4
Implementação Computacional

4.1.
Introdução

Neste capítulo é apresentada a formulação matemática do problema de otimização da disposição das linhas de ancoragem para minimizar os deslocamentos (*offsets)* das unidades flutuantes quando submetidas às condições ambientais. São apresentados os principais aspectos do algoritmo s*teady-state,* implementado neste trabalho, assim como a sua aplicação a diversos problemas de otimização de treliças e outras funções contínuas encontradas na literatura técnica. As configurações usadas em cada problema são detalhadas e comparações com as soluções conhecidas são discutidas.

4.2.
Formulação Matemática

O problema proposto neste trabalho consiste na minimização dos deslocamentos sofridos pelas unidades flutuantes quando submetidas a diferentes combinações de carregamentos provenientes das condições ambientais.

Uma das principais vantagens dos AG, quando comparados com os métodos clássicos, é permitir a manipulação de variáveis de projeto de forma contínua, discreta ou através de uma combinação de ambas. Neste trabalho as variáveis envolvidas no problema foram tratadas de forma contínua e, portanto, a distribuição "ótima" das linhas de ancoragem pode ser expressa como um problema contínuo de otimização sem restrições da seguinte maneira:

$$\text{Min } \sum_{i=1}^{m} \Delta_i^2(\alpha) = \sum_{i=1}^{m} [\Delta x_i^2(\alpha) + \Delta y_i^2(\alpha)] \tag{4.1}$$

sujeito a:

$$\alpha_{j\,\min} \leq \alpha_j \leq \alpha_{j\,\max}, \quad j = 1...n \tag{4.2}$$

onde Δ é o deslocamento resultante da unidade flutuante, que pode ser decomposto nas suas componentes Δx e Δy, para um dado conjunto de condições ambientais; $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ é um vetor que contém todas as variáveis de projeto, i. e., os azimutes das linhas de ancoragem; n é o número de variáveis independentes de projeto; m é o número de combinações das condições ambientais; e, finalmente, as desigualdades mostradas na Equação (4.2) são as restrições laterais.

A Figura 4.1 ilustra uma unidade flutuante ancorada por 8 linhas e suas respectivas disposições (α_i) a serem consideradas no problema de otimização.

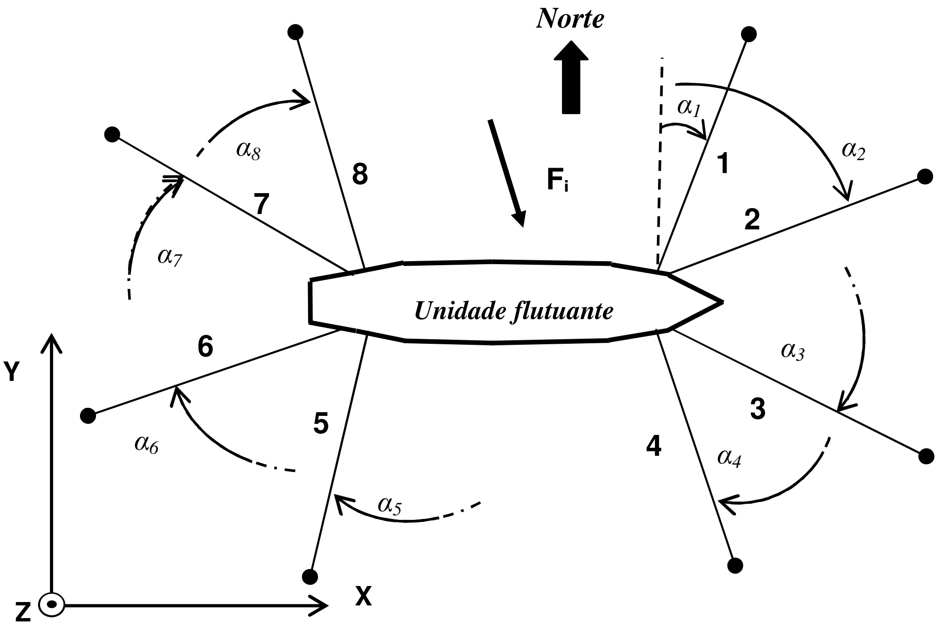

Figura 4.1 – Representação de um sistema de ancoragem com 8 linhas.

A minimização dos deslocamentos (também conhecidos como *offsets* estáticos) implica em se determinar uma disposição "ótima" das linhas de ancoragem (Figura 4.2).

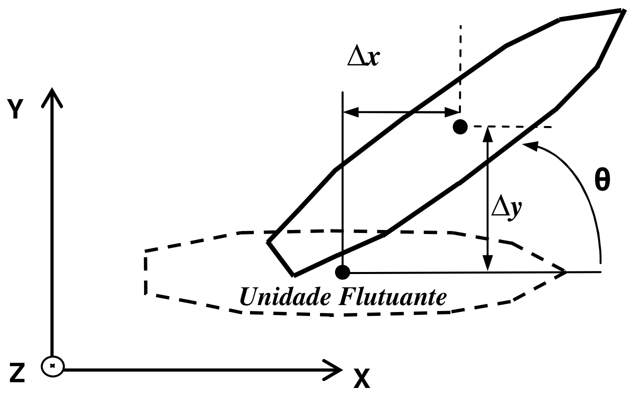

Figura 4.2 – Deslocamentos sofridos por uma unidade flutuante sob a ação de cargas externas.

4.3. Codificação das Variáveis de Projeto

Existem várias formas de se representar as variáveis de projeto, tais como vetores binários, vetores de números reais e listas de permutação, dentre outras. Neste trabalho as variáveis de projeto foram codificadas utilizando-se vetores de números binários de comprimento fixo, construídos com o alfabeto binário {0,1} e concatenados de ponta a ponta para formar um único vetor de comprimento maior. Essa estrutura concatenada representa o cromossomo. Assim, cada cromossomo contém todas as variáveis de projeto.

O comprimento total de cada cromossomo é calculado aplicando-se a Equação (3.6).

4.4. Cálculo dos Deslocamentos

Um conjunto de condições ambientais compreende as ações de ventos, ondas e correntes marinhas. Em geral, para análises estáticas, como é o caso do presente trabalho, as condições ambientais são aproximadas por valores médios, constantes ao longo do tempo, que atuam sobre a unidade flutuante. Esta força externa equivalente (Fi) caracteriza o tipo de análise conhecida como "quase-estática". No cálculo dos deslocamentos das unidades flutuantes, as linhas de ancoragem são tratadas como molas não-lineares (Figura 4.3) que impõem forças de restauração na unidade. Tais forças, expressas como uma função da distância horizontal entre a âncora e o ponto de conexão no extremo superior da linha, são descritas por meio de curvas de restauração, as quais são geradas utilizando-se a equação da catenária.

Obtidas as forças desequilibradas, uma nova posição de equilíbrio estático da unidade flutuante é calculada resolvendo-se o seguinte sistema de equações não-lineares:

$$\boldsymbol{K}\,\boldsymbol{d} = \boldsymbol{F}_i - \boldsymbol{R}_{int} \tag{4.3}$$

onde $\boldsymbol{K}$ é a matriz de rigidez; $\boldsymbol{d}$ é o vetor de deslocamentos que se deseja obter; $\boldsymbol{F}_i$ corresponde às forças externas devido a cada conjunto de condições ambientais; e $\boldsymbol{R}_{int}$ é a resultante das forças internas (de restauração) considerando-se todas as linhas de ancoragem.

