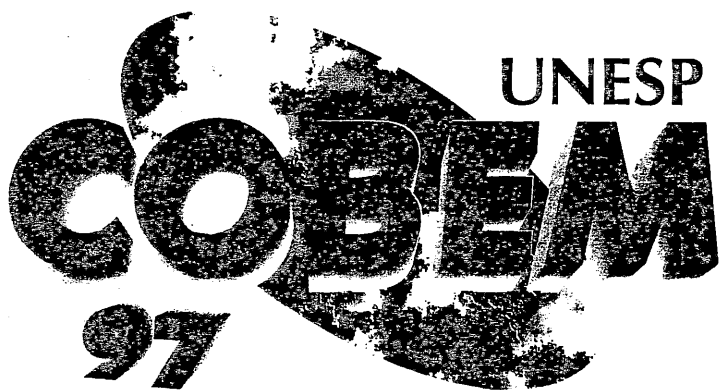


SEM
21/03
/68

XIV CONGRESSO BRASILEIRO DE ENGENHARIA MECÂNICA



14th BRAZILIAN CONGRESS OF MECHANICAL ENGINEERING

08 a 12 de Dezembro de 1997

December 8-12th 1997

Centro de Convenções Obeid Plaza Hotel
Bauru - SP / Brasil

ORGANIZAÇÃO

unesp 

PROMOÇÃO


Associação Brasileira

ALGORITMOS GENÉTICOS E SUAS POSSIBILIDADES DE APLICAÇÕES EM ENGENHARIA MECÂNICA/GENETIC ALGORITHMS AND ITS APPLICATIONS IN MECHANICAL ENGINEERING

86350
Eduardo Vila Gonçalves Filho & Creusa Sayuri Tahara

Universidade de São Paulo - Escola de Engenharia de São Carlos - Departamento de Engenharia Mecânica
Av. Dr. Carlos Botelho, 1465 - 13560-250 - São Carlos - SP, Brasil E-mail: evila@sc.usp.br - tahara@sc.usp.br

Abstract

Genetic algorithm(GA) is a potential tool for the resolution of combinatorial problems with the advantages of simplicity, flexibility and robustness in the solution. There are some drawbacks in this kind of algorithms, mainly because a GA algorithm is usually tailor-made for most of the application problems. This paper describes some possibilities of application for genetic algorithms in the field of engineering activity as production planning, assembly line balancing, layout design, cell formation, production scheduling and shows a specific application for determination of the maximum profit velocity in a lathe operation

Keywords

Genetic algorithm, engineering, otimization, combinatorials problems/ Algoritmo genético, engenharia, otimização, problemas combinatoriais.

1. INTRODUÇÃO

Os problemas de otimização são resolvidos por meio de muitas técnicas, sendo algumas de simples utilização com resultados aproximados e outras mais sofisticadas envolvendo maior tempo computacional mas fornecendo ótimos resultados. Contudo, muitos problemas, principalmente aqueles com características combinatorias, encontram dificuldade de solução devido a complexidade computacional. Estes problemas são bem resolvidos com a utilização do algoritmo genético(AG).

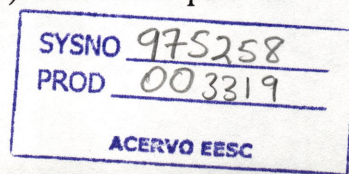
Neste trabalho são tratadas aplicações do algoritmo genético na engenharia. Na seção 2 é apresentado um breve resumo da teoria de algoritmos genéticos; na seção 3 algumas aplicações são descritas, na seção 4 detalha-se uma aplicação do AG em um problema de usinagem de metais e na seção 5 é apresentada a conclusão do trabalho.

2. ALGORITMO GENÉTICO

Algoritmo genético é um método computacional de busca baseado em mecanismos de evolução natural e genética. Estes mecanismos combinam a sobrevivência dos mais ajustados com uma estrutura de troca aleatória de informações.

