando o recozimento simulado ao problema do posicionamento	42
4.1	Introdução	42
4.2	O recozimento simulado	43
4.2.1	Diretrizes na aplicação de recozimento simulado	44
4.3	Heurísticas probabilísticas completas	45
4.4	Heurísticas para os sub-problemas do posicionamento	45
4.4.1	Ordem de posicionamento	46
4.4.2	Rotação das formas	46
4.4.3	Translação das formas	47
4.5	Geração de configurações de posicionamento	50
4.5.1	Fator de cristalização	52
4.6	Avaliação da função objetivo a partir de parâmetros de posicionamento	54
4.6.1	Desempate de soluções com custo idêntico	55
4.7	Conclusões	56
5	Implementação do processo	58
5.1	Introdução	58
5.2	Operações geométricas sobre polígonos	60
5.2.1	Representação computacional de polígonos	60
5.2.2	Rotação	64
5.2.3	Algoritmo de soma de Minkowski para polígonos convexos	65
5.3	Armazenamento e reuso de resultados geométricos intermediários	68
5.3.1	Polígonos de Obstrução	69
5.3.2	Regiões livres de colisão	71
5.4	Conclusões	74

6	Resultados	76
6.1	Introdução	76
6.2	Implementação	76
6.3	Visualização do processo de otimização	76
6.4	Problemas propostos	77
6.4.1	Pequeno Puzzle	78
6.4.2	Tangram	79
6.4.3	Recipientes irregular	79
6.4.4	Recipiente não-convexo	80
6.4.5	Recipiente com furo	80
6.5	Avaliação da influência de parâmetros do processo	81
6.5.1	Influência nos resultados do parâmetro de resfriamento geométrico α	81
6.5.2	Influência do Desempate de soluções com custo idêntico	82
6.6	Problemas adaptados da literatura	82
6.6.1	Problema <i>Albano</i>	83
6.6.2	Problema <i>Jackobs1</i>	84
6.6.3	Problema <i>Jackobs2</i>	84
6.6.4	Problema <i>Fu</i>	88
6.6.5	Problema <i>shapes0</i>	89
6.6.6	Problema <i>dagli</i>	89
6.7	Conclusões	89
7	Conclusões e considerações finais	93
	Apêndice A – Recozimento Simulado	96
A.1	Descrição	96
A.2	Algoritmo	97
A.3	Programas de resfriamento	98

A.3.1	Temperatura Inicial	99
-------	-------------------------------	----

Referências		101
--------------------	--	------------

Lista de Figuras

1.1	Duas possíveis apresentações para um problema de otimização de posicionamento.	17
1.2	O polígono de obstrução $A - B$ que representa a área obstruída pelo polígono A no posicionamento do polígono B	20
1.3	Posicionamento gerado pela repetição de um padrão.	20
2.1	Um problema de empacotamento e sua solução ótima.	26
2.2	A variação da função objetivo no problema dual.	28
2.3	A variação da função objetivo no problema primal.	29
2.4	(a) Um problema de posicionamento relaxado com uma solução ótima inválida.	29
3.1	Um conjunto de translações de uma forma representado como a região formada por todas as possíveis translações de seu ponto de referência.	33
3.2	A região livre de obstrução (hachuradas) para o posicionamento de uma forma em um recipiente com obstáculos.	34
3.3	A região de posicionamento interno (<i>inner-fit polygon</i>) para uma dada forma e seu recipiente.	34
3.4	Exemplo de geração de polígono de obstrução para polígonos convexos.	36
3.5	Problema cuja solução ótima não pode ser alcançada por uma heurística <i>larger first</i>	39
3.6	Um problema de posicionamento rotacional.	39
3.7	Exemplo simples de problema cuja solução ótima não pode ser alcançada por uma heurística <i>bottom-left</i>	40
4.1	Problema resolvido com ordem gerada por recozimento simulado.	46

4.2	Um problema de posicionamento rotacional resolvido com recozimento simulado aplicado somente ao sub-problema de rotação. . .	47
4.3	Uma solução válida e seu grafo de conectividade.	48
4.4	Exemplo de solução conexa e não-conexa.	49
4.5	Exemplo de posicionamento de uma forma durante uma etapa do processo de otimização.	51
4.6	Avaliação da função objetivo do posicionamento.	55
5.1	Forma não-poligonal aproximada por um polígono.	59
5.2	Polígono e uma seqüência de pontos que o representa.	61
5.3	Sobreposição e intersecção de dois polígonos.	61
5.4	Intersecção de dois segmentos com coordenadas inteiras produzindo um ponto com coordenadas não-inteiras.	62
5.5	Aproximação com coordenadas inteiras da intersecção de dois segmentos.	62
5.6	Algoritmo para rotação aproximada de polígonos.	65
5.7	Ordenação de dois vetores. O vetor e_1 está a <i>direita</i> de e_2	66
5.8	Algoritmo para classificação de dois vetores.	67
5.9	Algoritmo para construção do polígono de obstrução.	68
6.1	Exemplo de uma seqüência de espectros de valores da função objetivo e temperaturas para o recozimento simulado.	77
6.2	Solução final de um pequeno puzzle com quatro polígonos.	78
6.3	Espectro de função objetivo para o pequeno puzzle.	78
6.4	Solução ótima de um tangran com sete formas.	79
6.5	Solução “alternativa” para o <i>tangran</i> , obtida mediante um aumento de 5% na folga do recipiente.	79
6.6	Espectros de função objetivo para o tangran.	80
6.7	Solução ótima de puzzle com 24 formas com recipiente irregular. .	80
6.8	Espectros de função objetivo para o puzzle de recipiente irregular.	81
6.9	Puzzle com recipiente não-convexo.	81

6.10	Puzzle com recipiente furado.	82
6.11	Espectros de função objetivo para um “resfriamento rápido” do problema tangran.	83
6.12	Espectros de energia para a função objetivo original	83
6.13	Espectros de energia com a função objetivo modificada por um fator de escala encontrado com uma busca de profundidade 8.	84
6.14	(a) Resultado obtido por Gomes e Oliveira ^[1] para o problema chamado <i>Albano</i> . (b) Resultado obtido pelo processo de otimização proposto neste trabalho.	85
6.15	Espectros de função-objetivo para o problema mostrado na figura 6.14 (<i>Albano</i>).	85
6.16	(a) Resultado obtido por Gomes e Oliveira ^[1] para o problema <i>Jackobs1</i> . (b), (c) e (d) Resultados obtidos pelo processo proposto.	86
6.17	Espectros de função objetivo para o problema mostrado na figura 6.16 (<i>Jackobs1</i>).	86
6.18	(a) Resultados obtidos por Gomes e Oliveira ^[1] para o problema <i>Jackobs2</i> . (b) e (c) Resultados obtidos neste trabalho.	87
6.19	Espectros de função objetivo para o problema da figura 6.18 (<i>Jackobs2</i>).	87
6.20	(a) Resultados obtidos por Gomes e Oliveira ^[1] para o problema <i>Fu</i> . (b) and (c) Resultados obtidos neste trabalho.	88
6.21	Espectros de valores para a função objetivo do problema da figura 6.20 (<i>Fu</i>).	88
6.22	(a) Resultado obtido por Gomes e Oliveira ^[1] para o problema <i>shapes0</i> . (b) Resultado obtido neste trabalho.	90
6.23	Espectros de valores para função objetivo do problema da figura 6.22 (<i>shapes0</i>).	91
6.24	(a) Resultado obtido por Gomes e Oliveira ^[1] para o problema <i>dagli</i> . (b) Melhor resultado obtido neste trabalho.	91
6.25	Espectros de valores para função objetivo do problema da figura 6.24 (<i>dagli</i>).	92
A.1	Distribuição de Boltzman.	97

A.2 Algoritmo de recozimento simulado.	100
--	-----

1 Introdução aos problemas de posicionamento

1.1 Introdução

A diminuição de custos é procurada por todos os ramos da indústria moderna. Algumas vezes em busca da competitividade, outras buscando aumento da margem de lucros, o controle de custos recebe atenção especial da alta direção.

Dentre as diversas formas de se encontrar economia de custos, uma das mais importantes consiste na diminuição dos custos variáveis de produção, aqueles que têm relação direta com a quantidade produzida. Um dos exemplos mais diretos dos custos variáveis é a matéria-prima.

Em diversos ramos da indústria a redução de custos pode vir através da otimização de uso da matéria-prima. As indústrias de couro, tecidos, chapas e folhas de metal, estampagem, entre outras, são exemplos de ramos industriais em que a otimização do posicionamento das formas para corte na matéria-prima é fundamental para a otimização da fabricação.

A otimização do posicionamento consiste no problema de se encontrar o posicionamento ótimo das formas a serem fabricadas sobre folhas de matéria-prima. Esse posicionamento é ótimo quando se obtém a máxima área utilizada de matéria-prima, resultando na mínima área desperdiçada de material, tendo todos os limites das formas respeitados, isto é, sem sobreposição.

Exemplos como o da indústria do couro tornam o problema ainda mais complicado, dado que a matéria-prima, o couro, pode possuir irregularidades que não podem existir nos produtos, como rasgos e furos. Estes então devem ser isolados e retirados do recipiente, que é a região do espaço que será utilizado para o posicionamento dos objetos.

Na indústria têxtil, assim como na estampagem, a área de tecido (ou metal) utilizada para se fabricar peças é o que se procura otimizar: quanto menor a área

de material desperdiçada, menor será o desperdício de matéria-prima.

Os problemas de otimização de posicionamento se apresentam sob duas grandes categorias:

- Posicionamento de uma quantidade variável de formas em um recipiente de dimensões fixas.
- Posicionamento de uma quantidade fixa de formas em um recipiente de dimensões variáveis.

Um exemplo da primeira categoria é o problema de empacotamento, no qual o maior número de objetos deve ser acondicionado em um recipiente de dimensões fixas.

Um exemplo da segunda categoria é o já citado planejamento de corte de material da indústria têxtil, no qual o menor comprimento possível de uma faixa de tecido deve ser empregado para se obter uma determinada demanda de peças.

Esta classificação está diretamente relacionada ao conceito de dualidade em pesquisa operacional. Arbitrariamente, neste trabalho, os problemas de posicionamento em recipientes com dimensões fixas serão denominados problemas *primais*, enquanto que os problemas de posicionamento de recipientes de dimensões variáveis serão denominados problemas *duais* (vide figura 1.1).

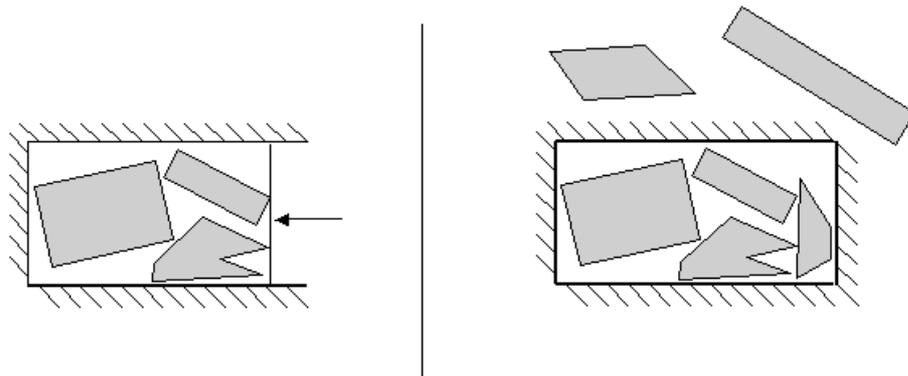


Figura 1.1: Duas possíveis apresentações para um problema de otimização de posicionamento.

Uma outra categoria de problemas de posicionamento relacionada à otimização é a categoria dos problemas de *decisão*. Nesta categoria, deseja-se determinar se uma dada meta de desempenho é atingível para um determinado problema de posicionamento. Por exemplo, o problema de decisão relacionado ao supra-citado problema da indústria têxtil é o problema de se determinar se é possível

produzir uma determinada demanda de peças utilizando somente uma quantidade pré-determinada de tecido. Nota-se que se existir um algoritmo eficiente para resolver um problema de decisão ligado a um problema de otimização, este mesmo pode ser empregado para resolver por dicotomia o problema de otimização original. Por outro lado, um algoritmo que resolva de forma ótima o problema de otimização pode ser empregado para resolver o problema de decisão relacionado. Para tanto, basta verificar se a solução ótima obtida pelo algoritmo de otimização é melhor ou igual à proposta pelo problema de decisão.

Problemas de decisão têm grande importância na teoria da complexidade (vide [2]). O estudo de complexidade em problemas de decisão de posicionamento mostram que mesmo versões restritas desses problemas são classificadas como NP-Difíceis. Um problema pertencente à classe de problemas NP-Difícil é um problema cuja solução pode ser empregada para resolver *todos* os problemas da classe NP. Na prática, isso implica que ainda não há solução algorítmica eficiente para resolver problemas de posicionamento. Como será visto na seção 1.2, isso se reflete na ampla adoção de heurísticas para abordar estes problemas.

1.2 Revisão de trabalhos com problemas de posicionamento

Os problemas de posicionamento sempre atraíram muita atenção na literatura, chegando a motivar edições inteiras de periódicos voltados para a pesquisa operacional. A grande diversidade de problemas e de abordagens faz com que se tenha disponível hoje uma grande quantidade de material original sobre o tema. Uma revisão recente sobre o trabalho na área pode ser encontrada em [3].

Dyckhoff^[4] apresenta uma proposta de classificação abrangente dos problemas de posicionamento. Esta classificação foi posteriormente revisada e estendida por Wäscher *et al.* em [5].

Em seu trabalho Wäscher *et al.* classificam os problemas de acordo com as seguintes características:

- Quanto à dimensão do problema: uma, duas, três ou mais dimensões.
- Quanto ao objetivo do problema: maximização de saída (por exemplo, quantidade de peças produzidas) ou minimização de entrada (por exemplo, quantidade de matéria-prima bruta). Esta classificação está diretamente ligada à caracterização de problemas primais e duais apresentadas na seção

1.1.

- Quanto aos recipientes: número de recipientes, um ou vários, de dimensões fixas ou com uma dimensão variável.
- Quanto à variedade de geometria das formas: as geometrias das formas no conjunto são congruentes, fracamente heterogêneos ou fortemente heterogêneos.
- Quanto às restrições de geometria de formas: as formas são geometricamente simples (retangulares ou circulares) ou geometricamente irregulares.

Neste trabalho, Wäscher *et al.* apontam que trabalhos que tratam de problemas de posicionamento com formas irregulares são escassos na literatura especializada, particularmente os que envolvem maximização de saída.

Do mesmo modo, a classe de problemas de posicionamento bidimensional para a qual se tem a maior quantidade de referências disponível é o problema de posicionamento de formas regulares, particularmente os envolvendo minimização de entrada. Esta opinião também é compartilhada por outros autores ^[4, 6].

Embora estudados em menor quantidade, problemas de posicionamento de formas irregulares também são contemplados em diversos estudos na literatura. Em ^[7], Dowsland e Dowsland fazem uma revisão do progresso no tratamento computacional destes problemas, ressaltando que a indústria na grande maioria dos casos recorre a especialistas humanos para abordá-los.

Na abordagem deste tipo de problema, um conceito muitas vezes empregado é o de polígono de obstrução (*no-fit polygon*) (vide figura 1.2) , cujo conceito foi introduzido por Adamowicz e Albano em ^[8]. O polígono de obstrução representa a área obstruída por uma forma no posicionamento de outra. Ele pode ser gerado a partir da trajetória de um ponto de referência da forma a ser posicionada, quando esta “desliza” ao longo da forma obstrutora. Vários trabalhos sobre posicionamento bidimensional empregam o conceito de polígono de obstrução (inclusive este), como o de Li e Milenkovic ^[9, 10] e o de Burke ^[11], que tratam do problema de posicionamento exclusivamente translacional.

No contexto de um projeto temático da Textile/Clothing Technology Corporation e do Harvard Center for Textile and Apparel Research, Milenkovic *et al.* produziram uma grande quantidade de material sobre posicionamento de formas irregulares com minimização de entrada (problemas duais) com aplicações para a indústria têxtil. Como parte dessa produção, podemos mencionar ^[12], que trata

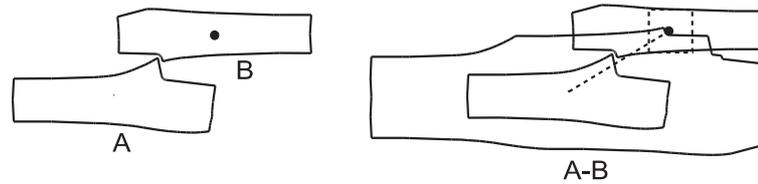


Figura 1.2: O polígono de obstrução $A - B$ que representa a área obstruída pelo polígono A no posicionamento do polígono B .

do problema da compactação local de leiautes de corte de tecido para melhorar leiautes produzidos manualmente. Este trabalho é expandido em ^[13] onde é feita uma abordagem algorítmica para o posicionamento rotacional otimizado de um número pequeno de formas (a abordagem adotada é prática para até três formas). Neste mesmo contexto, Daniels ^[14] aplicou uma abordagem algorítmica ao problema da decisão, resolvendo o problema de determinar se um conjunto de formas pode ou não ser posicionado em um recipiente. Mais uma vez, esta abordagem é prática apenas para um pequeno número de formas.

O fato de abordagens puramente algorítmicas encontrarem dificuldades para tratar um número grande de formas não é surpreendente, visto que, como mencionado na seção 1.1, estudos de complexidade ^[14, 15] mostram que problemas de posicionamento tendem a ser NP-Difícil. Assim, abordagens algorítmicas têm custo computacional que cresce exponencialmente com o tamanho dos problemas, e são conseqüentemente práticas apenas para problemas de tamanho restrito.

Um exemplo de abordagem algorítmica do problema de posicionamento simplificado é o da geração de um padrão periódico de posicionamento de formas com máxima compactação (vide figura 1.3). Pesquisas sobre a versão translacional deste problema podem ser encontradas em ^[16] e ^[17].

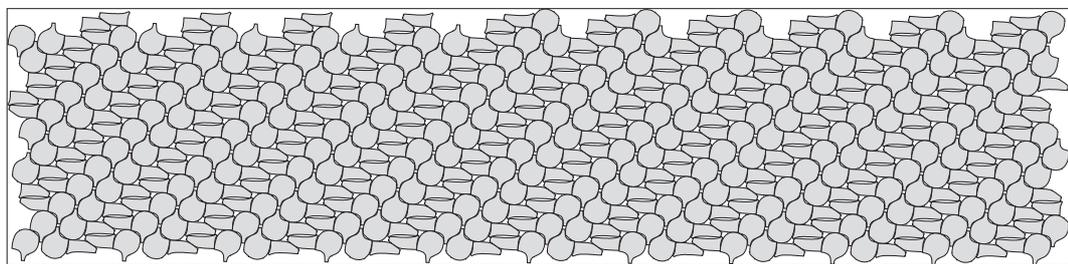


Figura 1.3: Posicionamento gerado pela repetição de um padrão.

Uma outra maneira de se contornar as limitações de complexidade dos problemas de posicionamento é a de se adotar algoritmos de aproximação. Algoritmos de aproximação buscam soluções para problemas de otimização que se mantenham dentro de certos limites de desempenho, quando comparadas às soluções

ótimas. Flávio Miyazawa e Yoshiko Wakabayashi ^[18] pesquisaram algoritmos de aproximação para o posicionamento translacional de formas ortogonais em uma, duas e três dimensões.

Algoritmos de aproximação ainda se restringem a técnicas para as quais é possível provar matematicamente limites inferiores de desempenho para o pior caso. Esta restrição na prática mostra-se irrelevante para a indústria, que se interessa mais por dados de desempenho de casos práticos e na sua comparação com posicionamentos gerados manualmente. Assim, abordagens heurísticas para problemas de posicionamento têm se mostrado bastante atraentes.

Abordagens heurísticas envolvem regras de posicionamento empíricas, que muitas vezes mimetizam o comportamento de especialistas humanos na geração de leiautes. As regras de posicionamento podem ser determinísticas ou probabilísticas. Heurísticas probabilísticas tendem a ser mais versáteis e geralmente obtêm melhores resultados. Por outro lado, estas possuem um comportamento menos previsível que as suas contrapartes determinísticas.

Um tipo de abordagem heurística determinística para problemas puramente translacionais é a *bottom-left*, na qual as formas são ordenadas de acordo com algum critério (geralmente em ordem decrescente de tamanho) e posicionadas em seqüência no recipiente, na posição mais abaixo-à-esquerda disponível. Trabalhos com este tipo de heurística podem ser encontrados em ^[19] que trata do posicionamento de retângulos, e em ^[20], que trata do posicionamento de formas irregulares.

Abordagens heurísticas probabilísticas geralmente são baseadas em meta-heurísticas probabilísticas de otimização. As meta-heurísticas mais comuns são:

- Busca *tabu*.
- Algoritmos Genéticos.
- Recozimento Simulado.

Um dos primeiros trabalhos a aplicar recozimento simulado ao problema do posicionamento é o de Dowsland em ^[21], que abordou o posicionamento exclusivamente translacional de retângulos e formas irregulares. Neste trabalho, Dowsland permite a sobreposição temporária de formas enquanto o processo de otimização ocorre. A sobreposição de formas causa uma correção na função objetivo a ser otimizada, num processo conhecido classicamente como penalização externa.

O problema da geração de padrões repetitivos de empacotamento de formas idênticas com rotação foi tratado com recozimento simulado e penalização externa por Jain *et al.* em [22].

Heckman e Lengauer tratam o posicionamento exclusivamente translacional de formas irregulares heterogêneas com recozimento simulado e penalização externa em [23]. Neste trabalho, Heckman e Lengauer introduzem uma heurística para limitar o movimento de certas formas quando estes movimentos não contribuem mais para a evolução do processo de otimização.

Trabalhos com heurísticas probabilísticas que não empregam penalização externa ainda são minoria na literatura, mas técnicas de otimização híbridas, que combinam heurísticas probabilísticas sem penalização externa com heurísticas determinísticas e algoritmos de otimização local vêm sendo estudadas com interesse crescente, sobretudo em problemas de leiaute translacional.

Um dos primeiros trabalhos a seguir esta linha é o de Burke e Kendall [11], no qual os autores tratam o posicionamento exclusivamente translacional com uma heurística derivada de *bottom-left* governada pelo recozimento simulado.

Na mesma linha, podemos citar Jakobs [24] que usou um processo híbrido combinando heurísticas *bottom-left* com algoritmos genéticos para determinar a seqüência de posicionamento de elementos em problemas com recipiente de dimensões variáveis. No seu trabalho, as formas dos elementos são aproximadas por retângulos, e a determinação de rotações é feita com uma busca local.

Ainda nessa campo, é digno de nota o trabalho de M'Hallah *et al.*, que emprega algoritmos genéticos para determinar a ordem de um posicionamento *bottom-left* de formas irregulares em [25]. Sua técnica foi aplicada em problemas de recipientes com dimensões abertas (problema dual). Uma variação deste trabalho foi feita por Hiffi e Hallah em [26] em que a estratégia híbrida é aplicada a problemas com recipientes de formas variáveis. De acordo com a estratégia adotada, a posição dos elementos é determinada de forma seqüencial por meio de uma heurística determinística, enquanto que a ordem de posicionamento é produzida por algoritmos genéticos. Neste trabalho, um algoritmo de compactação local determinístico é aplicado ao leiaute gerado em cada iteração.

Gomes e Oliveira tratam o problema de posicionamento translacional dual combinando heurísticas probabilísticas e determinísticas. Em [27], é adotada uma heurística *bottom-left* para posicionar uma seqüência de formas, enquanto que a ordem deste posicionamento é obtida através de heurísticas determinísticas e

probabilísticas. Em ^[1], Gomes e Oliveira apresentada uma técnica híbrida, na qual o recozimento simulado é empregado para trocar a posição de formas em um dado leiaute. Um algoritmo de separação, baseado em programação linear, é empregado em seguida para remover eventuais sobreposições, e o leiaute obtido é submetido a um algoritmo de compactação local, também baseado em programação linear. Caso a sobreposição não possa ser removida, a solução é ignorada. As técnicas de programação linear adotadas neste trabalho são semelhantes às sugeridas por Milenkovic em ^[12].

Outro trabalho digno de nota é o de Faina ^[28], no qual o recozimento simulado é empregado no planejamento de cortes de guilhotina para geração de formas retangulares.

Avaliações mais recentes do desempenho de heurísticas probabilísticas aplicadas ao problema do posicionamento podem ser encontradas em ^[29] e em ^[30].

1.3 Conclusões

Como visto, problemas de posicionamento são abundantes e bastante diversificadas na indústria.

A grande maioria das abordagens computacionais a problemas de posicionamento tratam de problemas puramente translacionais, e em particular, de formas ortogonais.

As abordagens adotadas variam de algorítmicas exatas, passando por algoritmos de aproximação, heurísticas determinísticas e heurísticas probabilísticas. Pela natureza NP-Difícil dos problemas de posicionamento, quanto mais complexo o problema tratado, menos algorítmica é a abordagem adotada.

Desta maneira, verifica-se que trabalhos que tratam de problemas de posicionamento que permitem rotações e translações para formas heterogêneas tendem a adotar abordagens baseadas em heurísticas probabilísticas. Frequentemente nestes trabalhos – em particular nos que tratam de problemas para dimensões fixas – é feito um “relaxamento” das restrições originais do problema conhecida como *penalização externa*.

Nenhum autor até o momento aplicou o recozimento simulado sem a utilização de penalização externa em problemas de posicionamento não-uniforme rotacional para recipientes de dimensões fixas. Esta é justamente a proposta deste trabalho.

2 O problema de posicionamento translacional e rotacional em recipientes com dimensões fixas

2.1 Introdução

Neste capítulo serão estudadas as características do problema de posicionamento translacional e rotacional em recipientes com dimensões fixas.

Serão apresentadas motivações para o estudo deste problema em particular na seção 2.2.

Apresenta-se na seção 2.3 uma definição precisa do problema, bem como sua classificação diante dos demais problemas de empacotamento estudados na literatura. É discutida também a complexidade do problema e a necessidade de abordagens heurísticas.

Segue-se a seção 2.5, que apresenta o problema do posicionamento em recipientes com dimensões variáveis como a variante *dual* do problema aqui estudado. Trata-se de um problema que possui uma cobertura significativamente mais ampla na literatura, e são apontadas as dificuldades intrínsecas do problema de posicionamento em recipientes de dimensões fixas quando comparado à sua variante dual.

