

Proposta de aplicação de um algoritmo genético para solução do problema do corte bidimensional em lâminas de vidro

F. Costa, N. C. F. Canto, R. J. Sassi

Resumo— O projeto consiste no estudo e desenvolvimento de métodos de solução, baseados em computação evolutiva, para Problemas de Satisfação de Restrições (PSR), os quais abrangem uma variedade de problemas de otimização combinatória. Em particular, é proposto o desenvolvimento de um algoritmo genético para solução e otimização do problema de corte de lâminas de vidro, o qual consiste em determinar o arranjo de peças a serem cortadas maximizando a utilização da lâmina e respeitando as restrições impostas pelo fluxo de produção e cronogramas de entregas.

Palavras-chave— Problemas de Satisfação de Restrição, Algoritmo Genético, Problemas de Corte Bidimensional.

IV. INTRODUÇÃO

Os PSR são caracterizados por uma estrutura comum que consiste em encontrar um conjunto de objetos cujo estado precisa satisfazer um número de restrições ou limitações [1][2]. Os PSR frequentemente apresentam alta complexidade e requerem uma combinação de heurísticas e métodos de busca para que possam ser solucionados em um tempo razoável (complexidade de tempo polinomial) [3]. São vários os sistemas de produção que envolvem os PSR, alguns exemplos são: alocação de tarefas; problemas logísticos de empacotamento e processos de corte de materiais [4][5]. Devido ao potencial das aplicações práticas para otimizar processos industriais e às dificuldades para obtenção de soluções exatas, os PSR têm sido objetos de intensas pesquisas nas áreas Pesquisa Operacional (PO) e Inteligência Artificial (IA), uma vez que o estudo de tais problemas fornece uma base comum para análise e solução de outros problemas que pertencem à mesma categoria. Neste contexto, a computação evolutiva tem ganhado espaço no cenário acadêmico e sendo utilizada cada vez mais na solução de tais problemas. A computação evolutiva compreende um grupo de métodos computacionais que incorporam mecanismos evolutivos presentes na natureza e descritos pela teoria da evolução natural [6][7]. Como exemplo, tem-se os algoritmos genéticos (AG) os quais têm se mostrado uma ferramenta promissora para análise e solução dos PSR [8]. O projeto de Iniciação Científica em desenvolvimento visa o estudo e aperfeiçoamento de métodos de solução, baseados em computação evolutiva, para solução de PSR oriundos de sistemas de produção. Especificamente é proposto o desenvolvimento de um algoritmo genético para solução do

problema de corte bidimensional em lâminas de vidro.

V. PROBLEMAS DE CORTE

Problemas de corte são uma importante classe entre os problemas de satisfação de restrição [9][10]. Em particular, o problema do corte bidimensional consiste no arranjo e corte de peças de diferentes tamanhos e formatos com o objetivo de minimizar as perdas de material. As abordagens tradicionais para o corte bidimensional podem ser divididas em duas categorias: programação linear (PL) e métodos heurísticos (MH), sendo os métodos heurísticos os mais populares na solução de problemas reais. Entretanto, a utilização de Algoritmos Genéticos (AG) na solução de problemas de restrição tem crescido e a técnica tem se tornado uma importante ferramenta para modelagem e solução dessa classe de problemas.

III. O CORTE EM LÂMINAS DE VIDRO

O trabalho propõe o desenvolvimento de um AG para a solução de um problema real de uma empresa de corte em lâminas de vidro. O problema consiste em maximizar a utilização das lâminas padrão. As lâminas padrão têm o formato 6000 x 3210 (mm) e são utilizadas para produção de peças retangulares de diferentes dimensões. Atualmente, os leilotes de corte são produzidos por *softwares* que acompanham as máquinas operatrizes, estes *softwares* oferecem um conjunto de algoritmos de otimização baseados em PL e MH e que produzem resultados que variam, dependendo do número de lâminas utilizadas e peças a serem cortadas em determinado pedido, mas que, em média, ficam em torno de 5% do total de material utilizado. A tabela I apresenta um exemplo de pedido encaminhado à produção e a tabela II apresenta valores típicos de desperdício gerado por esses algoritmos.

