

Um algoritmo genético híbrido para o problema de corte industrial bidimensional

Ademir Aparecido Constantino* e Augusto Mendes Gomes Junior

Departamento de Informática, Universidade Estadual de Maringá, Av. Colombo, 5790, 87020-900, Maringá, Paraná, Brasil.

*Author for correspondence. e-mail: ademir@din.uem.br

RESUMO. O objetivo deste trabalho é a implementação de um algoritmo eficaz que solucione o problema de corte bi-dimensional, ou seja, que encontre uma solução muito boa (viável) e rápida para este problema, otimizando o uso das chapas para diminuir o desperdício gerado com os cortes que são feitos sobre a mesma. Para a sua implementação, foram utilizados Algoritmos Genéticos como uma técnica de otimização, e o *Bottom-Left* como uma técnica de encaixe das peças na chapa retangular. Alguns testes foram executados e os resultados ficaram no máximo 10% acima da solução ótima.

Palavras-chave: minimizar desperdício, problemas de corte, algoritmos genéticos, *Bottom-Left*.

ABSTRACT. A hybrid genetic algorithm for two-dimensional industrial cutting.

The implementation of an algorithm to solve the two-dimensional cutting problem is provided. A good and fast solution for this problem has to be found and the use of rectangular shapes must be optimized to reduce the wasted area made by cuts about the rectangular shapes. Genetic Algorithms were used as an optimization technique and the Bottom-Left method was employed to put the objects in the rectangular shape. The system was checked for results, which were at least 10% over the optimum solution.

Key words: minimizing wasted area, cutting problems, genetic algorithms, Bottom-Left.

Introdução

O problema de corte bi-dimensional é um problema de otimização combinatória (Goa, 2001), que consiste, basicamente, na determinação de padrões de corte de unidades de material de maneira a produzir um conjunto de unidades menores, satisfazendo determinadas restrições. Normalmente, o objetivo nesses problemas é minimizar o desperdício de material envolvido no processo. Esse tipo de problema surge em muitas aplicações práticas, tais como: corte de bobinas de papel (Maculan *et al.*, 1988), de tecido (Richter, 1992), de bobinas de alumínio (Helmberg, 1995), de chapas de madeira (Morabito e Arenales, 1998), de vidro; entre outras.

Numa indústria, cortes e empacotamentos constituem componentes importantes na formação do custo final dos produtos, e qualquer redução de custos é sempre bem-vinda para as empresas e indústrias. Porém, trata-se de um problema de otimização combinatória NP-difícil que na prática são extremamente difíceis de serem resolvidos por algoritmos exatos para a obtenção da solução ótima. Sendo que esta solução requer muito tempo de

processamento computacional. Para contornar essa restrição de tempo utiliza-se algoritmos heurísticos, que nem sempre garantem a solução ótima, mas garantem uma solução muito boa com baixo esforço computacional.

O objetivo deste trabalho é a implementação de um algoritmo eficaz que solucione o problema, ou seja, que encontre uma solução muito boa (viável) e rápida para este problema, otimizando o uso das chapas para diminuir o desperdício gerado com os cortes que são feitos sobre a mesma.

Quanto ao problema de corte, há dois tipos de corte que podem ser feitos: corte guilhotinado e corte não-guilhotinado. No corte guilhotinado (Morabito e Arenales, 1998), cada corte é feito por toda a extensão da chapa retangular, gerando assim duas chapas retangulares de menor tamanho do que a original. Geralmente, esse tipo de corte é chamado de guilhotinado de 2-estágios, pois o corte é feito primeiro com a guilhotina na vertical, e depois rotaciona a guilhotina 90° e faz o corte na horizontal. A Figura 1 ilustra o corte guilhotinado. Já no corte não-guilhotinado (Jakobs, 1993), cada corte é feito exatamente sobre o tamanho da peça, sem haver a descaracterização da chapa retangular,

como acontece no corte guilhotinado. A Figura 2 ilustra o corte não-guilhotinado.