Na seção 2.6 é discutida a *penalização externa*, que é freqüentemente utilizada com heurísticas probabilísticas no problema do posicionamento (e em problemas com domínios complexos em geral). Esta técnica consiste em relaxar as restrições de não-colisão do problema original enquanto é aplicada uma correção à função objetivo das soluções inválidas, de modo a penalizar a sua escolha. São apresentados os problemas intrínsecos desta técnica no posicionamento de formas irregulares e as razões pelas quais esta técnica é evitada neste trabalho.

2.2 Motivação do estudo

Conforme visto na seção 1.2, devido à sua complexidade, existe uma carência de trabalhos em problemas de posicionamento irregular em recipientes de dimensões fixas que permitam translações e rotações das formas.

Esta carência pode ser explicada pela existência de inúmeros problemas de corte de material nos quais o tratamento pelo posicionamento rotacional simplesmente não se aplica.

Um exemplo clássico de problema de corte de material, o caso da indústria têxtil, raramente admite rotação livre nas formas, visto que estas devem possuir um alinhamento pré-determinado com os padrões do tecido. Além disso, o comprimento da peça bruta de tecido é tão grande que ele é mais bem abordado com técnicas que tratam do problema de recipiente com comprimento infinito [7].

Em outros casos, como é o do corte de metal na indústria mecânica, as peças a serem produzidas não possuem quaisquer restrições de rotação, e as peças brutas possuem dimensões limitadas. No entanto, ainda assim estes problemas de corte não são tratados como problemas de posicionamento rotacional, pois nestes casos o custo de um processo de corte irregular é comparável ao custo do material desperdiçado, sendo vantajoso considerá-los como problemas mais restritos, ou seja, como problemas de posicionamento retangular ou problemas de corte de material por guilhotina [6].

Ainda assim, um estudo mais extensivo encontra diversos ramos da indústria nos quais a otimização de posicionamentos rotacionais com recipientes de dimensões fixas é de suma importância.

Considere-se, por exemplo, o problema da indústria de couro. Nesta indústria, formas são recortadas de peças brutas de couro. Verifica-se que neste caso o valor do custo do material desperdiçado supera facilmente o custo de um processo de corte irregular. Ainda mais grave, até a data presente, retalhos de corte de couro são classificados no Brasil como resíduos de classe C I (resíduos sólidos perigosos) [31], de modo que o seu descarte exige um processo específico de custo significativo.

A demanda insatisfeita da indústria por técnicas para o posicionamento rotacional em recipientes de dimensões fixas motiva este trabalho, que trata do problema de posicionamento em seu modo mais irrestrito, o posicionamento translacional e rotacional de formas heterogêneas irregulares (convexas ou não-convexas) em recipientes irregulares (também convexas ou não-convexas).

2.3 Definição do problema

O problema estudado neste trabalho é o posicionamento rotacional de múltiplas formas em um único recipiente de dimensões fixas (problema primal).

Ele pode ser definido como o problema de, dado um recipiente (um polígono convexo ou não-convexo) e um conjunto de n formas (polígonos convexos ou não-convexos), determinar um subconjunto do conjunto de formas e as transformações (translações e rotações) que ao serem aplicadas às respectivas formas, os posicione dentro do recipiente sem sobreposições, e de modo a maximizar a área ocupada (vide figura 2.1).

Definição 2.1 *Existe sobreposição entre formas quando a intersecção entre elas é não-vazia.*

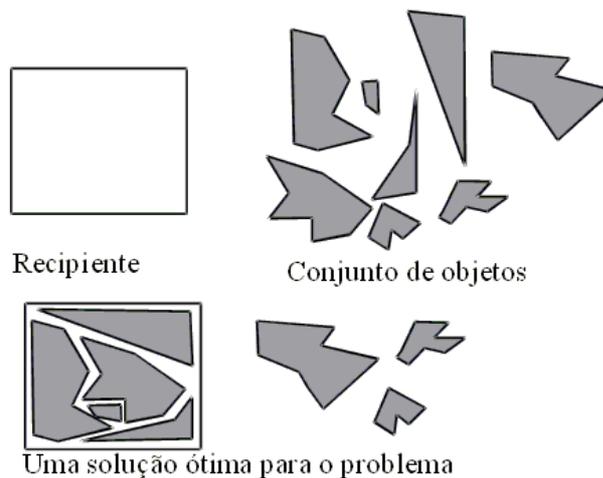


Figura 2.1: Um problema de empacotamento e sua solução ótima.

Na classificação sugerida por Wascher *et al.* em [5], este é um problema de maximização de saída, com um único recipiente, cujas formas têm geometrias fortemente heterogêneas. Quanto à dimensionalidade, definida por Wascher *et al.* como “o número mínimo de parâmetros necessários para se caracterizar a posição de um objeto em um leiaute”, verifica-se que a permissão de livre rotação adiciona um grau de liberdade ao problema de posicionamento bidimensional, o que o transforma em um problema com 3 dimensões. Assim, ele é classificado por Wascher *et al.* como um Problema de Mochila Simples 3D (*Simple Knapsack Problem*).

2.4 Complexidade

Como mencionado na seção 1.1, é demonstrado que mesmo versões restritas do problema (por exemplo, que lidam apenas com retângulos) são NP-Difíceis, o que significa que não se acredita que seja possível resolver algoritmicamente o problema para um número prático de formas ^[15]. Por este fato diversos autores (vide seção 1.2) voltaram-se para abordagens baseadas em heurísticas probabilísticas, que não garantem a determinação da solução ótima, mas podem, em um tempo razoável, encontrar uma solução aceitável. Heurísticas probabilísticas de otimização seguem o seguinte padrão: enquanto um critério de parada não for satisfeito, em cada etapa avalia-se a função a ser otimizada em um conjunto de pontos e aplicam-se algumas regras para determinar o conjunto de pontos a ser avaliado na próxima etapa.

2.5 Características da função objetivo e dualidade

O problema de posicionamento em recipientes de dimensões fixas possui um problema correlato dual que é o problema de, dado um conjunto de formas, encontrar o menor recipiente capaz de as conter sem que haja sobreposição. No levantamento feito por Wascher *et al.* ^[5], o problema dual encontra uma cobertura significativamente mais ampla na literatura.

Quando comparado à sua variante dual, o problema de posicionamento em recipientes de dimensões fixas possui uma característica que dificulta o seu tratamento, o fato de que ele é um problema de otimização de variáveis *contínuas* (as translações e rotações das formas) com uma função objetivo que assume valores *discretos* (a área total não-ocupada pelas formas).

A figura 2.2 mostra um problema de otimização dual de uma só variável (a rotação θ da forma mais a esquerda), no qual o objetivo é minimizar a largura total do recipiente $f(\theta)$. Como se pode ver, a largura mínima do recipiente varia de modo *contínuo* com a variável de otimização θ .

No caso do problema primal mostrado na figura 2.3, a função objetivo a ser minimizada é o espaço não ocupado no interior do recipiente. Como este só pode variar em função da área de formas sendo adicionadas ou retiradas do recipiente, a função objetivo assume somente uma quantidade finita de valores, tornando-se assim descontínua.

Esta característica dificulta a avaliação da sensibilidade da função objetivo

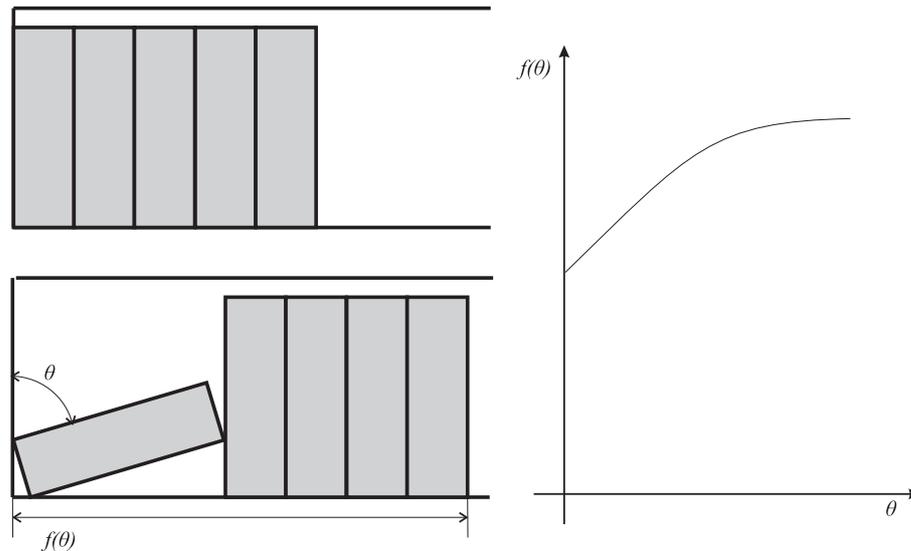


Figura 2.2: A variação da função objetivo no problema dual.

com relação às variáveis do problema.

2.6 A penalização externa

Para o problema abordado, o espaço de soluções válidas é um subconjunto de formas bem como um conjunto de transformações (translações e rotações) aplicadas a este sub-conjunto de modo que as restrições de que não podem haver sobreposições entre as formas e a de que estas devem estar totalmente contidas no recipiente sejam respeitadas. O espaço delimitado por estas restrições revela-se muito complexo (e mutável). Geralmente, quando confrontadas com espaços complexos, heurísticas probabilísticas “relaxam” as restrições originais do problema, permitindo pontos fora do espaço de soluções válidas e aplicando penalizações à função objetivo de pontos externos. Esta técnica clássica da otimização em domínios complexos é conhecida como penalização externa (*Exterior Penalty*) [32].

Embora tal abordagem leve a uma diminuição do custo computacional do processo de otimização, ela introduz um parâmetro empírico adicional, a quantidade de penalização a ser aplicada ao custo dos pontos externos.

Outro inconveniente da penalização externa é que o relaxamento das restrições do problema pode fazer com que o processo de otimização produza soluções inválidas, e a aplicação de penalização pode não garantir que soluções ótimas do problema relaxado sejam soluções válidas do problema original. Por exemplo,

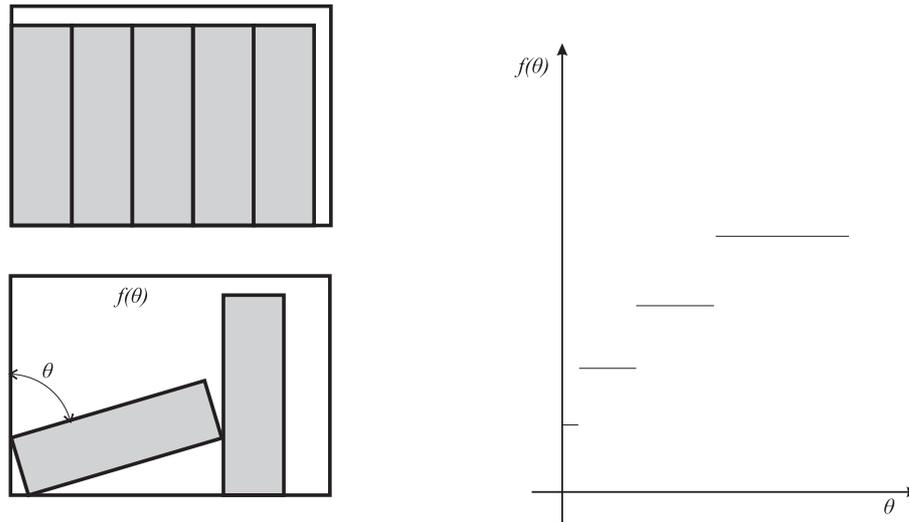
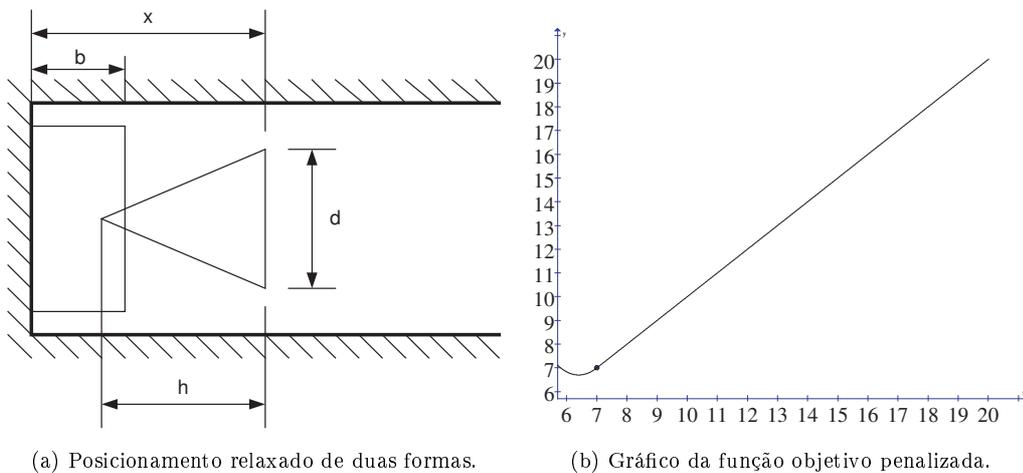


Figura 2.3: A variação da função objetivo no problema primal.

uma forma popular de penalização externa em problemas de posicionamento, a penalização externa baseada puramente em área sobreposta, não garante a validade de suas soluções ótimas.

Isso pode ser percebido facilmente no problema de posicionamento ilustrado na figura 2.4. Neste problema, deseja-se minimizar o comprimento total x do recipiente capaz de conter as duas formas. A única variável do problema é a posição da forma triangular, definida como a distância entre a sua aresta vertical e a aresta vertical à esquerda do recipiente. A função objetivo original é o próprio valor da variável $f(x) = x$, a restrição do problema, imposta pela condição de não sobreposição entre as formas é $x \geq b + h$ e a solução ótima para o problema é $x_{opt} = h + b$.



(a) Posicionamento relaxado de duas formas.

(b) Gráfico da função objetivo penalizada.

Figura 2.4: (a) Um problema de posicionamento relaxado com uma solução ótima inválida.

Aplicando-se a penalização externa baseada em área de sobreposição, tem-se um novo problema irrestrito no qual a função objetivo é:

$$f^*(x) = \begin{cases} x + k \cdot \overbrace{\frac{d}{2h}(b+h-x)^2}^{\text{sobreposição}} & \text{para } x < b+h \\ x & \text{para } x \geq b+h \end{cases}$$

onde k é um parâmetro que define a quantidade de penalização a ser aplicada por área de sobreposição. A função objetivo modificada é contínua de classe C_1 , de modo que tem-se

$$\begin{aligned} \frac{df}{dx} \Big|_{x=x_{opt}^*} = 0 &\Rightarrow 1 - \frac{kd}{h}(b+h-x_{opt}^*) = 0 \\ &\Downarrow \\ (b+h-x_{opt}^*) &= \frac{h}{kd} \end{aligned}$$

Ora, mas $(b+h-x_{opt}^*)$ é a própria medida do comprimento de sobreposição das formas na solução ótima do problema modificado. Assim, percebe-se que embora a sobreposição seja decrescente com k , ela é não-nula para qualquer valor finito de k .

Este efeito pode ser visto no trabalho de Heckman e Lengauer ^[23], em que problemas similares aos aqui tratados são abordados com recozimento simulado. Neste trabalho, Heckman e Lengauer apontam que o resultado bruto de seu processo de otimização pode conter soluções inválidas (que apresentam superposição entre as formas), sendo necessário fazer um pós-processamento dos resultados obtidos.

2.7 Conclusões

O problema do posicionamento translacional e rotacional de formas em recipientes de dimensões fixas, apesar de ser relativamente pouco estudado, é de grande relevância para a indústria.

O problema possui complexidade NP-Difícil, de modo que abordagens puramente algorítmicas não são viáveis para problemas de dimensões práticas. Assim sendo, abordagens heurísticas são recomendáveis para este tipo de problema.

O posicionamento em recipientes de dimensões fixas possui dificuldades intrínsecas quando comparado com o posicionamento em recipientes de dimensões variáveis. Em particular, embora ambos sejam problemas que atuem em domí-

nios contínuos, a função objetivo do problema com recipientes de dimensões fixas assume valores discretos. Esta característica dificulta a análise de sensibilidade das variáveis de otimização.

Abordagens heurísticas para o problema baseadas na técnica de *penalização externa*, que relaxa restrições do problema e aplica uma modificação na função objetivo de soluções inválidas, podem convergir para leiautes inválidos, que apresentam sobreposição de formas.

Conseqüentemente, é interessante avaliar a viabilidade de se abordar o problema do posicionamento com heurísticas que não empreguem a penalização externa, explorando o espaço original de soluções do problema.

3 O Polígono de obstrução e a decomposição do problema de posicionamento

3.1 Introdução

Como visto no capítulo anterior, é interessante criar uma heurística para o problema do posicionamento rotacional de formas irregulares em recipientes de dimensões fixas que não empregue a penalização externa.

O abandono da penalização externa implica em restringir o espaço de busca ao espaço original de soluções do problema, livre de superposições entre as formas. Para lidar com este complexo espaço, emprega-se aqui uma ferramenta muito utilizada em problemas de posicionamento translacional com recipientes de dimensões abertas, o polígono de obstrução (*no-fit polygon*).

A seção 3.2 introduz o conceito de *polígono de obstrução* como elemento-base para a construção de heurísticas que exploram o espaço de soluções do problema.

A seção 3.3 mostra como o uso do polígono de obstrução introduz a necessidade de se realizar um posicionamento *seqüencial* das formas no recipiente. Este posicionamento seqüencial leva a uma decomposição do problema em três sub-problemas: a ordenação das formas, a rotação das formas e a translação das formas. Expõem-se as dificuldades do tratamento determinístico de cada um destes sub-problemas.

3.2 Polígonos de obstrução e regiões livres de colisão

Polígonos de obstrução e regiões livres de colisão são conceitos relacionados a conjuntos de *translações de uma forma*, que são matematicamente representados por conjuntos de vetores. No entanto, para melhor compreensão das propriedades

e operações envolvidas com tais conceitos, as translações de uma forma serão associadas *regiões* do plano xy .

Para tanto, inicialmente observa-se que é possível associar à toda forma um *ponto de referência*, normalmente considerado como origem do sistema de coordenadas da forma (note que este ponto pode ser interno ou externo à forma). A região no espaço associada a um conjunto de translações desta forma é o conjunto de pontos que podem ser obtidos através da aplicação das translações do conjunto ao ponto de referência da forma (vide figura 3.1).

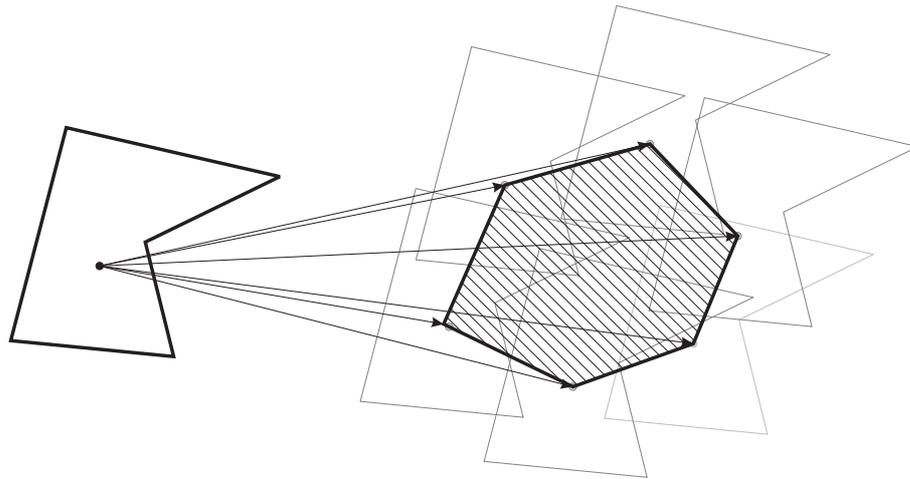


Figura 3.1: Um conjunto de translações de uma forma representado como a região formada por todas as possíveis translações de seu ponto de referência.

Considerando um dado recipiente e um conjunto de formas já posicionadas, conforme ilustrado na figura 3.2, deseja-se posicionar uma nova forma no interior do recipiente e sem colisão com as formas já posicionadas. Assim, apresenta-se a seguinte definição:

Definição 3.1 *Define-se região livre de obstrução como o conjunto de todas as translações, que, quando aplicadas a uma determinada forma, posicionam-na no interior de um recipiente sem que haja colisão com as formas já posicionadas.*

Como pode ser visto na figura 3.2, a região livre de obstrução pode ter geometrias variadas e até mesmo ser formada por múltiplas sub-regiões não-conexas.

Quando o recipiente encontra-se vazio, a região livre de obstrução corresponde a todas as translações que posicionam a forma no interior do recipiente. A essa região dá-se o nome de *região de posicionamento interno* (*inner-fit polygon*) (vide figura 3.3), e a sua determinação é o primeiro passo para encontrar a região livre de obstrução para uma forma.

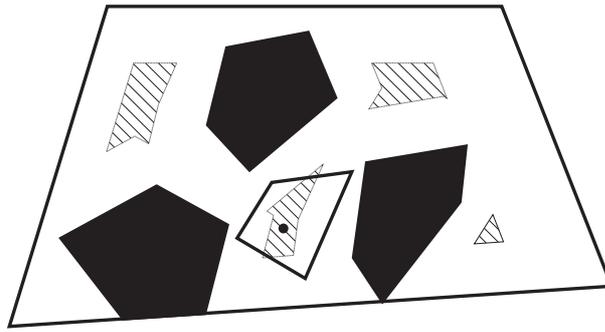


Figura 3.2: A região livre de obstrução (hachuradas) para o posicionamento de uma forma em um recipiente com obstáculos.

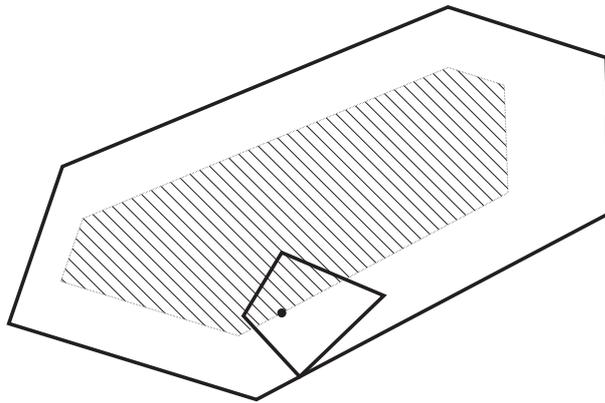


Figura 3.3: A região de posicionamento interno (*inner-fit polygon*) para uma dada forma e seu recipiente.

As regiões livres de colisão no posicionamento de uma forma são obtidas retirando-se da região de posicionamento interno todos os polígonos de obstrução gerados pelos obstáculos que já se encontram no recipiente.

O polígono de obstrução, conhecido na literatura como *no-fit polygon*, é a construção geométrica fundamental empregada nesta pesquisa. Através dela, pode ser determinada a região proibida ou obstruída no posicionamento de uma determinada forma induzida pela presença de um obstáculo (outra forma). Como visto na seção 1.2, o polígono de obstrução vem sendo aplicado a problemas de posicionamento exclusivamente translacionais com recipientes de dimensões abertas.

Definição 3.2 Define-se polígono de obstrução induzido por um polígono A a um polígono B denotado $A \ominus B$ como o conjunto de vetores de translação que, quando aplicados ao polígono B , o levam a uma sobreposição¹ com A .

Ou seja:

¹vide definição 2.1.

$$A \ominus B = \{O + \vec{v} \mid \exists b \in B \mid b + \vec{v} \in A\} \quad (3.1)$$

Os polígonos de obstrução são encontrados através do algoritmo de soma de Minkowski^[33] e do *polígono oposto*.

Definição 3.3 A soma de Minkowski entre dois polígonos A e B , denotada $A \oplus B$, é definida como o conjunto de pontos $\{O + \vec{v}_a + \vec{v}_b \mid O + \vec{v}_a \in A, O + \vec{v}_b \in B\}$.

Definição 3.4 Define-se o polígono oposto a um dado polígono B denotado $-B$ como o conjunto de pontos definido por $-B = \{O - \vec{v}_b \mid O + \vec{v}_b \in B\}$.

Obtém-se o polígono oposto invertendo-se o sinal de todas as coordenadas do polígono original.

A partir das definições acima, bem como da definição de polígono de obstrução (vide definição 3.2 na seção 3.2) percebe-se que

$$A \ominus B = A \oplus (-B) \quad (3.2)$$

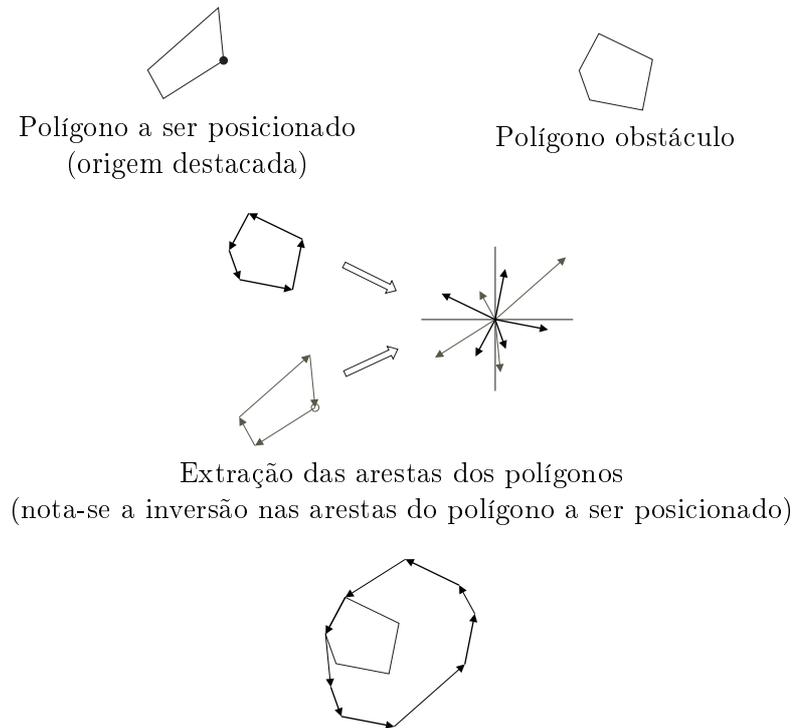
ou seja, o polígono de obstrução é gerado pela soma de Minkowski do obstáculo com o oposto do polígono a ser posicionado.

Somas de Minkowski podem ser calculadas facilmente para polígonos convexos. O resultado da soma de Minkowski de dois polígonos convexos é um polígono convexo construído a partir das arestas dos polígonos originais (vide figura 3.4). Polígonos não-convexos podem ser decompostos em uma etapa de pré-processamento do algoritmo, visto que as transformações aplicadas (de rotação e translação) não afetam tal decomposição.

