



Otimização do Projeto Estrutural via Algoritmo Genético

*Eng. Sandra Rita de Oliveira
Prof. Dr. Oscar Brito Augusto
Universidade de São Paulo
Depto. de Eng. Naval e Oceânica*

Resumo

Neste trabalho apresenta-se o Algoritmo Genético (AG), um algoritmo de programação matemática não linear, a ser utilizado em projetos otimizados de estruturas navais e oceânicas. O algoritmo foi aplicado ao projeto de uma Tampa de Escotilha e de um Vaso de Pressão. Os valores obtidos foram comparados com os resultados apresentados nos mesmos projetos, através da aplicação de alguns algoritmos consagrados da literatura. Embora os algoritmos utilizados no trabalho comparativo, em sua forma original, tenham sido propostos para operarem com variáveis contínuas, eles foram adaptados para o tratamento de problemas com variáveis discretas, uma vez que nos projetos estruturais em geral, é comum encontrar-se este tipo de variável. Os testes executados mostram o Algoritmo Genético como sendo um eficaz algoritmo exploratório, nas regiões do espaço de busca, devido a sua globalidade e flexibilidade.

Abstract

This work introduces the Genetic Algorithm (GA), a non linear programming algorithm used for the computer aided design of ocean structures. It was used in a Hatch Cover and a Pressure Vessel design. The results were compared to results obtained with some usual algorithms found in technical literature. However, these algorithms, originally, only work with continuous variables, then some improvements were incorporated in order to make possible the operation of discrete variables. The testes shown the Genetic Algorithm as a good algorithm of exploration in the search spaces due to its globality and flexibility.

1. Introdução

Está muito enraizado na engenharia oceânica, em particular, e na engenharia, em geral, os procedimentos de projeto baseados no método conhecido como *espiral de projeto*, onde os valores a serem definidos para vários parâmetros de projeto são, à princípio, admitidos, com base em experiências passadas, e melhorados,

segundo critérios específicos de performance, através de repetidos exercícios de modificações por tentativa e erro.

Com a introdução de formas estruturais inovadoras e o crescimento da complexidade dos requisitos de projeto, surge a necessidade de novos procedimentos capazes de satisfazer os requisitos de projeto em espaços mais

curtos de tempo. A tendência atual é a de substituir a prática convencional, que depende em larga escala da experiência passada, por um sistema automatizado de projeto no qual o problema em questão é formulado numericamente e, em consequência disto, tratado por técnicas puramente matemáticas, inclusive de otimização.

Estudos de otimização em projetos de estruturas de navios datam dos meados da década de 70 e um certo grau de sucesso tem sido atingido pelos diversos modelos propostos que utilizam principalmente a técnica de minimização seqüencial sem restrições (SUMT), envolvendo principalmente os parâmetros de projeto tratados como variáveis contínuas. No entanto, os parâmetros de projeto são inerentemente discretos - o número de enrijecedores em um painel reforçado do casco de um navio, por exemplo, é, inexoravelmente, uma variável discreta - indicando que uma abordagem mais adequada, para problemas de projeto, deveria ser encontrada na otimização global, permitindo a combinação de variáveis contínuas e discretas.

Devido à constante busca de melhorias das técnicas de projeto, deve ser considerada a possibilidade do uso de ferramentas mais modernas, mais robustas e mais rápidas, baseadas em outros conceitos de otimização. As considerações anteriores estimularam a presente pesquisa de otimização de estruturas de embarcações através do uso do algoritmo genético. Como o nome indica, o algoritmo genético tenta simular, em computador, os processos de evolução observados na natureza.

Esta nova abordagem difere das várias técnicas convencionais de otimização, as quais procuram melhorar sucessivamente, conjunto por conjunto, as variáveis independentes de um problema, gerando uma única solução com ótima

performance, que pode não ser necessariamente o ótimo global. O algoritmo genético objetiva melhorar, em paralelo, um número de conjuntos de variáveis independentes e, deste modo, simultaneamente melhorar os valores estimados de performance para um grupo de soluções.

2. O Algoritmo Genético

Organismos vivos têm a intrínseca capacidade de solucionar problemas. Eles exibem uma versatilidade que deixam o melhor programa de computador envergonhado, conseguindo suas habilidades através de mecanismos, aparentemente indiretos, de evolução e seleção natural.

