**UNIVERSIDADE DE PERNAMBUCO**

**ESCOLA POLITÉCNICA DE PERNAMBUCO**

**RELATÓRIO DE PRÁTICA DE ALGORITIMOS GENÉTICOS**

CARLOS HENRIQUE MACIEL - PPGEC

CRISTÓVÃO ZUPPARDO RUFINO - PPGES

1. Introdução

A motivação desse estudo é desenvolver e aprender como desenvolver um algoritmo inteligente baseado na técnica de computação evolucionária.

O objeto de estudo é o código-fonte da versão clássica de algoritmo genético.

O propósito é programar e testar um algoritmo genético. Os objetivos secundários são verificar a convergência do algoritmo, comparar os operadores e definir os melhores parâmetros que fazem com que o algoritmo convirja mais rápido.

É assumida a perspectiva do desenvolvedor durante o desenvolvimento do algoritmo, e a perspectiva do usuário para a melhor definição das variáveis do algoritmo para garantir que os resultados sejam alcançados no menor tempo, custo e maior qualidade.

O escopo que define esse trabalho é um projeto único para a implementação e teste do algoritmo genético.

Objetivo da Medição

Será realizada uma comparação do desempenho do algoritmo genético, a medida que mudamos os valores assumidos pelas variáveis.

* Melhor *fitness* por iteração.

Objetivo do estudo

**Implementar e testar** um algoritmo genético.

**Com o objetivo de** configurar os melhores operadores e definir os parâmetros que garantem uma convergência mais rápida do algoritmo.

**A respeito do** tamanho da população, tipo de cruzamento, taxa de cruzamento, tipo de mutação, taxa de mutação e tipo de seleção para sobrevivência.

**Do ponto de vista** **do** usuário dessa técnica.

**No contexto de** estudantes de um curso de Computação Inteligente do Mestrado de Engenharia da Computação e de Sistemas.

Questões

Aumentando o número de iterações é possível garantir a convergência do algoritmo genético? A partir de que número de iteração, há uma pequena variação no resultado obtido? Quais são os melhores parâmetros a serem definidos para um algoritmo genético para minimizar a função de teste Rastringin.

Métricas

A métrica utilizada é o melhor fitness por iteração, que é apresentada por valores absolutos contidos no intervalo entre 1050 e 0.

1. Planejamento do Experimento

Projeto

O experimento consiste na análise da convergência do algoritmo genético para cada caso de teste definido a partir da seleção dos parâmetros do algoritmo genético. Além disso, será realizado o teste estatístico não paramétrico de Wilcoxon para comparar os casos de testes definidos.

Erros

O tipo de erro utilizado no teste de hipótese não paramétrico de Wilcoxon é do tipo II, o que significa que é aceito o caso em que a hipótese nula é aceita, mesmo que realmente ela não seja verificada. Na prática, aceita-se os falsos positivos. Ou seja, o erro do tipo II aceita o caso em que um paciente realizará um tratamento de câncer mesmo que ele não tenha desenvolvido a doença, ao invés de, correr o risco de não tratar um paciente de câncer.

Mudanças

Para o experimento, não houve mudanças na arquitetura do sistema, na disponibilidade do processador, nem no manuseio de memória que pudessem tornar o ambiente de teste dinâmico.

Confiabilidade

Considera-se os resultados obtidos confiáveis.

Pareamento de Dados

Como se trata do mesmo algoritmo sendo executado antes e depois de algumas mudanças, podem realizar o pareamento dos dados sem problemas.

Complexidade

O algoritmo programado foi concebido na linguagem C, uma linguagem de nível intermediário (apenas um passo de tradução para linguagem de máquina), para que o desempenho do programa fosse o melhor possível.

O código gerado é bastante reduzido, porém modularizado, o nível de complexidade do programa na notação O(*Big O*) é O(n2). Porque o algoritmo genético tem uma complexidade linear e o algoritmo de ordenação utilizado tem uma complexidade quadrática.