TABELA I
EXEMPLO DE PEDIDO ENCAMINHADO PARA PRODUÇÃO

Cód. do Produto	Medidas (mm)	Quantidades
10010	720x220	27
10020	515x2150	43
10050	950x1410	15

TABELA II

VALORES TÍPICOS DE DESPERDÍCIO DE MATERIAL PRODUZIDOS POR *SOFTWARES* PROPRIETÁRIOS DE MÁQUINAS OPERATRIZES PARA CORTE DE LÂMINAS DE VIDRO

Lâmina	Área da Lâmina (m ²) – (A)	Desperdício
1	19,26	5,86%
2	19,26	3,08%
3	19,26	3,94%
4	19,26	5,50%

O objetivo é comparar os resultados do AG proposto com os resultados gerados pelos algoritmos do sistema e verificar se existirão ganhos significativos com a utilização de um algoritmo evolutivo. A seção seguinte apresenta os detalhes da arquitetura e codificação do AG proposto.

IV ALGORITMO GENÉTICO PARA OTIMIZAÇÃO DE CORTE EM LÂMINAS DE VIDRO

Inicialmente o AG proposto consistirá de uma versão que selecionará, a partir do pedido solicitado à produção, um número n de peças com o objetivo de minimizar as sobras na lâmina de vidro padrão. No problema em questão, trabalha-se com um grande número de peças a serem cortadas e o objetivo é encontrar uma combinação de peças, extraída do pedido, que minimize a sobra de material em cada lâmina. A tabela III mostra o exemplo de um pedido encaminhado para a produção contendo apenas 10 itens e quatro tipos de peças a serem cortadas.

TABELA III
EXEMPLO DE PEDIDO DE PEÇAS PARA CORTE

Peça	Quantidade	Código da Peça	Dimensões
0	3	5010	720x220
1	2	5020	515x2150
2	4	5050	950x1410
3	1	5090	250x1850

O pedido descrito na tabela III será representado por um vetor p que será utilizado para geração da população inicial do AG.

$$p = (a_0, a_0, a_0, a_1, a_1, a_2, a_2, a_2, a_2, a_3) \quad (1)$$

A. População inicial

Os passos seguintes descrevem o processo de geração da população inicial do AG. A população é formada por candidatos a solução do problema, ou seja, arranjos de peças extraídas do pedido p .

1. Criar uma cópia (p') do pedido p para manipulação;
2. Sortear aleatoriamente um item i do pedido p' ;
3. Remover o item sorteado i de p' e calcular

$$S = A - a_i \quad (2) \text{ sendo:}$$

A a área da lâmina padrão;

S a área ainda não utilizada;

a_i a área da peça i sorteada.

4. Verificar se $S > 0$ então colocar a peça a_i no novo indivíduo;

5. Repetir o processo até que o teste acima falhe ($S \leq 0$) ou que p' esteja vazio (não é possível encaixar mais nenhuma peça sem ultrapassar a área total da lâmina);

6. repetir os itens de 1 a 5 para gerar os n indivíduos da população;

Desta forma, teremos n indivíduos contendo um número variado de peças e respeitando a restrição de que a soma das áreas de todas as peças não ultrapasse a área da lâmina.

B. Codificação dos indivíduos

Neste exemplo, como o pedido só possui 10 itens, cada peça será codificada com 3 bits, 2 bits para representar o número da peça e 1 bit para representar a posição (em pé ou deitada);

TABELA IV
CODIFICAÇÃO DAS PEÇAS PRESENTES NOS INDIVÍDUOS

Peça	Codificação 1 (em pé)	Codificação 1 (deitada)
0	000	001
1	010	001
2	100	001
3	110	001

Alguns exemplos de indivíduos:

Indivíduo 1 - 010011100 indivíduo formado por 3 peças (peça 1 (em pé), peça 1 (deitada) e peça 2 (em pé)).

Indivíduo 2 - 110101011000 indivíduo formado por 4 peças (peça 3 (em pé), peça 2 (deitada), peça 1 (deitada) e peça 0 (em pé)).

Portanto, após gerar os n indivíduos no item 4, os indivíduos serão codificados para o formato binário e sorteado o último bit representando a posição da peça. Essa representação facilitará a aplicação dos operadores genéticos *crossover* e a mutação.