O princípio básico de funcionamento dos algoritmos genéticos é a utilização de operadores que realizam funções de reprodução, cruzamento e mutação, semelhante ao processo biológico. O algoritmo genético funciona iterativamente com uma população de possíveis soluções e cada iteração corresponde a uma nova geração de soluções, sendo que as mais aptas (que fornecem um maior valor para a função de ajuste) continuam para as

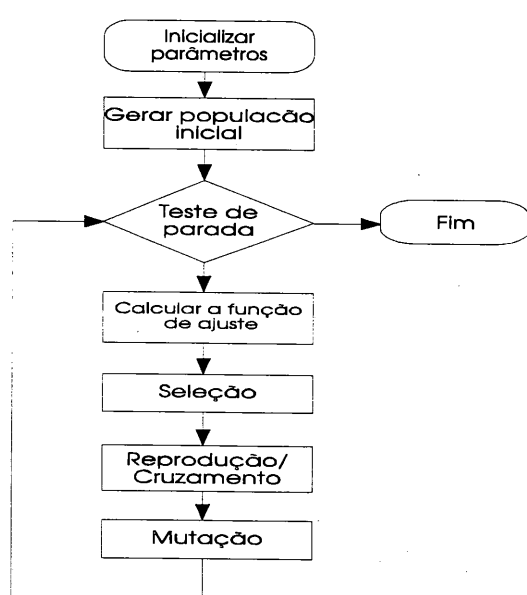
975258
270798



próximas gerações. Assim, o primeiro passo para a aplicação dos AGs é a representação das possíveis soluções no espaço de busca como uma sequência de símbolos, uma *string* de *bits*, números, etc. A solução estruturada é chamada de cromossomo artificial sendo que cada elemento dessa sequência é chamado gene e o seu possível valor é chamado alelo.

Inicia-se a otimização com um conjunto de soluções obtidas aleatoriamente ou por algum processo heurístico de forma que a população gerada seja representativa do espaço de busca. Novos candidatos a solução são, então, obtidos a partir da população atual por meio da aplicação de operadores genéticos artificiais. Com base em uma política de seleção natural, a população de soluções retorna ao seu tamanho original, sobrevivendo apenas as melhores soluções.

Os operadores genéticos comumente empregados são: cruzamento, mutação e reprodução. O operador cruzamento é utilizado para obter-se soluções a partir de partes trocadas de duas soluções selecionadas na população atual. O operador mutação é empregado para promover alterações arbitrárias em um ou mais elementos da *string* de uma solução. Este processo é controlado por uma probabilidade de mutação que em geral é pequena. A mutação garante uma chance de maior exploração no espaço de soluções. A política de seleção, que avalia a função de ajuste, mantém o número de soluções da população constante, pela sobrevivência das soluções que mais se aproximam da meta, as soluções mais adaptadas num paralelo com a teoria da evolução. Assim, com uma boa estruturação do problema e uma combinação conveniente dos operadores as soluções convergem em direção à solução ótima do problema. Um AG pode ser descrito em poucos passos, como mostra a figura 1 e o seguinte procedimento básico:



1. Formar a população inicial constituída de N soluções;
2. Teste de parada: número máximo de gerações
3. Calcular a função ajuste de todas as soluções na população atual;
4. Produzir uma nova solução
 - *selecionar pares na população atual;*
 - *produzir descendentes por cruzamentos*
 - *produzir mutação nos descendentes aleatoriamente*
 - *remover da população atual os menos ajustado*
 - *adicionar os novos descendentes.*
5. Escolher a melhor solução ajustada.
6. Retornar ao passo 2

Figura 1: Ciclo básico do algoritmo genético

O tema central das pesquisas em algoritmos genéticos é a robustez, um balanceamento entre eficiência e eficácia necessários para um bom desempenho em vários ambientes diferentes. Um ponto importante dos algoritmos genéticos é a sua flexibilidade e simplicidade se comparados aos algoritmos existentes. Em geral, problemas de natureza combinatorial são os mais apropriados para o emprego de AGs.

3. APLICAÇÕES NA ENGENHARIA

Os algoritmos genéticos estão sendo utilizados para resolver muitos problemas na área de engenharia. Vamos listar em seguida algumas aplicações como no planejamento e controle da produção, no projeto de produto, na manufatura celular, no seqüenciamento da produção.

