

ALGORITMOS GENÉTICOS PARA O PROBLEMA DE ANINHAMENTO NA INDÚSTRIA DE EMBALAGENS: UM ESTUDO COM MODELOS REDUZIDOS

ROBERTO SELOW, HEITOR S. LOPES, FLÁVIO NEVES JR.

Programa de Pós-graduação em Engenharia Elétrica e Informática Industrial - CPGEI

Centro Federal de Educação Tecnológica do Paraná – CEFET-PR

Av. 7 de setembro, 3165

80230-901 – Curitiba, PR

E-mails: selow@bol.com.br, hslopes@cpgei.cefetpr.br, neves@cpgei.cefetpr.br

Resumo— O problema de aninhamento de áreas planas é encontrado em diversas indústrias, com diferentes limitações e objetivos. Entre estas estão as indústrias de embalagem e de roupa, que usam partes irregulares côncavas e convexas. Uma peculiaridade da indústria de embalagens é o freqüente uso do mesmo tipo de peças para o processo de aninhamento. Neste artigo foi utilizado algoritmos genéticos (GA) para resolver o problema de aninhamento e o comportamento do GA é analisado com o aumento do número de peças. Usando-se modelos reduzidos pode-se observar a característica multimodal do problema. Observou-se ainda a tendência de crescimento deste até alcançar o tamanho real de um problema industrial. Os resultados obtidos sugerem a combinação do AG com outro método a fim diminuir o espaço da busca sem descartar arranjos viáveis e que podem ajudar à convergência a um ótimo global.

Abstract— The problem nesting parts in plane areas is found in several industries, with different restrictions and objectives. Among these are packing and clothing industries, which use irregular parts like concave and convex ones. A peculiarity of the packing industry is the frequent use of the same part type for each nesting process. In this paper we use genetic algorithms (GA) to solve the nesting problem and the behavior of the GA is analyzed as the number of parts increase. Using reduced models it can be observed the multimodal characteristic of the problem and the growth tendency until reaching the real size of an industrial problem. The results obtained suggest the combination of AG with other method in order to decrease the search space without discarding the promising combinations that can help the convergence to a global optimum.

Keywords— nesting problem, cutting stock, genetic algorithms, packing industry, optimization.

1 Introdução

O problema do aninhamento na indústria de embalagens pode ser definido como sendo a obtenção de um número máximo de embalagens num arranjo numa folha de papel de tamanho conhecido, com a menor perda possível de material. O objetivo deste processo é a otimização da produção por meio de uma minimização dos custos de produção. A figura 1 mostra um exemplo de quatro caixas que serão tiradas de uma folha de papel de tamanho padrão.

Um fator importante na busca de uma solução ótima para este problema é o número de peças que serão aninhadas na folha de papel. Há uma explosão combinatorial à medida que o número de peças aumenta, levando a custos computacionais inaceitáveis. Para problemas do mundo real, o número de peças é, usualmente, menor do que 20.

Algoritmos Genéticos – GA (Goldberg, 1989) têm sido utilizados satisfatoriamente nas últimas décadas para resolver problemas combinatoriais de alto grau de complexidade, em várias áreas da Engenharia (Lopes, 1999) e da Computação. Inclusive, o uso de AG para problemas similares ao descrito aqui, são mencionados por Chu e Beasley (1997) e também por Hopper e Turton (1999).

O objetivo deste trabalho é analisar o comportamento de AG num modelo reduzido do problema de aninhamento e então utilizá-lo para problemas do mundo real.

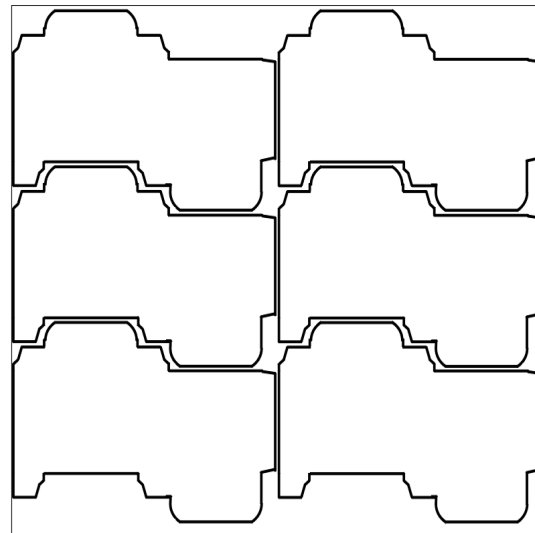


Figura 1. Arranjo de seis embalagens aninhadas em uma folha de papel 560 x 480 mm.