Assim, pode-se, através do polígono de obstrução, delimitar o espaço para as translações de uma forma, assumindo que *a sua rotação já esteja determinada*. Nota-se que a região válida para o posicionamento de uma forma pode ser vazia, resultado que corresponde à situação em que não há mais espaço no recipiente para a forma.

3.3 Decomposição do macro-problema de posicionamento

Como visto, o polígono de obstrução fornece a informação geométrica sobre a região “proibida” (obstruída por outras formas ou paredes do recipiente) para



Resultado: Polígono de obstrução gerado a partir das arestas dos dois polígonos

Figura 3.4: Exemplo de geração de polígono de obstrução para polígonos convexos.

uma *única* forma, dadas as posições das formas já posicionadas no recipiente. Isso implica que o posicionamento das formas deve ser feito de forma *seqüencial* ou seja, em primeiro lugar, as formas devem ser ordenadas de acordo com algum critério. Em seguida, elas devem ser posicionadas no recipiente de modo que cada forma seja posicionada sem sobreposição desta com qualquer uma das formas já posicionadas.

Assim, o macro-problema de se determinar o posicionamento das formas pode ser decomposto nos seguintes sub-problemas:

- Determinar uma ordem de posicionamento para as formas. Este sub-problema, no caso primal, trata da escolha de quais formas serão selecionadas para o posicionamento, visto que as últimas formas a serem posicionadas encontram um recipiente já obstruído. Além disso, ele exerce uma forte influência na posição final das formas selecionadas. Diversos trabalhos relacionados ao problema dual (nos quais o problema de seleção de formas inexistente) tratam *exclusivamente* da determinação da ordem de posicionamento, aplicando em seguida uma heurística *bottom-left* à seqüência de formas (por exemplo, em ^[27] e em ^[25]).

- Determinar as rotações de cada forma. Este sub-problema é o que caracteriza o posicionamento rotacional. Devido ao fato de que a geometria das formas foi assumida como sendo irrestrita, este é um problema combinatorial por si, e de difícil solução. Uma solução algorítmica para mínimos locais (com rotações limitadas) é apresentada em ^[12].
- Determinar para cada uma das formas a sua translação relativa às demais e ao recipiente. Este sub-problema é o mais estudado na literatura. Como já mencionado, pretende-se empregar o polígono de obstrução para determinar as regiões livres de colisão, reduzindo o problema à determinação do ponto nestas regiões livres sobre o qual a forma será posicionada.

3.3.1 Heurística para a ordem de posicionamento

Como visto na seção 3.2, a proposta de abordagem aqui estudada envolve posicionar as formas em configurações livres de sobreposição através do uso de polígonos de obstrução.

Embora o polígono de obstrução seja uma ferramenta eficiente para determinar rapidamente a área não-obstruída no posicionamento de uma forma, ela não resolve o principal problema combinatorial de determinar áreas não-obstruídas para o posicionamento simultâneo de múltiplas formas (visto que estas formas podem colidir entre si, dado que a posição das formas ainda não é determinada).

Assim, o uso do polígono de obstrução implica em um posicionamento *seqüencial* das formas no recipiente, sendo que a ordem em que este posicionamento se dá é um problema combinatorial por si de difícil solução.

A principal influência da ordem está evidentemente na seleção das formas que serão posicionadas no recipiente. As formas que forem posicionadas por último encontrarão um recipiente já bastante obstruído, e conseqüentemente, terão uma probabilidade menor de encontrarem uma região livre para serem posicionadas.

Além disso, observa-se que a ordem em que as formas do subconjunto selecionado são inseridas possui uma forte influência nas suas posições finais. Isso ocorre pelo modo como trabalham as heurísticas que determinam a translação das peças em posicionamentos seqüenciais. Estas heurísticas, para evitar lidar com a natureza combinatorial do problema de posição, geralmente consideram *somente* as formas já posicionadas no momento de determinar a posição de uma nova forma. Assim, elas buscam posicionar uma forma com base em uma *estimativa* das necessidades de espaço das formas ainda não posicionadas.

Desta maneira, ordenações nas quais as necessidades de espaço das peças ainda por serem posicionadas são mais previsíveis geram naturalmente leiautes satisfatórios. Como a forma de prever a necessidade de espaço é particular à heurística de translação, existe um forte acoplamento entre esta e a heurística de ordem de posicionamento.

Muitos trabalhos que tratam do problema da ordem no posicionamento seqüencial de formas (vide por exemplo [20, 27]) empregam heurísticas determinísticas baseadas na ordem decrescente de tamanho das formas. Estas heurísticas são baseadas em observação do comportamento de especialistas humanos na solução do problema que buscam definir a posição das formas maiores antes das formas menores.

Estas heurísticas são coletivamente denominadas heurísticas de maior primeiro (*larger first*). Elas variam entre si pela definição de “tamanho” adotada, sendo que esta definição pode ser baseada em:

- Área da forma.
- Área do casco convexo (*convex hull*) da forma.
- Área do menor retângulo com orientação definida que envolve a forma (*Bounding Box*).
- Comprimento em uma determinada direção
- Maior distância entre dois pontos na forma
- Uma combinação dos parâmetros acima

Esta heurística de ordenação relativamente simples atinge resultados surpreendentemente satisfatórios na maioria dos casos. Isso pode ser explicado à luz do exposto anteriormente, devido ao fato de que um número grande de formas com dimensões pequenas pode ser encaixado em áreas vazias de formas mais variadas do que um número pequeno de formas de dimensões grandes. Assim, a área livre no recipiente torna-se um bom indicador da necessidade de espaço para as formas pequenas, de modo que leiautes nos quais estas são posicionadas por último tendem a ser mais eficientes.

No entanto, como para toda heurística de otimização determinística, é relativamente simples apontar contra-exemplos de eficiência. A figura 3.5 mostra a melhor solução para um *puzzle* atingível através de uma heurística *larger first*.

Neste puzzle, a heurística de translação busca *sempre* posicionar a primeira forma ao lado de uma parede do recipiente.

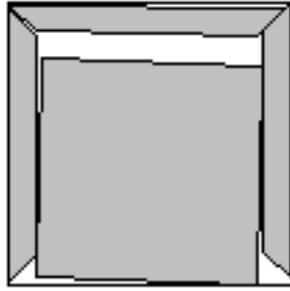


Figura 3.5: Problema cuja solução ótima não pode ser alcançada por uma heurística *larger first* .

3.3.2 Heurística para a rotação das formas

O problema da determinação da rotação das formas é precisamente o que caracteriza as apresentações rotacionais dos problemas de posicionamento. Esta determinação por si só já representa um problema de otimização significativamente complexo. Como pode ser visto na figura 3.6, as rotações de formas em um leiaute podem estar estreitamente vinculadas, de modo que a rotação de uma única forma pode afetar a disponibilidade de espaço livre para todas as suas vizinhas.

Como visto na seção 1.1, problemas de posicionamento que tratam da rotação livre de formas são minoria absoluta na literatura da área ^[5] e são a principal motivação neste trabalho para a busca por abordagens heurísticas probabilísticas. De fato, o sub-problema da rotação é o *único* neste trabalho que é abordado *exclusivamente* com o recozimento simulado.

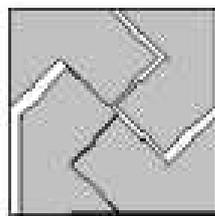


Figura 3.6: Um problema de posicionamento rotacional.

3.3.3 Heurística para a translação das formas

A abordagem tradicional para a translação de formas é a heurística determinística *bottom-left*.

Nesta abordagem, a seqüência de formas é posicionada no recipiente de modo que cada forma é sempre posicionada na posição “mais abaixo à esquerda” possível.

Como visto na seção 1.2, esta é uma abordagem que vem sido empregada com relativo sucesso em trabalhos da área (cita-se por exemplo ^[19, 20, 25, 27]).

Pode-se compreender a aplicação bem sucedida do *bottom-left* tendo em mente que, em primeiro lugar, trata-se de uma heurística bastante econômica em recursos computacionais. Além disso, por agrupar as formas junto a duas paredes do recipiente (esta é uma heurística aplicada quase exclusivamente a recipientes retangulares), ela mantém a área livre no recipiente relativamente desobstruída, de modo que uma variedade maior de formas possa ser posicionada.

Nota-se no entanto que é relativamente simples apontar problemas de posicionamento nos quais o ótimo global é inatingível via uma heurística *bottom-left* (vide figura 3.7).

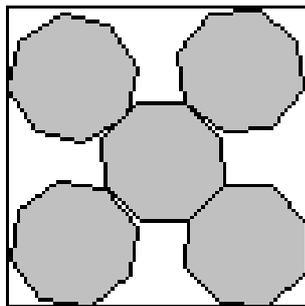


Figura 3.7: Exemplo simples de problema cuja solução ótima não pode ser alcançada por uma heurística *bottom-left* .

3.4 Conclusões

A exploração do espaço com restrições originais pode ser feita de forma eficiente com o uso do polígono de obstrução, que determina a região obstruída por uma

forma no posicionamento de outra. Através disso, é possível no posicionamento de uma forma determinar exatamente o espaço de translações válidas.

Como o polígono de obstrução é aplicável a somente uma forma por vez, o seu uso implica em tratar o posicionamento das formas de modo seqüencial. Além disso, é necessário que a rotação da forma a ser posicionada seja bem definida. Estas restrições levam à decomposição do problema de posicionamento nos três sub-problemas:

- Determinação da ordem de posicionamento das formas.
- Determinação das rotações das formas.
- Determinação das translações das formas.

Verifica-se que a abordagem com heurísticas determinísticas para cada um dos sub-problemas é deficiente, sendo trivial construir exemplos de problemas para os quais as heurísticas determinísticas levam a soluções sub-ótimas.

Assim, verifica-se a necessidade de empregar heurísticas probabilísticas para este problema.

4 Aplicando o recozimento simulado ao problema do posicionamento

4.1 Introdução

Conforme visto no capítulo 3, para que o processo de otimização tenha a possibilidade de convergir para uma solução ótima, é necessário empregar uma heurística probabilística.

O recozimento simulado (*Simulated Annealing*)^[34] é a meta-heurística probabilística empregada neste trabalho. Suas características estão expostas na seção 4.2 (mais detalhes sobre o recozimento simulado encontram-se no apêndice A). Na sub-seção 4.2.1 são apontadas diretrizes apontadas pela literatura^[35] a serem seguidas na aplicação do recozimento simulado.

Na seção 4.3 são definidas e discutidas heurísticas de otimização probabilísticas completas.

A seção 4.4 retoma a decomposição do problema de posicionamento realizada na seção 3.3, e aponta para cada um dos sub-problemas uma heurística probabilística de otimização completa baseada no recozimento simulado.

Na seção 4.5 é mostrado como são geradas as soluções a serem exploradas pelo recozimento simulado consideradas as heurísticas propostas na seção 4.4.

Finalmente, a seção 4.6 mostra como é feita a síntese do processo completo de avaliação da função objetivo a partir dos parâmetros produzidos pelas heurísticas definidas na seção 4.4.

4.2 O recozimento simulado

A meta-heurística probabilística adotada neste trabalho é o recozimento simulado (*Simulated Annealing*). O recozimento simulado é uma meta-heurística probabilística dita de *exploração local*. Uma heurística de otimização de exploração local analisa a cada iteração uma e somente uma solução do problema. O custo desta solução é comparado com o custo da solução da iteração imediatamente anterior, e baseando-se em regras da heurística, uma nova solução é gerada para a próxima iteração.

No caso do recozimento simulado, a solução do problema x_i^* avaliada na iteração $i + 1$ é escolhida aleatoriamente em uma *vizinhança* da solução atual $V(x_i)$. Por *vizinhança* de uma solução, entende-se um conjunto de soluções válidas relacionadas a uma determinada solução. A definição de vizinhança é diretamente dependente do problema de otimização abordado, e da maneira específica com a qual se decide aplicar o recozimento simulado a ele. Cabe notar que a distribuição da probabilidade de seleção das soluções dentro da vizinhança não é necessariamente uniforme.

A solução atual x_i pode ser substituída em uma determinada iteração pela solução analisada x_i^* de acordo com as seguintes regras:

- $f(x_i^*) \leq f(x_i)$ A solução analisada na iteração tem custo inferior ou igual ao da solução atual. Neste caso, a solução atual é imediatamente substituída pela solução analisada.
- $f(x_i^*) > f(x_i)$ A solução analisada na iteração tem custo superior ao da solução atual. Neste caso, a solução atual pode ser substituída pela solução analisada com probabilidade determinada pela diferença de custos e pela “temperatura” do processo.

A “temperatura” é um parâmetro do recozimento simulado que varia ao longo do processo de acordo com um “programa de resfriamento”. Genericamente, quanto maior for a temperatura, maior a probabilidade do recozimento simulado substituir uma solução de custo inferior por uma solução de custo superior (caminhando assim, na direção do gradiente ascendente de custos). É este o tipo de movimentação que permite ao processo escapar de mínimos locais. À medida em que o processo avança, a temperatura é gradativamente reduzida, permitindo assim que este se fixe em uma solução ótima.

Para maiores detalhes sobre o recozimento simulado e programas de resfriamento, vide o Apêndice A.

4.2.1 Diretrizes na aplicação de recozimento simulado

Herz e Widmer escreveram sobre técnicas gerais para aplicar meta-heurísticas a problemas de otimização [35]. Em seu trabalho, eles apontam as seguintes diretrizes a serem adotadas na aplicação de métodos de otimização de exploração local:

Diretriz 4.1 *Deve ser fácil gerar soluções válidas.*

Métodos de exploração local tipicamente avaliam o custo de um grande número de soluções do problema. É mister portanto, dispor de um mecanismo para geração de soluções válidas computacionalmente eficiente, sob pena do método tornar-se computacionalmente inviável.

Diretriz 4.2 *Para cada solução factível, deve existir um caminho ligando esta solução a uma solução ótima.*

Caso existam soluções sub-ótimas sem caminho para soluções ótimas, o processo pode ficar “preso” na vizinhança de tais soluções, de modo que o ótimo global não possa ser atingido.

Diretriz 4.3 *A vizinhança de uma dada solução deve ser relativamente próxima a ela.*

Uma maneira simples de garantir que cada solução possua um caminho para uma solução ótima é fazer com que o subconjunto de soluções explorado em cada etapa seja igual ao conjunto universo de soluções. No entanto, ao trabalhar deste modo, a tarefa de se selecionar a próxima solução a ser explorada torna-se consideravelmente mais difícil (no caso de um método de exploração local probabilístico, o processo reduzir-se-ia a uma exploração aleatória uniforme do espaço). É interessante, portanto, reduzir o subconjunto de soluções a ser explorado a uma vizinhança da solução anterior. Uma vantagem de se restringir à vizinhança da solução anterior é que se as modificações feitas na solução forem restritas o suficiente, pode ser bastante simples calcular o novo valor da função objetivo. Além do mais, se a topologia do espaço induzida pela função objetivo for relativamente suave, o custo de uma solução pode indicar seu grau de vizinhança com uma solução ótima.

Diretriz 4.4 *A topologia do espaço induzida pela função objetivo no espaço de soluções factíveis não deve ser demasiado plana.*

A ocorrência de regiões extensas do espaço com custo muito próximo prejudica o desempenho de métodos de exploração local, visto que estes geralmente utilizam a variação do custo ao longo da exploração para determinar os próximos passos. Para problemas em que tais regiões ocorram naturalmente, recomenda-se adotar uma heurística de “desempate” que modifique a função objetivo de modo a reduzir o custo de soluções que estariam mais próximas de uma solução ótima.

4.3 Heurísticas probabilísticas completas

Em problemas de otimização com variáveis contínuas é comum adotar-se simplificações do problema de modo a reduzir o espaço total a ser explorado.

No entanto, conforme visto na seção 3.3 do capítulo anterior, tais simplificações podem levar a heurísticas que são incapazes de atingir uma solução ótima do problema.

Neste trabalho, procurou-se buscar uma heurística *completa*, ou seja, uma heurística à qual a solução ótima do problema *sempre esteja acessível* (nota-se que uma meta-heurística completa neste sentido atende à diretriz 4.2 de Hertz e Widmer) [35].

Definição 4.1 *Uma heurística probabilística de otimização completa é uma heurística que possui uma probabilidade não-nula de atingir uma solução ótima para o problema.*

4.4 Heurísticas para os sub-problemas do posicionamento

Apresentam-se a seguir heurísticas probabilísticas baseadas no recozimento simulado para cada um dos sub-problemas apresentados na seção 3.3. Cada heurística consiste basicamente de uma representação do espaço de soluções válidas para o sub-problema e de uma regra de vizinhança, que dita como selecionar neste espaço uma solução vizinha a uma solução dada.

É preciso observar que quando estas heurísticas são empregadas *simultaneamente*, as regras de vizinhança são modificadas de modo que apenas a solução de

um dos sub-problemas seja modificada por iteração. Isso assegura que a diretriz 4.3 de Herz e Widmer seja atendida. De fato, verifica-se experimentalmente um melhor desempenho do processo quando apenas uma sub-solução é modificada por iteração.

4.4.1 Ordem de posicionamento

Devido às limitações das heurísticas determinísticas na seleção de ordem, é proposta aqui uma heurística probabilística baseada no recozimento simulado.

Nela, a ordenação inicial do posicionamento é escolhida ao acaso. A vizinhança (vide seção 4.2) de uma dada ordenação é qualquer ordenação obtida da original na qual a ordem de duas formas é permutada.

A figura 4.1 retoma o problema que ilustra a seção 3.3.1, mostrando como ele pode ser resolvido por uma heurística probabilística.

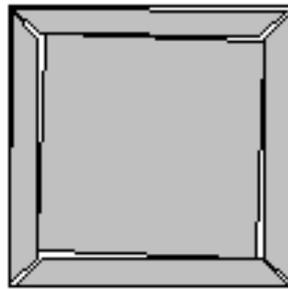


Figura 4.1: Problema resolvido com ordem gerada por recozimento simulado.

4.4.2 Rotação das formas

A rotação de cada uma das n formas é definida por um vetor de rotações $\Theta = \{\theta_1, \theta_2, \dots, \theta_n\}$, sendo que este vetor é resultado da otimização por recozimento simulado.

A vizinhança de um vetor de rotações Θ é definida como o conjunto de rotações nas quais todos os componentes θ_i são iguais ao original, *exceto um*. Assim, em uma etapa do recozimento simulado, apenas uma forma pode ter sua rotação modificada.

Esta definição de conjunto de vizinhança difere das tradicionais definições aplicadas ao recozimento simulado no sentido em que ela comporta intervalos *contínuos* de rotações, contendo conseqüentemente infinitas soluções. Tal característica emerge naturalmente do fato de que o espaço de rotações é por si mesmo

um espaço contínuo, e a segmentação deste espaço é muito custosa em recursos computacionais para ser aplicada efetivamente a uma heurística como o recozimento simulado. Uma definição de vizinhança semelhante neste aspecto será empregada também para tratar o problema de translação (vide seção 3.3.3).

Esta heurística revela-se bastante eficaz, mesmo quando aplicada em conjunto com heurísticas determinísticas simples de ordem e translação (vide figura 4.2).



Figura 4.2: Um problema de posicionamento rotacional resolvido com recozimento simulado aplicado somente ao sub-problema de rotação.

Quando esta heurística é utilizada em conjunto com a heurística de seleção de ordem de posicionamento, a vizinhança de uma solução é definida como o conjunto de soluções nas quais ou as rotações ou a ordem são idênticas à solução original, e o parâmetro restante está contido na vizinhança do parâmetro da solução original. Neste caso, em uma dada iteração, ou a rotação de uma forma é modificada, ou a ordem de posicionamento de duas formas é trocada.

A distribuição de probabilidade de seleção de uma solução nova de rotação na vizinhança da atual não é uniforme, mas depende do *fator de cristalização* da forma cuja rotação será alterada. O *fator de cristalização* é uma variável associada a cada forma que regula a amplitude dos seus movimentos. Seu objetivo é reduzir a amplitude de movimento de uma forma cuja movimentação historicamente leva a soluções rejeitadas. Detalhes sobre o fator de cristalização são dados na seção 4.5.1.

4.4.3 Translação das formas

A proposta deste trabalho para o sub-problema de translação é de mapear continuamente um subconjunto de translações válidas em um espaço simplificado que contenha restrições simples, de modo que seja trivial gerar soluções neste espaço.

O subconjunto de translações considerado pela heurística é o subconjunto que leva a *soluções conexas*. *Soluções conexas* serão definidas inicialmente definindo-se *grafo de conectividade* de uma solução.

Definição 4.2 *Duas formas distintas A e B são ditas contíguas se existe ao menos um ponto sobre o perímetro da forma A para o qual não há uma vizinhança que não contenha ao menos um ponto sobre o perímetro da forma B .*¹

Definição 4.3 *O grafo de conectividade associado a uma solução s de um problema de posicionamento é o grafo cujos vértices são compostos pelo recipiente P_0 e pelas formas P_i posicionadas no seu interior. A aresta $E_{i,j}$ existe se o polígono i é contíguo ao polígono j .*

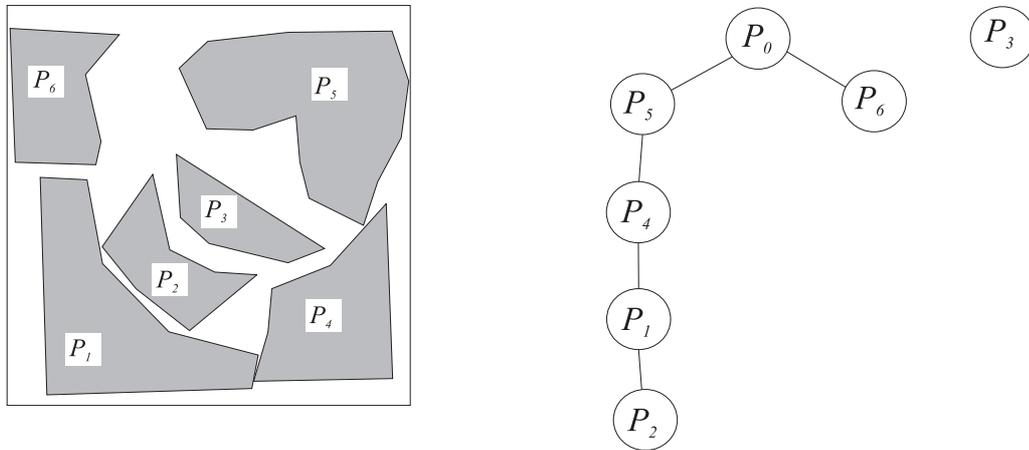


Figura 4.3: Uma solução válida e seu grafo de conectividade.

O grafo de conectividade contém a informação de quais polígonos na solução são contíguos entre si (vide figura 4.3).

Definição 4.4 *Uma solução s de um problema de posicionamento é dita conexa se o seu o grafo de conectividade associado for conexo.*

A definição 4.4 estabelece a categoria de soluções em que não há formas “soltas” no recipiente. Sempre existe contato, direto ou indireto, entre uma forma e o recipiente (vide figura 4.4).

Teorema 4.1 *É possível a partir de uma solução não-conexa de um problema de posicionamento que permita translação livre construir uma solução conexa de igual custo.*

¹Note que a definição é comutativa, ou seja, se A é contíguo a B , então B é contíguo a A . Note também que A não é contíguo a si mesmo.

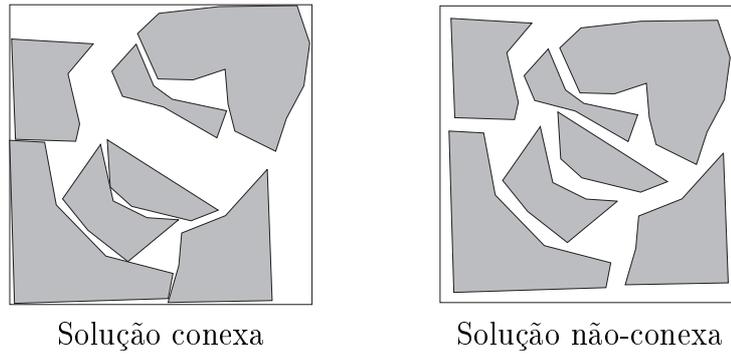


Figura 4.4: Exemplo de solução conexa e não-conexa.

Prova: Para cada sub-grafo conexo do grafo de conectividade da solução dada, é possível transladar as formas associadas até que ao menos uma delas torne-se contígua a uma forma associada a um outro sub-grafo conexo. Deste modo, é sempre possível construir a partir de uma solução cujo grafo de conectividade apresente $n > 1$ sub-grafos conexos uma solução com um grafo de conectividade com $n - 1$ sub-grafos conexos. Por indução, é possível construir uma solução conexa.

Corolário 4.2 *Todo problema de posicionamento primal que permita translação livre possui ao menos uma solução ótima conexa (observa-se que, como visto na seção 2.5, todo problema primal admite um conjunto finito de valores para a sua função objetivo, e conseqüentemente, possui soluções ótimas).*

Do corolário 4.2 decorre que é possível construir uma heurística *completa* (ou seja, que possa atingir uma solução ótima) que se restrinja somente à exploração exclusiva do subconjunto de translações que levam a soluções conexas.

A exploração restrita ao subconjunto de soluções conexas é feita através do posicionamento seqüencial das formas no recipiente. Para cada forma a ser posicionada é determinada a região livre de obstruções (área na qual, se posicionado, a forma não fica em sobreposição com nenhuma outra forma). A forma é posicionada no perímetro desta região. Isso garante que ele estará posicionado de modo contíguo a outro polígono ou ao recipiente.

Como a forma e o número de regiões livres de colisão para uma forma varia de acordo com a disposição das formas já inseridos no recipiente, é preciso mapear a posição relativa de posicionamento em variáveis independentes.

Os pontos sobre o perímetro de cada região livre (pontos nos quais a forma pode ser posicionada) são mapeados em um parâmetro uniforme $t \in [0, 1[$. Este

mapeamento é feito considerando um vértice de referência sobre o perímetro (neste trabalho, é escolhido o vértice com a menor coordenada x). Este vértice de referência corresponde ao ponto de posicionamento para $t = 0$. A partir deste vértice, o perímetro é percorrido em sentido anti-horário e os seus pontos são mapeados de modo uniforme no intervalo $[0, 1[$.

Resta a questão da decisão de em qual sub-região conexa livre posicionar a forma. Para tanto, as sub-regiões são ordenadas segundo algum critério arbitrário (neste trabalho foi empregada uma ordenação baseada na coordenada x do vértice mais a esquerda de cada área). Então, com base nesta ordenação, a cada região é atribuída uma divisão uniforme do intervalo $[0, 1[$. O parâmetro f de uma forma dita em qual área livre ela será posicionado.