O algoritmo genético faz com que seja possível explorar a mais ampla gama de soluções potenciais para um problema, de melhor forma que os programas convencionais. Além do mais, como pesquisadores investigam a seleção natural de programas sob controladas e bem entendidas condições, os resultados práticos que eles obtêm podem produzir algum discernimento de detalhes de como a vida e a inteligência evoluem no mundo natural.

A maioria dos organismos evolui principalmente através de dois processos primários: seleção natural e reprodução sexual. O primeiro determina quais membros da população sobrevivem para a reprodução e o segundo, assegura a mistura e a recombinação entre os genes dos seus descendentes. Ocorre a troca de material genético, quando há uma fusão entre o espermatozóide e o óvulo, combinando um cromossomo com outro. A recombinação ou mistura (*crossing over*) permite às criaturas evoluírem muito mais rapidamente do que se cada descendente tivesse uma simples cópia dos genes de um

dos pais, modificado ocasionalmente através de mutação.

As pessoas têm empregado uma combinação de cruzamento e seleção de raças por milênios, para criar melhores colheitas, apurar qualidades em cavalos, produzir flores ornamentais. Contudo, este não é um processo simples para ser traduzido para um programa de computador. O problema chave é a construção de um *código genético* que possa representar a estrutura de diferentes problemas, assim como o DNA representa a estrutura de uma pessoa ou de um rato.

Pode-se neste momento considerar-se uma importante questão:

O que na realidade são os Algoritmos Genéticos?

Os Algoritmos Genéticos são algoritmos de pesquisa baseados nos mecanismos de seleção e genética natural. Eles combinam a sobrevivência do mais apto entre, *indivíduos* ou projetos, caracterizados por estruturas de *strings*, com uma elaborada troca de informação aleatória, para formar um algoritmo de busca com as mesmas habilidades encontradas nos mecanismos da evolução natural. Em cada geração, um novo conjunto de criaturas artificiais (*strings*) é criado com base em subgrupos de melhor performance da geração anterior. Apesar da característica aleatória, algoritmos genéticos não fazem simplesmente uma busca aleatória. Eles eficientemente exploram informações históricas para especular novos pontos de busca com uma já esperada melhora na performance.

Desde a sua criação por Jonh Holland (1975), as principais metas nas pesquisas realizadas com os Algoritmos Genéticos são:

1. Abstrair e explicar rigorosamente os processos adaptativos dos sistemas naturais.

2. Projetar sistemas artificiais, *softwares*, que retenham os importantes mecanismos dos sistemas naturais

Este enfoque tem levado a importantes descobertas nos dois ramos da ciência, natural e artificial.

A questão fundamental no desenvolvimento de algoritmos genéticos tem sido *robustez*, o equilíbrio entre eficiência e eficácia necessária para a sobrevivência nos mais variados meios. As implicações da robustez para sistemas artificiais são múltiplas. Caso os sistemas artificiais possam ser feitos mais robustos, o custo de um novo projeto pode ser reduzido ou eliminado.

Um algoritmo genético simples (AG) e que produz bons resultados em muitos problemas práticos deve ser composto de três operadores básicos: reprodução, *crossover* e mutação.

Reprodução é um processo no qual *strings* individuais são copiados de acordo com os valores da função objetivo, f (biólogos chamam esta função de função performance). Copiar *strings* de acordo com seus valores de performance significa que *strings* com melhor performance têm uma maior probabilidade de contribuir com um ou mais descendentes nas próximas gerações. Este operador é uma versão artificial da seleção natural, ou, em outras palavras, ele produz um sobrevivente *Darwiniano* entre os mais aptos *strings*. Em populações naturais o mais apto é determinado pela habilidade da criatura de sobreviver aos seus predadores, pestes e outros obstáculos para atingir a fase adulta e, conseqüentemente, a reprodução. Neste caso, artificial, a função objetivo é o árbitro final que decidirá, de acordo com as suas performances, a "vida ou a morte" das criaturas formadas por *strings*.