2 Revisão da literatura

O problema do aninhamento ou corte de peças no plano tem sido amplamente estudado na literatura recente, dado que este é um problema bastante comum encontrado em vários ramos da indústria, tais como a indústria gráfica, de embalagens, siderúrgica, vestuário, móveis, etc. De acordo com Han & Na (1996) tal tipo de problema pode ser dividido em

dois grupos principais, de acordo com o número de peças de material que devem ser modeladas: o primeiro grupo considera somente peças de formato retangular, enquanto o segundo grupo utiliza peças de formatos irregulares.

Provavelmente, Gilmore e Gomory (1965, 1966) foram os precursores desta área de pesquisa. Eles utilizaram peças retangulares e lidaram com o problema utilizando técnicas de programação linear. Os mesmos autores também tiveram sucesso com problemas uni, bi e tridimensionais.

Posteriormente, Haims e Freeman (1970) estudaram o problema sem a restrição do número de peças a serem cortadas numa folha de um dado material. O método que eles desenvolveram consistia em obter um retângulo que englobava uma ou mais peças irregulares usando a menor área possível. Tal retângulo era chamado de “módulo”. Módulos eram, então, agrupados em uma folha de material por meio de programação dinâmica. Este algoritmo exigia que o módulo retangular fosse posicionado em um dos cantos da folha. Mais tarde, Adomowicz e Albano (1976) propuseram uma melhoria neste algoritmo, eliminando tal limitação. Este algoritmo foi usado na indústria da construção naval. Também Christofides e Whitlock (1977) apresentaram um algoritmo baseado na busca em árvores que utilizava programação dinâmica para obter cortes de peças retangulares em uma folha.

No que diz respeito a peças irregulares, o que é um problema muito mais complexo, Albano e Sappuppo (1980) propuseram uma técnica que usava métodos de busca heurística. Também utilizando métodos heurísticos, Nee (1984) propôs um algoritmo para o problema de aninhamento na indústria siderúrgica.

Dentre os primeiros artigos publicados utilizando Algoritmos Genéticos (AG) para este tipo de problema encontra-se o trabalho de Ismail e Hon (1992). Desde então, outros autores também se seguiram, como, por exemplo, Fujita e Gakkai (1993) que apresentaram uma abordagem híbrida utilizando AG e um método de otimização local. András et al (1996) desenvolveram um método baseado em AG onde a combinação de peças era representada em uma árvore.

Mais recentemente, Petridis et al (1998) usaram AG e peças representadas como polígonos. Eles também desenvolveram uma técnica específica para variar a função de *fitness* com o objetivo de facilitar a busca pelo algoritmo. Também, Hopper e Turton (1999) propuseram um método que combina AG com busca heurística para peças retangulares e compararam os resultados com busca randômica.

3 Metodologia

3.1 Introdução

Existem duas maneiras de representar peças em problemas de aninhamento: utilizando polígonos ou utilizando pequenos quadrados num esquema semelhante a uma grade. Uma vez que este trabalho é restrito ao estudo de modelos reduzidos, a representação em grade é apropriada. Esta abordagem provê uma representação simplificada para o problema e não gera sobrecarga computacional significativa entre as peças. Cada peça pode ter quatro graus de liberdade e rotação representadas por ϕ_i . Estes ângulos de rotação são: 0° , 90° , 180° and 270° . Cada peça é composta por pequenos quadrados chamados de elementos de área (EA). O conjunto de peças utiliza um sistema de coordenadas no qual a origem é relativa a cada peça e é descrita por três parâmetros: X_i , Y_i e o ângulo ϕ_i . Estes três parâmetros serão codificados no cromossomo do AG (descrito nas próximas seções). Cada peça, então, é descrita por seu conjunto de elementos de área utilizando dois vetores: CX_{jk} e CY_{jk} (como mostrado na figura 2).