3.1 Projeto de produto

O projeto de produto é cada vez mais reconhecido como uma atividade crítica que tem um impacto significativo no desempenho da empresa. Conseqüentemente, quando uma empresa considera um projeto de um novo produto é importante que ela utilize técnicas que forneçam soluções ótimas. Uma técnica recentemente utilizada neste processo é o AG.

O problema deve ser definido de forma a enquadrar-se nos moldes de um AG. Uma solução *string* candidata para o projeto do produto é definida pela especificação de um conjunto que contém informações específicas sobre as características do produto. A característica chave da proposta é que ela é flexível suficiente para acomodar critérios de ajuste para avaliar a solução candidata. A técnica explora o fato de que um bom modelo de produto contém algumas características (substrings) que são desejáveis e assim contribuem para sua alta avaliação.

Balakrishnan & Jacob (1995) apresentam resultados de 192 conjuntos de dados testados indicando que o AG proporciona soluções que são 99,13% e 99,92% do modelo do produto, ótimo quando os objetivos do problema são maximizar a escolha da forma e o bem estar do comprador. Estes resultados são estatisticamente significantes e 3% e 1% acima da solução identificada pelo modelo de programação dinâmica.

3.2 Layout de Fábrica

O *layout* de fábrica é um importante aspecto do projeto de um sistema de manufatura. O problema de encontrar um *layout* ótimo é difícil e as técnicas determinísticas não são computacionalmente viáveis Suresh *et al.* (1995).

No projeto de fábrica, o cromossomo é definido da seguinte forma: as posições na *string* significam a posição das instalações no *layout* e os valores em cada posição representam as instalações.

Particularmente na manufatura celular, Joines *et al.* (1995) sugerem um modelo de programação inteira empregando algoritmo genético para o projeto de manufatura celular. Os modelos de programação matemática possuem limitações críticas como: devido a não linearidade, muitas vezes não são agrupadas máquinas em células e peças em famílias; o número de células deve ser especificado; os modelos muitas vezes são intratáveis para problemas realísticos ou de porte maior. Para responder estas limitações, um AG foi desenvolvido. O cromossomo de representação descreve cada indivíduo da população. Os indivíduos são representados por uma coleção de atributos ou variáveis. Especificamente nesse problema, as primeiras m variáveis representam máquinas e as últimas n variáveis estão associadas às peças. A população inicial é obtida por um processo aleatório. Uma vantagem prática é a formação de famílias de peças e máquinas simultaneamente. O algoritmo foi testado em 17 grupos de dados e mostraram resultados eficientes.

3.3 Planejamento da produção

Freitas & Kirner (1989) fornecem um exemplo de aplicação do algoritmo genético para resolver o problema no planejamento da produção, otimizando a quantidade produzida de cada produto, maximizando o lucro. Num problema real, o número de variáveis em geral é muito grande, mas no exemplo correspondem a oito (produtos). O primeiro passo para a solução do problema é escolher a população inicial. Cada indivíduo desta população representa uma solução possível para o problema e sendo representado por uma *string* de oito números inteiros, indicando que quantidade deve ser produzida para cada tipo de produto.

Nota-se, nesses casos, que a existência de restrições dificulta a utilização de operadores genéticos e uma forma de contornar estas restrições é compor o problema com novos operadores ou critérios na função de avaliação que forcem a eliminação dos indivíduos da população que violam as restrições.

3.4 Seqüenciamento da Produção

As aplicações de algoritmos genéticos no seqüenciamento de um *flowshop* envolvem problemas combinatoriais de grande escala. A fim de aplicar um AG no seqüenciamento da produção existe o problema da representação do cromossomo, que em geral é feita por uma *string* binária (de 0 e 1).

No seqüenciamento o uso da permutação para representar a solução é mais natural, mas pode gerar cromossomos não válidos no processo de cruzamento. Para contornar este problemas foram sugeridos algumas alterações no processo de cruzamento e mutação. A seleção dos pares para reprodução foi feita por uma distribuição de probabilidade. Outras modificações foram feitas no operador mutação de modo a reduzir a taxa de convergência. A população inicial é obtida partindo-se de uma heurística e os resultados foram obtidos mais rapidamente sem perda da qualidade, Reeves (1995).