Métricas orientadas às questões

As métricas definidas são específicas para responder as questões de convergência e teste pareado.

A métrica selecionada é:

* Melhor *fitness* por iteração (*fitit*);

Validação das medições

As medições são válidas, pois as melhores soluções sempre são armazenadas na memória do programa, além disso, verificamos a integridade dessa memória, já que outros programas não tem permissão concedida pelo Sistema Operacional para invadir essa região e violá-la.

Coleta automatizada dos dados

Os dados são gerados e armazenados automaticamente pelo programa que executa o algoritmo. Os dados são armazenados em formato de tabela num arquivo de texto.

Nível de medição

Nominal.

Definição das Hipóteses

**Hipótese Nula (H0):** Como hipótese nula, assume-se que *fitit* de cada experimento não são diferentes.

H0: *fitie’* = *fitie’’* , tal que o *fitness* por iteração *fiti* em cada experimento *e* não são diferentes.

**Hipótese Alternativa (H1):** Como hipótese alternativa, assume-se que *fitit* de cada experimento são diferentes.

H0: *fitie’* ≠ *fitie’’* , tal que o *fitness* por iteração *fiti* em cada experimento *e* são diferentes.

Tratamento

O tratamento aplicado a esse estudo é a implementação modularizada e parametrizada do algoritmo, para facilitar a extração dos dados. As variáveis do algoritmo a serem analisadas são apresentadas nas seções seguintes, junto com os possíveis valores assumidos. Após a extração, os resultados de cada experimento foram salvo em arquivos separadamente. Cada experimento foi executado 30 vezes repetidamente.

Em seguida, os dados foram tratados e trabalhados por uma ferramenta estatística para geração dos gráficos de convergência. Por fim, os dados de cada experimento foram comparados um a um, utilizando uma ferramenta estatística para realização do teste não paramétrico pareado de Wilcoxon.

Objeto de Controle

Com o objetivo de implementar o algoritmo genético, o objeto de controle foi o pseudo código do algoritmo apresentado em [1]. Com o objetivo de analisar os parâmetros, foi utilizado como parâmetro de controle os valores da Tabela 1.

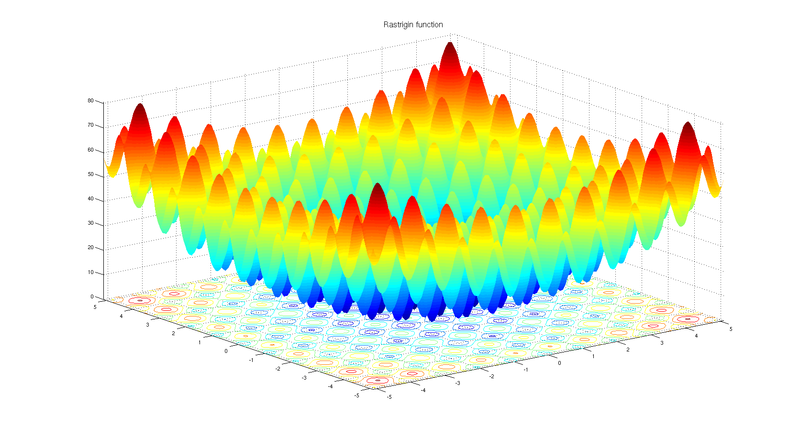
Tabela 1. Parâmetros de controle

|  |  |
| --- | --- |
| Variável |  |
| Tamanho da População | 30 indivíduos |
| Tipo de Cruzamento | 1 ponto |
| Taxa de Cruzamento | 0.9 |
| Tipo de Mutação | Uniforme |
| Taxa de Mutação | 0.1 |
| Tipo de Seleção para Sobrevivência | Elitismo |

Objetos do Experimento

Os objetos do experimentos são os possíveis valores que podem ser assumidos pelas variáveis para a minimização da função de teste. A função de teste utilizada foi a Rastringin, que é uma função bastante conhecida e relativamente difícil de se encontrar os mínimos globais devido a existência de um grande platô no qual as soluções recursivas geralmente demoram a percorrer e alcançar o mínimo global.