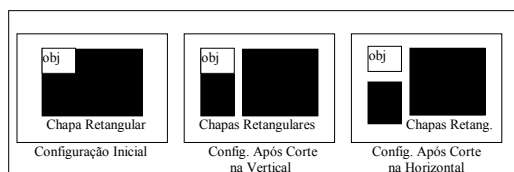


Figura 1. Corte guilhotinado

Para se desenvolver um algoritmo que resolva o problema de corte industrial, deve-se considerar a complexidade do problema, determinada pela geometria dos objetos a serem gerados, pois, os objetos podem ser regulares (retângulos) (Hopper e Turton, 1999) ou irregulares (círculos e outros) (Dowsland e Dowsland, 1995), sendo que os irregulares são mais difíceis de ser tratados devido a sua geometria.

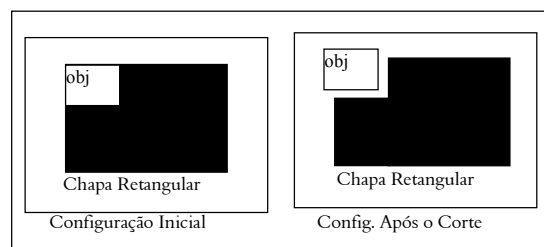


Figura 2. Corte não-guilhotinado

Um dos primeiros estudos sobre o problema de corte foi por Gilmore e Gomory (1961), que estudaram o problema de *cutting-stock* unidimensional. Posteriormente, foram propostos alguns métodos para a resolução do problema de corte bi-dimensional.

Um dos métodos mais utilizados é a geração de colunas (Rocha, 1997). Outro método utilizado, só que com menor frequência, é o *Simulated Annealing* (Ciurlizza e Chávez, 1998). Algoritmos Genéticos (Goldberg, 1989) é um método que também tem ganhado destaque nas publicações mais recentes. Os Algoritmos Genéticos vem sendo estudado e aplicado em muitos problemas de corte e empacotamento (Jakobs, 1996; Hopper e Turton, 1999).

Algoritmos Genéticos já foram utilizados para a resolução do problema de corte guilhotinado em (András *et al.*, 1996; Kröger, 1995); a resolução do *bin packing* em (Falkenauer, 1992 e Hwang, 1994); a solução da compactação, *bin packing* e *nesting problems* em (Goodman *et al.*, 1994). Para o problema de corte não-guilhotinado pode ser citado o trabalho de Hopper e Turton (1999), que utiliza algoritmos genéticos combinados com *Bottom-Left*, porém, os

autores não deixam claro os detalhes da implementação.

Há também um projeto temático na área de corte e empacotamento assistido por computador (Ceac, 1999). Este projeto engloba 4 universidades: USP - São Carlos, Ufscar - São Carlos, ITA - São José dos Campos e Inpe - São José dos Campos, Estado de São Paulo.

O problema tratado neste artigo consiste em encaixar um número pré-determinado de peças bi-dimensionais retangulares em uma chapa retangular, utilizando o corte não-guilhotinado, de modo que a área utilizada seja a mínima possível. Ou seja, a intenção deste problema é a minimização do desperdício que é gerado com o encaixe das peças na chapa retangular.

Como não há algoritmos exatos com baixo tempo computacional que resolvam este problema, há a necessidade da utilização de técnicas heurísticas para a sua implementação. Por este motivo, utilizou-se algoritmos genéticos como uma técnica heurística e forma de otimização da solução.

Material e métodos

Técnica de encaixe (*Bottom-Left*)

Na literatura podem ser encontrados alguns algoritmos para o encaixe das peças na chapa retangular. Os algoritmos mais utilizados são: o *Next-Fit*, o *First-Fit* e o *Bottom-Left*. O algoritmo genético proposto neste trabalho utiliza o algoritmo *Bottom-Left* como função de avaliação.