Depois da reprodução, segue-se, em dois passos, um simples *crossover*. Primeiro, membros de um *string* recém

reproduzido são cruzados aleatoriamente. Segundo, cada par de *strings* sofre o cruzamento da seguinte forma: seleciona-se, de forma aleatória, uniformemente distribuída, entre 1 e o comprimento do *string* menos 1 $[1, l - 1]$, uma posição k . Dois novos *strings* são criados através da troca de todos os caracteres entre as posições $k + 1$ e l , inclusive. Por exemplo, considere os *strings* A_1 e A_2 seguintes:

$$A_1 = 01 | 110$$

$$A_2 = 11 | 010$$

Suponha-se $k = 2$ o resultado da escolha de um número aleatório entre 1 e 4, (conforme indicado pelo símbolo de separação |). Como resultado, o *crossover* produz dois novos *strings* onde o símbolo (') significa que os *strings* são parte de uma nova geração:

$$A'_1 = 01010$$

$$A'_2 = 11110$$

Conforme pode-se notar os mecanismos de reprodução e *crossover* são surpreendentemente simples, envolvendo a geração de números aleatórios, cópias de *strings*, e a troca parcial entre *strings*. Contudo, a ênfase combinada de reprodução e a estruturação, através da aleatoriedade, da troca de informação do *crossover*, dá aos algoritmos genéticos muito do seu poder.

Neste caso então, qual é o objetivo do operador mutação? Sem trazer surpresas, há muita confusão sobre o papel da mutação em genética (tanto natural como artificial), mas qualquer que seja a causa desta confusão, os estudiosos do assunto vêem a mutação como um papel secundário na operação do algoritmo genético. A mutação é necessária, porque mesmo com a reprodução e *crossover*,

efetivamente buscando-se e recombinando-se parte do material, ocasionalmente eles podem se tornar por demais zelosos e perder material genético potencialmente útil (1's ou 0's em uma locação particular de um *string*). Em sistemas genéticos artificiais, o operador mutação protege o algoritmo de uma perda irreversível. No AG, a mutação é, com pequena probabilidade, uma alteração aleatória ocasional do valor de um *bit* em uma dada posição do *string*. Em um código binário, isto simplesmente significa mudar de um 1 para um 0 e vice-versa. A mutação propriamente dito, é uma "caminhada" aleatória no espaço do *string*. Quando escassamente usado, em conjunto, com reprodução e *crossover*, ele é uma medida de segurança contra perdas prematuras de importantes informações.

Outros operadores genéticos e planos reprodutivos têm sido abstraídos do estudo de exemplos biológicos. Contudo, os três operadores examinados (reprodução, *crossover* e mutação) têm provado ser tanto, computacionalmente, simples, quanto efetivo na solução de um grande número de importantes problemas de otimização.

3. Aplicações

Com o objetivo maior de verificar o comportamento do algoritmo em comparação a outros algoritmos tradicionalmente utilizados em otimização, utilizou-se o algoritmo genético na solução dos dois problemas descritos a seguir.

3.1 Tampa de Escotilha

Visando o mínimo peso, deseja-se obter as dimensões h e t_b , mostrados na figura 1, para a tampa de escotilha de porão de carga de uma embarcação. A largura da escotilha é $l_0 = 6.0m$ e a tampa é formada por segmentos de viga caixão,

fabricada em alumínio - módulo de elasticidade $E = 70\text{KN/mm}^2$, tensão de escoamento, $\sigma_e = 70\text{N/mm}^2$, coeficiente de Poisson, $\mu = 0.3$ - e largura de segmento de $b = 600\text{mm}$. A pressão, a ser suportada

pela escotilha, é de $p = 10\text{KN/m}^2$. Devem ser verificadas as restrições, a seguir, mencionadas.

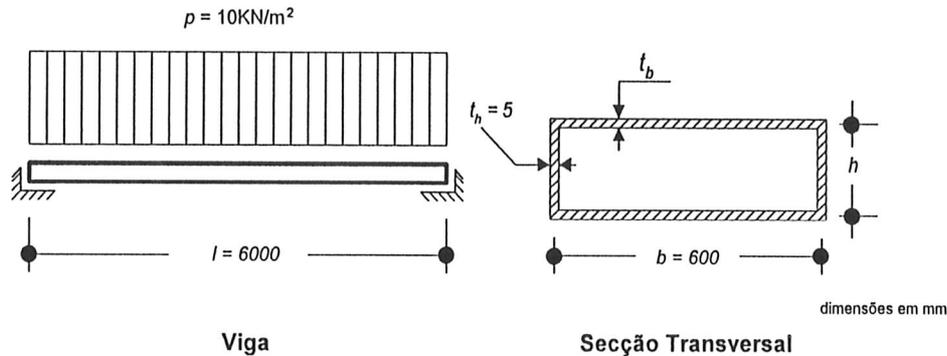


Figura 1 - Segmento de tampa de escotilha

- Máximo deslocamento, $g_1(h, t_b)$:

$$\delta = \frac{5(pb)l^4}{384EI} \leq \delta_{\text{lim}} = \frac{l}{500} = 12.0\text{mm}$$