Por exemplo, supondo que para uma forma em particular, com uma determinada orientação, existam quatro regiões livres, A_1, A_2, A_3 e A_4 . Supondo que o processo de ordenação produziu a seqüência (A_2, A_1, A_3, A_4) . Assim, a forma será posicionada na área:

$$\begin{aligned} A_2 & \text{ se } f \in [0, 1/4[, \\ A_1 & \text{ se } f \in [1/4, 1/2[, \\ A_3 & \text{ se } f \in [1/2, 3/4[, \\ A_4 & \text{ se } f \in [3/4, 1[\end{aligned}$$

Assim, a translação de cada forma é derivada dos parâmetros contínuos (t, f) , produzidos pelo processo de recozimento simulado. O posicionamento a partir destes parâmetros para uma forma é ilustrado pela figura 4.5.

Assim como na rotação (vide seção 3.3.2), a vizinhança de uma solução de translação é um conjunto de vetores $(t, f)_i$ nos quais *apenas um* par de parâmetros difere da solução original. Assim, o recozimento simulado modifica a translação de apenas uma forma.

4.5 Geração de configurações de posicionamento

A configuração inicial é gerada aleatoriamente.

A cada etapa, a configuração anterior é modificada de modo a gerar uma nova solução a ser explorada. Como já mencionado, a seleção de uma solução na vizinhança da solução atual consiste em modificar a solução atual em apenas um aspecto dentre ordem de posicionamento, rotação das formas e translação das formas.

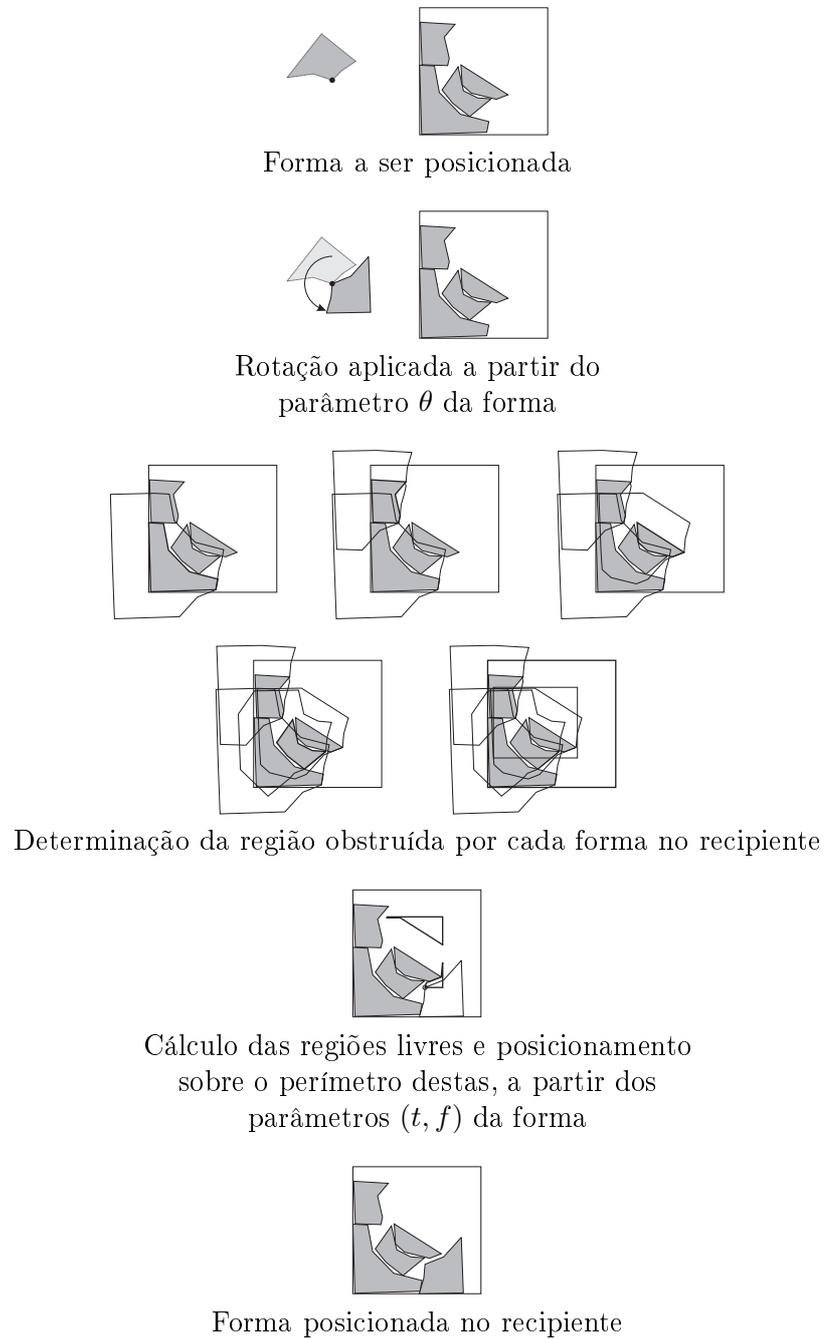


Figura 4.5: Exemplo de posicionamento de uma forma durante uma etapa do processo de otimização.

A seleção do aspecto a ser modificado é equi-provável, de modo que cada aspecto tem 1/3 de probabilidade de ser modificado.

No caso da modificação da ordem, duas peças são eleitas ao acaso, e sua ordem de posicionamento é permutada.

No caso da modificação da rotação ou translação, uma forma é escolhida aleatoriamente com uma distribuição uniforme. Como visto, à forma escolhida estão associados parâmetros $(\theta, (t, f))$, responsáveis pela rotação e translação da mesma. Estes parâmetros contínuos são modificados todos da mesma maneira. Seja w_i o parâmetro da forma a ser modificado. Como todos são definidos em $[0, 1[$, a modificação para gerar o parâmetro w_i^* é feita de acordo com a equação 4.1.

$$\begin{aligned} w_i^* &= (w_i + \Delta_{w_i} \bmod 1) \\ -1/2 &\leq \Delta_{w_i} < 1/2 \end{aligned} \tag{4.1}$$

O novo parâmetro w_i^* é gerado somando-se ao parâmetro original w_i um número aleatório Δ_{w_i} gerado entre -1/2 e 1/2, e, caso o resultado ultrapasse os limites, é somado ou subtraído de 1. O número aleatório Δ_{w_i} *não* é gerado de forma uniforme, mas com uma distribuição que depende do *fator de cristalização* associado ao parâmetro.

4.5.1 Fator de cristalização

É fácil perceber que no procedimento do recozimento simulado, soluções rejeitadas não contribuem para o progresso do processo de otimização. Desta maneira, a distribuição de probabilidade de seleção de soluções na vizinhança da solução atual é adaptada de modo a aumentar a probabilidade de se selecionar uma solução que será aceita.

Isso pode ser feito através de um mecanismo de realimentação adicionado ao recozimento simulado. Este mecanismo emprega o resultado da aceitação ou não da atual solução explorada para corrigir a distribuição da nova solução a ser explorada. É importante observar que esta correção deve manter uma certa anisotropia e simetria das distribuições de vizinhança, caso contrário o recozimento simulado pode se degenerar em um método de gradiente descendente. Um mecanismo semelhante foi adotado por Heckman e Lengauer em ^[23], onde eles dividiram as regiões de vizinhança da solução atual em zonas simétricas de acordo com a sua distância até a solução. Quando uma solução explorada em uma de-

terminada zona é aceita pelo recozimento simulado, a probabilidade de se gerar soluções em zonas com distâncias semelhantes é aumentada.

O mecanismo adotado neste trabalho é o fator de cristalização. O fator de cristalização é uma variável associada a um parâmetro de uma forma que está relacionada com a “amplitude” provável de movimento deste parâmetro quando ele é modificado em uma solução por outro parâmetro em sua vizinhança. O fator de cristalização está associado tanto aos parâmetros θ_i de rotação das formas (vide seção 3.3.2) quanto aos parâmetros (f_i, t_i) de translação (vide seção 3.3.3). Sua principal função é reduzir o número de soluções rejeitadas pelo recozimento simulado, através da redução na amplitude de movimento

De acordo com a heurística adotada neste trabalho, os parâmetros contínuos w_i são modificados através da adição de uma perturbação aleatória Δ_{w_i} sorteada no intervalo $[-1/2, 1/2[$. Sendo C_{w_i} o fator de cristalização associado ao parâmetro w_i , tem-se:

$$\Delta_{w_i} = \frac{\sum_{C_{w_i}} U(-1/2, 1/2)}{C_{w_i}} \quad (4.2)$$

onde $U(-1/2, 1/2)$ é uma variável aleatória de *distribuição uniforme* no intervalo $[-1/2, 1/2[$.

Como mostra a equação 4.2, Δ_{w_i} é o resultado da média de C_{w_i} sorteios uniformes entre $-1/2$ e $1/2$, onde C_{w_i} é o fator de cristalização associado ao parâmetro w_i .

Pode-se assim de acordo com as equações 4.1 e 4.2, obter do novo parâmetro w_i^* o seu valor esperado $\mu_{w_i^*}$ e sua variância $\sigma_{w_i^*}^2$:

$$\begin{aligned} \mu_{w_i^*} &= w_i \\ \sigma_{w_i^*}^2 &= \frac{1}{12C_{w_i}} \end{aligned}$$

Isso mostra que a distribuição do parâmetro modificado w_i^* é centrada no valor original w_i , de modo que a anisotropia do processo é respeitada. Por outro lado, percebe-se que a variância do parâmetro modificado decresce com o valor C_{w_i} . Assim, um parâmetro cujo fator de cristalização for maior sofrerá modificações de amplitudes menores.

A realimentação ocorre da seguinte maneira: no início, todos os parâmetros possuem o fator de cristalização 1. Quando a nova solução gerada pela modificação do parâmetro w_i é rejeitada, o fator de cristalização C_{w_i} é incrementado.

Quando a nova solução é aceita, o fator de cristalização C_{w_i} volta ao valor original 1.

Formas que atingem configurações que levam a valores baixos da função objetivo tendem a ter modificações em seus parâmetros rejeitadas pelo recozimento simulado, particularmente a baixas temperaturas do processo (visto que estas fatalmente levariam a valores maiores da função objetivo). Assim, a realimentação faz com que os fatores de cristalização dos seus parâmetros cresça, o que reduz a amplitude de movimentação de tais formas (nesta situação a forma é dita “cristalizada”).

Por outro lado, formas em configurações sub-ótimas possuem uma boa probabilidade de terem modificações em seus parâmetros aceitas pelo recozimento simulado. Neste caso, a realimentação diminui os fatores de cristalização de seus parâmetros, aumentando conseqüentemente a amplitude de movimentação destas formas.

4.6 Avaliação da função objetivo a partir de parâmetros de posicionamento

Como visto, o posicionamento das formas é determinado pelos seguintes parâmetros produzidos pelo processo de recozimento simulado:

- Ordem de posicionamento das formas
- Para cada forma, parâmetros $(\theta, (t, f))$, onde θ é a rotação da forma, f define a região livre na qual posicionar a forma e t define onde posicionar a forma na borda da região livre.

A função objetivo a ser minimizada é o espaço não ocupado no recipiente.

A avaliação da função objetivo ocorre da maneira descrita na figura 4.6. Observa-se que o cálculo da função objetivo incorre em extensas operações geométricas com as formas e o recipiente, nas etapas de determinação das regiões obstruídas por cada forma já inserida e na determinação das regiões livres de colisão. É importante para o desempenho do cálculo da função objetivo que estas etapas sejam executadas o mais rapidamente possível.

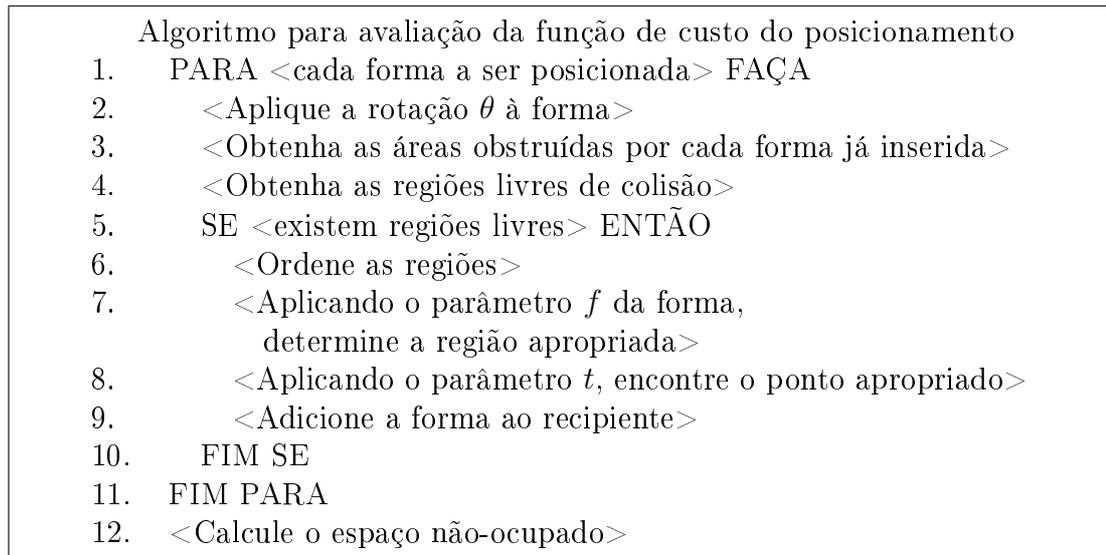


Figura 4.6: Avaliação da função objetivo do posicionamento.

4.6.1 Desempate de soluções com custo idêntico

Como mencionado na diretriz 4.4, o desempenho do algoritmo é prejudicado pela existência de soluções vizinhas de custo idêntico. Ora, como apontado na seção 2.5, a função objetivo do problema de posicionamento em recipientes de dimensões fixas apresenta regiões de custo constante. Isso ocorre devido ao fato de que todo conjunto de soluções distintas somente pelos parâmetros de formas que não foram posicionadas apresentam custo idêntico (visto que formas não posicionadas não afetam o espaço não-ocupado). Assim, é interessante adicionar uma heurística de desempate para determinar quais destas soluções estão mais próximas do ótimo.

É razoável considerar que a proximidade do ótimo está relacionada com a proximidade em uma solução de encaixar uma nova forma. Assim, para cada forma não posicionada, pode ser feita uma busca binária de profundidade limitada para se encontrar um fator de escala que aplicado à forma permite o seu posicionamento. Isso permite identificar situações onde uma forma está prestes a ser posicionada, e modificar a função objetivo para considerar esta situação.

A adição deste procedimento mitiga os problemas causados pelo caráter discreto da função objetivo do problema de posicionamento em recipientes de dimensões fixas descrito na seção 2.5, pois o seu principal efeito é o de introduzir nesta níveis de valor intermediários.

É importante observar que cada nível adicional de profundidade na busca pelo fator de escala implica em um novo cálculo da função objetivo descrita no algoritmo da figura 4.6. Assim, é importante balancear o ganho do desempenho

do recozimento simulado com um número maior de níveis de valores para a função objetivo contra o custo computacional adicional por iteração no cálculo da função objetivo.

4.7 Conclusões

A meta-heurística adotada neste trabalho para abordar o problema do posicionamento é o *recozimento simulado*.

O recozimento simulado é uma heurística de otimização probabilística de gradiente ascendente que explora aleatoriamente o espaço do problema, amostrando um grande número de soluções e avaliando nestas o valor da função objetivo.

Para um bom desempenho da heurística, é importante que esta seja *completa*, ou seja, que haja uma probabilidade não-nula de que esta visite uma solução ótima do problema.

Verifica-se que é possível criar heurísticas probabilísticas completas baseadas em recozimento simulado para cada um dos sub-problemas do posicionamento, a saber, a determinação de uma ordem de posicionamento das formas, a determinação da rotação das formas e a determinação das translações das formas.

Como no recozimento simulado as soluções rejeitadas não contribuem para o progresso do processo de otimização, criou-se um processo de realimentação que corrige as distribuições das soluções exploradas de modo a aumentar a probabilidade de que uma solução seja aceita, mas sem que o recozimento simulado perca a sua característica de gradiente ascendente. Este processo consiste em limitar modificações que no histórico do processo apresentam uma tendência de gerar soluções rejeitadas.

Os problemas com o caráter discreto da função objetivo do problema de posicionamento em recipientes com dimensões fixas (vide seção 2.5) podem ser mitigados através de uma modificação desta, que busca diferenciar soluções com custos idênticos estimando qual destas estaria mais próxima do ótimo. Isto é feito estimando-se o quão próxima uma solução está de ter uma nova forma encaixada em seu leiaute. Para tanto, para cada forma não posicionada em uma solução é realizada uma busca binária de profundidade limitada por fatores de escala que, quando aplicados às formas ainda não-encaixadas, possibilitariam o seu encaixe.

Avaliando-se o todo o processo do recozimento simulado, percebe-se que o

maior custo computacional deste reside no cálculo da função objetivo, o qual faz uso extenso de operações geométricas sobre as formas. Este custo é agravado quando se adota o processo de desempate de soluções idênticas, pois a quantidade de testes de posicionamento de formas aumenta linearmente com o número de níveis de profundidade das buscas adotadas.

Assim, embora a heurística proposta tenha a capacidade potencial de abordar com sucesso o problema do posicionamento estudado, é fundamental para a sua viabilidade que seja possível realizar as operações geométricas de determinação de polígonos de obstrução e áreas de colisão com o menor custo computacional possível.

5 Implementação do processo

5.1 Introdução

Conforme mencionado na seção 4.2, Meta-Heurísticas de otimização probabilísticas como o recozimento simulado realizam um grande número de avaliações da função objetivo, de modo que é crucial que o custo computacional do processo de avaliação descrito na seção 4.6 seja o menor possível.

De fato, quando se compara em uma iteração do recozimento simulado (vide apêndice A para mais detalhes) o custo computacional da avaliação da função objetivo com o das operações restantes, verifica-se que o custo destas é desprezível, e o desempenho final do processo depende exclusivamente do desempenho do algoritmo de avaliação da função objetivo.

Como se vê na figura 4.6, o custo computacional do algoritmo de avaliação da função objetivo está sobretudo no laço interno que determina a posição no recipiente em que uma dada forma deve ser inserida.

Esta determinação consiste basicamente em operações geométricas realizadas sobre a forma a ser posicionada, o recipiente e as formas já posicionadas.

Neste trabalho, as formas serão representadas por polígonos. A representação poligonal é uma representação *fechada* no que diz respeito às operações envolvidas no algoritmo de avaliação da função objetivo. Isso significa que as operações de rotação, subtração e geração polígonos de obstrução, quando aplicadas a formas poligonais, produzem somente formas poligonais. Além disso, dadas algumas restrições (vide adiante a seção 5.2.1), esta é uma representação computacionalmente eficiente, para a qual existem algoritmos estabelecidos para todas as operações geométricas necessárias neste trabalho.

Quanto a formas não-poligonais, estas podem ser aproximadas por polígonos com precisão arbitrária como mostra a figura 5.1.

Resumindo, as operações necessárias para se determinar a posição de um

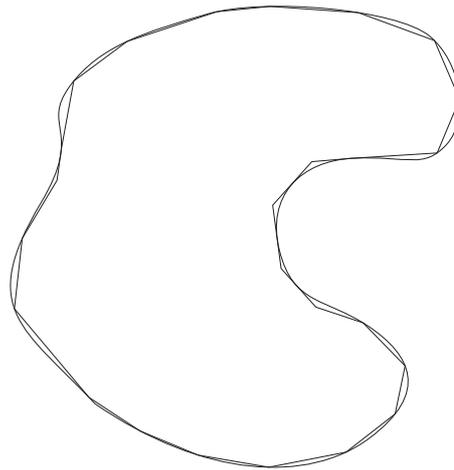


Figura 5.1: Forma não-poligonal aproximada por um polígono.

polígono, dado o recipiente, as formas poligonais já inseridas e dos seus parâmetros (θ, f, t) são:

- Aplicação da rotação θ_i à forma poligonal a ser posicionada
- Geração da região livre de obstrução
 - Geração da região de posicionamento interno do recipiente
 - Geração dos polígonos de obstrução para cada forma poligonal já posicionada
 - Subtração booleana dos polígonos de obstrução da região de posicionamento interno do recipiente
- Mapeamento da área livre de obstrução
- Determinação do ponto de posicionamento a partir do mapeamento e dos parâmetros f e t

A busca por eficiência computacional nestas operações se concentra em duas frentes:

- *Reduzir o custo computacional de cada operação.* Consiste em buscar representações e algoritmos eficientes para cada operação descrita acima. Isto será discutido na seção 5.2.
- *Usar de forma racional as operações.* Consiste em buscar armazenar resultados intermediários das operações e tentar identificar situações onde estes resultados possam ser reaproveitados. Isto será discutido na seção 5.3.

5.2 Operações geométricas sobre polígonos

As operações realizadas são de rotação, geração de polígonos de obstrução e determinação de região livre de obstrução. A determinação da região livre de obstrução é obtida subtraindo-se da região de posicionamento interno os polígonos de obstrução induzidos pelos polígonos já inseridos.

5.2.1 Representação computacional de polígonos

Polígonos são conjuntos de pontos delimitados por uma fronteira contínua composta por um número finito de segmentos de reta. É trivial observar que cada segmento de reta é delimitado por dois vértices, e cada vértice é compartilhado por dois segmentos. Assim, o polígono pode ser representado pela seqüência de seus vértices quando o seu perímetro é percorrido em um sentido convencional. A convenção mais comum é utilizar a seqüência de vértices obtida com o percurso no sentido *anti-horário* (vide figura 5.2).

$$P = \{p_1, p_2, p_3, \dots, p_n\}$$

Nota-se que a representação adotada não é única, pois a escolha do ponto inicial da seqüência é arbitrária. Assim,

$$P = \{v_1, v_2, v_3, \dots, v_n\} \equiv \{v_n, v_1, v_2, \dots, v_{n-1}\} \quad (5.1)$$

Ou seja, polígonos são invariantes quanto a *permutações circulares*¹ da seqüência de pontos que os representa. Da mesma forma, esta seqüência de vértices pode ser estendida para uma seqüência infinita, sem início nem fim $P = \{\dots, p_1, p_2, p_3, \dots, p_n, \dots\}$ na qual vale $p_i \equiv p_{i \bmod n}$. Neste caso, o vértice sucessor de p_n é o vértice $p_{n+1} \equiv p_1$ e do mesmo modo, o vértice antecessor de p_1 é o vértice $p_0 \equiv p_n$.

No plano xy , cada vértice pode ser representado univocamente por suas coordenadas (x, y) .

$$v_i = \begin{vmatrix} x_i \\ y_i \end{vmatrix} \quad (5.2)$$

¹Uma *permutação circular* de uma seqüência é obtida dividindo-se a seqüência em duas sub-seqüências e permutando-se a ordem destas.

Chega-se enfim à representação computacional mais usual para polígonos, uma seqüência de pares ordenados $P = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$.

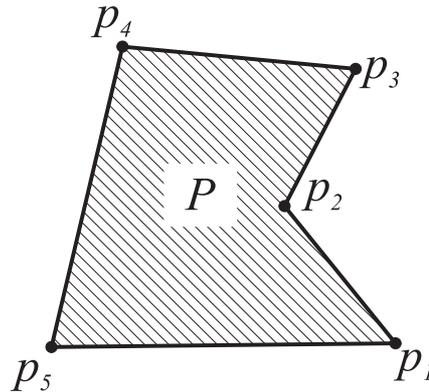


Figura 5.2: Polígono e uma seqüência de pontos que o representa.

É desejável, para assegurar o desempenho dos algoritmos que tratam os polígonos, empregar representações nativas de números em computadores com precisão finita para as coordenadas (x_i, y_i) . No entanto, verifica-se que polígonos com coordenadas representadas com precisão finita não formam um conjunto *fechado* quanto a operações booleanas de adição e subtração, ou seja, existem situações em que o resultado da operação booleana de dois polígonos representados por precisão finita *não* é representável por precisão finita.

De fato, as operações booleanas de adição e subtração de polígonos consistem basicamente em se determinar a intersecção entre as suas arestas (vide figura 5.3), quebrar as arestas nestes pontos de intersecção, classificar as regiões geradas e coletar as regiões de acordo com a operação que se deseja realizar ^[36].

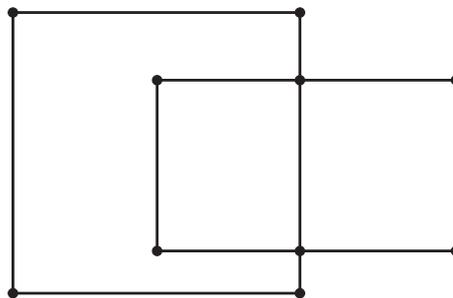


Figura 5.3: Sobreposição e intersecção de dois polígonos.

O problema geral da intersecção exata de arestas requer o uso de números racionais. Como visto na figura 5.4, mesmo vértices cujas arestas correspondem a pontos com coordenadas inteiras podem produzir intersecções com coordenadas

racionais².

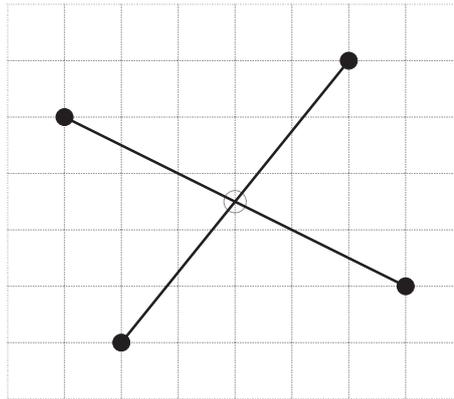


Figura 5.4: Intersecção de dois segmentos com coordenadas inteiras produzindo um ponto com coordenadas não-inteiras.

Pode-se mostrar que o uso de memória para se trabalhar com polígonos com coordenadas racionais cresce exponencialmente com o número de operações booleanas realizadas nos polígonos ^[37]. Para uma aplicação como a deste projeto, o custo computacional de operações booleanas exatas é impraticável, de modo que optou-se por adotar operações *aproximadas*, que mantêm todas as coordenadas inteiras (vide a figura 5.5).

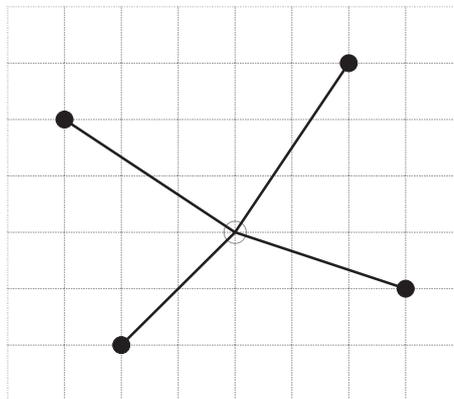


Figura 5.5: Aproximação com coordenadas inteiras da intersecção de dois segmentos.