Neste estudo, foi utilizado um número variável de peças nas simulações e a carga computacional para encontrar o ótimo (conhecido) foi anotada. Em todos os casos estudados, todos os ótimos possíveis eram previamente conhecidos. Isto permitiu uma avaliação precisa dos resultados alcançados pelo algoritmo implementado. Com um número relativamente pequeno de simulações, foi possível verificar o comportamento do AG à medida que o tamanho do problema crescia.

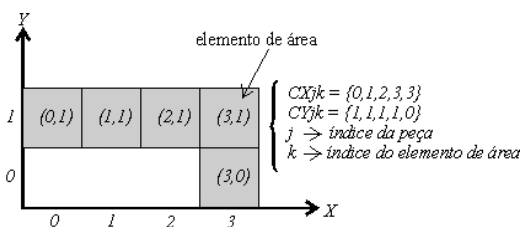


Figura 2. Representação e codificação das peças utilizando a abordagem de elementos de área (EA).

3.2 Codificação e espaço de busca

Com o objetivo de implementar o algoritmo de aninhamento com o AG, é necessário definir o espaço de busca no qual as peças serão movidas, os ângulos de rotação, o sistema de coordenadas para posicionamento de cada peça, o número de peças e a função de *fitness* para computar a qualidade de uma dada solução. Foi utilizado um espaço de busca

baseado nas coordenadas de origem de cada peça, de tal maneira a se manter constante independentemente do tamanho da peça. O espaço de busca foi limitado no intervalo $x=\{0..19\}$ e $y=\{0..19\}$ iniciando na origem da peça. Assim sendo, peças podem ter elementos de área, por exemplo, nas coordenadas $(x, y)=(22,0)$, desde que as coordenadas da origem estejam dentro do intervalo do espaço de busca. Para problemas do mundo real, todos os elementos de área das peças deveriam estar contidos dentro do espaço de busca, uma vez que não seria aceitável que parte de uma embalagem estivesse fora da folha de papel.

Cada peça pode ter quatro ângulos de rotação, 0° , 90° , 180° e 270° , como mostrado na figura 3.

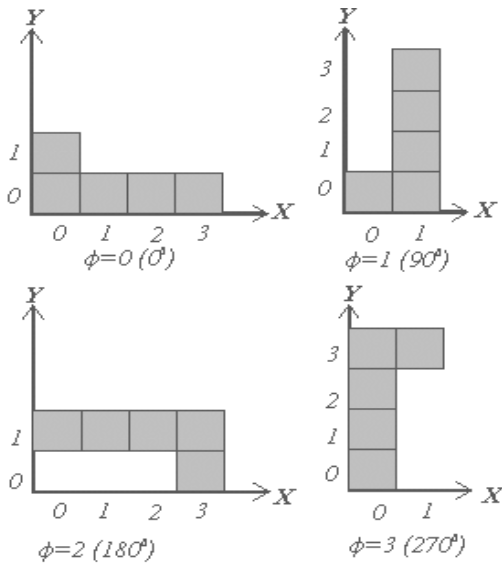


Figura 3. Ângulos de rotação (ϕ) possíveis para cada peça.

A posição de cada peça será calculada considerando a coordenada cuja origem caia na primeira quadrícula da grade, embaixo e à esquerda, dada pelas tangentes à peça. O cromossomo será composto pelo conjunto das coordenadas de origem (X_i , Y_i) e o ângulo (ϕ_i) para todas as peças, onde i é o índice de cada peça. Usando as informações codificadas no cromossomo, o AG vai computar o conjunto de coordenadas CX_{jk} e CY_{jk} , onde j é o índice de cada peça e k o índice dos elementos de área. O conjunto de coordenadas é utilizado para calcular a área total ocupada e a sobreposição entre peças, como mostrado na figura 4.

Nesta versão do algoritmo, o número de peças é dado pelo usuário para que seja avaliada a carga computacional para arranjos de 2, 4, 6 e 8 peças. Para problemas do mundo real, o número de peças será determinado pelo algoritmo propriamente dito, o qual tentará encontrar um arranjo com o maior número de peças numa folha de papel.