C. Fitness

O cálculo do *fitness* dos indivíduos da população será realizado com o auxílio do algoritmo de posicionamento de peças denominado *bottom-left* [11][12] desenvolvido em Java. O processo consiste em encaixar as peças do indivíduo na lâmina e calcular a área não utilizada da lâmina.

$$f(n) = A - \sum_{i=1}^x a_i \quad (3) \text{ Sendo:}$$

$f(n)$ – o *fitness* do indivíduo n ;

A – área da lâmina padrão;
 a_i – área de cada peça que compõe o indivíduo;
 x - o número de peças do indivíduo

D. Seleção

O método de seleção de indivíduos utilizado no AG proposto é denominado de Roleta (*roulette wheel*). Neste método, atribui-se a cada indivíduo uma probabilidade proporcional ao *fitness* relativo de cada indivíduo na população.

$$p_{selec} = \frac{f(n)}{\sum_{i=1}^N f(n_i)} \quad (4)$$

sendo:

p_{selec} = probabilidade do indivíduo n de ser selecionado;

$f(n)$ = o *fitness* do indivíduo n , dado pela equação (3);

$\sum_{i=1}^N f(n_i)$ = o somatório do *fitness* de todos os indivíduos.

E. Crossover

No algoritmo proposto utiliza-se o operador *crossover* de um ponto com uma taxa fixa de probabilidade de ocorrência entre os indivíduos. Para este operador, são selecionados dois indivíduos (pais) que dão origem após a recombinação a dois novos indivíduos (filhos). A figura 1 ilustra o processo de *crossover* aplicado a um par de indivíduos [].

		Ponto de corte
Pai 1	0 1 0	0 1 1 1 0 0
Pai 2	1 1 0	1 0 1 0 1 1 0 0 0
Filho 1	0 1 0	1 0 1 0 1 1 0 0 0
Filho 2	1 1 0	0 1 1 1 0 0

Fig. 1. Exemplo de *crossover* entre dois indivíduos da população.

Uma heurística é implementada ao processo de escolha do ponto de corte para evitar a formação de indivíduos que não atendam as restrições de área da lâmina padrão.

F. Mutação

O objetivo do operador mutação é inserir diversidade na população de indivíduos sem desestabilizá-la. Por isso, a taxa de mutação utilizada é pequena 1% do total de genes da população.

V RESULTADOS ESPERADOS

Espera-se que após um número n gerações o algoritmo genético proposto seja capaz de extrair do pedido original um número k de peças e alocá-las de forma otimizada na lâmina padrão. A figura 2 apresenta um exemplo de leiaute que deverá ser gerado pelo algoritmo em desenvolvimento.

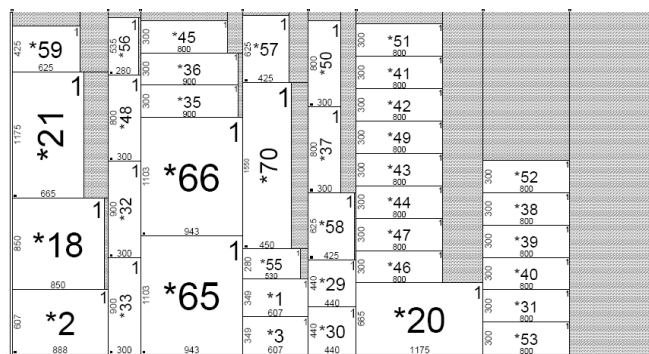


Fig. 2. Exemplo de leiaute produzido pelo algoritmo genético.

O modelo apresentado consiste da primeira versão do AG e os resultados obtidos irão orientar os próximos passos da pesquisa com possíveis alterações e melhorias no algoritmo com o objetivo de melhorar a performance do modelo.