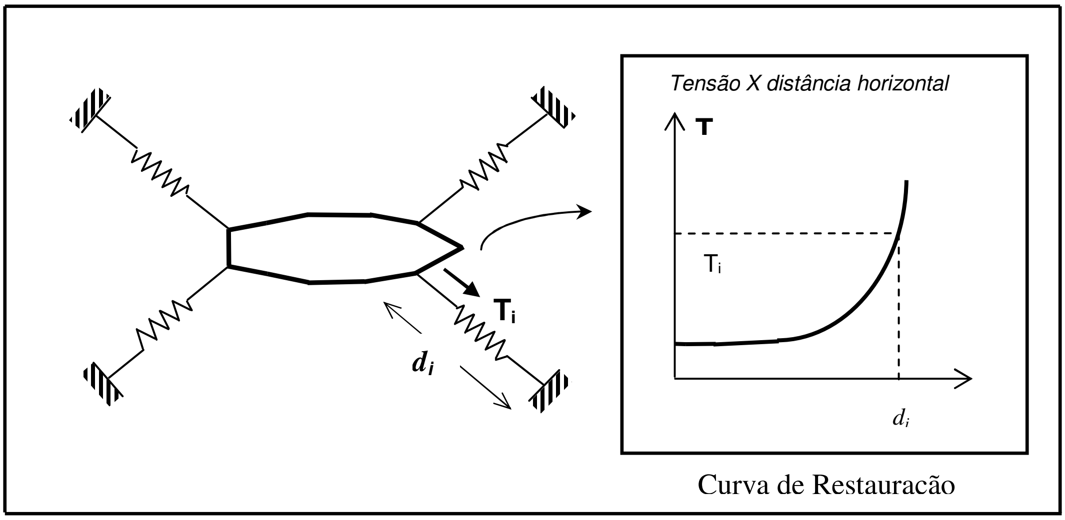

Figura 4.3 – Representação de linhas de ancoragem por meio de molas não–lineares.

A solução do sistema de Equações (4.3) conduz a uma nova posição da unidade flutuante. Esse cálculo é repetido até que a resultante dos deslocamentos obtidos seja menor do que uma dada tolerância (critério de convergência), isto é, até que a unidade alcance a sua posição final de equilíbrio estático.

4.5. Função *Fitness*

A função *fitness* ou de aptidão é um valor de qualidade que mede a eficiência da reprodução dos indivíduos em uma população, de acordo com o princípio de sobrevivência dos mais aptos [18]. Isto significa que um cromossomo com um valor mais elevado de *fitness* terá maiores probabilidades de ser selecionado como pai no processo de reprodução. Conseqüentemente, o problema de minimização da função objetivo pode ser transformado em um problema de maximização da função *fitness*, a qual é definida pelas seguintes expressões:

$$F_i = 1 - \frac{\phi_i}{\phi_{\max}} \tag{4.4}$$

$$\phi_i = \frac{\Delta_i^2}{\Delta_{avg}^2} \tag{4.5}$$

onde Δ_i^2 é a função objetivo (Equação (4.1)) e Δ_{avg}^2 é a média dos valores da função objetivo. De acordo com a Equação (4.4), a função *fitness* é normalizada para que os valores negativos possam ser excluídos.

4.6. Operador *Crossover*

Neste trabalho foi adotado o operador *crossover* simples ou *crossover* de um ponto (1X). Neste tipo de operador, apenas um ponto de cruzamento é escolhido aleatoriamente; assim os dois cromossomos selecionados como pais trocam seu material genético a partir desse ponto para dar origem a novos filhos, tal como ilustrado na Figura 3.6.

4.7. Operador Mutação

Neste trabalho, o operador mutação troca o valor de um determinado alelo de 0 para 1 e vice-versa, de acordo com um valor pré-definido de probabilidade (Pm), conforme ilustrado na Figura 3.7.

4.8. Fator *Generation* GAP

O fator GAP é um parâmetro que controla o número de indivíduos que será substituído a cada geração. Neste trabalho foi implementado o algoritmo *steady-state*, cuja principal característica é a substituição de um ou, no máximo, dois indivíduos por geração. Portanto, o valor de GAP utilizado no presente trabalho é obtido pela seguinte equação:

$$G = \frac{2}{n} \tag{4.6}$$

sendo *n* o tamanho total da população.

4.9. Algoritmo SSGA

Os principais passos computacionais do SSGA, implementados neste trabalho, são:

```
Início
< 1 > Inicialização de parâmetros: tamanho da  população,
      probabilidades de crossover e mutação, dentre outros;

< 2 > Semeação
     < 2.1 > População inicial é gerada aleatoriamente (utilizando
             uma representação binária);
     < 2.2 > População inicial é decodificada (Equações (3.9),
             (3.10) e (3.11));
     < 2.3 > Cálculo do offset;
     < 2.3 > Cálculo do valor fitness de cada indivíduo –
             Equação (4.4);

< 3 > Reprodução
     < 3.1 > Dois indivíduos são selecionados como pais (seleção por
             ranking);
     < 3.2 > Aplicação do operador crossover;
     < 3.3 > Aplicação do operador mutação;

< 4 > Atualização
     < 4.1 > Os dois novos indivíduos resultantes do processo de
             reprodução substituem os dois piores da população atual;

< 5 > Avaliação
     < 5.1 > Os dois novos cromossomos são decodificados;
     < 5.2 > Cálculo do offset;
     < 5.3 > O fitness dos dois novos cromossomos é calculado;

< 6 > Critério de parada
      Se satisfeito, ir para o passo 7; caso contrário, voltar
      para o passo 3(o critério de parada adotado foi o número
      máximo de iterações);

< 7 > Relatório (apresentação dos resultados obtidos);

Fim
```


4.10. Aplicação

Com o objetivo de testar a eficiência do algoritmo proposto, os procedimentos computacionais previamente descritos foram implementados em um programa de computação (escrito em linguagem C) e foram aplicados em vários problemas de otimização discreta de treliças e de otimização de funções contínuas, encontrados na literatura técnica.

Para obter resultados equivalentes e poder fazer as respectivas comparações, todos os problemas foram minimizados utilizando os mesmos parâmetros no que se refere a tamanho de população, probabilidade de mutação e número de iterações. Os resultados são mostrados nas seções a seguir.

4.10.1. Treliça de 2 Barras

Problema: minimizar o peso da estrutura ilustrada na Figura 4.4, sujeita ao carregamento indicado. Dados: σ_{adm} (tensão admissível) = 25 ksi, u_{adm} (deslocamento admissível) = 2 in, E (módulo de elasticidade) = 10^4 ksi, ρ (peso específico) = 0.10 lb/in^3, P = 100 kips.

Os valores discretos, correspondentes às áreas das seções transversais das barras e que serão atribuídos às variáveis de projeto, podem ser escolhidos a partir da seguinte lista de perfis disponíveis: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3.0, 3.2 e 3.4 (in^2). Este problema foi estudado por Fonseca e das Neves [30].

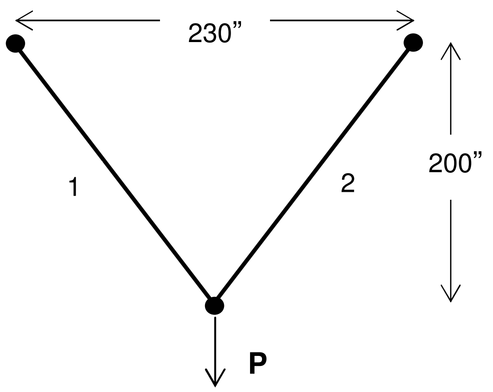

Figura 4.4 – Treliça de 2 barras.

Configurações do AG utilizado:

- População inicial aleatória
- Tamanho da população: 8
- Número máximo de iterações: 100
- Probabilidade de mutação Pm: 0.01
- Coeficiente de penalidade C_i, Equação (3.4): 11

Tempo de execução: 0.01 seg.

Após 100 iterações, o peso mínimo de 110.73 lb foi obtido na iteração 89. A Figura 4.5 mostra o processo de convergência discreta para o problema em estudo. Os resultados obtidos, bem como as respectivas comparações, são mostrados na Tabela 4.1.