Computacionalmente, números em ponto flutuante são muito utilizados para se determinar o ponto de intersecção aproximado entre duas arestas. Entretanto, quando deseja-se determinar o polígono resultante da operação booleana aplicada a dois polígonos, além da determinação do ponto de intersecção entre arestas, é necessário verificar se existe a coincidência entre os vértices dos dois polígonos.

²Mesmo os números de ponto flutuante disponíveis na maioria dos computadores contemporâneos não basta, visto que estes possuem precisão finita.

A coincidência entre dois números em ponto flutuante é verificada fazendo-se uso de uma tolerância:

$$|f1 - f2| < tol$$

onde $f1$ e $f2$ são os dois números em ponto flutuante que se deseja comparar e tol é a tolerância em ponto flutuante que indica se dois números estão suficientemente próximos. Esta verificação é conhecida como comparação por tolerância. Uma vez que dois valores são considerados iguais segundo a tolerância utilizada, um deles será substituído pelo outro. A comparação por tolerância é comutativa, mas não é associativa, implicando que o resultado obtido está associado à seqüência em que as comparações por tolerância foram feitas.

A aritmética intervalar é uma outra forma robusta para representar coordenadas. Cada coordenada que define um vértice é representada por um intervalo $[a, b]$, onde a é o limite inferior do intervalo e b é o limite superior. As operações aritméticas entre intervalos são definidas de modo que o resultado é um novo intervalo que contenha o maior e o menor resultados possíveis da operação quando esta é aplicada aos valores contidos nos intervalos originais, podendo ser necessárias para uma única operação intervalar até quatro operações em ponto flutuante. Assim, a cada operação uma imprecisão é acrescentada, mas que mantém o resultado final coerente independentemente da seqüência em que as operações foram feitas. Entretanto, dependendo do número de operações, o resultado final pode ser muito abrangente e conseqüentemente a sua utilidade duvidosa ^[38].

A representação de coordenadas por números inteiros em precisão limitada também é robusta. O resultado das operações aritméticas é arredondado para o número inteiro mais próximo, a fim de garantir o fechamento das operações aritméticas. A comparação é feita de modo direto sem a necessidade de nenhuma tolerância ou propagação de imprecisão.

A representação das coordenadas dos vértices dos polígonos deve ser escolhida de modo que a implementação das operações booleanas deve ser robusta, rápida, uso razoável de memória e tenha resultados úteis. Neste sentido optou-se pela representação de coordenadas por números inteiros.

Neste trabalho adotou-se a biblioteca *Polyboolean* para realizar as operações booleanas sobre polígonos. Esta é uma biblioteca de operações booleanas aproximadas sobre polígonos que utiliza números inteiros de precisão limitada para as coordenadas de vértices ^[39].

Dentre as características da biblioteca destaca-se:

- Coordenadas de vértices com precisão limitada a 20 *bits*. A biblioteca trata todas as coordenadas como um número binário de 20 *bits*. Operações de intersecção de vértices realizam aproximações para “encaixar” os vértices resultantes nas coordenadas inteiras. A precisão de 20 *bits* pode ser considerada suficiente para a aplicação em problemas de recorte e posicionamento.
- Polígonos representados por meio de listas ligadas de vértices. A seqüência de vértices que representa o polígono (vide acima) é representada internamente por meio de uma *lista circular duplamente ligada* de vértices. Isso significa que os vértices podem ser acessados eficientemente de forma seqüencial nas duas direções. Além disso, por ser a fila circular, o sucessor do último vértice é o primeiro.

5.2.2 Rotação

A rotação de um polígono de θ no sentido anti-horário pode ser realizada por uma transformação linear trivial dos vértices do polígono conforme mostrado abaixo:

$$v'_i = T \cdot v_i$$

$$T = \begin{vmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{vmatrix}$$

A dificuldade é que, como as coordenadas de cada vértice devem ser números inteiros (vide seção 5.2.1), é necessário fazer um arredondamento do resultado. Neste trabalho, os resultados são arredondados para o *inteiro mais próximo*, como mostra a equação 5.3.

$$\tilde{x} = \lfloor x + 1/2 \rfloor \tag{5.3}$$

Chega-se assim ao algoritmo para realizar rotações *aproximadas* de polígonos empregado neste trabalho.

Observa-se que o algoritmo 5.6 acessa os vértices de forma *seqüencial*, o que pode ser feito de forma eficiente com a estrutura de dados empregada (vide seção 5.2.1). O cálculo de $\cos\theta$ e $\sin\theta$ usa rotinas de ponto flutuante (geralmente executadas com um co-processador específico). Este é o *único* uso de ponto flutuante nas operações geométricas.

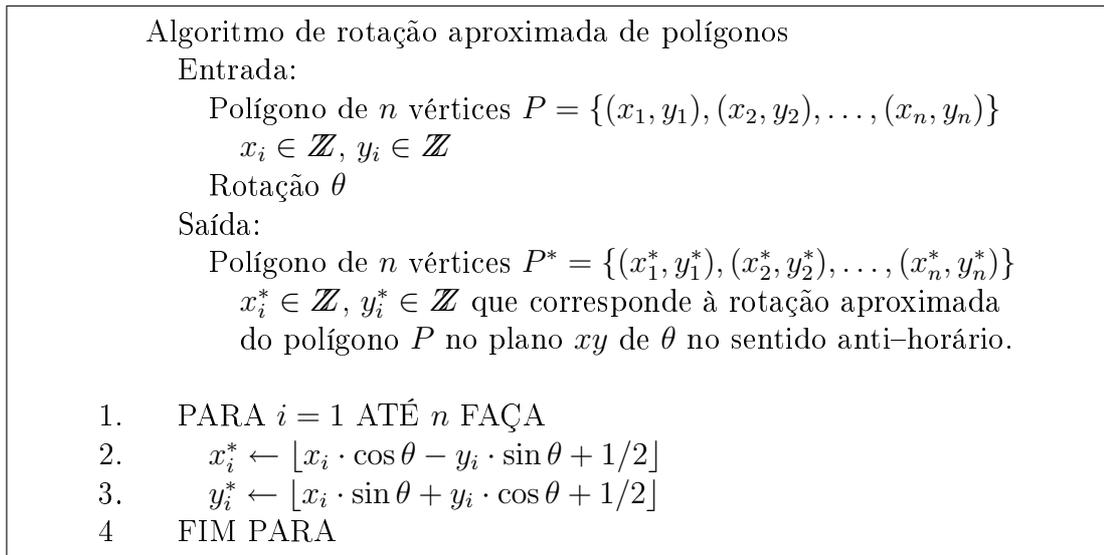


Figura 5.6: Algoritmo para rotação aproximada de polígonos.

5.2.3 Algoritmo de soma de Minkowski para polígonos convexos

Conforme visto na seção 3.2, o polígono de obstrução ($A \ominus B$) induzido por um polígono convexo A a um outro polígono convexo B é o polígono convexo gerado a partir das arestas dos polígonos A e B (sendo que as arestas do polígono A são invertidas).

5.2.3.1 Ordenação de vértices

Neste momento, é oportuno introduzir a notação vetorial para as arestas de um polígono. O vetor relacionado à i -ésima aresta \vec{e}_i de um polígono P é o vetor que *parte* do vértice v_i e vai até o próximo vértice na seqüência v_{i+1} . A seqüência $E(P)$ de arestas de um polígono P é.

$$\vec{e}_i = v_{i+1} - v_i \equiv \begin{vmatrix} x_{i+1} - x_i \\ y_{i+1} - y_i \end{vmatrix} \quad (5.4)$$

Como necessariamente $\sum \vec{e}_i = \vec{0}$ (visto que os vetores percorrem um circuito fechado), tem-se $\vec{e}_n = v_1 - v_n$, o que é natural, pois os polígonos são invariantes quanto a permutações circulares de sua seqüência de vértices.

O algoritmo de soma de Minkowski constrói o polígono convexo com os vértices dos dois polígonos originais. É fácil mostrar que a condição necessária e suficiente para que um polígono seja convexo é que, na sua seqüência de arestas, nenhum vetor esteja à *direita* do seu predecessor.

O ângulo do vetor \vec{e}_1 ao vetor \vec{e}_2 é o ângulo medido no sentido anti-horário a partir de \vec{e}_1 até \vec{e}_2 quando as origens dos dois vetores forem coincidentes. Por esta definição, é possível encontrar sempre dois valores de ângulo, sendo um positivo e outro negativo. Assim, o menor ângulo em módulo dentre os dois possíveis valores é chamado de ângulo interno. Nota-se que o valor do ângulo interno pode ser tanto positivo quanto negativo.

Definição 5.1 Diz-se que um vetor \vec{e}_1 está a direita de um vetor \vec{e}_2 quando o ângulo interno de \vec{e}_1 a \vec{e}_2 for negativo.

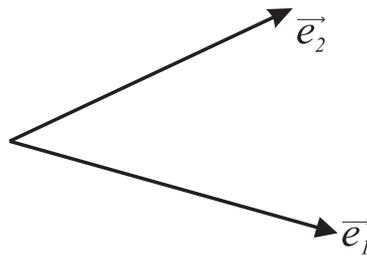


Figura 5.7: Ordenação de dois vetores. O vetor \vec{e}_1 está a *direita* de \vec{e}_2 .

Uma outra maneira de se determinar se um vetor \vec{e}_1 está à direita de outro vetor \vec{e}_2 é fazendo uso do produto vetorial. O resultado do produto vetorial de dois vetores é um vetor perpendicular aos dois cujo módulo é igual ao produto do módulo dos vetores vezes o seno do ângulo entre eles. Assim, o sentido do produto vetorial muda conforme o ângulo interno entre os vetores tem sentido horário ou anti-horário.

Como os vetores estão contidos no plano xy , vamos definir um produto escalar \otimes derivado do produto vetorial, que corresponde à componente z do produto vetorial:

$$\vec{e}_1 \otimes \vec{e}_2 \equiv (x_1, y_1) \otimes (x_2, y_2) = x_1 \cdot y_2 - x_2 \cdot y_1 \quad (5.5)$$

Assim, o vetor \vec{e}_1 está à direita do vetor \vec{e}_2 se e somente se $\vec{e}_1 \otimes \vec{e}_2 < 0$. Deste modo, pode-se construir um teste rápido para classificar dois vetores, conforme mostra o algoritmo da figura 5.8:

5.2.3.2 Montagem do polígono de obstrução

Como visto na figura 3.4, o polígono de obstrução $A \ominus B$, que representa a região proibida ao posicionamento de um polígono B por um obstáculo A , é construído

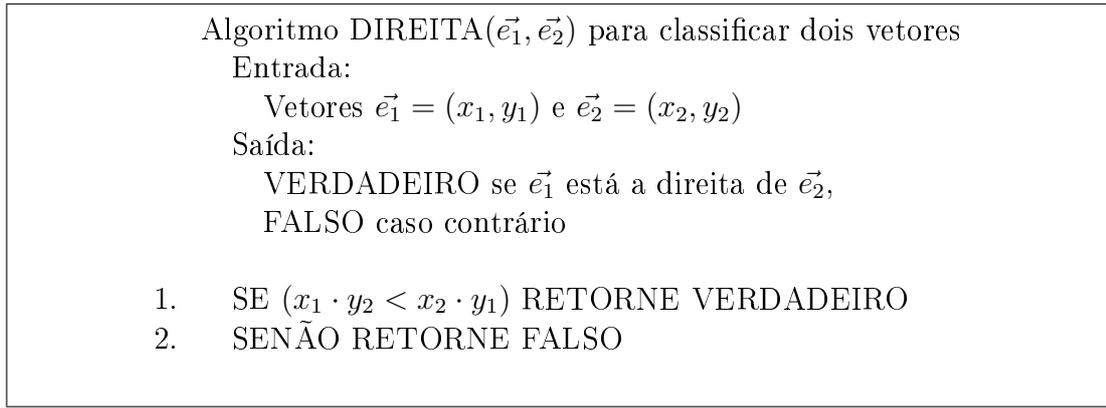


Figura 5.8: Algoritmo para classificação de dois vetores.

a partir das arestas dos polígonos A e B , sendo que as arestas do polígono A e as arestas do polígono B são tomadas *invertidas*.

Decorre da equação 5.4 que os vértices v_i de um polígono podem ser facilmente reconstruídos a partir da sua seqüência de arestas \vec{e}_i e do seu vértice inicial v_1 usando a expressão $v_{i+1} = v_i + \vec{e}_i$.

Dado que o polígono de obstrução é convexo, sejam as arestas dos polígonos originais $E(A) = \{\vec{e}_1^A, \vec{e}_2^A, \dots, \vec{e}_n^A\}$ e $E(B) = \{\vec{e}_1^B, \vec{e}_2^B, \dots, \vec{e}_m^B\}$, as arestas do polígono de obstrução $E(B \ominus A)$ são uma ordenação particular do conjunto de arestas $\{\vec{e}_1^A, \vec{e}_2^A, \dots, \vec{e}_n^A\} \cup \{-\vec{e}_1^B, -\vec{e}_2^B, \dots, -\vec{e}_m^B\}$ na qual nenhuma aresta está a direita de sua antecessora.

Vale observar que as seqüências de arestas $\{\vec{e}_1^A, \vec{e}_2^A, \dots, \vec{e}_n^A\}$ e $\{-\vec{e}_n^B, -\vec{e}_{n-1}^B, \dots, \vec{e}_1^B\}$ individualmente já possuem a ordenação desejada. Assim, a seqüência de arestas de $A \ominus B$ pode ser construída entremeando-se as duas seqüências de arestas de modo que a ordenação seja mantida.

Para tanto, inicialmente considera-se permutações circulares das duas seqüências $\{\vec{e}_1^{A*}, \vec{e}_2^{A*}, \dots, \vec{e}_n^{A*}\}$ e $\{-\vec{e}_n^{B*}, -\vec{e}_{n-1}^{B*}, \dots, \vec{e}_1^{B*}\}$ de modo que $-\vec{e}_n^{B*}$ esteja à direita de \vec{e}_1^{A*} , que por sua vez, esteja à direita de $-\vec{e}_1^{B*}$. O vértice inicial do polígono de obstrução é obtido por $a_i - b_j$, onde a_i é o vértice do qual parte a aresta \vec{e}_1^{A*} , e b_j é o vértice do qual parte a aresta $-\vec{e}_n^{B*}$.

Nota-se que o algoritmo descrito na figura 5.9 faz uso exclusivo de operações inteiras. Além disso, o acesso às coordenadas dos vértices dos polígonos é feita sempre de modo seqüencial, o que faz com que este algoritmo seja implementável de forma eficiente com a estrutura de dados empregada para representar polígonos (vide seção 5.2.1).

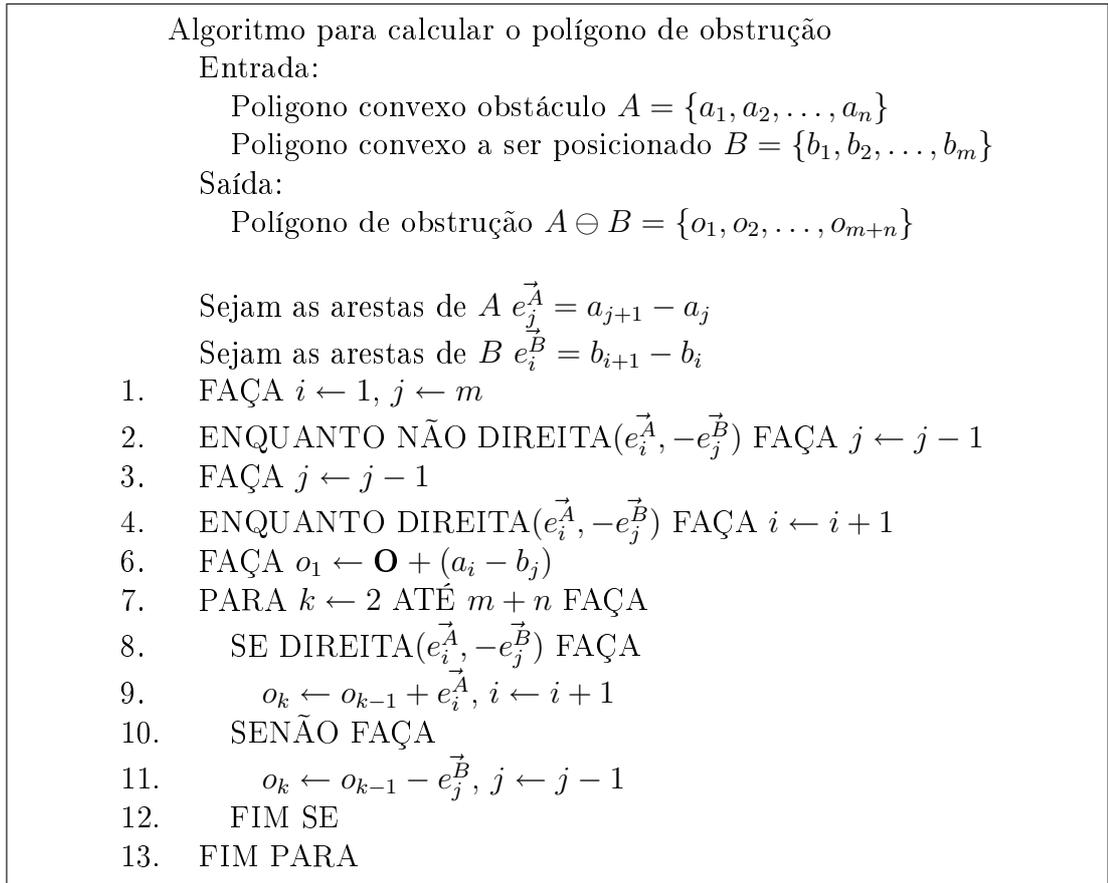


Figura 5.9: Algoritmo para construção do polígono de obstrução.

5.3 Armazenamento e reuso de resultados geométricos intermediários

Conforme mencionado previamente, o processo de cálculo da função objetivo descrito pelo algoritmo na figura 4.6 e na figura 4.5 são os responsáveis pela maior parte do custo computacional do processo de otimização, devido ao grande número de operações geométricas envolvidas na determinação dos polígonos de obstrução e das regiões livres de obstrução.

Deste modo, é interessante investigar o quanto do processamento realizado em uma iteração do processo de otimização pode ser reaproveitado na iteração seguinte.

As heurísticas de vizinhança adotadas (vide seções 4.4 e 4.5) asseguram que a solução avaliada em uma dada iteração difere da solução atual *apenas* quanto a:

- Rotação de uma forma
- Parâmetros de translação de uma forma

- Permutação da ordem de posicionamento de duas formas

Sendo que cada modificação tem probabilidade idêntica de ocorrer. Assim, verifica-se que a solução avaliada em uma iteração é bastante similar à solução atual, o que sugere que no processamento do cálculo da sua função objetivo deve ser possível re-utilizar dados intermediários obtidos no cálculo da solução atual.

5.3.1 Polígonos de Obstrução

Como visto na seção 4.6, em cada etapa do posicionamento, para determinar se a i -ésima forma pode ou não ser encaixada no recipiente, é necessário calcular os $i - 1$ polígonos de obstrução correspondentes às formas já posicionadas. Assim, em um problema com n formas, são necessários $(n^2 - n)/2$ polígonos de obstrução por iteração do processo de otimização.

No entanto, como visto acima, as modificações por iteração nos parâmetros das formas são limitadas, de modo que é interessante determinar a relação entre os polígonos de obstrução em uma iteração com os polígonos da iteração anterior.

Quando ocorre a modificação de translação de uma forma, um obstáculo A é substituído por um obstáculo $A + \vec{t}$, onde \vec{t} representa a modificação de posição do obstáculo entre as iterações. Observa-se que

$$A + \vec{t} = \{a + \vec{t} \mid a \in A\} \quad (5.6)$$

O polígono de obstrução gerado no posicionamento da forma B passa a ser portanto $(A + \vec{t}) \ominus B$. Substituindo na equação 3.1 (vide a definição 3.2) chega-se a:

$$(A + \vec{t}) \ominus B = \{O + \vec{v} \mid \exists b \in B \mid b + \vec{v} \in (A + \vec{t})\}$$

Da equação 5.6 observa-se que $b + \vec{v} \in (A + \vec{t}) \iff b + \vec{v} + \vec{t} \in A$, de modo que chega-se a

$$(A + \vec{t}) \ominus B = \{O + \vec{v} \mid \exists b \in B \mid b + \vec{v} \in A\} + \vec{t}$$

Donde

$$(A + \vec{t}) \ominus B = (A \ominus B) + \vec{t} \quad (5.7)$$

ou seja, o polígono de obstrução produzido por um obstáculo ao qual foi aplicada uma translação \vec{t} é igual à translação do polígono de obstrução original.

Observando que a translação de um polígono é um processo muito mais simples do que o cálculo completo do polígono de obstrução (vide seção 5.2.3), verifica-se que é oportuno manter uma tabela com os polígonos de obstrução gerados quando as formas encontram-se na sua posição original. Estes polígonos de obstrução primitivos ficam armazenados em memória, sendo re-calculados somente quando a rotação do obstáculo ou da forma a ser posicionada é alterada. Define-se a tabela de polígonos de obstrução primitivos Op como mostra a equação 5.8.

$$Op = \begin{vmatrix} 0 & P_2 \ominus P_1 & P_3 \ominus P_1 & \dots & P_n \ominus P_1 \\ P_1 \ominus P_2 & 0 & P_3 \ominus P_2 & \dots & P_n \ominus P_2 \\ P_1 \ominus P_3 & P_2 \ominus P_3 & 0 & \dots & P_n \ominus P_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_1 \ominus P_n & P_2 \ominus P_n & P_3 \ominus P_n & \dots & 0 \end{vmatrix} \quad (5.8)$$

Onde P_i corresponde à i -ésima forma do problema na sua posição original. Observa-se que uma forma não pode ser obstáculo em seu próprio posicionamento, de modo que os polígonos de obstrução $P_i \ominus P_i$ não têm utilidade no processo de otimização, e a diagonal da matriz Op pode permanecer nula.

Finalmente, recordando a definição de *polígono oposto* (vide definição 3.4), é trivial mostrar que:

$$A \ominus B = -(B \ominus A) \quad (5.9)$$

Ou seja, o polígono de obstrução induzido pelo obstáculo A no posicionamento da forma B é o *polígono oposto* ao polígono de obstrução induzido pelo obstáculo B no posicionamento da forma A . Assim, pode-se dizer que a matriz Op definida na equação 5.8 é “anti-simétrica” (no sentido de que os polígonos que se encontram abaixo da diagonal principal são os opostos dos polígonos que se encontram acima).

Como visto na seção 3.2, o polígono oposto pode ser obtido facilmente através da inversão das coordenadas do polígono original. Deste modo, em um problema de posicionamento com n formas, em qualquer etapa do processo de otimização existem no máximo $\frac{n^2-n}{2}$ polígonos de obstrução primitivos em memória.

Assim, a menos de modificações na rotação das formas, não há custo computacional apreciável com o cálculo dos polígonos de obstrução. No caso de modificações de rotação, que ocorrem em $1/3$ das iterações, a rotação de *apenas uma forma* é modificada, o que invalida $n - 1$ polígonos de obstrução da tabela Op . Assim, o número esperado de cálculo de polígonos de obstrução por iteração é $(n - 1)/3$.

5.3.2 Regiões livres de colisão

O cálculo das regiões livres de colisão no posicionamento da forma P_i consiste basicamente em subtrair os $i - 1$ polígonos de obstrução gerados pelas formas precedentes da região de posicionamento interno do recipiente.

Considerando

$$A_j^i = A_{j-1}^i - (P_j \ominus P_i)$$

onde A_0^j é a região de posicionamento interno do recipiente no posicionamento da forma P_i , verifica-se que a região livre de obstrução no posicionamento da forma P_i é a região A_{i-1}^i .

Assim, no posicionamento da i -ésima forma, são necessárias $i - 1$ operações booleanas de subtração de polígonos, de modo que num problema com n formas, o número total de operações booleanas é:

$$N_{tot} = \frac{n^2 - n}{2} \approx \frac{n^2}{2} \quad (5.10)$$

para valores grandes de n

Pergunta-se qual seria o possível ganho em se manter em memória as regiões intermediárias A_j^i . Para tanto, é necessário estudar em que situações se faz necessário refazer operações booleanas, e qual a extensão destas operações.

É fácil perceber que as regiões A_j^i são modificadas apenas quando um dos polígonos de obstrução utilizado na sua construção é alterado. Esta modificação pode ocorrer de três maneiras:

- Translação de um obstáculo
- Rotação de um obstáculo
- Rotação da forma a ser posicionada

Observe que o polígono de obstrução não depende da translação da forma a ser posicionada. Quando ocorre a modificação na translação da k -ésima forma, todos os polígonos de obstrução do tipo $P_k \ominus P_i$ são modificados. Deste modo, uma modificação na translação da k -ésima forma afeta *somente* as regiões $A_j^i, j \geq k$. Deste modo, o número de operações booleanas a serem refeitas quando ocorre uma modificação na translação de uma forma de ordem de posicionamento k_t é:

$$N_t = \frac{(n - k_t)^2 + n - k_t}{2} \approx \frac{(n - k_t)^2}{2} \quad (5.11)$$

No caso da modificação ocorrer na *rotação* da k -ésima forma, o raciocínio é semelhante, no entanto, além dos polígonos de obstrução $P_k \ominus P_i$ serem modificados, os polígonos $P_j \ominus P_k$ também o são (visto que uma rotação na forma a ser posicionada afeta os seus polígonos de obstrução). Assim, além das regiões $A_j^i, j \geq k$ tornarem-se inválidas, todas as regiões A_j^k também precisam ser refeitas. Deste modo, o número de operações booleanas a serem refeitas quando ocorre uma modificação na rotação de uma forma de ordem de posicionamento k_r é:

$$N_r = \frac{(n - k_r)^2 + n - k_r}{2} + k_r - 1 \approx \frac{(n - k_r)^2}{2} \quad (5.12)$$

Para o caso de rotação simples, dado que todas as formas podem ser selecionadas com probabilidade idênticas, tem-se $E \langle k_r \rangle \approx n/2$ e $E \langle k_r^2 \rangle \approx n^2/3$, visto que k_r segue uma distribuição uniforme entre 1 e n . Deste modo, o número esperado de operações booleanas a serem refeitas no caso de uma modificação da rotação de uma forma é:

$$E \langle N_r \rangle \approx \frac{n^2}{6} \quad (5.13)$$

A distribuição dos índices k_t de modificações em translações é mais complexa, visto que uma modificação de translação pode ser oriunda tanto de uma modificação nos parâmetros (f, t) quanto de uma permutação na ordem do posicionamento de duas formas.