REFERÊNCIAS

- [1] C. BRODERICK, A. MARY, C. CARLOS e M. ERIC, "Using Constraint Programming to solve Sudoku Puzzles", *Third 2008 International Conference on Convergence and Hybrid Information Technology*, 926-931, 2008.
- [2] B. CRAWFORD, M. ARANDA, C. CASTRO e E. MONFROYUSING, "Constraint Programming to solve Sudoku Puzzles", *Third International Conference on Convergence and Hybrid Information Technology*, pp. 926-931, 2008.
- [3] A. N. BERNARD, "Representation Selection for Constraint Satisfaction: A Case Study Using n-Queens", *IEEE Expert*, 885, 16-23, 1990.
- [4] H. BAUMGÄRTEL, D. CHRYSLER, "Distributed Constraint Processing for Production Logistics", *IEEE Intelligent System*, 40-48, 2000.
- [5] W. SHEN, "Distributed Manufacturing Scheduling Using Intelligent Agents", *IEEE Intelligent System*, pp. 88-94, 2002.
- [6] J. H. HOLLAND, "Adaptation in Natural and Artificial Systems", The University of Michigan Press, Ann Arbor, MI, 1975.
- [7] J. R. KOZA, "Genetic Programming: On the Programming of Computers by Means of Natural Selection". MIT Press, 1992.
- [8] W. BANTZHAFF, J. R. KOZA, C. RYAN, L. SPECTOR, C. JACOB, "Genetic programming.", *Intelligent Systems and Their Applications* IEEE, 15, 3, 74 – 84, 2000.
- [9] E. HOPPER, B. TURTON, "A Genetic Algorithm for a 2D Industrial Packing Problem", *Computers and Industrial Engineering*, Cardiff, v. 37, n. 1, p. 375-378, 1999.
- [10] H. DYCKHOFF, "A typology of cutting and packing problems", *European Journal of Operational Research*, 44(2), 145–159, 1990.
- [11] T. F. GONZALES, "Handbook of Approximation Algorithms And Metaheuristics", Chapman & Hall/CRC, 2007.
- [12] C. COTTA, M. SEVAUX, K. SÖRENSEN, *Adaptive and Multilevel Metaheuristics*, Kenneth, 2008.

Flávio Moreira da Costa: Técnico em informática e redes de comunicação pela Escola SENAI "Suíço-brasileira" (2005), atualmente é estudante do 4º ano do curso de Ciência da Computação pela

Universidade Nove de Julho. Participa de em projetos de iniciação científica relacionado a Inteligência Artificial (Algoritmos Genéticos). Tem experiência nas áreas de Análise e Desenvolvimento de Sistemas Computacionais voltados para Gestão Administrativa e de Produção.

Nilton Canto: graduado em Física pela Universidade de São Paulo (1997), concluiu o mestrado em Engenharia Elétrica pela Universidade Presbiteriana Mackenzie em 2002 e o doutorado em Engenharia Elétrica pela Universidade de São Paulo em 2008. É membro da sociedade Brasileira de Computação (SBC), membro do Institute of Electrical and Electronics Engineers (IEEE) e membro da IEEE Computer Society. Atualmente é professor da Universidade Nove de Julho, do Centro Universitário Unisantanna e do Colégio Etapa. Possui mais de 30 softwares educativos desenvolvidos e em suas atividades profissionais interagiu com cinco colaboradores em co-autorias de trabalhos científicos. Em seu currículo Lattes os termos mais frequentes na contextualização da produção científica, tecnológica e artístico-cultural são: Córtex Visual, Sincronismo, Osciladores Acoplados, Ferramentas de Autoria, Sistemas Dinâmicos, Softwares Educativos, Tutores Inteligentes, Algoritmos Adaptativos, Algoritmos Genéticos e Sistemas Cognitivos.

Renato José Sassi: bacharel em Ciências Econômicas pela Faculdade de Economia Finanças e Administração de São Paulo (1987), especialista (pós-graduação Lato Sensu) em Administração de Empresas (área de concentração Análise de Sistemas) pela Fundação Escola de Comércio Álvares Penteado (1988), especialista (pós-graduação Lato Sensu) em Didática do Ensino Superior pelo Centro Universitário Sant'anna (1996), mestre em Administração de Empresas (Gestão de Negócios) pelo Centro Universitário Sant'Anna (1999) e doutor em Engenharia Elétrica pela Escola Politécnica (POLI) da Universidade de São Paulo (2006). Atualmente é pesquisador associado do Grupo de Inteligência Computacional, Modelagem e Neurocomputação (ICONE) do Laboratório de Sistemas Integráveis (LSI) da Escola Politécnica da Universidade de São Paulo (USP), docente e pesquisador do Programa de Mestrado em Engenharia de Produção da Universidade Nove de Julho (UNINOVE). Membro do Institute of Electrical and Electronics Engineers (IEEE) e membro da Computational Intelligence Society (IEEE).