A função Rastrigin[2] utilizada tem 30 variáveis, sendo que o intervalo de f(x) encontra-se entre 0 e 1050, o intervalo de x encontra-se entre -5,12 e 5,12. A Figura 1 representa a função Rastrigin com 2 variáveis.



**Figura 1. Rastrigin com 2 variáveis**

Os possíveis valores assumidos pelas variáveis dos algoritmos são apresentados na Tabela 2.

Sujeitos do Experimento

Sistema operacional Linux, Arch-Linux, Kernel 3.5, gcc 4.6, libc x.y, 4GB de memória, Processador Intel Core 2 Duo 1.86MHz.

Variáveis Independentes

Todas as variáveis que não estão sobre o nosso controle.

Variáveis Dependentes

As variáveis analisadas no estudo: tamanho da população, tipo de cruzamento, taxa de cruzamento, tipo de mutação, taxa de mutação e tipo de seleção para sobrevivência.

Concepção dos Ensaios

Combinando os possíveis valores assumidos pelas variáveis, chegamos a 48 combinações possíveis, ou 48 experimentos. Para cada experimento, executamos o algoritmo 30 vezes.

1. Operação

Preparação

Antes de Começar o experimento, nós compilamos o código-fonte na arquitetura de teste com uma versão de compilador compatível, para eliminar a possibilidade de erro de processamento interno.

Análise

A análise do experimento é a comparação dos dados coletados a partir dos ensaios do objeto do experimento e ensaios de objetos de controle com o objetivo de verificar se a hipótese nula pode ser rejeitada. O estudo analisa a convergência do algoritmo de acordo com as suas variáveis.

Como nós não podemos dizer a origem dos dados gerados no experimento, se eles satisfazem o Teste de Normalidade de Kolmogorov-Smirnov, e se descendem de distribuições de mesma variância, obtido com o Teste de Fisher, nós utilizamos o Teste Não Paramétrico Pareado de Wilcoxon. O Teste de Wilcoxon é um teste estatístico que compara se as medianas de duas amostras, e analisa se suas diferenças são significantes.

Ameaças a Validade

Validade Interna

O algoritmo é executado na mesma máquina, sob as mesmas condições iniciais. Logo, o ambiente do experimento é estável.

Validade da Conclusão

Um mentor estava acompanhando a execução do estudo para garantir a correta coleta dos dados. O Teste Não Paramétrico Pareado de Wilcoxon pode ser considerado um teste adequado, já que permite que os dados sejam tratados com uma rigidez menor, quando não tem-se tantas informações sobre as amostras.

Validade da Construção

A construção do algoritmo foi baseada na implementação mais simples do Algoritmo Genético[1].

Validade Externa

Os resultados desse estudo estão limitados às restrições acima.

Execução

Os ensaios são organizados e classificados segundo a Tabela 2.

Tabela 2. Definição dos parâmetros das variáveis

|  |  |  |  |
| --- | --- | --- | --- |
| Variável |  |  |  |
| Tamanho da População | 30 indivíduos | 50 indivíduos | - |
| Tipo de Cruzamento | 1 ponto | 2 pontos | Aleatório |
| Taxa de Cruzamento | 0.9 | 0.7 | - |
| Tipo de Mutação | Uniforme | Gaussiana | - |
| Taxa de Mutação | 0.1 | 0.3 | - |
| Tipo de Seleção para Sobrevivência | Elitismo | Roleta | - |

Na Tabela 3, rotulamos cada experimento.