O tamanho do espaço de busca para um arranjo de N peças é dado pela equação 1:

$$P(N) = [x \cdot y \cdot n\phi]^N \quad (1)$$

onde:

$$\left\{ \begin{array}{l} P(N) = \text{espaço de busca para } N \text{ peças;} \\ x, y = \text{coordenadas dos eixos;} \\ n\phi = \text{número de possíveis ângulos de rotação} \\ \text{(neste trabalho, limitado a 4);} \\ N = \text{número de peças.} \end{array} \right.$$

Para o modelo reduzido estudado neste trabalho, o espaço de busca cresce à medida que o número de peças aumenta, conforme mostrado na tabela 1.

Tabela 1- Crescimento do espaço de busca como função do número de peças.

N	2	4	6	8
P(N)	$2,56 \times 10^6$	$6,55 \times 10^{12}$	$1,67 \times 10^{19}$	$4,29 \times 10^{25}$
Para todas N, $x=20$, $y=20$ e $n\phi=4$				

3.3 Função de fitness

O objetivo do algoritmo é a maximização da função de *fitness* $f(s)$ definida pela equação 2. O termo $IA(s)$ representa o índice de área e $IS(s)$ o índice de sobreposição. A variável s representa a solução atual sob avaliação.

$$f(s) = 1 - \frac{IA(s) + IS(s)}{2} \quad (2)$$

O índice $IA(s)$, por sua vez, é normalizado no intervalo $[0..1]$ e é definido pelas equações 3 e 4 sendo válido para $\{ \forall E > AI \}$.

$$IA(s) = \{1 - R\}^{IG(gen)} \quad (3)$$

onde:

$$R = \frac{(E - \max_{j,k} (CX_{jk}) \cdot \max_{j,k} (CY_{jk}))}{(E - AI)} \quad (4)$$

O termo E é a área total ocupada pelas peças, baseando-se nas suas coordenadas de origem, assim sendo, é expresso pela equação 5:

$$E = x \cdot y \quad (5)$$

O termo $\max_{j,k} (CX_{jk}) \cdot \max_{j,k} (CY_{jk})$ é a área retangular que envolve as peças. Para o caso