O *Bottom-Left* toma uma lista com as peças a serem encaixadas e prossegue com o processo de encaixe das peças colocando a primeira no canto inferior esquerdo e as demais são encaixadas, sempre que possível, na parte inferior com a seguinte prioridade: primeiramente tenta-se encaixar a peça o mais próximo da base (eixo x), e depois deve ser movimentada mais a esquerda possível da chapa, sem sobrepor nenhuma peça. Depois que todas as peças são encaixadas é possível determinar o desperdício. A Figura 3 mostra um exemplo de como funciona o processo de encaixe desta técnica.

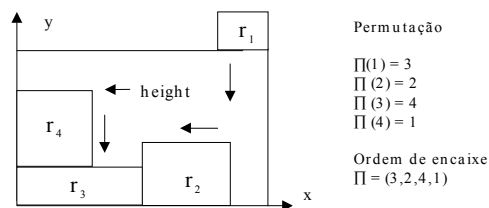


Figura 3. Ilustração do Bottom-Left

O desperdício é calculado através da altura da solução obtida. A altura de uma solução corresponde ao canto superior da peça que está em um nível mais alto. A melhor solução é aquela que apresenta a menor altura. Como o objetivo deste trabalho é a minimização do desperdício, logo, deseja-se a minimização da altura.

Algoritmos genéticos

Algoritmos Genéticos (Goldberg, 1989) são métodos de otimização e busca inspirados nos mecanismos de evolução de população de seres vivos. Os algoritmos baseados nesta técnica seguem o princípio da seleção natural e sobrevivência do mais apto. De acordo com Charles Darwin, “Quanto melhor um indivíduo se adaptar ao seu meio ambiente, melhor será sua chance de sobreviver e gerar descendentes”.

As técnicas de busca e otimização geralmente apresentam:

- um espaço de busca, onde estão todas as possíveis soluções do problema;
- uma função objetivo (geralmente chamada de função de aptidão), que é utilizada para avaliar as soluções produzidas, associando a cada uma delas uma nota.

A otimização consiste em achar uma solução que corresponda ao ponto de máximo ou mínimo da função objetivo. Se a função for de maximizar, deseja-se encontrar o ponto de máximo, caso contrário, deseja-se encontrar o ponto de mínimo. Como deseja-se a minimização do desperdício que é gerado no encaixe das peças na chapa retangular, espera-se encontrar o ponto de mínimo da função objetivo.

Um pseudo-código que exemplifica o funcionamento de um algoritmo genético é descrito abaixo.

Seja $S(t)$ a população de cromossomos na geração t
 $t \rightarrow 0$

inicializar $S(t)$

avaliar $S(t)$

Enquanto o critério de parada não for satisfeito faça

$t \rightarrow t+1$

selecionar $S(t)$ a partir de $S(t-1)$

aplicar *crossover* sobre $S(t)$

aplicar mutação sobre $S(t)$

avaliar $S(t)$ //aplicação do *Bottom-Left*

fim enquanto

A estrutura do algoritmo segue um padrão bastante conhecido, porém, a etapa de avaliação é feita por um algoritmo de encaixe, que nesse caso é

utilizado o *Bottom-Left*. Por essa razão tem-se um algoritmo híbrido.

Representação dos cromossomos

Para este problema de corte, o cromossomo é representado por um vetor que contém a ordem na qual as peças serão encaixadas. Um exemplo do vetor pode ser dado pela Figura 4.

3	7	2	5	8	1	6	4
---	---	---	---	---	---	---	---

Figura 4. Representação do cromossomo para o problema de corte

Cada número do vetor corresponde ao número da peça que será encaixada. As informações de cada peça, representadas pela altura e largura, também são armazenadas em uma lista.

Seleção

A seleção dos cromossomos para cruzamento é realizada sobre um percentual dos melhores cromossomos. Neste trabalho utilizou-se uma variação entre 30% e 60%. Essa técnica é conhecida como Elitismo (Lacerda e Carvalho, 1999).