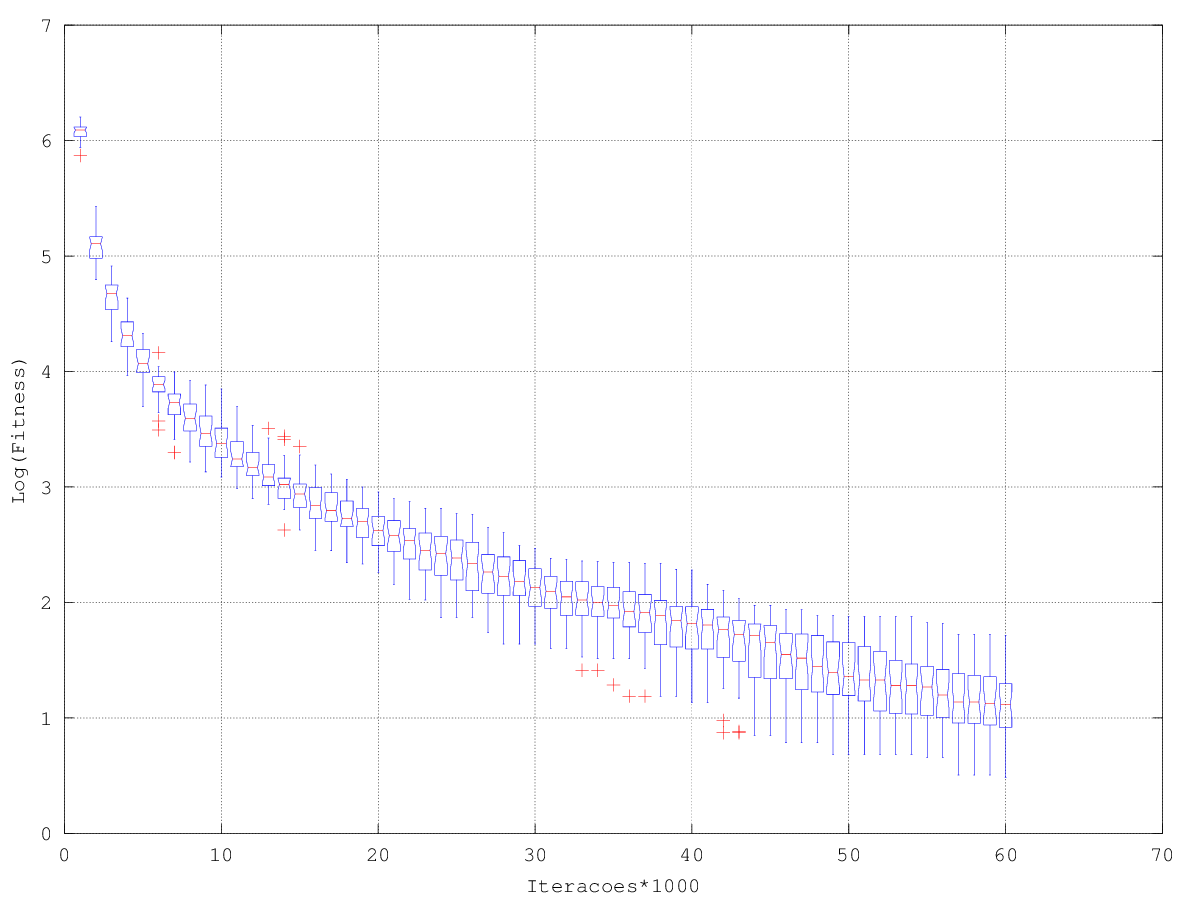
Tabela 3. Identificação de cada experimento