No caso de uma modificação nos parâmetros de translação (f, t) , o comportamento do índice da forma modificada k_f é idêntico ao índice da modificação em rotações k_r , de modo que $E \langle k_f \rangle \approx n/2$ e $E \langle k_f^2 \rangle \approx n^2/3$. Assim, o número esperado de operações booleanas a serem refeitas no caso de uma modificação nos parâmetros de translação N_f é:

$$E \langle N_f \rangle \approx \frac{n^2}{6} \quad (5.14)$$

Quando ocorre a permuta de ordem de posicionamento entre duas formas, o efeito da modificação nos polígonos de obstrução ocorre a partir da *primeira* forma permutada. Assim, se as áreas de ordem k_1 e k_2 forem permutadas, o valor de k_t a ser usado na equação 5.11 é $\min \{k_1, k_2\}$.

Este número *não* segue uma distribuição uniforme. A probabilidade de que em uma permutação de ordem de posicionamento as $k - 1$ primeiras formas tenham a ordem intocada e de que a k -ésima forma seja permutada é dada por:

$$\begin{aligned} P_n(k) &= \frac{(n-k+1)(n-k)}{n(n-1)} \cdot \frac{2}{n-k+1} \quad k \leq n-1 \\ &\Downarrow \\ P_n(k) &= \frac{2(n-k)}{n^2-n} \quad k \leq n-1 \end{aligned} \quad (5.15)$$

Onde o primeiro termo é a probabilidade de que nenhuma das duas formas permutadas estejam entre as $k - 1$ primeiras formas e o segundo é a probabilidade de que uma das formas permutadas seja exatamente a k -ésima forma.

Deste modo, sendo k_p a ordem da *primeira* forma a ser permutada, tem-se:

$$\begin{aligned} E \langle k_p \rangle &= \sum_{k=1}^{n-1} k P_n(k) = \frac{n+1}{3} \approx \frac{n}{3} \\ E \langle k_p^2 \rangle &= \sum_{k=1}^{n-1} k^2 P_n(k) = \frac{n^2+n}{6} \approx \frac{n^2}{6} \end{aligned} \quad (5.16)$$

Substituindo-se os valores de 5.16 em 5.11, conclui-se que o número esperado de operações booleanas a serem refeitas no caso de uma permutação da ordem de posicionamento de duas formas N_p é:

$$E \langle N_p \rangle \approx \frac{n^2}{4} \quad (5.17)$$

Observa-se que este valor é superior ao obtido em 5.13 e em 5.14, mostrando que a permutação de ordem de formas é a modificação que incorre no maior número de operações booleanas de subtração.

Dado que cada uma das modificações, tanto de rotação, quanto de parâmetros de translação ou de permutação de ordem de posicionamento têm uma probabilidade igual de ocorrer, o número esperado de operações booleanas a serem refeitas em uma iteração N_m é:

$$E \langle N_m \rangle = \frac{1}{3} (E \langle N_r \rangle + E \langle N_f \rangle + E \langle N_p \rangle) \approx \frac{7}{36} n^2$$

Comparando-o com o valor em 5.10,

$$\frac{E \langle N_m \rangle}{E \langle N_{tot} \rangle} \approx \frac{7}{18} \quad (5.18)$$

de modo que chega-se afinal à resposta da pergunta feita no início desta seção. Mantendo-se em memória as regiões livres de colisão intermediárias A_j^i , é possível aproveitar aproximadamente 11/18 das operações booleanas realizadas na iteração precedente, ou aproximadamente 61%.

5.4 Conclusões

É de capital importância para o processo de otimização que o custo computacional deste seja o menor possível. Verifica-se que o maior custo computacional na heurística aqui proposta está incorrido nas operações geométricas sobre as formas quando da avaliação da função objetivo do problema (vide seção 4.6).

Adotou-se neste trabalho a representação poligonal para formas, tratando com aproximações problemas que envolvam formas não-poligonais.

Operações geométricas exatas sobre polígonos possuem um custo computacional elevado demais para serem viáveis no processo de otimização aqui estudado. Sendo assim, foi adotada uma representação *aproximada*, que emprega coordenadas inteiras. A biblioteca *Polyboolean* ^[39] foi eleita para tratar das operações booleanas sobre polígonos.

Verificou-se que é possível realizar as operações de rotação e geração de polígonos de obstrução com esta representação a um baixo custo computacional.

Dado que as heurísticas de vizinhança adotadas neste processo realizam modificações limitadas na solução atual (vide seção 4.5), investigou-se a possibilidade e o ganho em se re-utilizar resultados de operações geométricas processadas em iterações anteriores.

Verificou-se que a adoção de um *cache* para os polígonos de obstrução pode reduzir o número de operações de Minkowski em uma iteração de $(n^2 - n)/2$ para $(n - 1)/3$, sendo n o número de formas do problema.

Além disso, verificou-se que a adoção de um *cache* para os resultados inter-

mediários das operações booleanas nos cálculos das áreas livres de colisão tem o potencial de economizar 61% das operações geométricas por iteração.

6 Resultados

6.1 Introdução

A seção 6.2 discute sumariamente alguns detalhes da implementação que foi realizada.

A seção 6.3 trata dos *espectros de função objetivo*, uma técnica de visualização desenvolvida neste trabalho para acompanhar a evolução do processo de otimização.

A seção 6.4 trata da avaliação do comportamento do processo quando este é submetido a problemas de posicionamento especialmente construídos para verificar propriedades específicas que ele em teoria possui.

A seção 6.5 trata da avaliação da sensibilidade do desempenho do processo de otimização a alguns de seus parâmetros.

Finalmente, a seção 6.6 compara resultados adaptados da literatura com os aqui obtidos.

6.2 Implementação

O método de otimização foi implementado no *software* `polygonFit`.

O *software* `polygonFit` foi escrito em linguagem C++, e emprega uma versão modificada da biblioteca *PolyBoolean* (vide seção 5.2.1).

O programa de resfriamento adotado nos exemplos expostos aqui foi o de resfriamento geométrico com $\alpha=0.95$ (vide apêndice A).

6.3 Visualização do processo de otimização

Para ilustrar o progresso do processo de otimização, foram adotadas seqüências de espectros do valor da função objetivo (energia) para diferentes valores de tem-

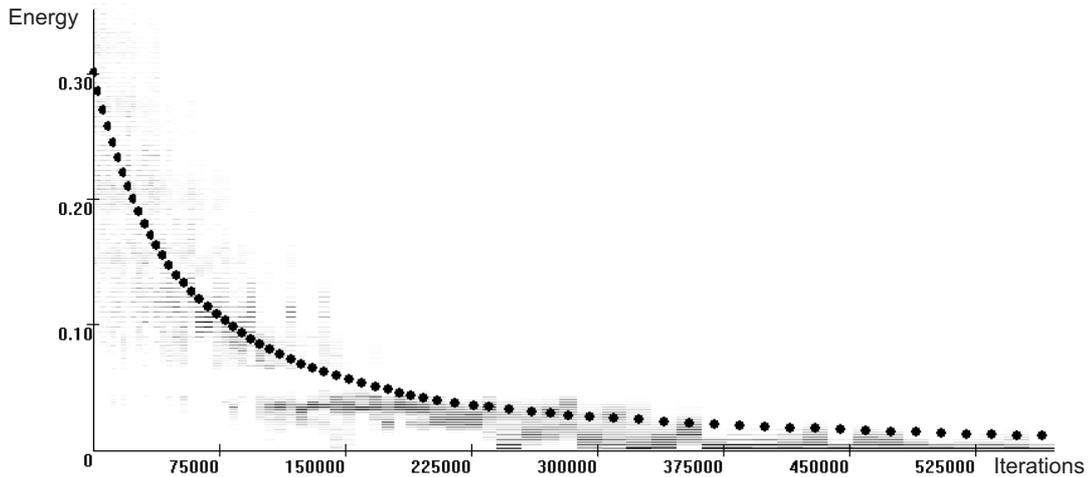


Figura 6.1: Exemplo de uma seqüência de espectros de valores da função objetivo e temperaturas para o recozimento simulado.

peratura. Assim, para uma dada temperatura, um espectro em tons de cinza com a distribuição dos valores da função objetivo é traçado. Estes espectros (na forma de barras horizontais) são combinados, com a temperatura (na forma de pontos) *versus* o número de iterações. Quanto mais escuras as barras horizontais, maior a freqüência daquele nível de energia no processo de otimização (vide figura 6.1).

A figura 6.1 foi gerada durante o processo de otimização do problema descrito na seção 6.4.2. Pode-se ver que, à medida que o programa de resfriamento progride, a temperatura diminui, levando o sistema a níveis menores de energia. A transição entre níveis no entanto não é suave. O sistema no exemplo possui duas “fases” bem pronunciadas. Isso geralmente corresponde a estados de macro-organizações no sistema (por exemplo, a transição visível na figura do exemplo corresponde ao posicionamento das duas maiores formas no problema da seção 6.4.2).

6.4 Problemas propostos

Foram criados problemas cuja solução ótima global é conhecida para avaliar a capacidade do processo de convergir para o ótimo global do problema (lembrando que a heurística criada é *completa* no sentido da definição 4.1).

Todos os problemas estudados possuem uma solução em que todos os polígonos podem ser encaixados no recipiente. Em todos os problemas, a área total do recipiente é 10% maior do que a soma total da área dos polígonos.

Observa-se ainda que exemplos presentes nas seções 3.3 e 4.4, ilustrados pelas

figuras 4.1, 3.6 e 3.7, criados para ilustrar dificuldades específicas do problema abordado foram todos resolvidos com a heurística proposta.

6.4.1 Pequeno Puzzle

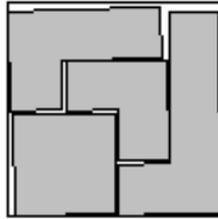


Figura 6.2: Solução final de um pequeno puzzle com quatro polígonos.

Este problema é um puzzle relativamente simples com quatro formas não-convexas. As formas não-convexas são decompostas em uma etapa de pré-processamento. A decomposição não afeta o resultado final. A figura 6.2 mostra a solução final para este problema e a figura 6.3 mostra os espectros de valores da função objetivo. Baseando-se nestes espectros, pode-se distinguir duas fases do processo. A primeira fase é caracterizada por valores de função objetivo maiores do que 0.15. A segunda começa por volta de 250000 iterações, e possui valores de função objetivo menores do que 0.15. A convergência final ocorre somente após 350000 iterações, mas uma solução ótima é encontrada muito mais cedo, com aproximadamente 250000 iterações (aproximadamente quatro minutos em um processador Pentium 4 de 2 GHz).

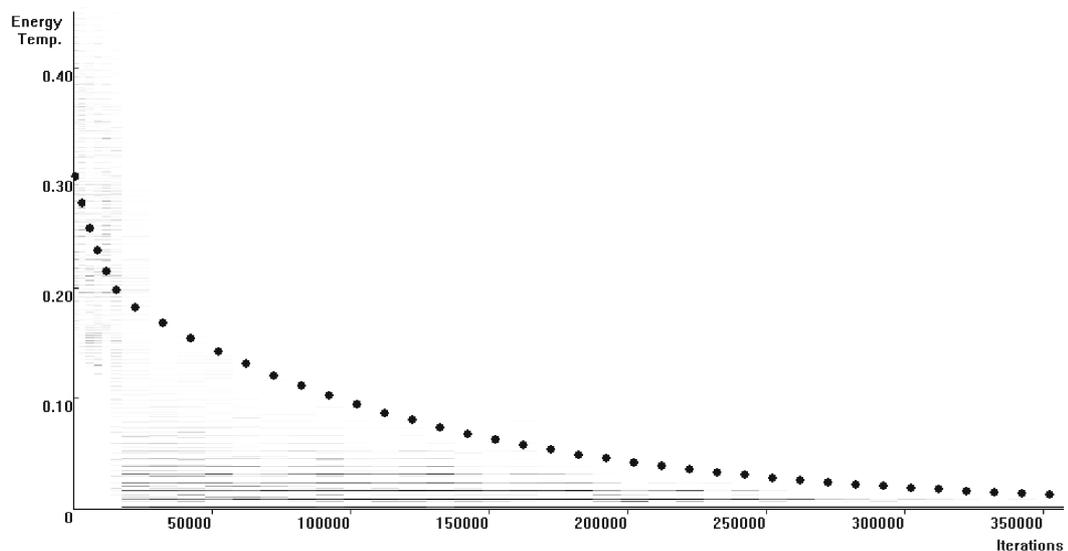


Figura 6.3: Espectro de função objetivo para o pequeno puzzle.

6.4.2 Tangram

Posicionamento de sete formas não-congruentes. A figura 6.4 mostra a solução final para este problema, e a figura 6.6 mostra os espectros de função objetivo. Este problema é particularmente difícil, apresentando múltiplos mínimos locais de valores próximos. Isso pode ser visto na figura 6.5, em que um recipiente com uma folga 5% maior admite uma solução alternativa com todas as formas posicionadas.

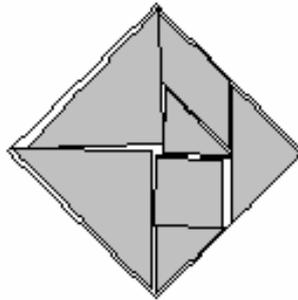


Figura 6.4: Solução ótima de um tangran com sete formas.

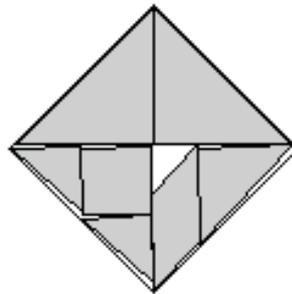


Figura 6.5: Solução “alternativa” para o *tangran*, obtida mediante um aumento de 5% na folga do recipiente.

Neste problema em particular, o sistema passa por uma transição de fase quando é encontrada a posição das duas maiores formas. A convergência ocorre depois de 525000 iterações, mas uma solução ótima é encontrada já a partir de 135000 iterações (aproximadamente vinte e cinco minutos em um processador Pentium 4 de 2 GHz).

6.4.3 Recipientes irregular

Posicionamento de 24 formas em recipiente irregular. Este exemplo ilustra a capacidade do processo de lidar com recipientes de formato não-retangular. Os

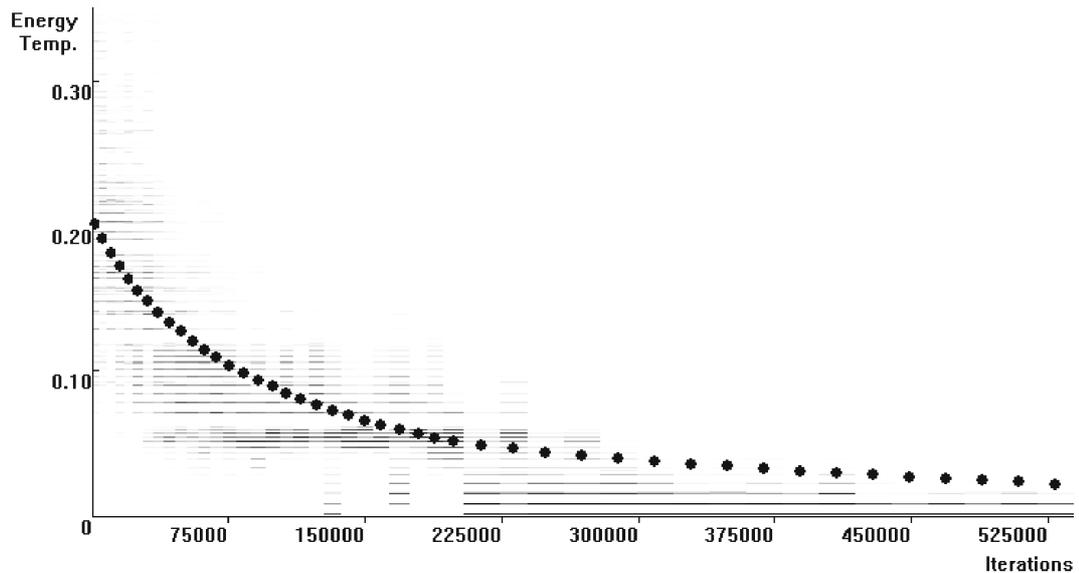


Figura 6.6: Espectros de função objetivo para o tangran.

espectros de energia para este problema na figura 6.8 mostra uma convergência bastante suave quando comparada à convergência do puzzle da seção 6.4.2. Isso decorre do maior número de formas e da inexistência de discrepâncias de tamanho entre elas.

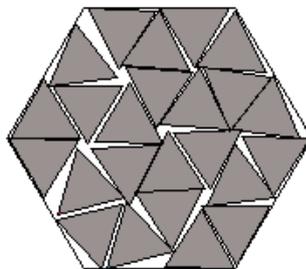


Figura 6.7: Solução ótima de puzzle com 24 formas com recipiente irregular.

6.4.4 Recipiente não-convexo

O problema da figura 6.9 ilustra a capacidade do processo de tratar problemas com recipientes não convexos.

6.4.5 Recipiente com furo

O problema da figura 6.10 ilustra a capacidade do processo de tratar problemas com recipientes dotados de furos.

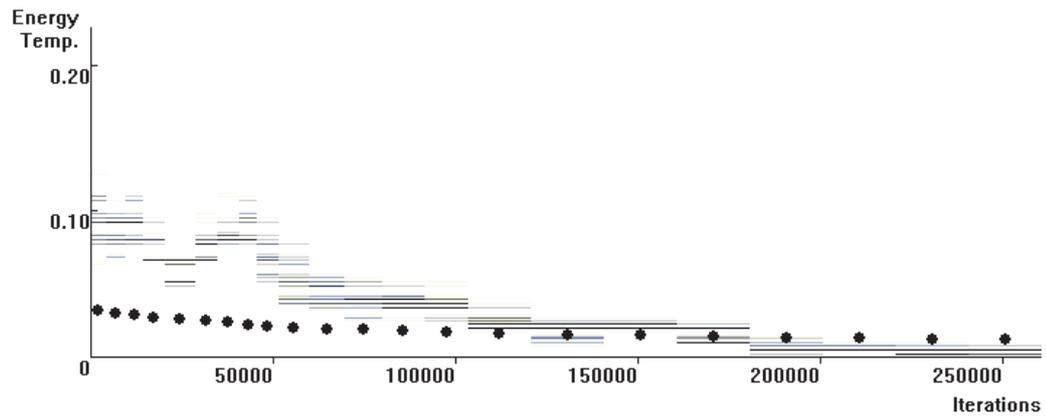


Figura 6.8: Espectros de função objetivo para o puzzle de recipiente irregular.

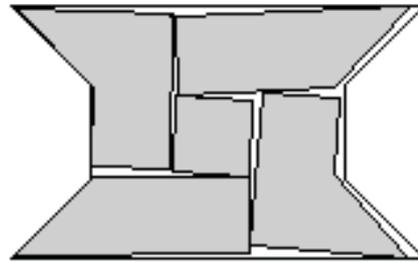


Figura 6.9: Puzzle com recipiente não-convexo.

6.5 Avaliação da influência de parâmetros do processo

Os problemas a seguir visam avaliar a influência de parâmetros do processo no seu desempenho.

6.5.1 Influência nos resultados do parâmetro de resfriamento geométrico α

Para verificar a influência do parâmetro α , foi feito um processamento do problema *tangran* com $\alpha = 0.90$, levando a um resfriamento mais rápido. O resultado é mostrado na figura 6.11.

Como pode ser visto na figura 6.11, embora o processo leve menos iterações para convergir, ele *não* converge para o ótimo global, ficando “congelado” em uma configuração sub-ótima.

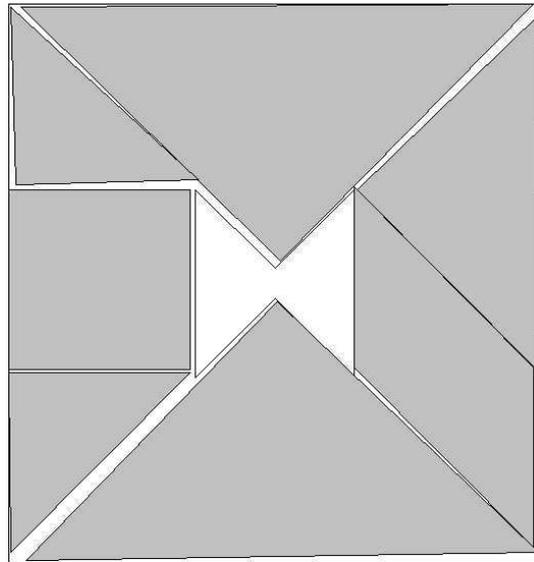


Figura 6.10: Puzzle com recipiente furado.

6.5.2 Influência do Desempate de soluções com custo idêntico

O efeito da modificação da função objetivo descrita na seção 4.6.1 pode ser visto nas figuras 6.13 e 6.12 para o problema descrito na seção 6.4.1.

Nota-se que a modificação da função objetivo acelera a convergência do processo. É preciso pesar contudo, o custo computacional adicional gerado pelas iterações de busca pelo fator de escala. Para o exemplo extremo ilustrado na figura 6.13, o tempo computacional do processo sem modificação é *menor*, apesar dele levar mais iterações do recozimento simulado para convergir.

6.6 Problemas adaptados da literatura

Os seguintes problemas são considerados *benchmark* na literatura, e foram abordados por Gomes e Oliveira em ^[1]. Todos os problemas na forma original são problemas de recipientes com uma dimensão variável e um conjunto pré-determinado de formas a ser posicionado no recipiente. O objetivo do problema é minimizar a dimensão variável do recipiente. Assim, estes problemas são variantes *duais* dos problemas abordados neste trabalho (vide seção 2.5), como o são a maioria dos problemas abordados na literatura ^[5]. Para permitir uma comparação entre os resultados, serão criados problemas com recipientes de dimensões fixas de modo que estas dimensões correspondam às dimensões mínimas encontradas por Gomes e Oliveira. O conjunto de formas a ser encaixado é o mesmo utilizado nos

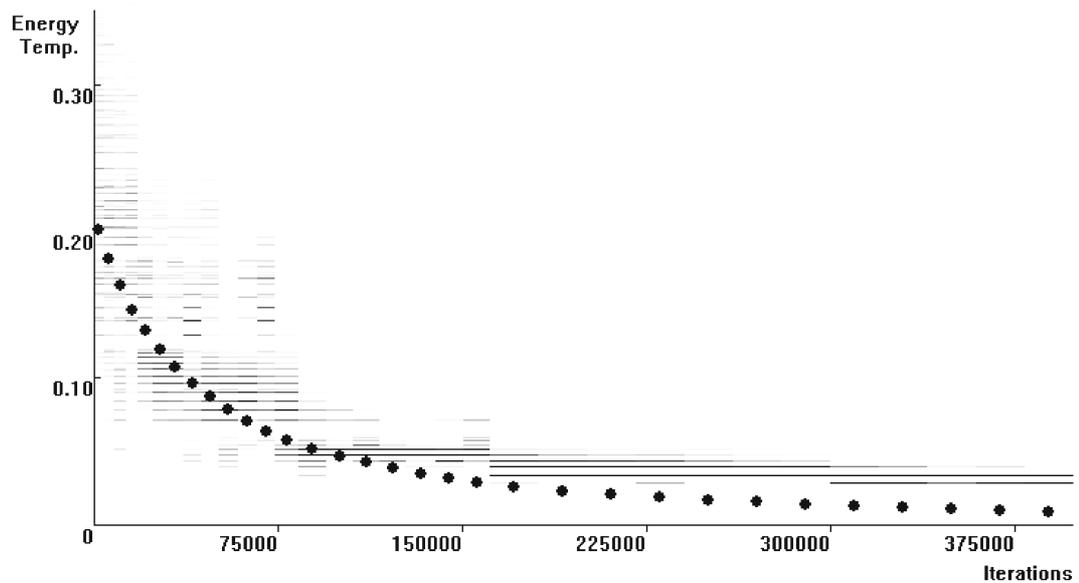


Figura 6.11: Espectros de função objetivo para um “resfriamento rápido” do problema tangran.

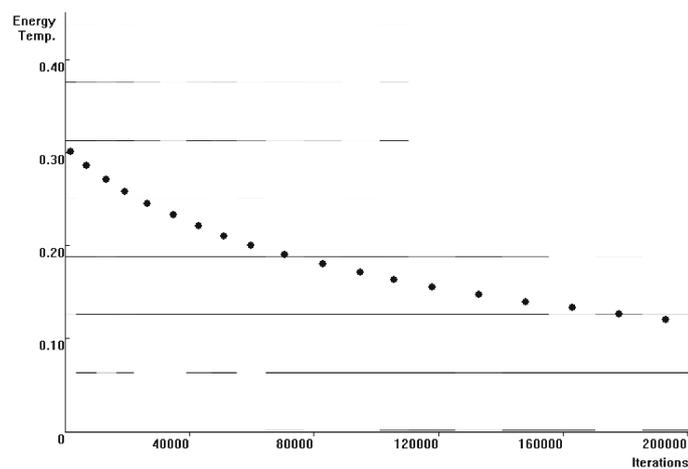


Figura 6.12: Espectros de energia para a função objetivo original

problemas originais. Naturalmente, as soluções ótimas destes problemas possuem um nível de aproveitamento igual ao dos problemas originais.

Outra diferença nos problemas abordados por Gomes e Oliveira é que estes *não* permitem rotação livre das formas. Alguns possuem a rotação da forma pré-definida, enquanto que outros permitem rotações finitas de 90° .

6.6.1 Problema *Albano*

O problema *Albano* é um problema com 24 formas. Trata-se de um problema real oriundo da indústria têxtil, que permite duas orientações para as formas: 0° e 180° . O resultado apresentado na figura 6.14.(a) foi produzido por Gomes e

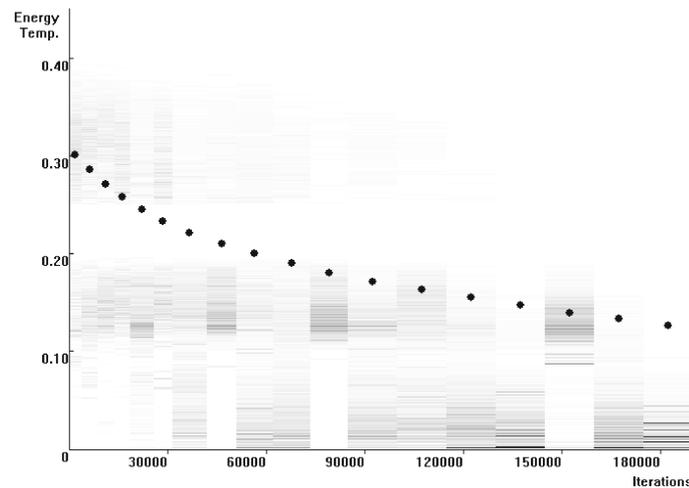


Figura 6.13: Espectros de energia com a função objetivo modificada por um fator de escala encontrado com uma busca de profundidade 8.