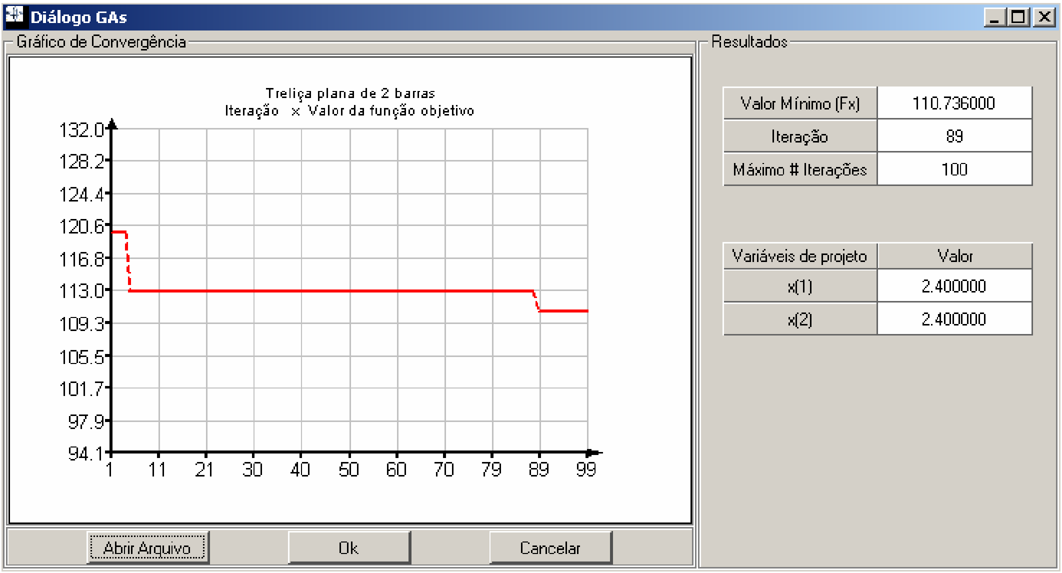

Figura 4.5 – Gráfico de convergência para a otimização discreta da treliça de 2 barras.

Tabela 4.1 – Comparação de resultados obtidos com a treliça de 2 barras.

Variáveis (in^2)	**SSGA**	**Ref. [30]**
A_1	2.40	2.40
A_2	2.40	2.40
Peso (lb)	**110.73**	**110.73**

4.10.2. Treliça de 3 Barras

Problema: minimizar o peso da estrutura ilustrada na Figura 4.6, sujeita ao carregamento indicado. Dados: σ_{adm} = 147.15 Mpa, u_{adm} = 5.08 mm, E = 2.008 x 10^5 Mpa, ρ = 7.85 x 10^3 Kgf/m^3, P = 100 kips. As barras 1 e 2 apresentam a mesma área de seção transversal, isto é, $A_1 = A_2$. Os valores discretos, correspondentes às áreas das seções transversais das barras e que serão atribuídos às variáveis de projeto, podem ser escolhidos a partir da seguinte lista de perfis disponíveis: 1.2, 1.4, 1.6, 1.8, 2.0, 2.2, 2.4, 2.6, 2.8, 3.0, 3.2, 3.4, 3.6, 3.8, 4.0 e 4.4 (cm^2). Este problema foi estudado por Rajeev e Krishnamoorthy [31].

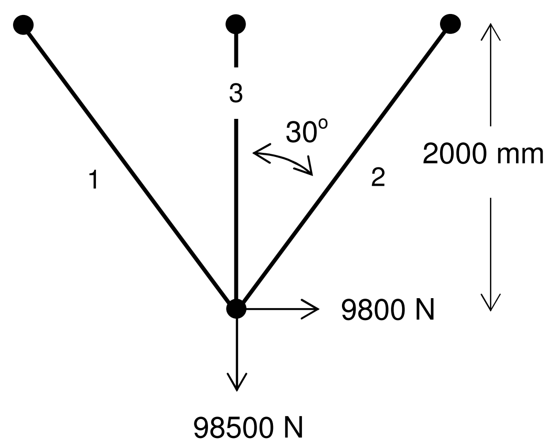

Figura 4.6 – Treliça de 3 barras.

Configurações do AG utilizado:

- População inicial aleatória
- Tamanho da população: 10
- Número máximo de iterações: 200
- Probabilidade de mutação Pm: 0.01
- Coeficiente de penalidade C_i, Equação (3.4): 11

Tempo de execução: 0.016 seg.

Após 200 iterações, o peso mínimo de 14.76 kgf foi obtido na iteração 158. A Figura 4.7 mostra o processo de convergência discreta para este exemplo. Os resultados obtidos, bem como as respectivas comparações, são mostrados na Tabela 4.2.Tabela 4.2

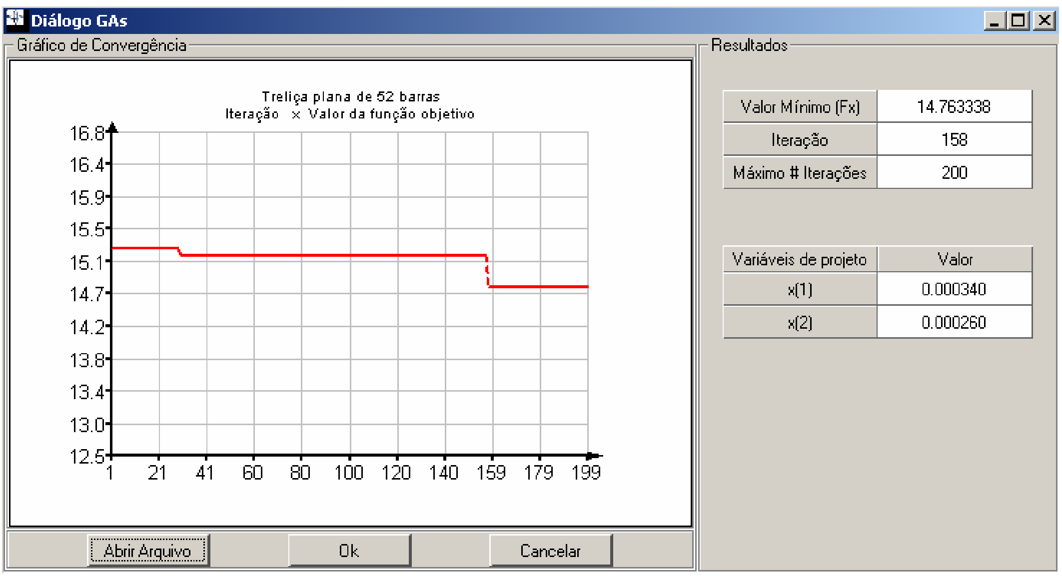

Figura 4.7 – Gráfico de convergência para a otimização discreta da treliça de 3 barras

Tabela 4.2 – Comparação de resultados obtidos com a treliça de 3 barras.

Variáveis (cm^2)	**SSGA**	**Ref. [31]**
A_1	2.6	2.4
A_2	2.6	2.4
A_3	3.4	3.8
Peso (Kgf)	**14.76**	14.77

4.10.3. Treliça de 10 Barras

Problema: minimizar o peso da estrutura ilustrada na Figura 4.8 sujeita ao carregamento indicado. Dados: σ_{adm} = 25 ksi, u_{adm} = 2 in, E = 10^4 ksi, ρ = 0.10 lb/in^3, P = 100 kips.