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Experimento** | **Tam. População** | **Taxa Mutação** | **Taxa Cruzamento** | **Tipo Cruzamento** | **Tipo Mutação** |
| 1 | 30,0 | 0,1 | 0,9 | 1,0 | 1,0 |
| 2 | 30,0 | 0,1 | 0,9 | 1,0 | 2,0 |
| 3 | 30,0 | 0,1 | 0,9 | 2,0 | 1,0 |
| 4 | 30,0 | 0,1 | 0,7 | 2,0 | 2,0 |
| 5 | 30,0 | 0,1 | 0,7 | 1,0 | 1,0 |
| 6 | 30,0 | 0,1 | 0,7 | 1,0 | 2,0 |
| 7 | 30,0 | 0,1 | 0,9 | 2,0 | 1,0 |
| 8 | 30,0 | 0,1 | 0,9 | 2,0 | 2,0 |
| 9 | 30,0 | 0,1 | 0,9 | 1,0 | 1,0 |
| 10 | 30,0 | 0,1 | 0,7 | 1,0 | 2,0 |
| 11 | 30,0 | 0,1 | 0,7 | 2,0 | 1,0 |
| 12 | 30,0 | 0,1 | 0,7 | 2,0 | 2,0 |
| 13 | 30,0 | 0,5 | 0,9 | 1,0 | 1,0 |
| 14 | 30,0 | 0,5 | 0,9 | 1,0 | 2,0 |
| 15 | 30,0 | 0,5 | 0,9 | 2,0 | 1,0 |
| 16 | 30,0 | 0,5 | 0,7 | 2,0 | 2,0 |
| 17 | 30,0 | 0,5 | 0,7 | 1,0 | 1,0 |
| 18 | 30,0 | 0,5 | 0,7 | 1,0 | 2,0 |
| 19 | 30,0 | 0,5 | 0,9 | 2,0 | 1,0 |
| 20 | 30,0 | 0,5 | 0,9 | 2,0 | 2,0 |
| 21 | 30,0 | 0,5 | 0,9 | 1,0 | 1,0 |
| 22 | 30,0 | 0,5 | 0,7 | 1,0 | 2,0 |
| 23 | 30,0 | 0,5 | 0,7 | 2,0 | 1,0 |
| 24 | 30,0 | 0,5 | 0,7 | 2,0 | 2,0 |
| 25 | 50,0 | 0,1 | 0,9 | 1,0 | 1,0 |
| 26 | 50,0 | 0,1 | 0,9 | 1,0 | 2,0 |
| 27 | 50,0 | 0,1 | 0,9 | 2,0 | 1,0 |
| 28 | 50,0 | 0,1 | 0,7 | 2,0 | 2,0 |
| 29 | 50,0 | 0,1 | 0,7 | 1,0 | 1,0 |
| 30 | 50,0 | 0,1 | 0,7 | 1,0 | 2,0 |
| 31 | 50,0 | 0,1 | 0,9 | 2,0 | 1,0 |
| 32 | 50,0 | 0,1 | 0,9 | 2,0 | 2,0 |
| 33 | 50,0 | 0,1 | 0,9 | 1,0 | 1,0 |
| 34 | 50,0 | 0,1 | 0,7 | 1,0 | 2,0 |
| 35 | 50,0 | 0,1 | 0,7 | 2,0 | 1,0 |
| 36 | 50,0 | 0,1 | 0,7 | 2,0 | 2,0 |
| 37 | 50,0 | 0,5 | 0,9 | 1,0 | 1,0 |
| 38 | 50,0 | 0,5 | 0,9 | 1,0 | 2,0 |
| 39 | 50,0 | 0,5 | 0,9 | 2,0 | 1,0 |
| 40 | 50,0 | 0,5 | 0,7 | 2,0 | 2,0 |
| 41 | 50,0 | 0,5 | 0,7 | 1,0 | 1,0 |
| 42 | 50,0 | 0,5 | 0,7 | 1,0 | 2,0 |
| 43 | 50,0 | 0,5 | 0,9 | 2,0 | 1,0 |
| 44 | 50,0 | 0,5 | 0,9 | 2,0 | 2,0 |
| 45 | 50,0 | 0,5 | 0,9 | 1,0 | 1,0 |
| 46 | 50,0 | 0,5 | 0,7 | 1,0 | 2,0 |
| 47 | 50,0 | 0,5 | 0,7 | 2,0 | 1,0 |
| 48 | 50,0 | 0,5 | 0,7 | 2,0 | 2,0 |