- resistência nos pontos de máxima sollicitação

flange, fibra externa, $g_2(h, t_b)$:

$$\sigma = \frac{(pb)l^2}{12W} \leq \sigma_{\text{max}} = 70\text{MPa}$$

alma, linha neutra, $g_3(h, t_b)$:

$$\tau = \frac{\left(\frac{pbl}{2}\right)m}{t_h I} \leq \tau_{\text{max}} = \frac{70}{\sqrt{3}}\text{MPa}$$

- não ocorrência de flambagem localizada dos flanges, $g_4(h, t_b)$:

$$\frac{b}{t_b} \leq \frac{4\pi}{\sqrt{12(1-\mu^2)}} \sqrt{\frac{E}{\sigma}}$$

Dados geométricos:

- Área: $A = 2(ht_h + bt_b)$

- Inércia: $I = (t_h h + 3t_b b) \frac{h^2}{6}$

- Módulo de resistência: $w = 2 \frac{I}{h}$

- Momento estático na linha neutra: $m = \frac{1}{4}bt_b h + \frac{1}{8}h^2 t_h$

- Função de mérito: $f(h, t_b) = A$

Por simplicidade, o peso das anteparas transversais e dos reforçadores não foram considerados no problema.

3.2 Vaso de Pressão

O outro problema refere-se à otimização do projeto estrutural do vaso de pressão, mostrado na figura 2. Este problema foi resolvido por diferentes métodos, Fu, Fenton e Cleghorn, (1991), Li e Chou, (1994), Thierauf e Cai, (1997). A função objetivo combina os custos de material, de conformação e de soldagem do vaso de pressão.

O conjunto de restrições está de acordo com os respectivos códigos ASME. O problema de otimização do projeto é formulado a seguir:

- minimizar

$$f(\mathbf{X}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

- sujeito a

$$g_1(\mathbf{X}) = x_1 - 0.0193x_3 \geq 0$$

$$g_2(\mathbf{X}) = x_2 - 0.00954x_3 \geq 0$$

$$g_3(\mathbf{X}) = \pi x_3^2 x_4 + \frac{4}{3} \pi x_3^3 - 750.0 \cdot 1728.0 \geq 0$$

$$g_4(\mathbf{X}) = 240.0 - x_4 \geq 0$$

$$1.000 \leq x_1 \leq 1.375$$

$$0.625 \leq x_2 \leq 1.000$$

As variáveis de projeto x_3 e x_4 são contínuas e as variáveis x_1 e x_2 são valores discretos, múltiplos de 0.0625 in.

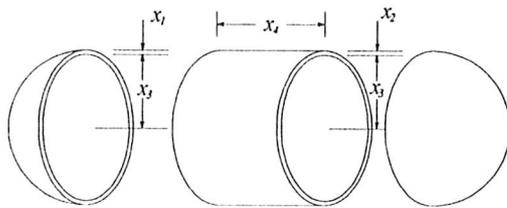


Figura 2 - Vaso de pressão

4. Resultados

Os resultados obtidos através das aplicações, em cada um dos exemplos, podem ser vistos nas tabelas 1 e 2. Comparou-se os resultados obtidos com o

Algoritmo Genético, com os resultados obtidos através da aplicação dos algoritmos de Box, Nelder-Mead e Hooke-Jeeves, o que não é uma tarefa trivial. Os algoritmos dependem de parâmetros iniciais que se adequam às suas particularidades. Os algoritmos de Nelder-Mead e Hooke-Jeeves, necessitam de um conhecimento prévio do problema por parte do usuário, que deve definir um ponto de partida, o qual tem grande influência na qualidade do resultado final. O algoritmo genético, assim como o de Box, têm a grande vantagem da não necessidade de um ponto de partida.

4.1 Tampa de Escotilha

Na figura 3 mostram-se, para o problema da Tampa de Escotilha, as funções de restrições e as curvas de nível para a superfície de resposta, utilizando-se o fator de resposta $r_k = 1000$, com as funções $f(\mathbf{X})$ e $g(\mathbf{X})$ não normalizadas. O fator de resposta adotado correlacionado ao fato de as funções, de mérito e de restrições, não estarem normalizadas não garante que o mínimo da superfície de resposta se localize no interior da região viável, resultando em modificações, durante o processo de busca, do fator de resposta e conseqüentemente da superfície de resposta.