Os valores discretos, correspondentes às áreas das seções transversais das barras, foram obtidos a partir do Manual do Instituto Americano de Construções de Aço (*American Institute of Steel Construction Manual*). Foram considerados os seguintes valores: 1.62, 1.80, 1.99, 2.13, 2.38, 2.62, 2.63, 2.88, 2.93, 3.09, 3.13, 3.38, 3.47, 3.55, 3.63, 3.84, 3.87, 3.88, 4.18, 4.22, 4.49, 4.59, 4.80, 4.97, 5.12, 5.74, 7.22, 7.97, 11.5, 13.5, 13.9, 14.2, 15.5, 16.0, 16.9, 18.8, 19.9, 22.0, 22.9, 26.5, 30.0 e 33.5 (in^2). Este problema foi estudado por Rajeev e Krishnamoorthy [31].

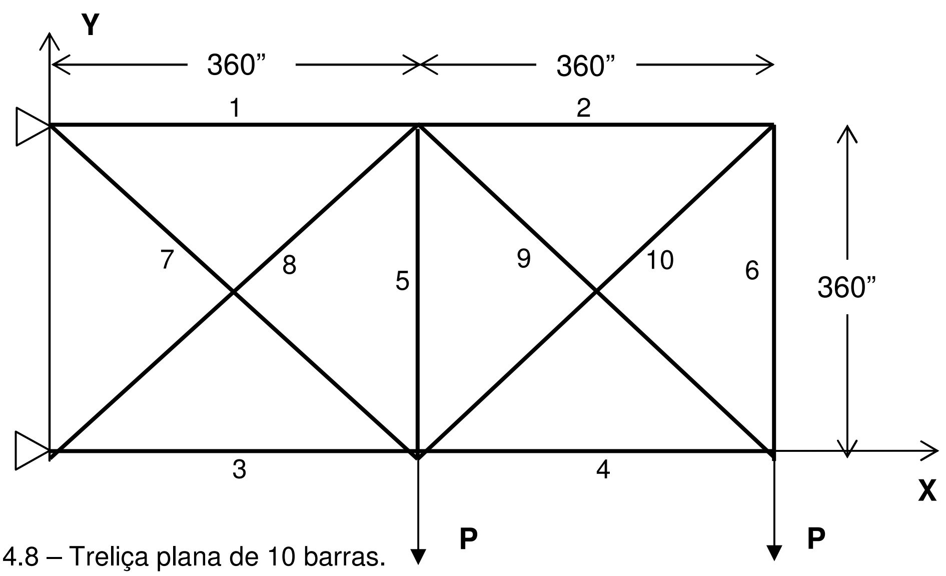

Figura 4.8 – Treliça plana de 10 barras.

Configurações do AG utilizado:

- População inicial aleatória
- Tamanho da população: 40
- Número máximo de iterações: 10000
- Probabilidade de mutação Pm: 0.01
- Coeficiente de penalidade C_i, Equação (3.4): 11

Tempo de execução: 19.84 seg.

A Figura 4.9 mostra o processo de convergência discreta para este exemplo. O resultados obtidos no processo de otimização, juntamente com as respectivas comparações, são apresentados na Tabela 4.3.

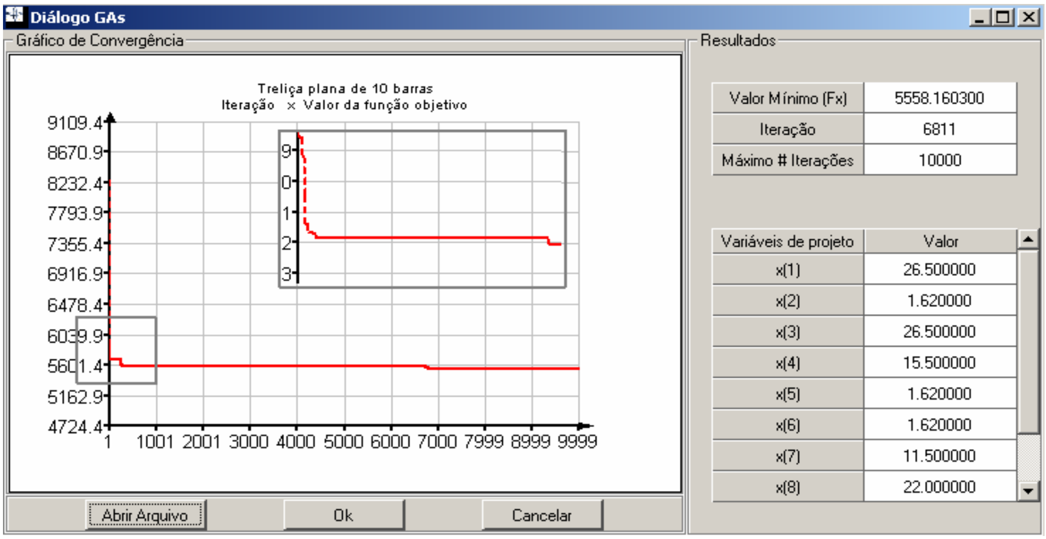

Figura 4.9 – Gráfico de convergência para a otimização discreta da treliça de 10 barras.

Tabela 4.3 – Comparação de resultados obtidos com a treliça de 10 barras.

Variáveis (in^2)	**SSGA**	**Ref [31]**
A_1	26.50	33.50
A_2	1.62	1.62
A_3	26.50	22.00
A_4	15.50	15.50
A_5	1.62	1.62
A_6	1.62	1.62
A_7	11.50	14.20
A_8	22.00	19.90
A_9	22.00	19.90
A_{10}	1.80	2.62
Peso (lb)	**5558.16**	5613.84

4.10.4. Treliça Espacial de 25 Barras

Problema: minimizar o peso da estrutura ilustrada na Figura 4.10 sujeita ao carregamento indicado. Dados: σ_{adm} = 40000 psi, u_{adm} = 0.35 in, E = 10^7 psi, ρ = 0.10 lb/in^3.

A otimização deve ser realizada mantendo-se a relação de simetria da estrutura em relação aos planos y-z e x-z. Oito variáveis de projeto são usadas para dimensionar os 25 elementos da treliça, conforme ilustrado na Tabela 4.4.

Os valores discretos, correspondentes às áreas das seções transversais das barras e que serão atribuídos às variáveis de projeto, podem ser escolhidos a partir da seguinte lista de perfis disponíveis: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.1, 1.2, 1.3, 1.4, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.8, 3.0, 3.2 e 3.4 (in^2). Os carregamentos aplicados são mostrados na Tabela 4.5. Este problema foi estudado por Wu e Chow [18].

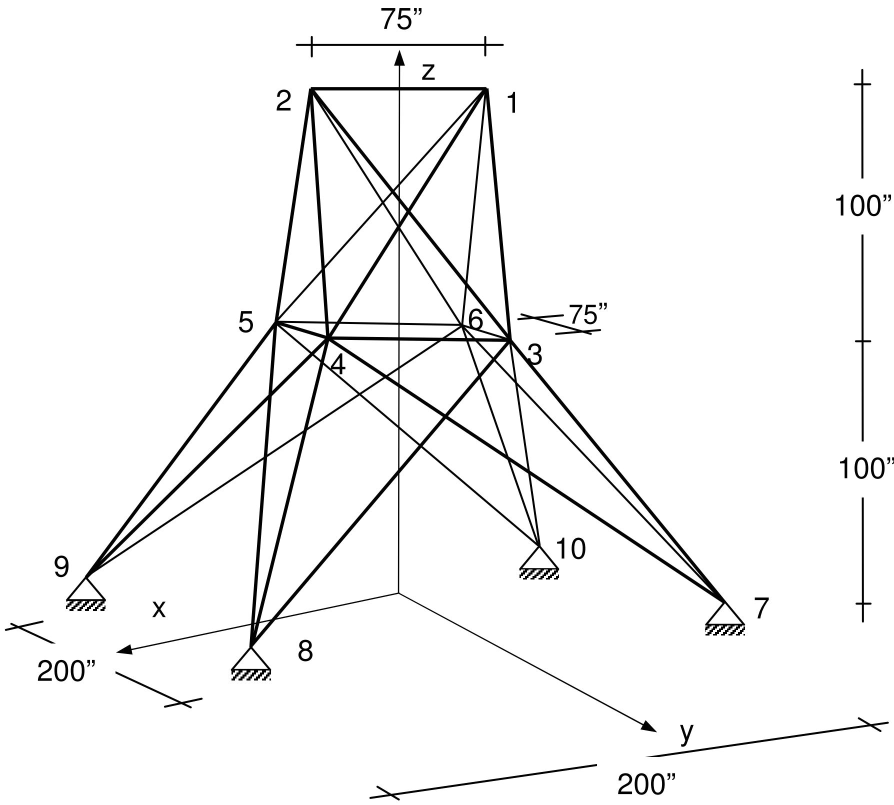

Figura 4.10 – Treliça espacial de 25 barras.