Crossover

Neste problema de corte, o operador de *crossover* gera um número aleatório entre 1 e a metade do primeiro cromossomo pai C_1 . Este número gerado é chamado de “ponto de corte”. E será o índice que indicará em que ponto do cromossomo C_1 começará o cruzamento. A partir deste ponto de corte, pega-se metade do cromossomo C_1 e coloca no início do cromossomo filho que será gerado F_1 . Depois ele percorre o segundo cromossomo pai C_2 , verificando se a peça está no cromossomo F_1 ou não. Se não estiver, ele coloca na primeira posição vazia do cromossomo F_1 . Se a peça estiver, ele verifica se a próxima peça do cromossomo C_2 está ou não no cromossomo F_1 . E assim sucessivamente, até que o cromossomo F_1 esteja completo. A Figura 5 ilustra um exemplo da aplicação deste tipo de *crossover*, onde os números em negrito nos cromossomos C_1 e C_2 indicam que eles foram utilizados para a geração do cromossomo F_1 (Figura 5).

Ponto de Corte: 3	
Cromossomo pai C1:	3 6 5 8 1 2 7 4
Cromossomo pai C2:	5 7 6 2 8 4 3 1
Cromossomo filho F1:	5 8 1 2 7 6 4 3

Figura 5. Ilustração do crossover

Quando o cromossomo que será gerado pelo *crossover* estiver pronto, aplica-se a função de mutação nele. Neste trabalho foram testados outros *crossover*, porém, este foi o que apresentou o melhor resultado.

Mutação

Este operador funciona invertendo o valor de algum gene do cromossomo. Para este problema de corte, o operador inverte as dimensões de uma peça, ou seja, ele inverte a altura e o comprimento de uma peça. A Figura 6 ilustra um exemplo deste operador.

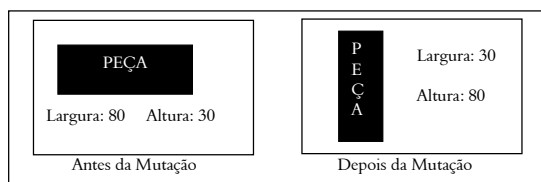


Figura 6. Exemplo de Mutação

A mutação melhora a diversidade dos cromossomos de uma população, mas por outro lado, destrói a informação contida no cromossomo. Por este motivo, deve ser utilizada em pequena proporção, geralmente entre 0,5% e 5%.

Após o cromossomo ter passado pela mutação, aplica-se nele o *Bottom-Left* para avaliar se o cromossomo gerado é bom ou não.

Função Avaliativa

Através de uma técnica de encaixe das peças na chapa retangular (*Bottom-Left*), é possível quantificar o desperdício gerado de um dado cromossomo. Esse valor é utilizado para definir a aptidão do cromossomo.

Resultados

Alguns testes foram realizados com dados obtidos da OR-Library (Beasley, 1990). Os mesmos dados também foram utilizados por Hopper e Turton (1999), servindo assim de base para a comparação com os resultados deste algoritmo. Infelizmente não há na literatura pesquisada nenhuma outra base de dados em que a comparação possa ser feita.

Os arquivos utilizados para os testes estão divididos em categorias, sendo que cada categoria contém o número de peças, a largura da chapa retangular e a altura da solução ótima.

Para a realização dos testes o algoritmo foi implementado em C++ Builder. A Figura 7 ilustra a interface gráfica do sistema. Para facilitar a implementação foi adotado o ponto de origem (0,0)

como sendo o canto superior esquerdo da tela. Portanto, o preenchimento da área disponível ocorre de cima para baixo.

Os melhores resultados obtidos são mostrados na Tabela 1 juntamente com os resultados de Hopper e Turton (1999), assim como a altura ótima de cada categoria. A coluna identificada pela sigla **GABL** corresponde aos resultados obtidos com o algoritmo proposto. As Figuras 7, 8 e 9 ilustram, as melhores soluções encontradas pelo algoritmo para as Categorias 1, 3 e 5, respectivamente.