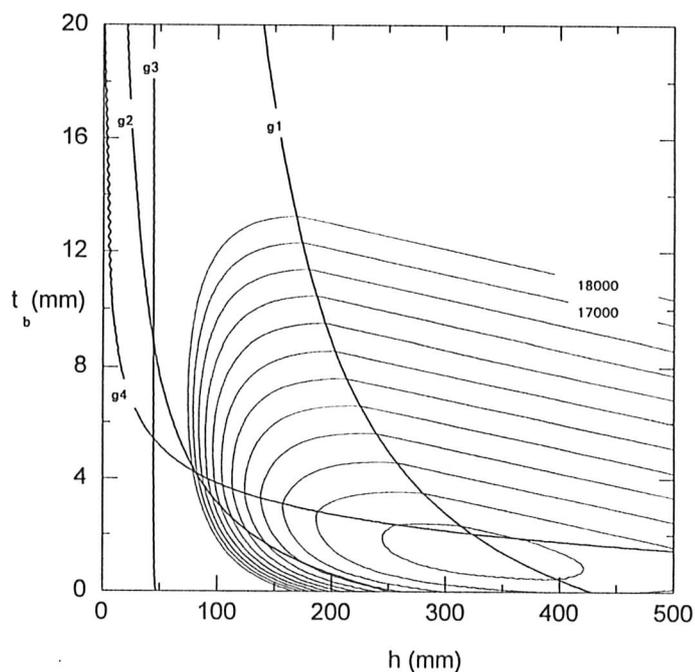


Figura 3 - Função de resposta - Tampa de Escotilha
 Fator de resposta $r_k = 1000$. Funções não normalizadas

Tabela 1 - Resultados para o projeto da Tampa de Escotilha

ALGORITMO	PONTO DE PARTIDA		RESULTADOS		FUNÇÃO DE MÉRITO
	h_0	tb_0	h	tb	
AG	-	-	314.515	3.224	7014
BOX	-	-	323.886	2.932	6757
	200	5	323.813	2.932	6757
NELDER-MEAD	300	5	323.813	2.932	6757
	300	2	323.813	2.932	6757
HOOKE-JEEVES	200	5	298.276	3.688	7408
	300	5	298.276	3.688	7408
	300	2	372.704	2.750	7027

Tolerância: 10^{-5} e passo de 20 para a variável t_b , tanto para a montagem do simplex, algoritmo de Nelder-Mead, quanto passo inicial, para o algoritmo de Hooke-Jeeves. Região de busca, $h \in [1, 500]$ e $t_b \in [0, 20]$.

00007

4.2 - Vaso de pressão

Tabela 2 - Resultados para o projeto do Vaso de Pressão

	Fu, Fenton e Cleghorn	Li, H. e Chou	Thierauf e Cai	AG
f_{min}	8048.6	7127.3	7006.9	7632.4
x_1	1.125	1.000	1.000	1.000
x_2	0.625	0.625	0.625	0.625
x_3	48.380	51.250	51.812	50.007
x_4	111.745	90.991	84.591	112.604
g_1	0.191	1.011	0.000	0.035
g_2	0.163	0.136	0.131	0.148
g_3	75.875	18759.754	15.000	112456.112
g_4	128.255	149.009	155.409	127.396

5. Conclusões & Recomendações

Pelos resultados apresentados pode-se verificar a característica fundamental do algoritmo genético: explorar regiões onde encontram-se os valores ótimos e, eventualmente, os atingir. O comportamento convergente sem a garantia da optimalidade é um problema que vem sendo encontrado pelos usuários do Algoritmo. Consegue-se constatar que o AG busca, rapidamente, interessantes áreas do espaço de variáveis, mas têm se mostrado um método fraco, quando comparado com os demais, na escalada para o ótimo. Isto não reduz sua utilidade. Muito pelo contrário, nos métodos “mais” convergentes não se garante a globalidade, e, em muitas situações, não se consegue ter a flexibilidade do GA. Adicionalmente, muitos métodos são limitados a um estreita classe de problemas, como o tratamento isolado de variáveis discretas e variáveis contínuas. Como resultado, o Algoritmo Genético pode ser usado quando técnicas convencionais apresentam dificuldades naquele específico problema.

Seria interessante utilizarem-se esquemas híbridos, iniciando-se a busca com o algoritmo genético, para a descoberta de importantes regiões do problema, e, uma vez conhecidas estas regiões, aplicam-se, localmente, os métodos convencionais de programação matemática, para definirem-se os ótimos locais. Desta forma pode-se combinar a globalidade e o paralelismo do GA com o comportamento “mais” convergente das técnicas locais.