Tabela 4.4 – Grupos de elementos da treliça espacial de 25 barras.

Grupo	**Elemento (Nó inicial – Nó final)**
A_1	1 (1,2)
A_2	2 (1,4), 3 (2,3), 4(1,5), 5(2,6)
A_3	6(2,5), 7(2,4), 8(1,3), 9(1,6)
A_4	10(3,6), 11(4,5)
A_5	12(3,4), 13(5,6)
A_6	14(3,10), 15(6,7), 16(4,9), 17(5,8)
A_7	18(3,8), 19(4,7), 20(6,9), 21(5,10)
A_8	22(3,7), 23(4,8), 24(5,9), 25(6,10)

Tabela 4.5 – Detalhe de cargas da treliça espacial de 25 barras.

Nó	**P_x (lb)**	**P_y (lb)**	**P_z (lb)**
1	1000	-10000	-10000
2	0	-10000	-10000
3	500	0	0
4	600	0	0

Configurações do AG utilizado:

- População inicial aleatória
- Tamanho da população: 60
- Número máximo de iterações: 30000
- Probabilidade de mutação Pm: 0.01
- Coeficiente de penalidade C_i, Equação (3.4): 60

Tempo de execução: 174.78 seg.

Após 30000 iterações o valor mínimo de 485.04 lb foi obtido na iteração 20465. A Figura 4.9 mostra o processo de convergência do problema de otimização discreta da treliça espacial de 25 barras. Os resultados obtidos são mostrados na Tabela 4.6, bem como as comparações com os resultados da referência [18].

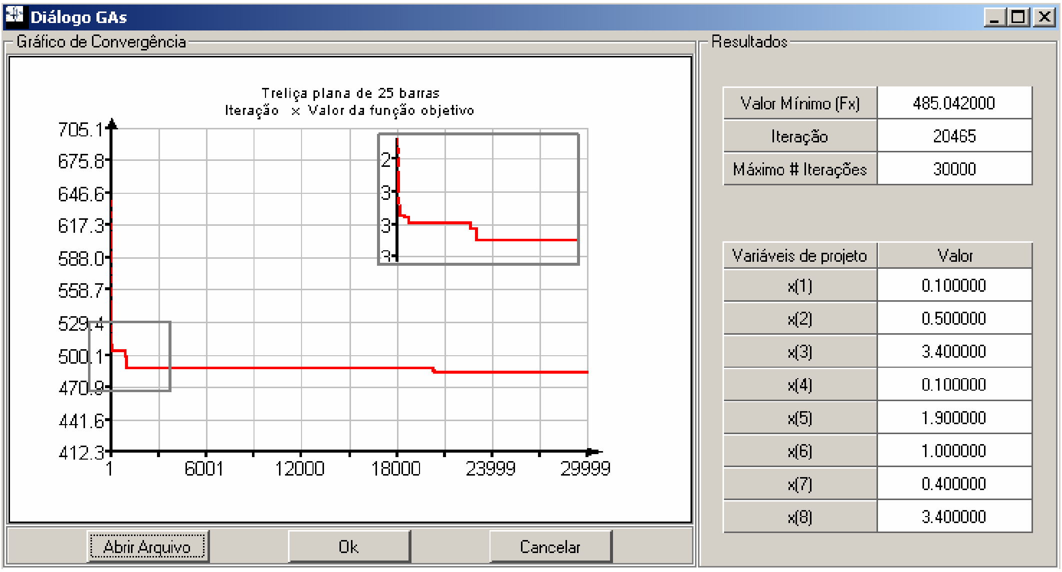

Figura 4.11 – Gráfico de convergência para a otimização discreta da treliça espacial de 25 barras.

Tabela 4.6 – Comparação de resultados obtidos com a treliça espacial de 25 barras.

Variáveis (in^2)	**SSGAs**	**Ref. [18]**
A_1	0.1	0.1
A_2	0.5	0.5
A_3	3.4	3.4
A_4	0.1	0.1
A_5	1.9	1.5
A_6	1.0	0.9
A_7	0.4	0.6
A_8	3.4	3.4
Peso (lb)	**485.04**	486.29

4.10.5. Treliça Plana de 52 Barras

Problema: minimizar o peso da estrutura ilustrada na Figura 4.12 sujeita ao carregamento indicado. Dados: σ_{adm} = 180 Mpa, E = 2.07 x 10^5 Mpa, ρ = 7860.0 kgf/m^3, P_x = 100 kN, P_y = 200 kN.

Os 52 elementos foram distribuídos em 12 grupos. Os valores discretos, correspondentes às áreas das seções transversais das barras e que podem ser atribuídos às variáveis de projeto, são mostrados na Tabela 4.7. Este problema foi estudado por Wu e Chow [18].

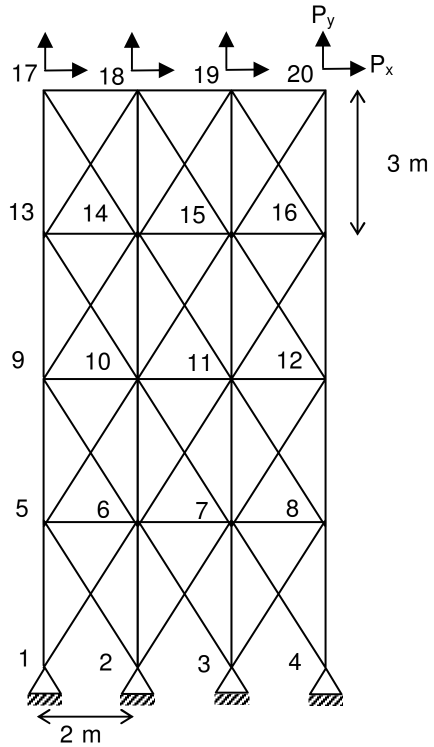

Figura 4.12 – Treliça plana de 52 barras.

Configurações do AG utilizado:

- População inicial aleatória
- Tamanho da população: 60
- Número máximo de iterações: 60000
- Probabilidade de mutação Pm: 0.01
- Coeficiente de penalidade C_i, Equação (3.4): 60

Tempo de execução: 1138.06 seg.

Após 60000 iterações o peso mínimo de 1963.75 kgf foi obtido na iteração 41109. As Figura 4.13 e Figura 4.14 mostram os gráficos de convergência do processo de otimização. Uma comparação dos resultados obtidos com os da referência [18] é apresentada na Tabela 4.8. Observe-se que este problema foi resolvido utilizando-se duas versões do operador *crossover*: de um ponto (1X) e de dois pontos (2X), respectivamente, conforme ilustrado na Tabela 4.8.

Tabela 4.7 – Seções disponíveis para o exemplo da treliça de 52 barras.