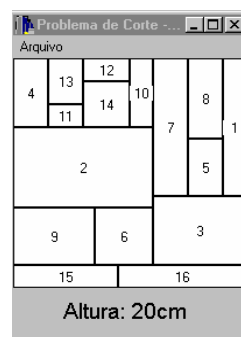


Figura 7. Melhor solução encontrada para a categoria C1 - 20 cm

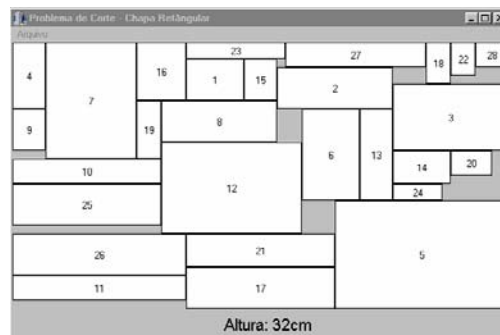


Figura 8. Melhor solução encontrada para a categoria C3 - 32 cm

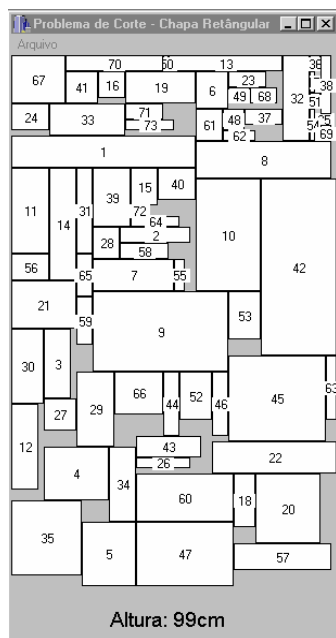


Figura 9. Melhor solução encontrada para a categoria C5 - 99 cm

Tabela 1. Comparação dos resultados

Categoria	Número de Peças	Largura da Chapa (cm)	Altura Ótima (cm)	Hopper e Turton (cm)	GABL. (cm)
C1	16 ou 17	20	20	-----	20
C2	25	40	15	16	16
C3	28 ou 29	60	30	32	32
C4	49	60	60	63	64
C5	73	60	90	-----	99
C6	97	80	120	127	135

Visualizando a tabela, observa-se que o algoritmo implementado consegue a solução ótima para a primeira categoria, mas a cada categoria que se sucede, a diferença entre a solução ótima e a solução encontrada pelo algoritmo aumenta. Esta é uma consequência natural observada nos problemas de otimização combinatória, à medida que o espaço de busca aumenta.

Discussão

Para verificar a qualidade das soluções, foram feitos testes com os mesmos dados utilizados por Hopper e Turton (1999). Nestes testes, a solução ótima era conhecida, pois as peças foram geradas a partir de um retângulo. Após a execução dos testes, chegou-se a conclusão de que as soluções encontradas pelo algoritmo implementado neste trabalho são muito boas. Comparando com Hopper e Turton (1999), observa-se, apenas, uma diferença maior no custo da solução para a última categoria.

Mas, no geral, o algoritmo se portou bem e as soluções encontradas ficaram em média entre 6% e 10% acima da solução ótima, e com um tempo de processamento que variou entre 30 e 150 segundos em um computador Pentium 233MHz com 32 Mb RAM.

O objetivo deste trabalho foi atingido e os resultados obtidos encorajam a continuar na investigação dessa vertente de pesquisa.