Os algoritmos genéticos são usados também para o seqüenciamento de *job shops*.

3.5 Balanceamento da Produção

Uma linha de montagem é um meio de produção em massa amplamente adotado e o balanceamento da linha de montagem é um problema que tem sido muito estudado, como por exemplo utilizando AGs, exemplificado em Suresh *et al.* (1996). Nesse caso, as estações de trabalho são indexadas de 1 a N. A solução é uma lista de N conjuntos $S_1, \dots, S_N$ cada um contendo uma ou mais tarefas. Os elementos de S_i , com $1 \leq i \leq N$, são tarefas alocadas a *i*-ésima estação. A população inicial é gerada pela associação aleatória de tarefas às estações de trabalho, sujeitas às restrições de precedência e tempo de ciclo. Assim, a população inicial é construída somente com soluções viáveis.

Suresh *et al.* (1996) apresentam duas versões de um AG para resolver o problema de balanceamento da linha de montagem. Em um dos casos, o algoritmo genético trabalha somente com soluções viáveis, no outro, é permitido uma porcentagem de soluções não viáveis. A presença de soluções não viáveis permite uma suavização no espaço de busca e ajuda a escapar de certos mínimos locais. Os resultados demonstram que o AG fornece uma convergência para boas soluções depois de poucas gerações. Assim, o AG pode ser utilizado para obter uma rápida solução aproximada que seria melhorada por outras técnicas.

3.6 Aplicações Gerais

O processo de seleção de ferramentas de forma ótima pode ser realizado com o auxílio de AGs, Mizugaki *et al.* (1994). O conjunto de ferramentas são codificados com a posição de cada *bit* correspondendo a uma ferramenta. O dígito 1 indica o uso da ferramenta e 0 a não utilização. O critério de seleção proposto considera o número de ferramentas, o total de tempo de máquina, inclusive com a influencia de trocas de ferramentas e a área impossível de corte. A utilidade deste método pode ser confirmado por simulações.

Em Pearson *et al.* (1995) e Goldberg (1989) podem ser encontradas aplicações de algoritmos genéticos em várias áreas como: Classificação; Otimização; Treinamento e Aprendizado; Biologia e Biotecnologia; Controles; Reconhecimento de Imagens; Teoria de Decisão, Medicina, Ciências Sociais, etc.

4. VELOCIDADE DE MÁXIMO LUCRO EM OPERAÇÕES DE TORNEAMENTO

Toda operação de usinagem deve ser realizada escolhendo-se uma velocidade de corte compreendida entre as velocidades de mínimo custo V_{ce} (Velocidade econômica de corte) e máxima produção, V_{cmax} . Estas duas velocidades dão origem ao chamado intervalo de máxima eficiência. As velocidades econômica de corte e de máxima produção são calculadas de forma simples pois as fórmulas para o cálculo são dadas por duas equações explícitas, isto é, existe uma equação que dá V_{ce} e V_{cmax} em função dos diversos parâmetros que influenciam a operação tais como: parâmetros de Taylor, tempo de troca de ferramenta, salário do operador, etc.. Pode-se provar que dentro do intervalo de máxima eficiência existe uma terceira velocidade de corte, denominada velocidade de máximo lucro V_p , em que o lucro líquido (valor de venda - custo de produção) por peça é máximo. Trabalhando-se nessa velocidade, o lucro por minuto p (\$/min) pode ser calculado pela equação (Hitomi, 1979):

$$p = \frac{r_n - \frac{K_m \times \lambda}{v} - \frac{K_t \times \lambda}{C^{1/n}} v^{\frac{1}{n}-1}}{t_p + \frac{\lambda}{v} + \frac{t_c \times \lambda}{C^{1/n}} v^{\frac{1}{n}-1}} - k_l \quad (1)$$

em que

p = lucro por minuto de usinagem (\$/min)

n, C = parâmetros de Taylor;

t_c = tempo de troca da ferramenta (min/aresta)

K_m = custo máquina (\$/min)

v = velocidade de corte (m/min)