No.	in^2	mm^2	No.	in^2	mm^2
1	0.111	71.613	33	3.840	2477.414
2	0.141	90.968	34	3.870	2496.769
3	0.196	126.451	35	3.880	2503.221
4	0.250	161.290	36	4.180	2696.769
5	0.307	198.064	37	4.220	2722.575
6	0.391	252.258	38	4.490	2896.768
7	0.442	285.161	39	4.590	2961.284
8	0.563	363.225	40	4.800	3096.768
9	0.602	388.386	41	4.970	3206.445
10	0.766	494.193	42	5.120	3303.219
11	0.785	506.451	43	5.740	3703.218
12	0.994	641.289	44	7.220	4658.055
13	1.000	645.160	45	7.970	5141.925
14	1.228	792.256	46	8.530	5503.215
15	1.266	816.773	47	9.300	5999.988
16	1.457	940.000	48	10.850	6999.986
17	1.563	1008.385	49	11.500	7419.340
18	1.620	1045.159	50	13.500	8709.660
19	1.800	1161.288	51	13.900	8967.724
20	1.990	1283.868	52	14.200	9161.272
21	2.130	1374.191	53	15.500	9999.980
22	2.380	1535.481	54	16.000	10322.560
23	2.620	1690.319	55	16.900	10903.204
24	2.630	1696.771	56	18.800	12129.008
25	2.880	1858.061	57	19.900	12838.684
26	2.930	1890.319	58	22.000	14193.520
27	3.090	1993.544	59	22.900	14774.164
28	3.130	2019.351	60	24.500	15806.420
29	3.380	2180.641	61	26.500	17096.740
30	3.470	2238.705	62	28.000	18064.480
31	3.550	2290.318	63	30.000	19354.800
32	3.630	2341.931	64	33.500	21612.860

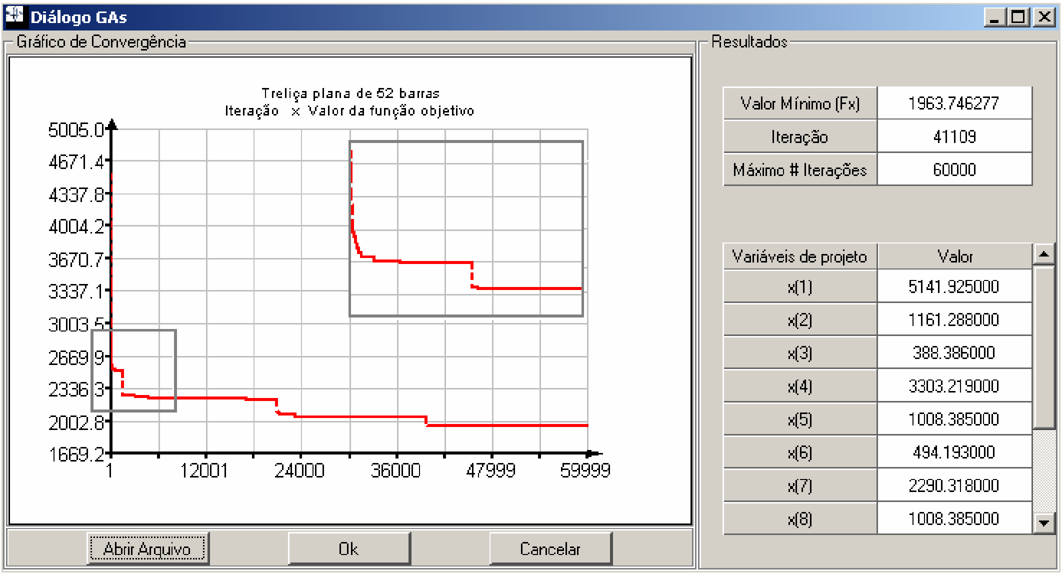

Figura 4.13 – Gráfico de convergência para a otimização discreta da treliça plana de 52 barras – *crossover* de um ponto (1X).

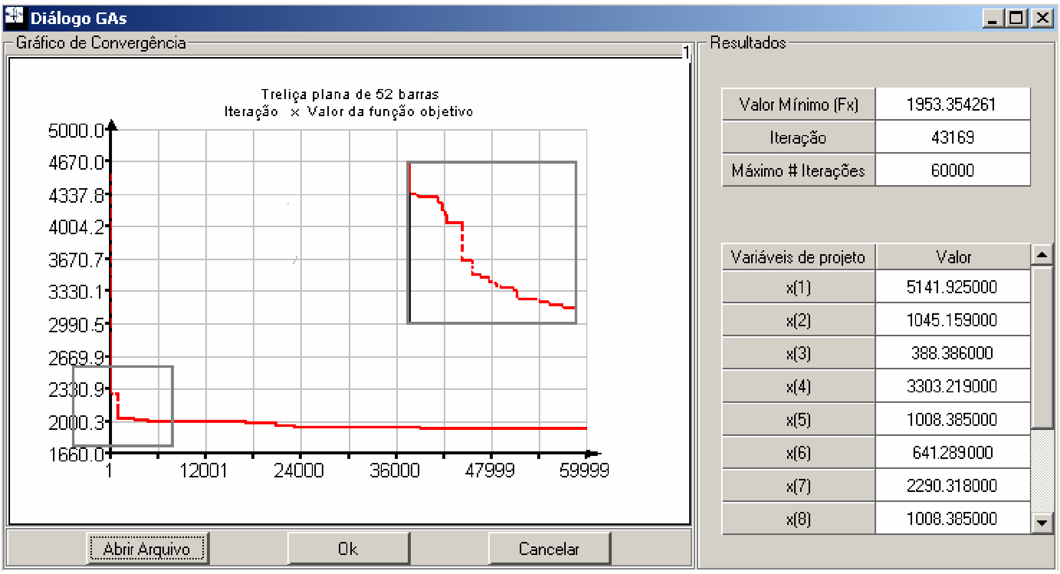

Figura 4.14 – Gráfico de convergência para a otimização discreta da treliça plana de 52 barras – *crossover* de dois pontos (2X).

Tabela 4.8 – Comparações de resultados obtidos com a treliça plana de 52 barras.

Variáveis	SSGA		Ref. [18]	
	IX	2X	IX	2X
A_1(1-4)	5141.925	5141.925	3703.218	4658.055
A_2(5-10)	1045.159	1045.159	2722.575	1161.288
A_3(11-13)	285.161	388.386	1858.061	645.160
A_4(14-17)	3303.219	3303.219	3206.445	3303.219
A_5(18-23)	1161.288	1008.385	1008.385	1045.159
A_6(24-26)	494.193	645.289	1008.385	494.193
A_7(27-30)	2290.318	2290.318	2477.41	2477.414
A_8(31-36)	1008.385	1008.385	1008.385	1045.159
A_9(37-39)	363.225	494.193	388.386	363.225
A_{10}(40-43)	1690.319	1283.86	2477.414	1696.771
A_{11}(44-49)	1008.385	1161.288	1008.385	1045.159
A_{12}(50-52)	388.386	388.386	1008.385	792.256
Peso (Kgf)	**1963.75**	**1953.35**	2294.521	1970.142

4.10.6. Funções Contínuas

Nesta seção, o algoritmo proposto é testado com relação às funções contínuas apresentadas no trabalho desenvolvido por André *et al.* [24].

Os gráficos de convergência são ilustrados nas Figuras 4.15 a 4.19, e as comparações dos resultados obtidos com a referência [24] são mostrados na Tabela 4.9.

- **F1**

$$f(x) = 2(x-0.75)^2 + \sin(5\pi x - 0.4\pi) - 0.125$$

$$onde \quad 0 \le x \le 1$$

População inicial aleatória

Tamanho da população: 60

Número máximo de iterações: 1000

Probabilidade de mutação Pm: 0.05

Tempo de execução: 0.34 seg.

Após 1000 iterações, o valor mínimo de -1.12323 foi obtido na iteração 98.