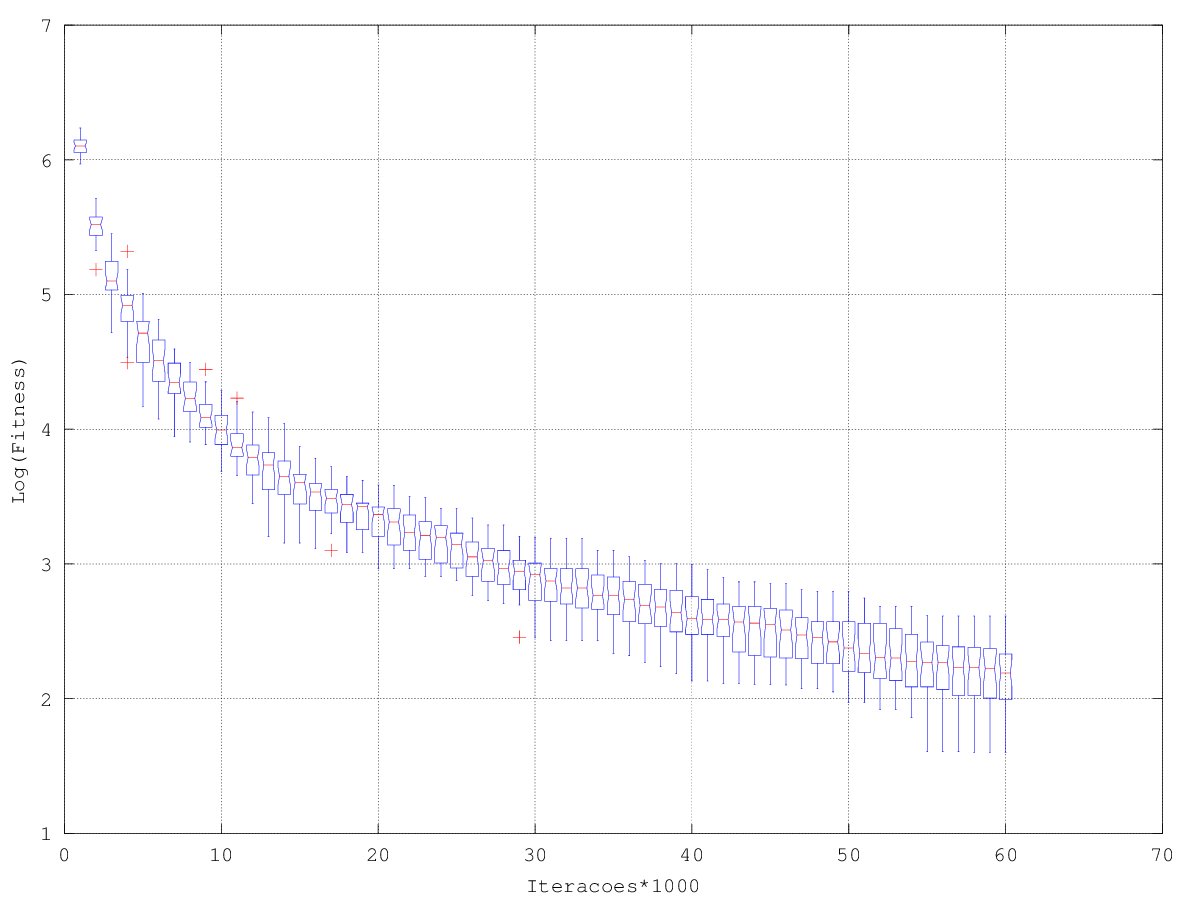
Desenvolvemos dois scripts para gerar os resultados, um que executa o experimento 30 vezes e outro que define os possíveis valores assumidos pelas variáveis. De forma que o último, executa o primeiro através da definição dos parâmetros.