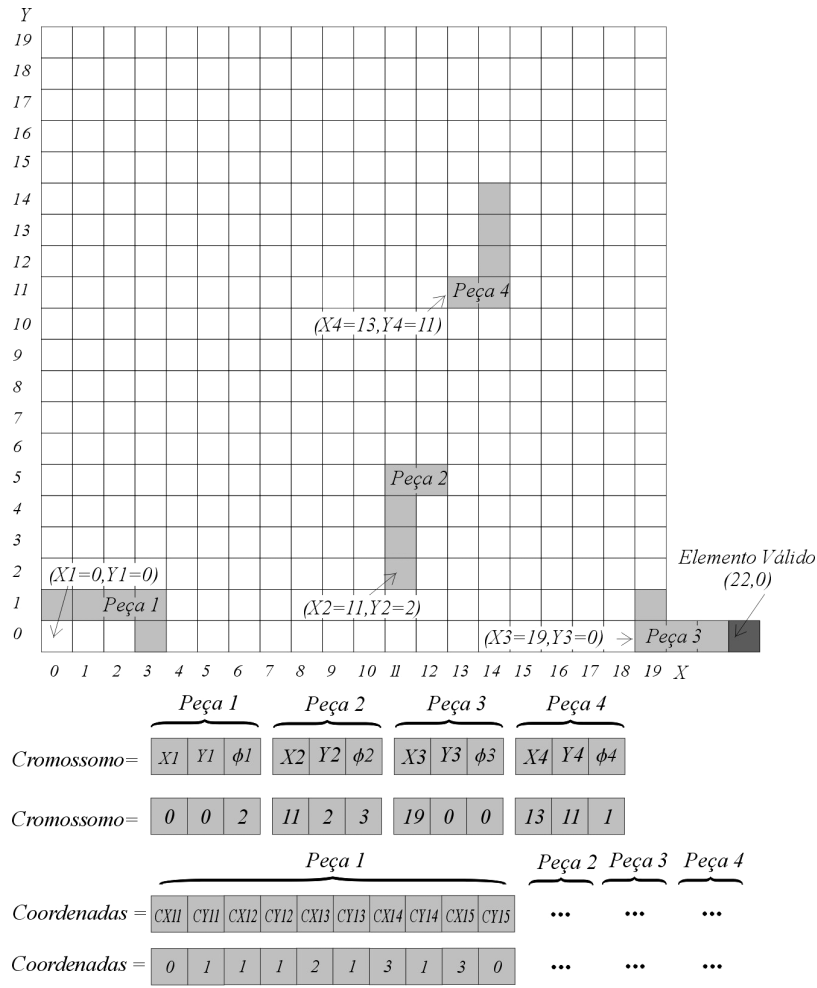


Figura 4. Exemplo de codificação e posicionamento de 4 peças no espaço de busca.

sob estudo, as peças têm um formato que as permite encaixar umas nas outras sem espaços vazios no arranjo. O índice para a área ideal (área ótima para o arranjo) é chamado de AI e é expresso pela equação 6, onde N é o número de peças e EA o elemento de área de cada peça.

$$AI = N.EA \quad (6)$$

Durante o processo de busca o índice $IA(s)$ foi variado dinamicamente, usando o expoente $IG(s)$.

Esta técnica, inspirada no trabalho de Bojarczuk, Lopes e Freitas (2000), permite um compromisso entre a busca de um arranjo ótimo e a manutenção da diversidade genética. Assim é possível manter a diversidade genética da população do AG adaptada dinamicamente ao longo das gerações (Petridis et al, 1998). Na primeira geração, $IG(0)=0,4$, enquanto que na última $IG(maxgen)=1,0$. Para qualquer outra geração, $IG(gen)$ é dado pela equação 7, onde gen é o número da geração atual e $maxgen$ o número máximo de gerações.

$$IG(gen) = \frac{(0.6.gen + 0.4.maxgen)}{maxgen} \quad (7)$$

De maneira análoga, o índice $IS(s)$ também é normalizado no intervalo $[0..1]$ e é definido pela equação 8 na qual $S(s)$ é o índice de sobreposição e SM o valor máximo possível de sobreposição, definido pela equação 9.

$$IS(s) = \left(\frac{S(s)}{SM} \right)^{IG(gen)} \quad (8)$$

$$SM = \frac{(EA.N.(N-1))}{2} \quad (9)$$

Assim sendo, a solução ótima, na qual a função de fitness $f(s)$ se iguala a 1, ocorrerá quando o índice $IA(s)$ for nulo, o qual, por sua vez, será nulo quando $E - \max_{j,k} (CX_{jk}) . \max_{j,k} (CY_{jk}) = AI$ e o índice $IS(s)$ também for nulo, significando que não há nenhuma área em sobreposição.

4 Resultados

Arranjos com dois, quatro, seis e oito peças foram avaliados. Para cada arranjo, as peças tinham quatro, oito, doze e dezesseis elementos de área (EA). O número EA não modifica o número de avaliações necessárias para o AG alcançar o ótimo (conhecido). Em cada situação, dez diferentes rodadas foram executadas com diferentes sementes randômicas, resultando, então em 1600 simulações computacionais. Em todos os experimentos, o número de gerações foi entre 200 e 1000, enquanto que o número de indivíduos (possíveis soluções) da população foi de 100 a 400. Utilizando tais parâmetros, 150 casos obtiveram sucesso e o número médio de avaliações é mostrado na tabela 2. Para até 8 peças, os valores mostrados na tabela 2 e figura 5 foram obtidos das simulações computacionais. Para os demais valores foi feita uma extrapolação com um ajuste exponencial.