Oliveira ^[1]. Como pode ser visto na figura 6.14.(b), o processo proposto encontrou um leiaute alternativo, mas com o mesmo aproveitamento de material. A figura 6.15 apresenta os espectros de função-custo para a otimização deste problema. Pode-se observar que o processo de otimização encontra uma transição de fase quando as duas maiores formas são posicionadas.

6.6.2 Problema *Jackobs1*

O problema *Jackobs1* é um problema artificial com 25 formas, que comporta três admissíveis rotações para estas, 0^0 , 90^0 e 180^0 . O resultado da figura 6.16.(a) foi obtido por Gomes e Oliveira ^[1]. O processo proposto produziu leiautes diferentes, como ilustrado pelas figuras 6.16.(b), (c) e (d). A figura 6.17 mostra os espectros de função-objetivo associados. Pode-se observar que uma solução ótima é descoberta relativamente cedo, após 50.000 iterações.

6.6.3 Problema *Jackobs2*

O problema *Jackobs2* é um problema artificial com 25 formas que admite três possíveis rotações para estas: 0^0 , 90^0 e 180^0 . O resultado mostrado na figura 6.18.(a) foi produzido por Gomes e Oliveira ^[1]. O processo de otimização proposto encontrou leiautes diferentes, como mostrado nas figuras 6.16.(b) e (c). Os espectros de função objetivo estão na figura 6.17 Soluções ótimas são encontradas a partir de 175000 iterações.

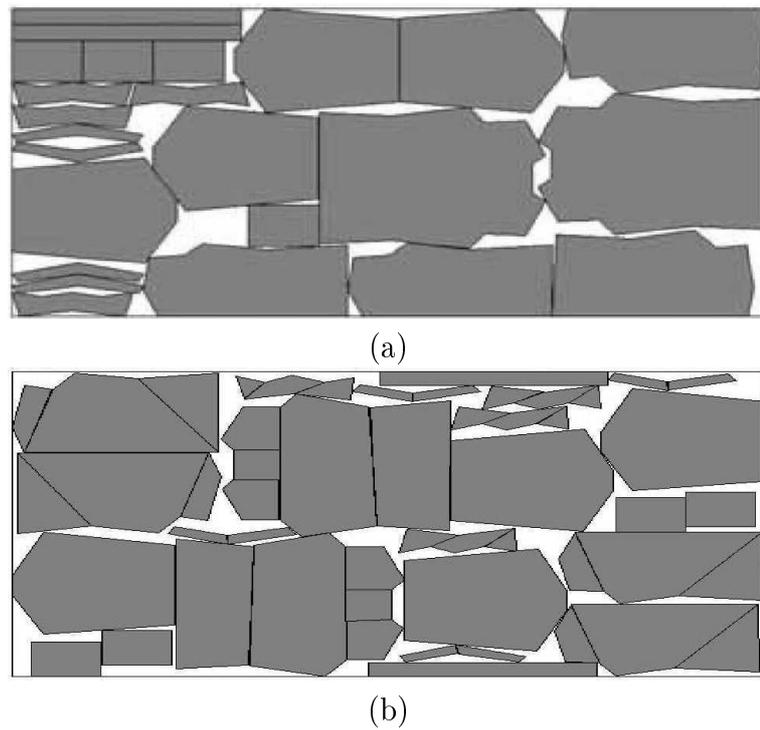


Figura 6.14: (a) Resultado obtido por Gomes e Oliveira ^[1] para o problema chamado *Albano*. (b) Resultado obtido pelo processo de otimização proposto neste trabalho.

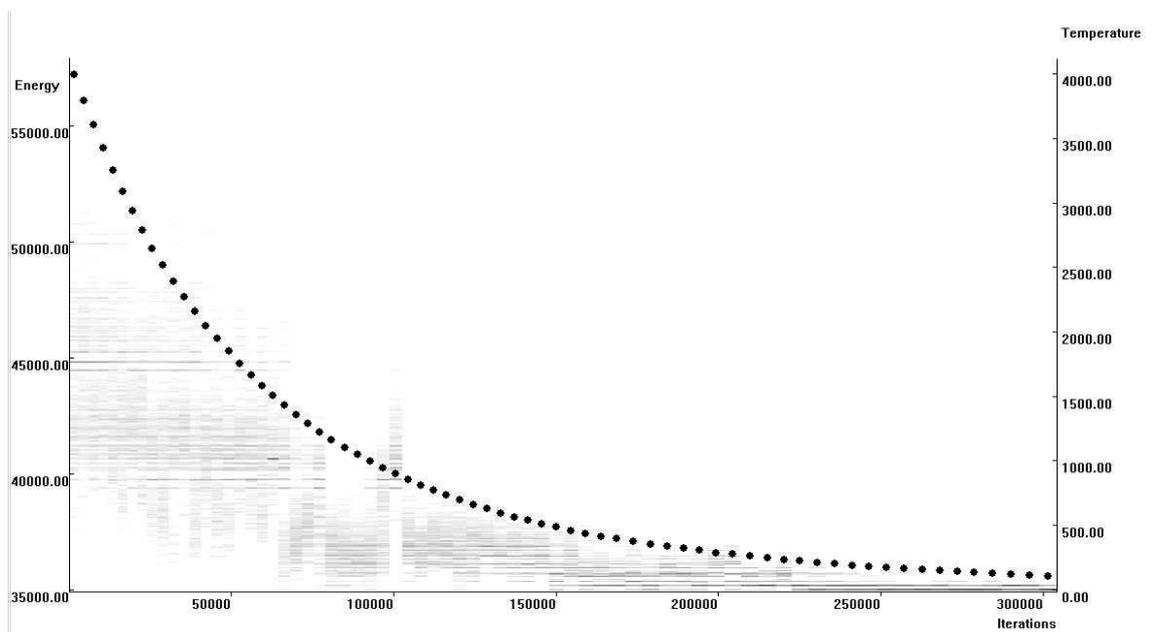


Figura 6.15: Espectros de função-objetivo para o problema mostrado na figura 6.14 (*Albano*).

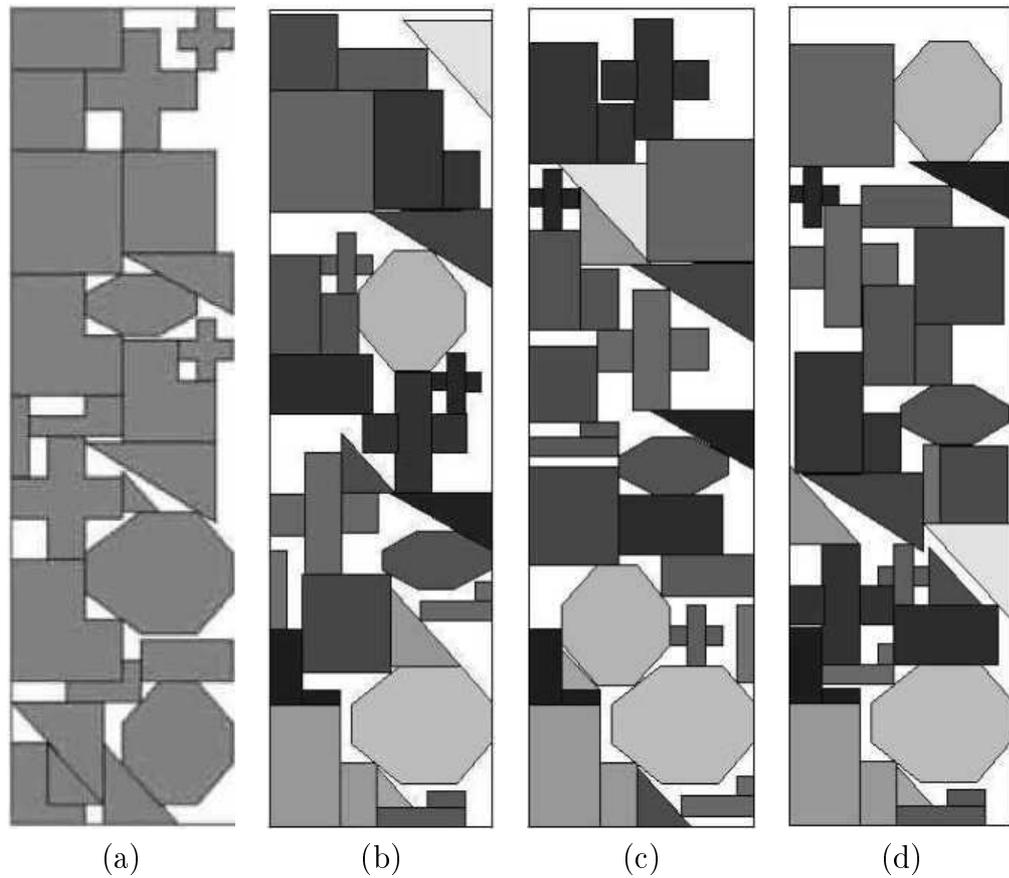


Figura 6.16: (a) Resultado obtido por Gomes e Oliveira ^[1] para o problema *Jackobs1*. (b), (c) e (d) Resultados obtidos pelo processo proposto.

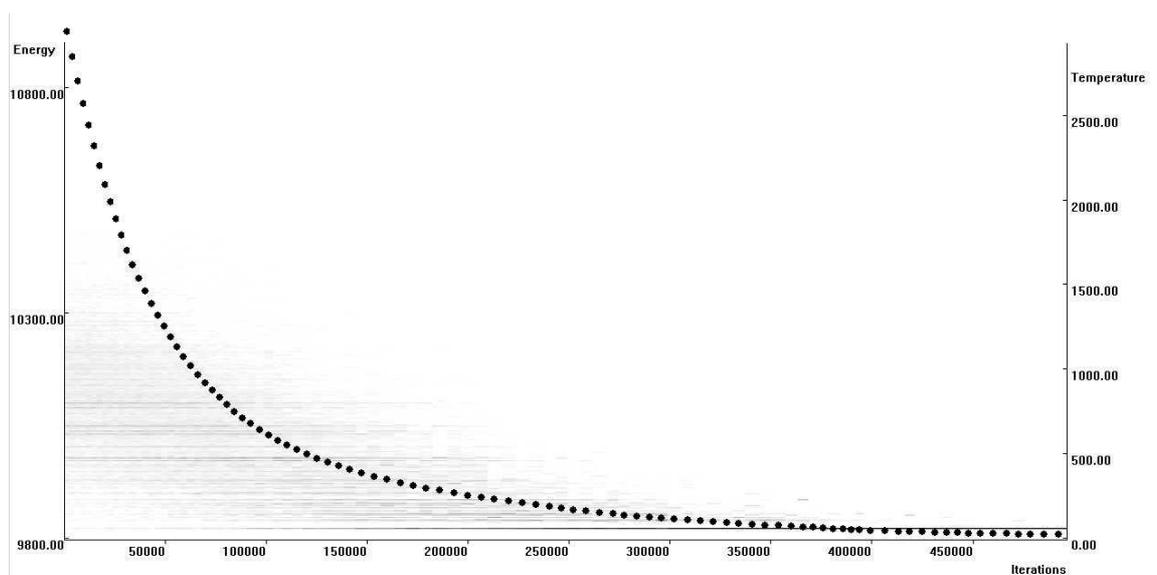


Figura 6.17: Espectros de função objetivo para o problema mostrado na figura 6.16 (*Jackobs1*).

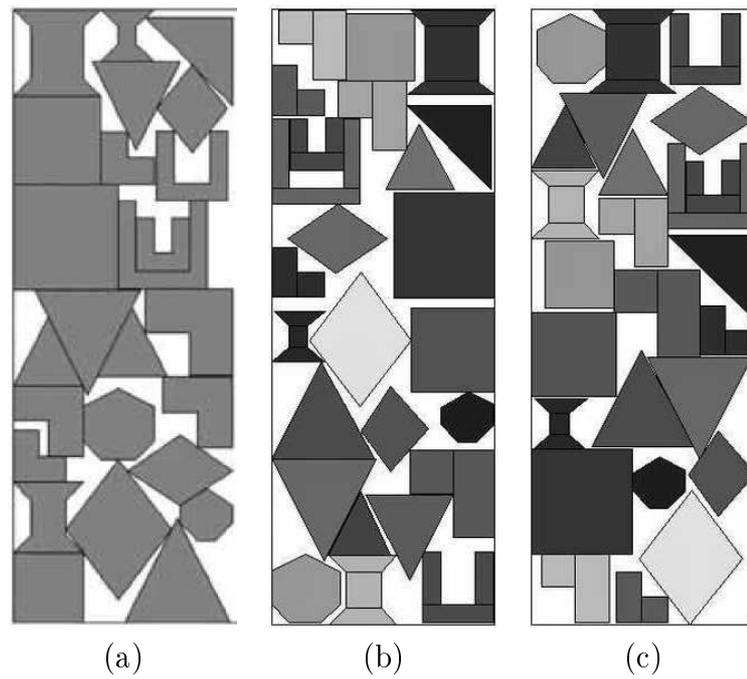


Figura 6.18: (a) Resultados obtidos por Gomes e Oliveira ^[1] para o problema *Jackobs2*. (b) e (c) Resultados obtidos neste trabalho.

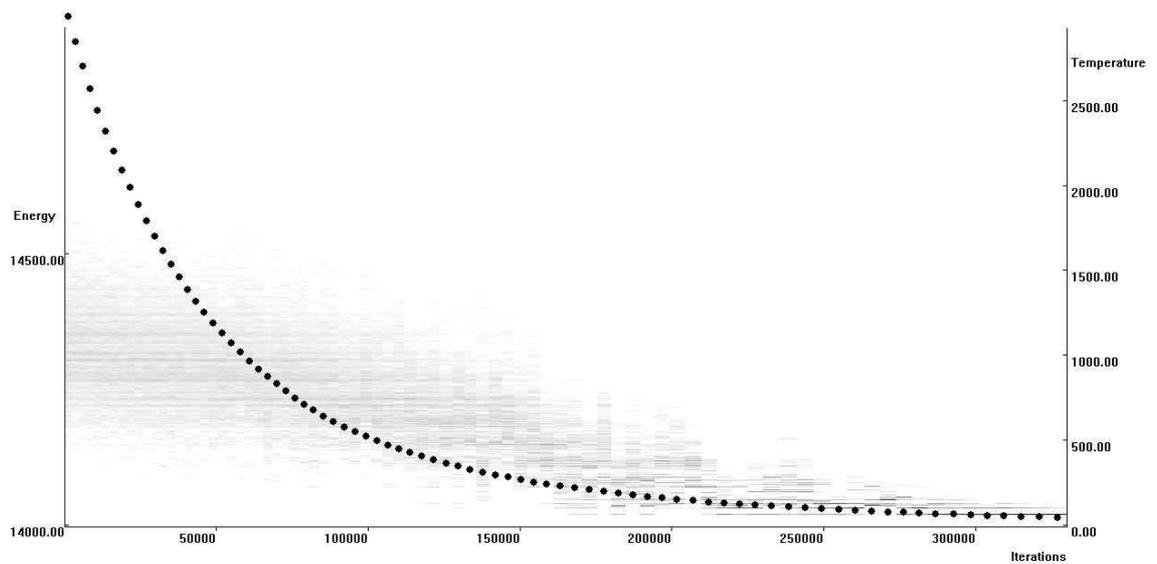


Figura 6.19: Espectros de função objetivo para o problema da figura 6.18 (*Jackobs2*).

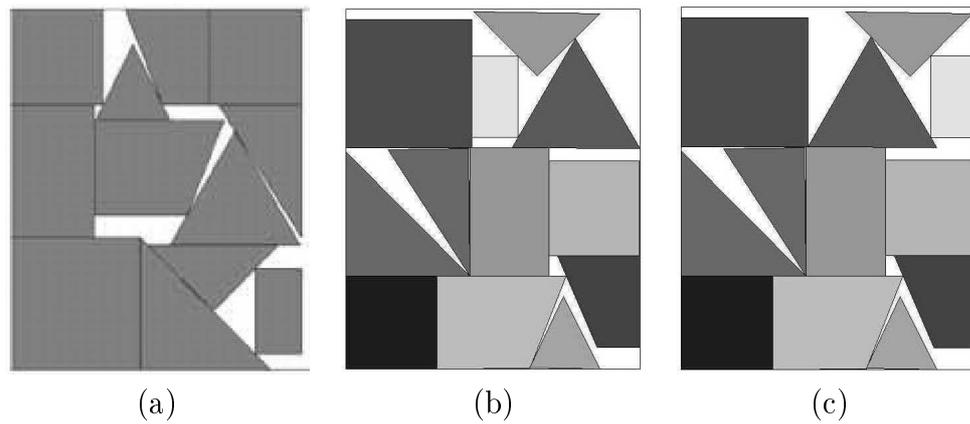


Figura 6.20: (a) Resultados obtidos por Gomes e Oliveira ^[1] para o problema *Fu*. (b) and (c) Resultados obtidos neste trabalho.

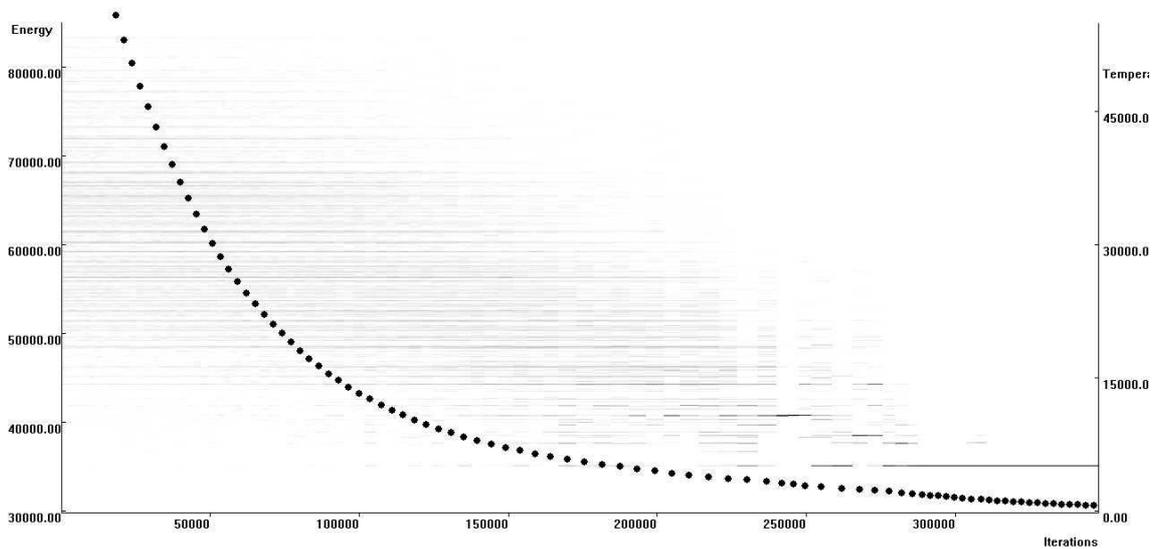


Figura 6.21: Espectros de valores para a função objetivo do problema da figura 6.20 (*Fu*).

6.6.4 Problema *Fu*

O problema *Fu* é um problema artificial com 12 formas, para as quais ele admite três rotações possíveis: 0° , 90° e 180° . O resultado da figura 6.20.(a) foi obtido por Gomes e Oliveira ^[1]. Os resultados obtidos por este trabalho estão apresentados nas figuras 6.20.(b) e (c). A figura 6.21 mostra os espectros de função objetivo do problema. Verifica-se que a convergência do processo neste problema em particular foi mais irregular, com os espectros evidenciando potenciais convergências para mínimos locais em baixas temperaturas.

6.6.5 Problema *shapes0*

O problema *shapes0* é um problema artificial com 43 formas com rotações fixas. O resultado da figura 6.22.(a) foi obtido por Gomes e Oliveira ^[1]. Os resultados obtidos por este trabalho estão apresentados na figura 6.22.(b). A figura 6.23 mostra os espectros de função objetivo do problema.

6.6.6 Problema *dagli*

O problema *dagli* é um problema derivado da indústria têxtil com 30 formas, para as quais ele admite duas rotações possíveis: 0^0 , e 180^0 . O resultado da figura 6.24.(a) foi obtido por Gomes e Oliveira ^[1]. O *melhor* resultado atingido pelo processo apresentado neste projeto está representado na figura 6.24.(b). Este problema é de particular interesse, pois trata-se do único dos problemas da literatura para o qual o processo, embora tenha visitado soluções ótimas, *não* convergiu para um ótimo global. Isso fica evidenciado nos seus espectros de energia ilustrados na figura 6.25. Nota-se que apesar do histórico do processo indicar que ele visita soluções ótimas, a convergência final se dá em um mínimo local sub-ótimo, o que é um indicador de excessiva velocidade de resfriamento.

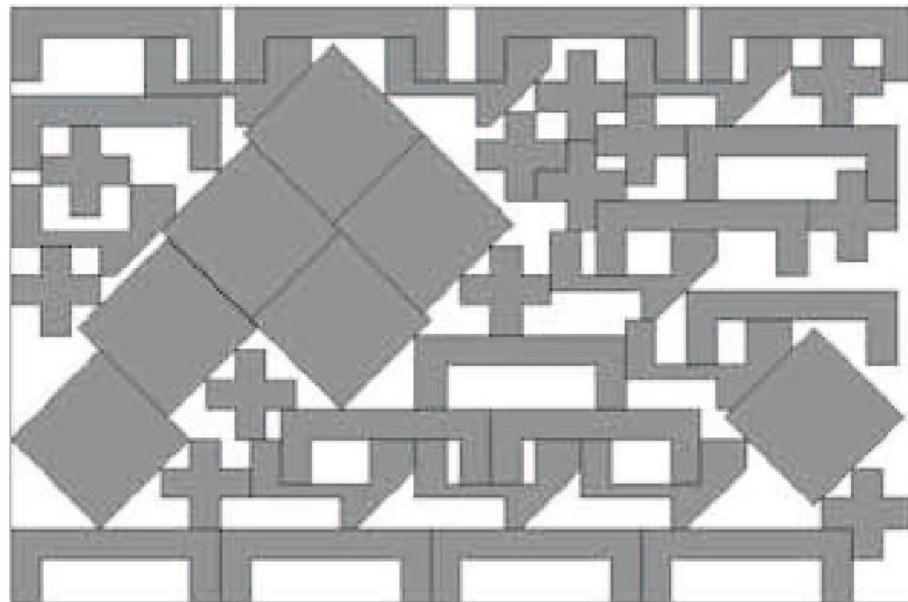
6.7 Conclusões

Comprovou-se através de problemas especialmente construídos que o processo proposto é capaz de tratar adequadamente problemas de posicionamento translacional e rotacional para formas convexas e não-convexas, com recipientes convexas e não-convexas, e até mesmo dotados de furos. Conforme o esperado, o processo se comporta adequadamente, gerando somente leiautes válidos, ou seja, leiautes em que as formas estão completamente inseridas no recipiente e sem sobreposições entre as formas.

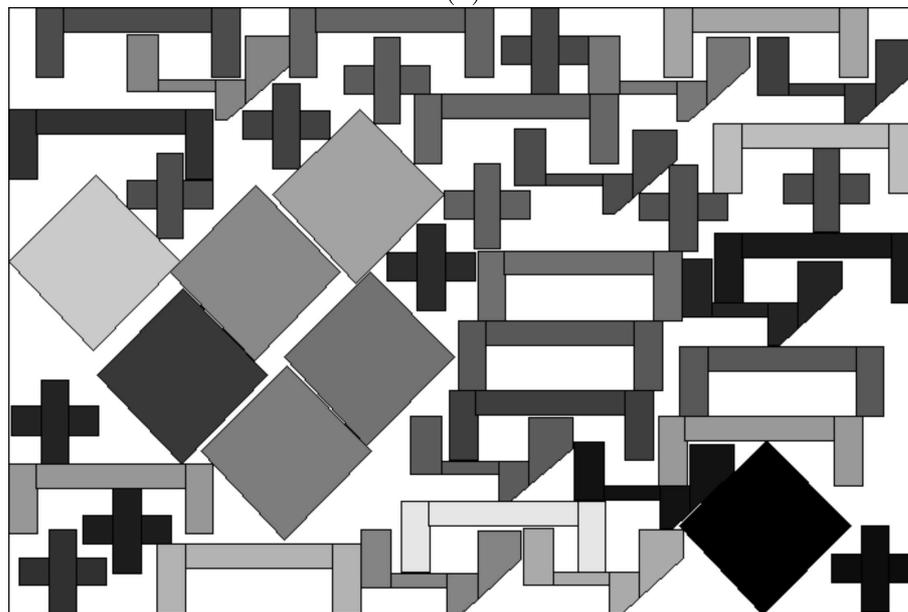
A estratégia de se pesquisar uma heurística *completa* (vide seção 4.3) foi validada por resultados em que o algoritmo converge para o ótimo global do problema mesmo na presença de mínimos locais de valor muito próximo.

A influência de parâmetros do processo como a velocidade de resfriamento e a quantidade de passos na busca binária para desempatar soluções foi avaliada. Comprovou-se que o processo responde a variações nestes parâmetros conforme o esperado.

A eficiência dos leiautes produzidos foi comparada com problemas *benchmark*



(a)



(b)

Figura 6.22: (a) Resultado obtido por Gomes e Oliveira ^[1] para o problema *shapes0*. (b) Resultado obtido neste trabalho.