Deve-se salientar que o que se apresentou neste trabalho faz parte de uma pesquisa que tem como objetivo implementar o algoritmo genético como uma ferramenta adicional a um programa de síntese de estruturas de embarcações.

Agradecimentos. Os autores são gratos à FAPESP, Fundação de Amparo à Pesquisa do Estado de São Paulo, pelo apoio concedido a este projeto de pesquisa.

6. Referências Bibliográficas

- Augusto, O . B., 1996 (1), "Processos de Busca não Linear para o projeto da Estrutura do Navio" BT/PNV/23, Departamento de Engenharia Naval e Oceânica - Escola Politécnica - USP.
- Augusto, O . B., 1996 (2), "Projeto Estrutural de um Navio Patrulha Costeiro" BT/PNV/26, Departamento de Engenharia Naval e Oceânica - Escola Politécnica - USP.
- Bouchard, E. E., 1988; Kidwell, G. H. ; Rogan, J. E. "The Application of Artificial Intelligence Technology to Aeronautical System Design" AIAA-88-4426, AIAA/AHS/ASEE Aircraft Design Systems and Operations Meeting, Atlanta, September.
- Box, M. J., 1965, "A New Method of Constrained Optimization an Comparison with other Methods", Computer Journal, Vol. 8, No. 1, April, pp. 42-52.
- Cella, A. ;Soosaar,K., 1973, "Discrete Variables in Structural Optimization" in Optimun Structural Design - Theory and Applications, John Willey & Sons.
- Davidor, Y., 1989, "Genetic Algorithms for Order Dependent Process Applied to Robot Path Planning", Unpublished Ph.D. dissertation, Imperial College, University of London.
- Davis, L., 1991, "Handbook of Genetic Algorithms", Van Nostrand Reinhold.
- Fu, J., Fenton, R. G., Cleghorn, W. L., 1991, "A Mixed Integer-Discrete-Continuous Programming Method and its Application to Engineering Design Optimization", Engineering Optimization, N. 17, pp. 263-280.
- Goldberg, D.E., Samtani, M.P., 1986, "Engineering Optimization via Genetic Algorithms" Proceedings of the Ninth Conference on Eletronic Computation, 471 - 482.
- Goldberg, D. E., 1989, "Genetics Algorithms in Search, Optimization, and Machine Learning", Addison-Wesley.
- Grefenstette, J. J., 1986, "Optimization of Control Parameters for Genetics Algorithms", IEEE Transactions on Systems, Man, and Cybernetics, SMC-16(1), pp. 122-128.
- Hillis, W. D., 1987, "The Connection Machine", Scientific American, 255(6).
- Holland, J. H., 1975, "Adaptation in Natural and Artificial Systems" Ann Arbor: The University of Michigan Press,
- Holland, J. K., 1992, "Genetic Algorithms", Scientific American, 267(1), 66-72.
- Hooke, R., Jeeves, T. A., 1961, "Direct Search Solution of Numerical and Statistical Problems", Journal of the Association of Computing Machinery, Vol. 8, April, pp. 212-229.
- Jenkins, W.M., 1997, "On the Application of Natural Algorithms to Structural Design Optimization" Engineering Structures, vol. 19, No. 4, 302 -308.
- Kitamura, K., 1972, "Studies on Optimization of Ships Structures - Optimum Design of Longitudinal Members of Tanker" J.S.N.A., No. 132.
- Li, H. L., Chou, C. T., 1994, "A Global Approach for Nonlinear Mixed Discrete Programming in Design

- Optimization”, Engineering Optimization, N. 22, pp. 109-122.
- Mayer-Kress, G., 1989, “A Nonlinear Dynamical System Approach to International Security”, Technical Report LA-UR-89-1355, Los Alamos National Laboratory.
- Nelder, J. A., Mead, R., 1965, “A Simplex Method for Function Minimization”, Computer Journal, Vol. 7, No. 4, Jan., pp. 308-313.
- Okada, T. ;Neki, I., 1992, “Utilization of Genetic Algorithm for Optimization the Design of Ship Hull Structures” J.S.N.A., vol. 171, June.
- Thierauf, G., Jianbo, C., 1997, “Parallel Evolution Strategy for Solving Structural Optimization”, Engineering Structures, Vol. 19, N.4, pp. 318-324.