Tabela 2 – Número de avaliações em função do número de peças.

nº de peças	Avaliação / Extrapolação	nº de avaliações
2	A	8980
4	A	31640
6	A	72322
8	A	170689
10	E	366399
12	E	699126
14	E	1208527
16	E	1934268
18	E	2916011
20	E	4193419

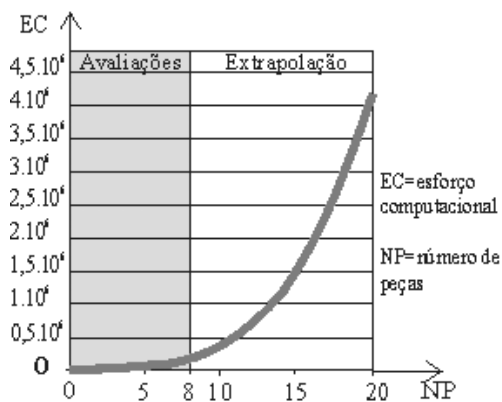


Figura 5. Esforço computacional (número de avaliações em função do número de peças.

Um exemplo de arranjo ótimo com seis peças em L é mostrado na figura 6.

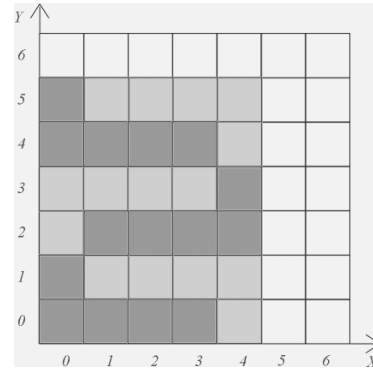


Figura 6. Exemplo de um arranjo ótimo com seis peças em L.

Um experimento adicional foi feito para o modelo reduzido com duas peças: busca exaustiva do arranjo ótimo no espaço de busca $P(2) = 160000$. Nesta simulação, foram utilizados os seguintes parâmetros: $EA = 8$, $x = [0..10]$, $y = [0..10]$, $\phi_1 = \phi_2 = [0..4]$. Por motivos de simplificação da implementação foram realizados laços do tipo “for” aninhados, onde cada possibilidade foi testada e um valor de fitness para cada situação obtido. A ordem dos laços aninhados foi a seguinte: x_1 (menos aninhado), x_2 , y_1 , y_2 , ϕ_1 , ϕ_2 (mais aninhado). Com este procedimento obteve-se um gráfico relacionando “número do teste x fitness”. Pode-se observar na figura 7 a característica multimodal do problema, tendo vários pontos de sub-ótimos (somente parte dos 160000 pontos foram plotados para melhor visualização).

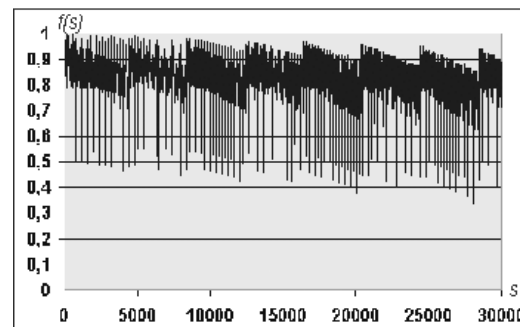
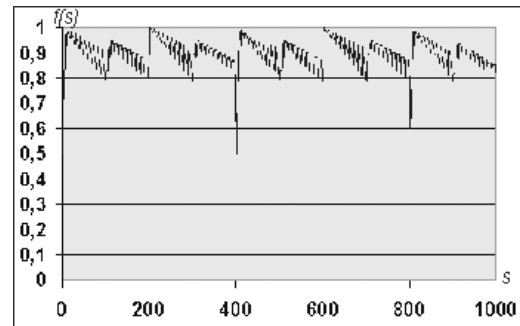


Figura 7. Multimodalidade do problema sendo mostrados inúmeros sub-ótimos. A figura superior é um trecho em escala expandida da figura inferior.

5 Discussão e Conclusões

A complexidade do problema aumenta exponencialmente à medida que o número de peças também aumenta. Esta complexidade e as características multimodais aumentam a dificuldade inerente a casos do mundo real. Isto ocorre pois pontos de sub-ótimos se comportam como atratores, induzindo uma busca mais localizada do AG, o que causa convergência prematura. O comportamento mostrado na figura 7 para um modelo reduzido provavelmente pode ser extrapolado para problemas de maior dimensionalidade. Qualquer método de otimização, matemático ou heurístico, dificilmente poderá lidar eficientemente com uma multimodalidade tão acentuada sem ser induzido em direção a ótimos locais.