Referências

- ANDRÁS, P. *et al.* A genetic solution for the cutting stock problem. In: PROCEEDINGS OF THE FIRST ON-LINE WORKSHOP ON SOFT COMPUTING, 1996, Nagoya University. *Proceedings...* Nagoya: Nagoya University, p. 87-92, 1996
- BEASLEY, J.E. OR-LIBRARY: Distributing Test Problems by Electronic Mail. *J. Oper. Res. Soc.*, Hamps, v. 41, p. 1069-1072, 1990
- CEAC - *Cortes e empacotamento assistidos por computador*, 1999. Disponível em: <<http://www.lac.inpe.br/po/projects/ceac/fapesp3.html>>. Acesso em: 21 mar 2001
- CIURLIZZA, A. G.; CHÁVEZ, B. C. Uso del Simulated Annealing para Minimizar el Desperdicio de Tela. In: CONGRESO LATINO-IBEROAMERICANO DE INVESTIGACIÓN OPERATIVA, 9., Buenos Aires. *Anais...*[CD-ROM] Buenos Aires, 1998, p. 1-10.
- DOWSLAND, K. A.; DOWSLAND, W. B. Solution Approaches to Irregular Nesting Problems. *Eur. J. Opera. Res.*, Hamps, v. 84, p. 506-521.
- FALKENAUER, E. A genetic algorithm for bin packing and line balancing. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 1992, Nice, 1992. *Proceedings...* Nice, 1992, p. 1186-1192.
- GILMORE, P. C.; GOMORY, R. E. A linear programming approach to the cutting-stock problem. *Opera. Res.*, Hamps, v. 9, p. 848-859, 1961.
- GOA. *Combinatorial Optimization*, 2001. Disponível em: <<http://goa.pos.dcc.unicamp.br/otimo>>. Acesso em: 15 dez 2001
- GOLDBERG, D. E. *Genetic algorithms in search optimization & machine learning*, USA: Addison-Wesley, 1989.
- GOODMAN, E. D. *et al.* A genetic algorithm approach to compaction, bin packing, and nesting problems. Michigan State University, 1994 (Technical Report).
- HELMBERG, C. Cutting aluminum coils with high length variables. *Ann. Oper. Res.*, Bussum, v. 57, p. 175-189, 1995.
- HOPPER, E.; TURTON, B. A Genetic Algorithm for a 2D Industrial Packing Problem. *Computers and Industrial Engineering*, Cardiff, v. 37, n. 1, p. 375-378, 1999.
- HWANG, S. M. On solving rectangle bin packing problems using genetic algorithms. In: IEEE INTERNATIONAL CONFERENCE ON SYSTEMS,

- MAN, AND CYBERNETICS, 1984, New York, *Proceeding...* New York, 1994, p. 1583-1590.
- JAKOBS, S. On Genetic Algorithms for the Packing of Polygons. *Eur. J. Opera. Res.*, Hamps, v. 88, p. 165-181.
- KRÖGER, B. Guillotineable bin-packing: A genetic approach. *Eur. J. Opera. Res.*, Hamps, v. 84, n. 3, p. 645-661, 1995.
- LACERDA, E. G. M.; CARVALHO, A. C. P. L. F. Introdução aos algoritmos genéticos. In: CONGRESSO NACIONAL DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 1999, Rio de Janeiro. *Anais...* Rio de Janeiro, 1999, v. 2, p. 51-126.
- MACULAN, N. *et al.* Problemas de otimização relacionados ao corte de bobinas de papel. In: COLÓQUIO BRASILEIRO DE MATEMÁTICA, 16, 1988, Rio de Janeiro. *Anais...* Rio de Janeiro, 1988, p. 159-182.
- MORABITO, R.; ARENALES, M. Otimização das Operações de Corte de Chapas na Indústria de Móveis. In: OFICINA NACIONAL DE PCE, XX CNMAC, 2, 1998, Curitiba. *Anais...* Curitiba, 1998, p. 43-52.
- RICHTER, K. Solving sequential interval cutting problems via dynamic programming. *Euro. J. Opera. Res.*, Hamps, v.57, p. 332-338, 1992.
- ROCHA, M. N. *Otimização de corte de barras*. 1997. Dissertação (Mestrado) - Universidade Federal de Viçosa, Viçosa, 1997
- Received on October 18, 2001.*
- Accepted on November 26, 2001.*