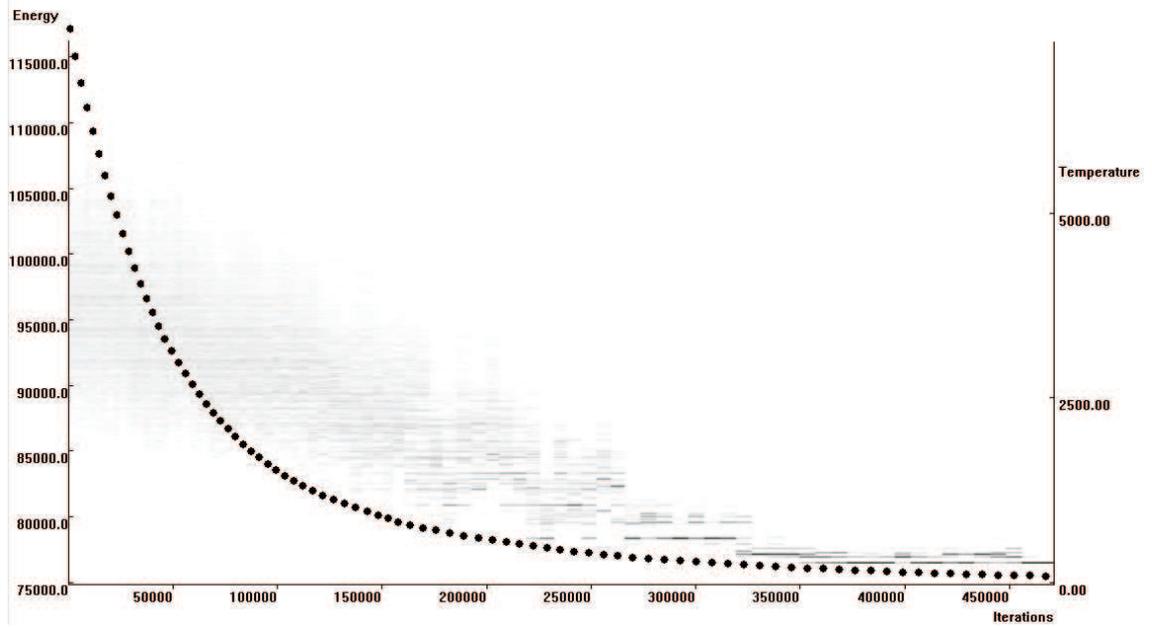


Figura 6.23: Espectros de valores para função objetivo do problema da figura 6.22 (*shapes0*).

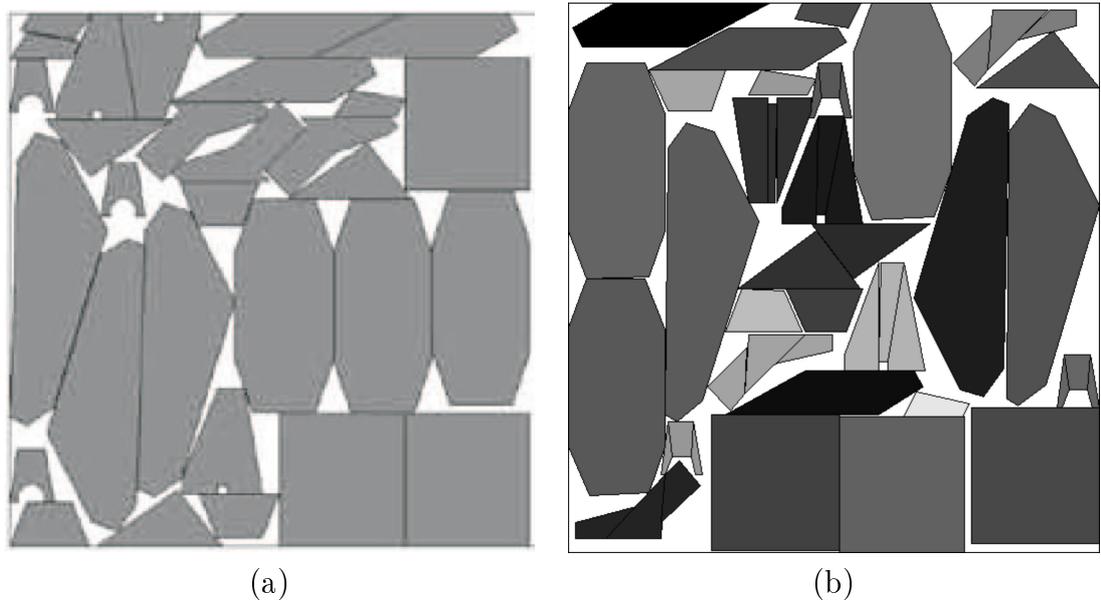


Figura 6.24: (a) Resultado obtido por Gomes e Oliveira ^[1] para o problema *dagli*. (b) Melhor resultado obtido neste trabalho.

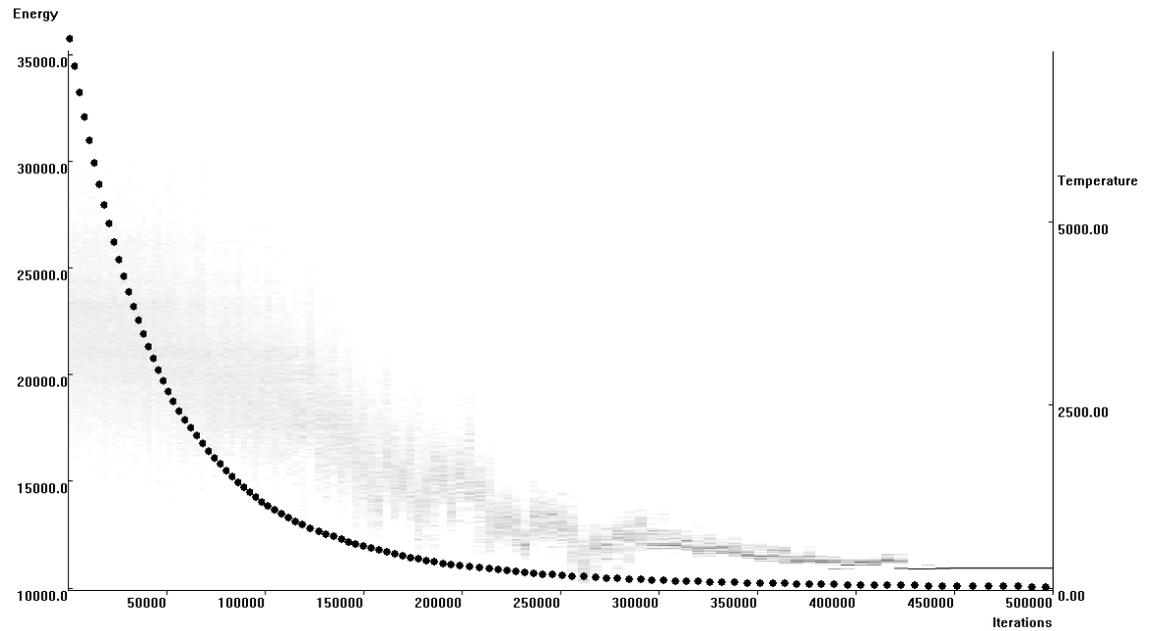


Figura 6.25: Espectros de valores para função objetivo do problema da figura 6.24 (*dagli*).

da literatura da área. Esta comparação foi limitada pelo foco da literatura da área nas variantes *duais* dos problemas de otimização, ou seja, nos problemas com recipientes com uma dimensão variável. Ainda assim foi possível construir variantes *primais* dos problemas apresentados pela literatura de modo que as soluções ótimas dos problemas primais tenham aproveitamento igual ao das soluções apresentadas na literatura.

Os leiautes produzidos pelo processo aqui proposto atingiram o nível de aproveitamento desejado (alcançando assim, soluções ótimas globais). Em um dos casos (vide seção 6.6.6), o processo visitou soluções ótimas globais, mas *não* convergiu em direção a estas, sugerindo um programa de resfriamento inapropriado.

7 Conclusões e considerações finais

O problema aqui tratado é o problema do posicionamento de formas irregulares com translações e rotações livres em recipientes irregulares com dimensões limitadas.

Problemas desta natureza possuem uma grande relevância para a indústria (vide seção 2.2), mas são minoria na literatura especializada em problemas de corte e empacotamento [5]. Este problema de otimização apresenta dificuldades únicas, em particular o fato de que trata-se de um problema de otimização em um domínio de variáveis contínuas com uma função objetivo que assume valores discretos (vide seção 2.5).

A complexidade dos problemas de posicionamento leva à impraticabilidade de abordagens puramente algorítmicas, e à busca por soluções heurísticas, e em particular, heurísticas probabilísticas.

A maior parte das abordagens do problema do posicionamento de formas irregulares em recipientes limitados emprega a técnica conhecida como *penalização externa*, que consiste em permitir a exploração de soluções “externas” do problema (ou seja, soluções que violam as restrições originais do problema) enquanto que uma correção aplicada à função objetivo “penaliza” a escolha destas soluções.

Embora esta seja uma técnica de otimização muito empregada em problemas de domínios complexos^[32], é possível apontar características do problema de posicionamento irregular que fazem com que esta técnica encontre dificuldades, como a geração de leiautes inválidos (vide seção 2.6).

Assim, este trabalho desenvolveu um processo de otimização para o problema do posicionamento rotacional e translacional de formas irregulares em recipientes de dimensões fixas baseado em heurísticas probabilísticas sem o uso da penalização externa, ou seja, realizando a busca diretamente no espaço original do problema.

Para tratar o domínio original do problema, o domínio em que não ocorrem sobreposições entre as formas, foi empregado o *polígono de obstrução*, que determina a região obstruída por uma forma no posicionamento de outra. Além disso, o problema do posicionamento das formas foi decomposto nos sub-problemas:

- Determinação da ordem de posicionamento das formas.
- Determinação das rotações das formas.
- Determinação das translações das formas.

Para cada um dos sub-problemas foi desenvolvida uma heurística baseada no *Recozimento Simulado*. O recozimento simulado é uma meta-heurística de otimização probabilística de *gradiente ascendente*, o que significa que ela pode evitar mínimos locais do problema e convergir para soluções globalmente ótimas.

Todas as heurísticas criadas para os sub-problemas foram projetadas de modo a possuírem uma probabilidade de convergir para uma solução globalmente ótima.

Para mitigar os efeitos do comportamento discreto da função objetivo do problema foi introduzido nesta um fator de “desempate” que busca diferenciar soluções com valores idênticos através de uma estimativa de quão próxima está uma determinada solução de conseguir encaixar uma nova forma em seu leiaute.

A implementação do processo foi estudada de modo a minimizar o seu custo computacional. Isso foi obtido através de algoritmos eficientes para operações geométricas sobre as formas, bem como o reaproveitamento de resultados geométricos intermediários.

O processo de otimização proposto foi implementado no *software fitPolygon* e o seu desempenho foi avaliado.

Nesta avaliação foi comprovada a eficácia do processo em produzir leiautes válidos, assim como a sua capacidade de convergir consistentemente para soluções ótimas globais em problemas nos quais estes ótimos eram conhecidos.

A comparação de resultados deste trabalho com resultados publicados na literatura foi limitada pelo foco desta nas variantes *duais* dos problemas de otimização de posicionamento, ou seja, nos problemas com recipientes com uma dimensão variável.

Para contornar esta limitação, construiu-se variantes *primais* dos problemas apresentados pela literatura de modo que as soluções ótimas dos problemas primais tenham aproveitamento igual ao das soluções apresentadas na literatura.

Nestes problemas, o processo aqui proposto foi eficaz em reproduzir os níveis de aproveitamento de material dos resultados apresentados na literatura, comprovando assim a validade da proposta de utilizar o recozimento simulado na otimização do posicionamento translacional e rotacional de formas irregulares em recipientes de dimensões fixas sem empregar técnicas de penalização externa.

Apêndice A – Recozimento Simulado

Recozimento Simulado (*Simulated Annealing*)^[34] é a meta-heurística probabilística estudada neste trabalho. Ela foi escolhida devido à sua capacidade de “escapar” de mínimos locais, que acontecem freqüentemente neste problema. Cabe também mencionar que o processo de recristalização que inspirou o recozimento simulado é um exemplo natural do problema de empacotamento.

A.1 Descrição

O recozimento simulado surgiu a partir do algoritmo de Metropolis da físico-química, que simula o processo de recristalização de átomos em um metal durante o processo de recozimento (resfriamento controlado e gradual). Durante o processo de recozimento, os átomos migram naturalmente para posições que minimizam a energia total do sistema, mesmo que durante este processo o sistema tenha que passar por configurações temporárias de alta energia. A observação deste comportamento sugeriu a aplicação da simulação deste processo a problemas de otimização combinatorial.

Recozimento simulado é um método de otimização de exploração local¹ de gradiente ascendente (*hill-climbing*), o que significa que ele pode escapar de mínimos locais permitindo a exploração do espaço em direções que levam a um acréscimo da função custo. Ele trabalha aplicando modificações aleatórias no ponto de avaliação da função de custo. Se a modificação leva a um ponto de custo menor, ela é automaticamente mantida. Caso contrário, a modificação ainda pode ser mantida com uma probabilidade obtida da distribuição de Boltzman (A.1).

$$P(\Delta E) = e^{-\frac{\Delta E}{kt}} \quad (\text{A.1})$$

onde $P(\Delta E)$ é a probabilidade de que o processo de otimização mantenha uma modificação na solução explorada que levou a um acréscimo ΔE da função de

¹Métodos de otimização de exploração local são métodos que exploram o universo de soluções de forma seqüencial, saltando de uma solução para outra vizinha, possivelmente mais interessante.

custo, k é um parâmetro do algoritmo (análogo à constante de Stefan-Boltzman) e t é a “temperatura” instantânea do processo. Esta temperatura é definida por um programa de resfriamento, e é o principal parâmetro de controle do algoritmo. Vários programas de resfriamento foram avaliados para o problema estudado.

A figura A.1 mostra a forma da distribuição de Boltzman. Observa-se que a probabilidade de um determinado estado ocorrer decresce com a energia do estado. No entanto, conforme a temperatura aumenta, o “declive” da curva $P(\Delta E)$ diminui.

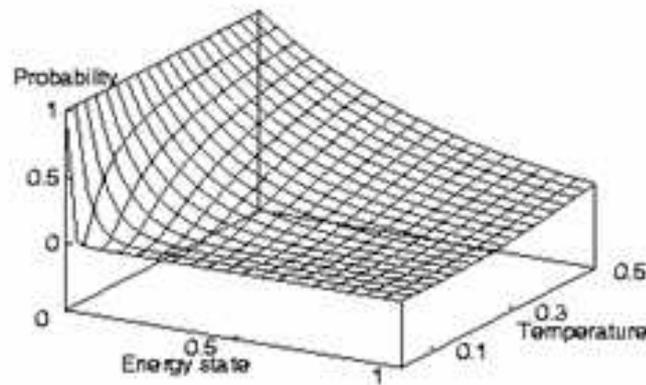


Figura A.1: Distribuição de Boltzman.

A.2 Algoritmo

Seja o problema de minimizar a função $f(X)$, onde x é um ponto no domínio de f que atenda a todas as restrições do problema. O algoritmo inicia-se com uma solução válida aleatória x_0 . Em seguida, a cada iteração, ele aplica uma transformação à solução, produzindo uma nova possível solução x^* na vizinhança de x . O acréscimo do custo $\Delta E = f(x^*) - f(x)$ é avaliado. Se este acréscimo é negativo (ou seja, a nova solução tem um custo menor), a nova solução é automaticamente adotada. Se este acréscimo é positivo (ou seja, a nova solução tem um custo maior), é gerado um número aleatório r entre 0 e 1. Se r for menor do que a probabilidade calculada por (A.1), x^* substitui x como a nova solução corrente. Senão, x^* é descartado.

Como pode ser visto na figura A.2, o algoritmo realiza iterações em uma temperatura fixa até que um critério de parada específico esteja atendido. Este critério tenta determinar se o sistema atingiu o “equilíbrio térmico” em uma determinada temperatura. Geralmente, é definido um número máximo de modificações mantidas e um número máximo de iterações em uma temperatura. Quando um

dos critérios é satisfeito, o algoritmo procede para a próxima temperatura no programa de resfriamento. O critério de parada global geralmente é definido pelo próprio programa de resfriamento. Quando o programa de resfriamento termina, o algoritmo se encerra. Existem, no entanto, critérios para interromper o algoritmo antes do fim do programa de resfriamento. Um critério adotado frequentemente é o de identificar um estado “congelado” do algoritmo limitando o número máximo de iterações sem progresso significativo.

A.3 Programas de resfriamento

Os seguintes programas de resfriamento ^[40] foram avaliados:

Resfriamento logarítmico Este resfriamento é o único para o qual se tem uma demonstração de convergência para o mínimo global com uma dada escolha de parâmetros i_0 e c . Mas é em geral lento demais para o problema estudado.

$$t_i = \frac{c}{\log(i + i_0)} \quad (\text{A.2})$$

Resfriamento linear Este programa de resfriamento tende a ser rápido demais nos momentos finais do algoritmo, o que leva ao “congelamento” do sistema em configurações não ótimas.

$$t_i = t_{i-1} - \Delta t \quad (\text{A.3})$$

Resfriamento geométrico Este resfriamento leva a bons resultados, permitindo ao sistema estabilizar-se em uma configuração ótima a medida que ele esfria. O parâmetro α é escolhido ao redor de 0.95, embora a determinação do valor apropriado ainda seja um problema difícil.

$$t_i = \alpha \cdot t_{i-1} \quad (\text{A.4})$$

Resfriamento adaptativo Este programa de resfriamento é uma variação do resfriamento geométrico, que tenta resolver o problema da determinação de α . Ele o faz através da avaliação do “calor específico” do sistema usando o desvio padrão do valor da função de custo σ_i durante as iterações em uma temperatura fixa.

$$t_i = \alpha_i \cdot t_{i-1} \quad (\text{A.5})$$

$$\alpha_i = e^{-\frac{\lambda \cdot t_{i-1}}{\sigma_{i-1}}} \quad (\text{A.6})$$

A.3.1 Temperatura Inicial

A temperatura inicial é definida de modo a evitar que o processo fique preso em um conjunto de soluções sub-ótimas no início do processo. Assim, ela deve ser selecionada de modo que praticamente todas as soluções testadas no início do algoritmo sejam aceitas. Por outro lado, uma temperatura inicial elevada demais provoca um processamento desnecessário de iterações do processo.

Uma maneira de se estimar a temperatura inicial apropriada é considerá-la como ^[23]:

$$T_{inicial} = \frac{-3\sigma_E}{\ln(P)} \quad (\text{A.7})$$

Onde σ_E é o desvio padrão da função objetivo obtido através de algumas iterações nas quais toda solução é aceita, e P é um parâmetro arbitrário, tomado entre 0.85 e 0.5.

Algoritmo de otimização por recozimento simulado

1. $x \leftarrow$ <Solução inicial aleatória>
2. $i \leftarrow 0$
3. ENQUANTO <Critério de parada global não satisfeito> FAÇA
4. $t \leftarrow Resfriamento(i)$
5. $i \leftarrow i + 1$
6. ENQUANTO <Critério de parada local não satisfeito> FAÇA
7. $x^* \leftarrow$ ModificaçãoAleatória(x, t)
8. $\Delta E = f(x^*) - f(x)$
9. SE $\Delta E < 0$ ENTÃO
10. $x \leftarrow x^*$
11. SENÃO
12. $a \leftarrow$ <Número aleatório gerado entre 0 e 1>
13. SE $a < e^{-\Delta E/kt}$ ENTÃO
14. $x \leftarrow x^*$
15. FIM SE
16. FIM ENQUANTO
17. FIM ENQUANTO
18. <Exibe Solução x>

Figura A.2: Algoritmo de recozimento simulado.

Referências

- 1 GOMES, A. M.; OLIVEIRA, J. F. Solving irregular strip packing problems by hybridising simulated annealing and linear programming. *European Journal of Operational Research*, v. 171, p. 811–829, 2006.
- 2 GAREY, M. R.; JOHNSON, D. S. *Computers and Intractability – A Guide to the Theory of NP-Completeness*. San Francisco, EUA: Freeman, 1979.
- 3 LODI, A.; MARTELLO, S.; MONACI, M. Two-dimensional packing problems: A survey. *European Journal of Operational Research*, v. 141, p. 241–525, 2002.
- 4 DYCKHOFF, H. A typology of cutting and packing problems. *European Journal of Operational Research*, v. 44, n. 2, p. 145–159, 1990.
- 5 WÄSCHER, G.; HAUSSNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. *Faculty of Economics and Management Magdeburg - Working Paper Series*, 2005.
- 6 DOWSLAND, K. A.; DOWSLAND, W. B. Packing problems. *European Journal of Operational Research*, v. 56, p. 2–14, 1992.
- 7 DOWSLAND, K. A.; DOWSLAND, W. B. Solution approaches to irregular nesting problems. *European Journal of Operational Research*, v. 84, p. 506–521, 1995.
- 8 ADAMOWICZ, M.; ALBANO, A. Nesting two dimensional shapes in rectangular modules. *Computer aided design*, v. 8, n. 1, p. 27–33, 1976.
- 9 LI, Z.; MILENKOVIC, V. A compaction algorithm for non-convex polygons and its application. In: *9th Annual Symposium on Computation Geometry*. [S.l.: s.n.], 1992.
- 10 MILENKOVIC, V.; DANIELS, K. M.; LI, Z. Placement and compaction of nonconvex polygons for clothing manufacture. In: *Fourth Canadian Conference on Computational Geometry*. [S.l.: s.n.], 1992. p. 236–243.
- 11 BURKE, E.; KENDALL, G. Applying simulated annealing and the no fit polygon to the nesting problem. In: *Proceedings of the World Manufacturing Congress*. [S.l.: s.n.], 1997. p. 27–30.
- 12 MILENKOVIC, V. Rotational polygon overlap minimization and compaction. *Computational Geometry*, v. 10, p. 305–318, 1998.
- 13 MILENKOVIC, V. Rotational polygon containment and minimum enclosure using only robust 2d constructions. *Computational Geometry*, v. 13, p. 3–19, 1999.

- 14 DANIELS, K. M. *Containment Algorithms for Nonconvex polygons with applications to Layout*. Tese — Center for Research in Computing Technology, Harvard University, Cambridge, Massachusetts, May 1995.
- 15 FOWLER, R. J.; PATERSON, M.; TANIMOTO, S. L. Optimal packing and covering in the plane are np-complete. *Inf. Process. Lett.*, v. 12, n. 3, p. 133–137, 1981.
- 16 STOYAN, Y. G.; PATSUK, V. A method of optimal lattice packing of congruent oriented polygons in the plane. *European Journal of Operational Research*, v. 124, p. 204–216, 2000.
- 17 MILENKOVIC, V. Densest translational lattice packing of non-convex polygons. In: *Proceedings of the 16h Annual ACM Symposium on Computational Geometry*. [S.l.: s.n.], 2000.
- 18 MIYAZAWA, F. K.; WAKABAYASHI, Y. Algoritmos de aproximação para problemas de empacotamento. *Concurso de Teses e Dissertações – Soc. Brasileira de Computação*, 1998. 1o. Prêmio CTD'98.
- 19 BURKE, E. K.; KENDALL, G.; WHITWELL, G. A new placement heuristic for the orthogonal stock-cutting problem. *Operations Research*, v. 52, n. 4, p. 655–671, July - August 2004.
- 20 DOWSLAND, K. A.; VAID, S.; DOWSLAND, W. B. An algorithm for polygon placement using a bottom-left strategy. *European Journal of Operational Research*, v. 141, p. 371–381, 2002.
- 21 DOWSLAND, K. A. Some experiments with simulated annealing techniques for packing problems. *European Journal of Operational Research*, v. 68, p. 389–399, 1993.
- 22 JAIN, P.; FEYNES, P.; RICHTER, R. Optimal blank nesting using simulated annealing. *Transactions of the ASME*, v. 114, p. 160–165, March 1992.
- 23 HECKMANN, R.; LENGAUER, T. A simulated annealing approach to the nesting problem in the textile manufacturing industry. *Annals of Operations Research*, v. 57, p. 103–133, 1995.
- 24 JAKOBS, S. On genetic algorithms for the packing of polygons. *European Journal of Operational Research*, v. 88, p. 165–181, 1996.
- 25 M'HALLAH, R.; BOUZIRI, A.; JILANI, W. Layout of two dimensional irregular shapes using genetic algorithms. *Lecture Notes In Computer Science*, v. 2070, p. 403–411, 2001.
- 26 HIFI, M.; M'HALLAH, R. Hybrid algorithm for the two-dimensional layout problem: the cases of regular and irregular shapes. *International Transaction in Operational Research*, v. 10, p. 195–216, 2003.
- 27 GOMES, A. M.; OLIVEIRA, J. F. A 2-exchange heuristic for nesting problems. *European Journal of Operational Research*, v. 141, p. 359–370, 2002.
- 28 FAINA, L. An application of simulated annealing to the cutting stock problem. *European Journal of Operational Research*, v. 114, p. 542–556, 1999.

- 29 BOUNSAYTHIP, C. K. *Algorithmes Heuristiques et Evolutionnistes: Application à la Résolution du Problème de Placement de Formes Irrégulières*. Tese — University of Science and Technology of Lille, Octobre 1998. Em Francês.
- 30 HOPPER, E.; TURTON, B. An empirical investigation of meta-heuristic and heuristic algorithms for a 2d packing problem. *European Journal of Operational Research*, v. 128, n. 1, p. 34–57, January 2001.
- 31 ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. *NBR 100004*: Resíduos sólidos. Rio de Janeiro, 2005.
- 32 VANDERPLAATS, G. N. *Numerical Optimization Techniques for Engineering Design: With Applications*. New York: Mcgraw-Hill, 1984. 123-128 p.
- 33 AGARWAL, P. K.; FLATO, E.; HALPERIN, D. Polygon decomposition for efficient construction of minkowski sums. In: *European Symposium on Algorithms*. [S.l.: s.n.], 2000.
- 34 KIRKPATRICK, S.; GELLAT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, p. 671–680, 1983.
- 35 HERZ, A.; WIDMER, M. Guidelines for use of meta-heuristics in combinatorial optimization. *European Journal of Operational Research*, v. 151, p. 247–252, 2003.
- 36 VATTI, B. R. A generic solution to polygon clipping. *Communications of the ACM*, v. 35, n. 7, p. 56–63, 1992.
- 37 MILENKOVIC, V. J. Shortest path geometric rounding. *Algoritmica*, v. 27, n. 1, p. 57–86, 2000.
- 38 TSUZUKI, M. S. G. *Solucionando Sistemas Polinomiais não Lineares pelo Método do Poliedro Projetado Implementado em Arimética Intervalar*. Tese de Livre Docência — Escola Politécnica da Universidade de São Paulo, São Paulo, novembro 2000.
- 39 LEONOV, M. V. Comparison of the different algorithms for polygon boolean operations. 1998. Disponível em: <<http://abntex.codigolivres.org.br>>. Acesso em: 21 dez. 2006.
- 40 STEINHÖFEL, K.; ALBRECHT, A. A.; WONG, C. K. On various cooling schedules for simulated annealing applied to the job shop problem. In: *RANDOM*. [S.l.: s.n.], 1998. p. 260–279.