$r_n = r_u - m_c$ = valor de venda - custo da matéria prima (\$/pc)

t_p = tempo improdutivo (min/peça)

K_t = custo da ferramenta por aresta de corte (\$/aresta)

$K_l = K_d + K_i$ = Custo de mão de obra (K_d) e custo administrativo (*overhead*) (K_i) (\$/min)

$\lambda = \frac{\pi \times D \times c}{1000 \times s}$, sendo D o diâmetro da peça (mm), c o comprimento de corte (mm) e s o avanço por volta (mm/volta)

A velocidade de máximo lucro V_p é obtida derivando-se a expressão (1) em relação à velocidade v e igualando o resultado a zero. A expressão resultante pode ser vista em (2).

$$(1-n)(K_t \times t_p + r_n \times t_c)V_p^{\frac{1}{n}} + \lambda(K_i - K_m \times t_c)V_p^{\frac{1}{n}-1} - n \times C^n(r_n + K_m \times t_p) = 0 \quad (2)$$

Como pode ser observado, não é possível isolarmos V_p , a não ser para valores particulares da constante de Taylor n , o que significa que necessitamos fazer diversos cálculos de forma iterativa afim de obter o ponto de máximo. Assim, é natural que se desenvolvam algoritmos computacionais para obtenção rápida deste ponto de máximo. Embora este não seja um problema que possa ser considerado difícil para um algoritmo genético, o objetivo aqui é usá-lo para mostrar a potencialidade dessa nova ferramenta de otimização.

4.7.2 O Algoritmo Genético

4.7.2.1 Função Objetivo

A expressão (1) é usada como função objetivo. Cada candidata a solução será avaliada pelo valor de p . A solução ótima deverá apresentar o valor máximo para p .

4.7.2.2 Codificação

Nesta aplicação vamos assumir que a velocidade de corte nunca será maior que 512 m/min. Assim podemos usar uma codificação binária para a velocidade de corte v , composta por uma string de nove dígitos binários(0 e 1). Por exemplo, a string (010011011) representa a velocidade de 434 m/min.

4.7.2.3 Operadores Genéticos

Os operadores genéticos utilizados foram os operadores clássicos de cruzamento e mutação. Por exemplo, assumindo que os elementos da população (100110010) e (010111011) tivessem sido selecionados para sofrerem cruzamento e que a sexta posição na *string* definisse o ponto de cruzamento então os dois descendentes produzidos do cruzamento seriam (100111011) e (010110010). O operador mutação atua sobre cada descendente podendo ou não produzir uma mutação. Uma mutação significa simplesmente escolher uma determinada posição na *string* e mudar o valor do dígito. Supondo que a sétima posição na string (010110010) tivesse sido sorteada e, que a mutação deveria ocorrer, então a mutação produzida seria a nova *string* (010110110). Nesta aplicação, a probabilidade de ocorrer o cruzamento foi fixada em 0,95 e a probabilidade de ocorrer mutação em 0,005.

4.7.2.4 Parâmetros do Algoritmo

Os seguintes parâmetros foram utilizados:

Tamanho da população: 100

Número de gerações: 40

Probabilidade de cruzamento: 0,95

Probabilidade de mutação: 0,005

4.7.2.5 Resultados

O algoritmo foi aplicado a um exemplo proposto por Hitomi (1979). Os dados do exemplo podem ser vistos na tabela 1.

Tabela 1 - Dados numéricos para cálculo de V_p

PARÂMETROS	DESCRIÇÃO	VALOR
Taylor	Constante de inclinação n	0,23
	Constante C	430
Usinagem	Profundidade de corte d (mm)	1,00
	Avanço (mm/rev.)	0,20
Peça	Diâmetro D (mm)	50,00
	Comprimento de usinagem L (mm)	200,00
Tempo	Tempo improdutivo t_p (min/pç)	0,75
	Tempo de troca ferramenta t_c (min/aresta)	1,50
Custo	mão de obra direta K_d (\$/min)	0,15
	Custo administrativo k_i (\$/min)	0,35
	Custo máquina K_m (\$/min)	0,05
	Custo ferramenta K_t (\$/aresta)	2,50
	custo matéria prima m_c (\$/pç)	20,00
	valor de venda r_u (\$/pç)	32,00