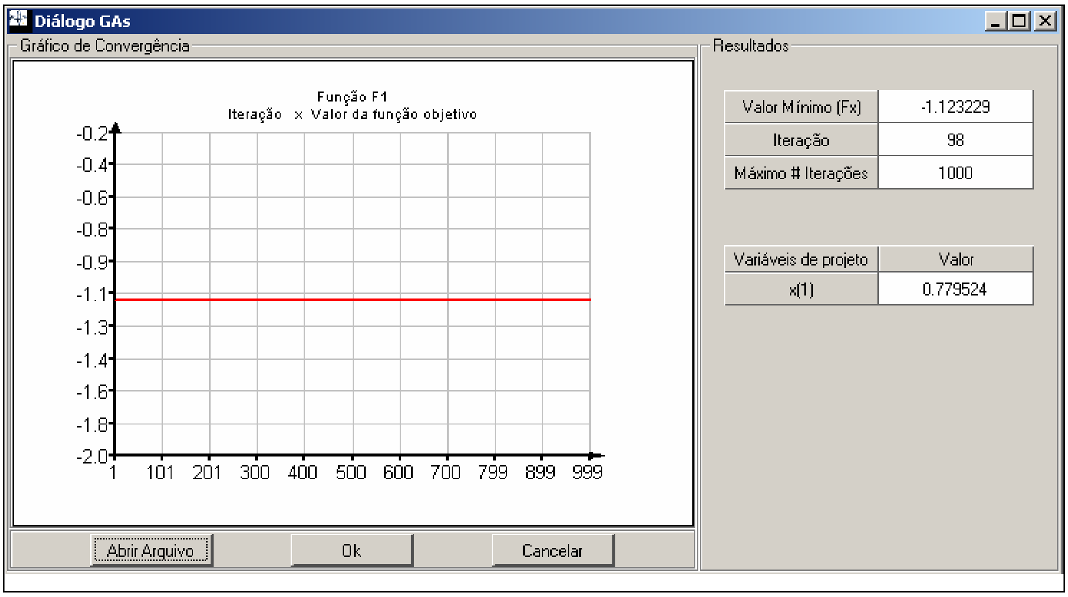

Figura 4.15 – Gráfico de convergência do processo de otimização da Função F1.

- **F3**

$$f(x) = -\sum_{j=1}^{5}\{j \sin[(j+1)x + j]\}$$

$$onde \quad -10 \le x \le 10$$

População inicial aleatória

Tamanho da população: 60

Número máximo de iterações: 1000

Probabilidade de mutação Pm: 0.05

Tempo de execução: 0.45 seg.

Após 1000 iterações, o valor mínimo de -12.03125 foi obtido na iteração 397.

- **Goldprice**

$$f(x, y) = [1 + (x + y + 1)^2 (19 - 14x + 3x^2 - 14y + 6xy + 3y^2)] \cdot$$
$$[30 + (2x - 3y)^2 (18 - 32x + 12x^2 + 48y - 36xy + 27y^2)]$$

$$onde \quad -2 \le x \le 2 \quad e \quad -2 \le y \le 2$$

População inicial aleatória

Tamanho da população: 60

Número máximo de iterações: 1000

Probabilidade de mutação Pm: 0.01

Tempo de execução: 0.64 seg.

Após 1000 iterações, o valor mínimo de 3.00000 foi obtido na iteração 149.

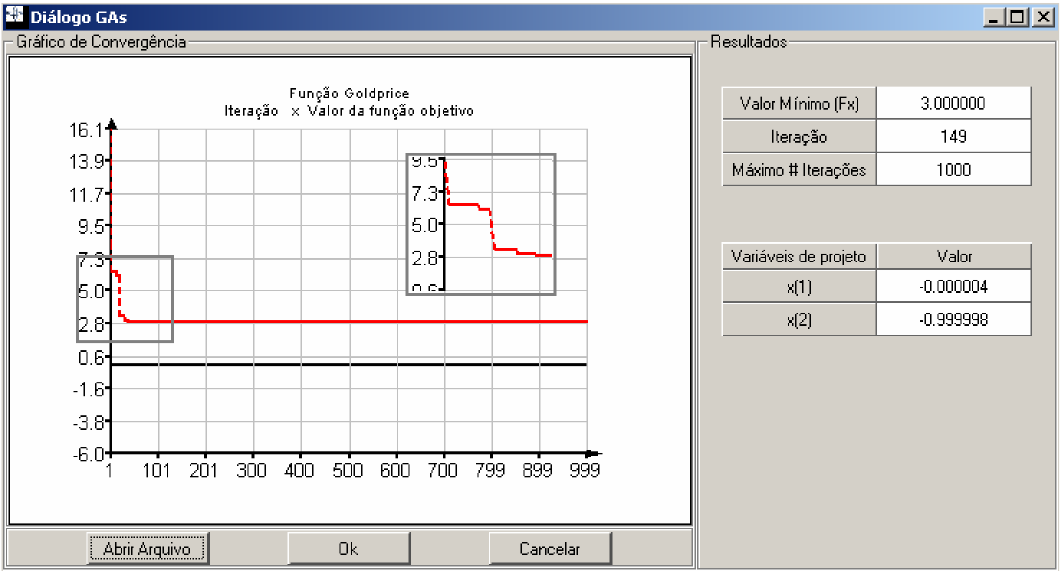

Figura 4.16 – Gráfico de convergência do processo de otimização da Função Goldprice.

- **Quartic**

$$f(x, y) = \frac{x^4}{4} - \frac{x^2}{2} + \frac{x}{10} + \frac{y^2}{2}$$

$$onde \quad -10 \leq x \leq 10 \quad e \quad -10 \leq y \leq 10$$

População inicial aleatória

Tamanho da população: 60

Número máximo de iterações: 500

Probabilidade de mutação Pm: 0.01

Tempo de execução: 0.35 seg

Após 500 iterações, o valor mínimo de -0.35239 foi obtido na iteração 411.

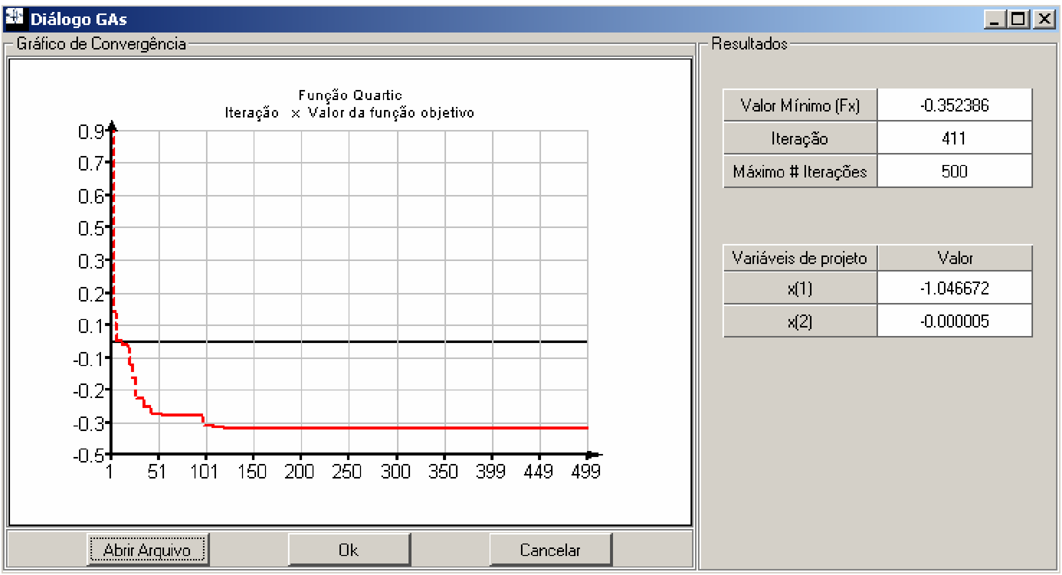

Figura 4.17 – Gráfico de convergência do processo de otimização da Função Quartic.