Um outro fator importante é relacionado à dificuldade de computar os parâmetros de qualidade para uma determinada solução. Isto é verdadeiro uma vez que, para problemas do mundo real, o número possível de embalagens pode alcançar centenas.

O resultados obtidos com este estudo fornecem informações importantes para um estudo seguinte, já em andamento, que busca substituir as peças em tamanho reduzido por peças em tamanho real.

Como conclusão final, com o objetivo de melhorar a eficiência do algoritmo para problemas com dimensionalidade do mundo real, parece ser necessário combinar o AG com uma outra técnica matemática ou heurística.

6 Referências Bibliográficas

- Adamowicz, M. & Albano, A. (1976) A solution of the rectangular cutting-stock problem. *IEEE Transactions on Systems, Man and Cybernetics*, **6**(4): 202-210.
- Albano, A. & Sapuppo, G. (1980) Optimal allocation of two-dimensional irregular shapes using heuristic search methods. *IEEE Transactions on Systems, Man and Cybernetics*, **10**(5): 242-248.
- András, P. et al (1996) A genetic solution for the cutting stock problem. *Proceedings of the first on-line workshop on soft computing*, Nagoya University, pp. 87-92.
- Bojarczuk, C.C., Lopes, H.S. & Freitas, A.A. (2000) Genetic programming for knowledge discovery in chest-pain diagnosis. *IEEE Engineering in Medicine and Biology Magazine*, **19**(4): 38-44.
- Chu, P.C. & Beasley, J.E. (1997) A genetic algorithm for the generalized assignment problem. *Computers in Operations Research*, **24**(1): 17-23.
- Christofides, N. & Whitlock, C. (1977) An algorithm for two-dimensional cutting problems. *Operations Research*, **25**(1): 30-44.
- Fujita, K. & Gakkai, S. (1993) Approach for optimal nesting algorithm using genetic algorithm and local minimization algorithm. *Transactions of the Japanese Society of Mechanical Engineers, Part C*, **59**(564): 2576-2583.
- Gilmore, P.C. & Gomory, R.E. (1965) Multistage cutting stock problems of two and more dimensions. *Operations Research*, **13**: 94-120.
- Gilmore, P.C. & Gomory, R.E. (1966) The theory and computation of knapsack functions. *Operations Research*, **14**(61): 1045-1074.
- Goldberg, D.E., (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading: Addison-Wesley.
- Goodman, E.D. (1996) *An Introduction to GALLOPS - The Genetic Algorithms Optimized for Portability and Parallelism System*. East Lansing, USA: Genetic Algorithms Research and Applications Group (GARAGE), Department of Computer Science, Michigan State University.
- Haims, M.J., & Freeman, H. (1970) Amultistage solution of the template layout problem. *IEEE Transactions on Systems Science and Cybernetics*, **6**(2): 145-151.
- G-C Han & S-J Na (1996) Two-stage approach for nesting in two-dimensional cutting problems using neural network and simulated annealing. *Journal of Engineering Manufacture*, **210**: 509-519.
- Hopper, E. & Turton, B. (1999) A genetic algorithm for a 2D industrial packing problem. *Computers & Industrial Engineering*, **37**: 375-378.
- Ismail, H. S. & Hon, K. K. B. (1992) New approach for the nesting of two-dimensional shapes for press tool design. *International Journal of Production Research*, **30**(4): 825-837.
- Lopes, H.S. (1999) Algoritmos genéticos em projetos de engenharia: aplicações e perspectivas futuras. In: *Anais do SBAI'99 - IV. Simpósio Brasileiro de Automação Inteligente*. São Paulo, 08-10/setembro/1999, pp. 64-74.
- Nee, A. Y. C. (1984) A heuristic algorithm for optimum layout of metal stamping blanks. *Annals of CIRP*, **33**(1): 317-320.
- Petridis, V.; Kazarlis, S.; Bakirtzis, A. (1998) Varying fitness functions in genetic algorithm constrained optimization: the cutting stock and unit commitment problems. *Cybernetics*, **28**(5): 629-640.