O resultado dado pelo algoritmo é mostrado na tabela 2. Para facilitar a compreensão do que ocorre com um algoritmo genético, foram colocados os dez indivíduos mais aptos da última geração, isto é, da 40ª geração. Pode-se observar que o indivíduo 8 correspondente à *string* (010110001), representa a velocidade de corte de 282 m/min e permite obter um lucro máximo de 7,6568 (\$/min). A velocidade de máximo lucro dada por Hitomi é de 278 m/min. Porém, calculando-se o lucro obtido com esta velocidade constata-se que ele é de 7,6536 \$/min, abaixo do valor dado pela velocidade de 282 m/min calculada pelo AG. Portanto, o valor dado pelo algoritmo produz um melhor resultado final.

Tabela 2 - Resultados do AG

INDIVÍDUO	STRING	VELOCIDADE(M/MIN)	LUCRO(\$)
8	010110001	282	7,6568
69	010110001	282	7,6568
43	010110001	282	7,6568
67	010110001	282	7,6568
66	010110001	282	7,6568
10	010110001	282	7,6568
47	010110001	282	7,6568
19	010110001	282	7,6568
65	000110001	280	7,6555
41	000110001	280	7,6555

5. CONCLUSÃO

Os algoritmos genéticos podem ser considerados como uma poderosa ferramenta para a solução de problemas combinatoriais de grande porte, podendo ser aplicados nas mais diversas áreas do conhecimento. Sua utilização requer uma função de ajuste (função objetivo) que possa avaliar a qualidade de uma solução proposta e o uso de operadores genéticos adequados, muitas vezes desenvolvidos especificamente para a aplicação em questão. Estes requisitos podem tornar um algoritmo genético dependente do domínio de aplicação. Os AGs, quando aplicados apresentam-se de forma simples e flexível e possuem robustez diante de ótimos locais, podendo ser utilizados mesmo com poucos recursos computacionais, permitindo maiores avanços, inclusive em soluções combinadas com outras técnicas.

6. REFERÊNCIAS BIBLIOGRÁFICAS

BALAKRISHNAN, P.V. & JACOB, V.S. Genetic algorithms for product design. **Management science**, v. 42, n. 8, p. 1105-1117, 1996.

CHU, P.C. & BEASLEY, J. E. A genetic algorithm for the generalised assignment problem. **Computers Operation Research**, v. 24, n. 1, p. 17-23, 1997.

FREITAS, A.A & KIRNER, C. Introdução à algoritmos genéticos. WORKSHOP SOBRE REDES NEURAIIS. In: Anais do Workshop sobre redes neurais. São Carlos: UFSCar, p. 71-89, set/ 199.

GOLDBERG, D.E. *Genetic Algorithms in search, optimization, and machine learning*. Addison-Wesley, 1989.

HITOMI, K., Manufacturing systems engineering. Taylor & Francis Ltd. London, 1979.

JOINES, F.A, CULBRETH, T.C., KING, R.E. Manufacturing cell design: an integer programming model emplying genetic algorithms. **IIE Transactions**, n. 28, p. 69-85, 1996.

MIZUGAKI, Y., HAO, M., SAKAMOTO, M. MAKINO, H. Optimal tool selection based on genetic algorithm in a geometric cutting simulation. In: **Annals of the CIRP**, v. 43, n. 1, p. 433-436, 1994.

PEARSON, D.W., STEELE, N.C., ALBRECHT, R. F. Artificial Neural Nets and Genetic Algorithms. **Proceedings of the International Conference in Alès**, France, 1995. Springer-Verlag Wien New York.

REEVES, C. R. A genetic algorithm for flowshop sequencing. **Computers Operation Research**, v. 22, n. 1, p. 5-13, 1995.

SURESH, G. VINOD, V.V., SAHU, S. A genetic algorithm for facility layout. **International Journal of Production Research**, v. 33, n. 12, p. 3411-3423, 1995.

SURESH, G. VINOD, V.V., SAHU, S. A geneticalgorithm for assembly line balancing. **Production Planning & control**, v. 7, n. 1, p. 38-46, 1996.