- **Shubert**

$$f(x, y) = \sum_{i=1}^{5} i\cos[(i+1)x+i] \cdot \sum_{i=1}^{5} i\cos[(i+1)y+i]$$

$$onde \quad -10 \le x \le 10 \quad e \quad -10 \le y \le 10$$

População inicial aleatória

Tamanho da população: 60

Número máximo de iterações: 650

Probabilidade de mutação Pm: 0.01

Tempo de execução: 0.58 seg.

Após 650 iterações, o valor mínimo de -186.73091 foi obtido na iteração 633.

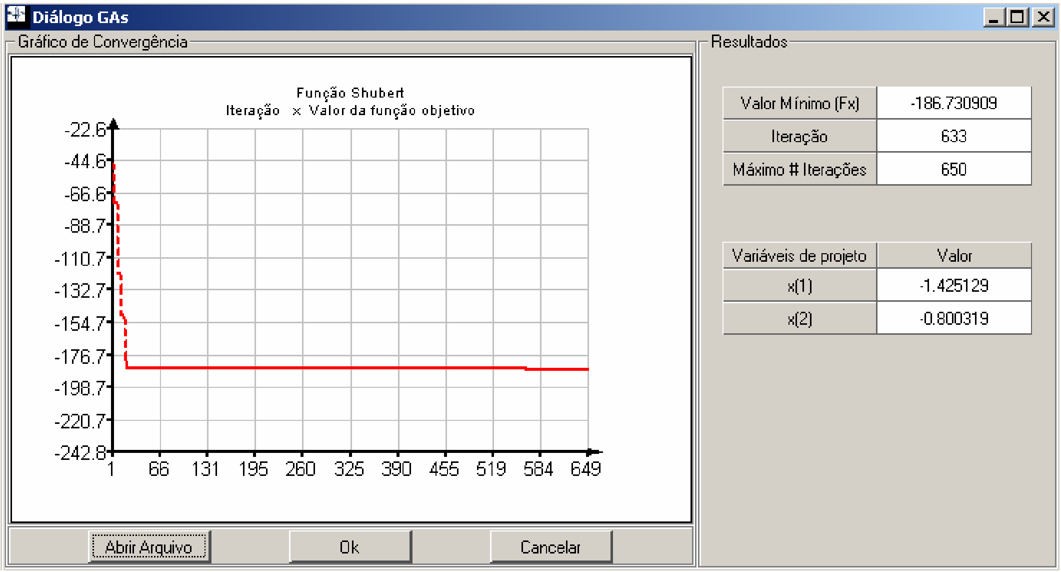

Figura 4.18 – Gráfico de convergência do processo de otimização da Função Shubert.

- **Brown3**

$$f(x) = \sum_{i=1}^{19} [(x_i^2)^{(x_{i+1}^2+1)} + (x_{i+1}^2)^{(x_i^2+1)}]$$

$$onde \ x = (x_1, \ldots, x_{20})^T \quad e \quad -1 \le x_i \le 4 \ \ para \ 1 \le i \le 20$$

População inicial aleatória

Tamanho da população: 100

Número máximo de iterações: 120000

Probabilidade de mutação Pm: 0.01

Tempo de execução: 1270.8 seg.

Após 120000 iterações, o valor mínimo de 0.010848 foi obtido na iteração 95624.

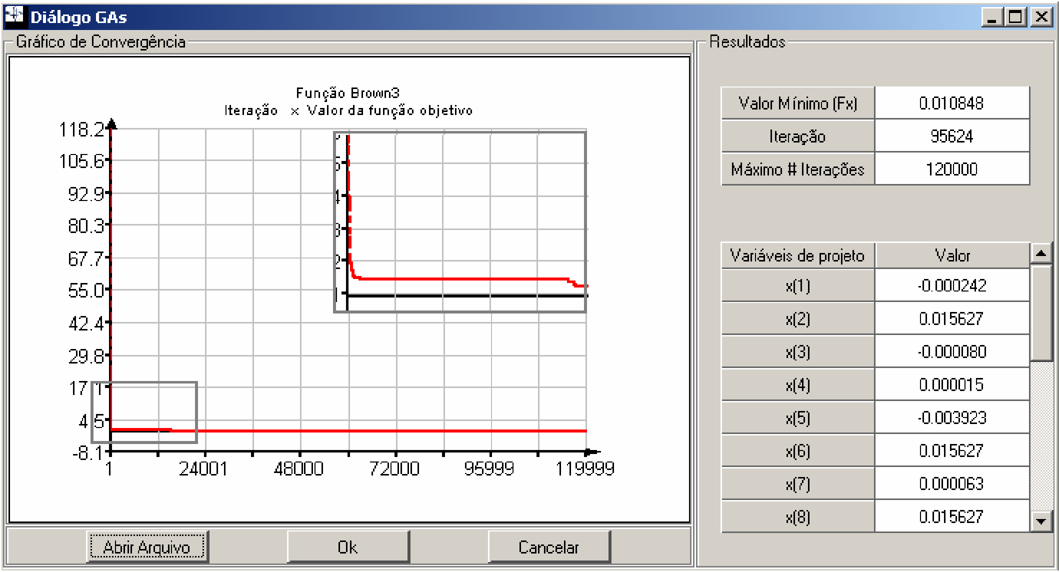

Figura 4.19 – Gráfico de convergência do processo de otimização da Função Brown3.

Tabela 4.9 – Comparações dos resultados obtidos com as funções contínuas.

Função	**Mínimo Teórico**	**SSGA**		**Ref. [24]**	
		Iteração	**Mínimo**	**Iteração**	**Mínimo**
F1	-1.12323	98	**-1.12323**	784	**-1.12323**
F3	-12.03125	397	**-12.03125**	744	-12.03120
Goldprice	3.00000	149	**3.00000**	4632	3.00028
Quartic	-0.35239	411	**-0.35239**	3168	-0.35238
Shubert	-186.73091	633	**-186.73091**	2364	-186.72802
Brown3	0.00000	95624	**0.010848**	106589	0.67464

4.10.7. Comparações com o algoritmo do ICA

Com o objetivo de se ter um outro parâmetro de comparação e verificação da eficiência do algoritmo desenvolvido no presente trabalho, as funções contínuas da seção anterior foram minimizadas utilizando-se um algoritmo genético disponibilizado pelo grupo ICA (Inteligência de Computação Aplicada) do Departamento de Engenharia Elétrica da PUC-Rio [32]. Neste algoritmo os operadores de *crossover* e mutação funcionam com taxas diferentes ao longo do processo de convergência. Essas taxas indicam a probabilidade com que tais operadores serão selecionados e executados.

A Tabela 4.10 apresenta as comparações dos resultados obtidos.

Tabela 4.10 – Comparação de resultados de funções contínuas com referencia [32].

Função	**Mínimo Teórico**	**SSGA**			**ICA [32]**		
		Tempo (seg)	**Iter.**	**Mín.**	**Iter.**	**Mín.**	**Tempo (seg)**
F1	-1.12323	0.34	98	**-1.12323**	104	**-1.12323**	2.06
F3	-12.03125	0.45	397	**-12.03125**	166	**-12.03125**	1.91
Goldprice	3.00000	0.64	149	**3.00000**	390	**3.00000**	1.88
Quartic	-0.35239	0.35	411	**-0.35239**	253	**-0.35239**	0.88
Shubert	-186.73091	0.58	633	**-186.73091**	644	**-186.73091**	1.27
Brown3	0.00000	1270.8	95624	0.010848	12000	**0.00000**	1458.5