Convergência do Algoritmo

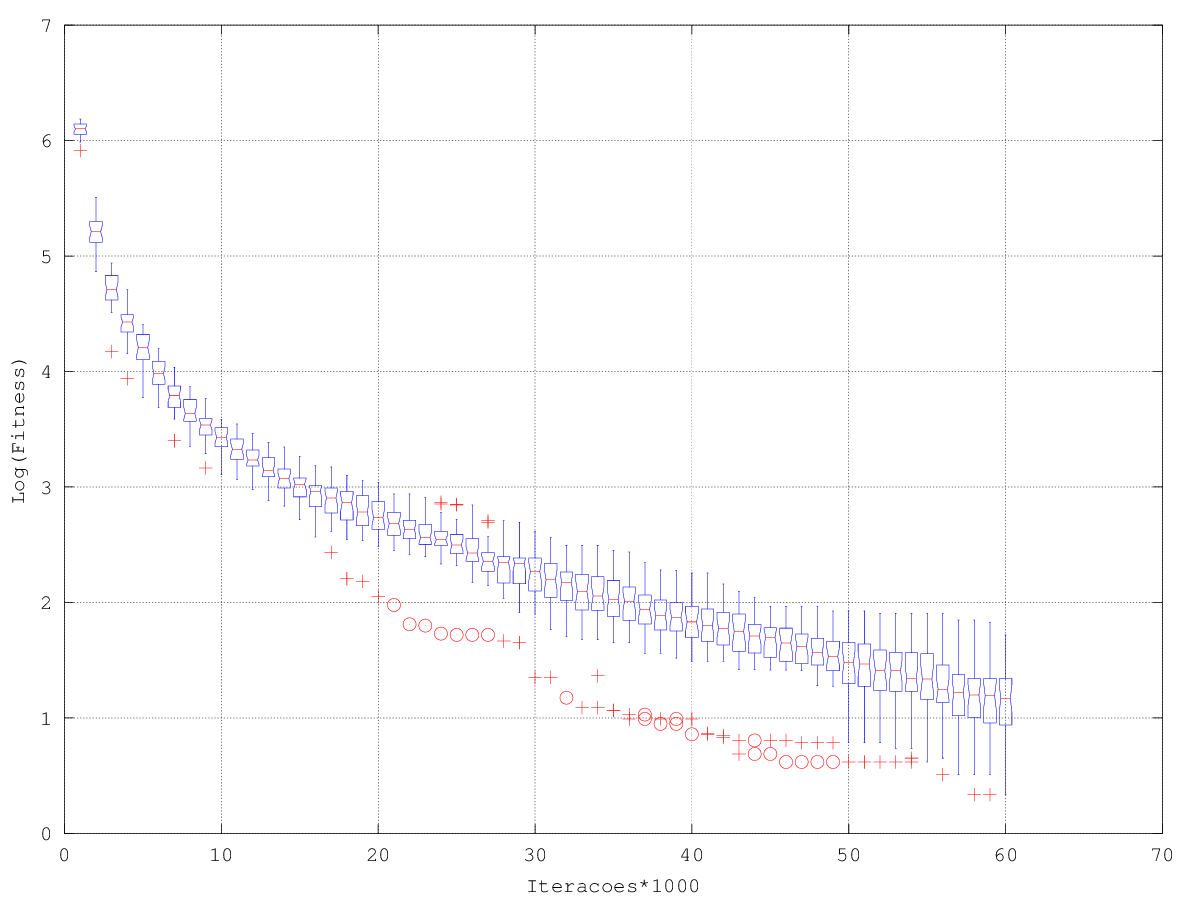
As Figuras a seguir apresentam todos os Boxplots obtidos para os 48 experimentos.



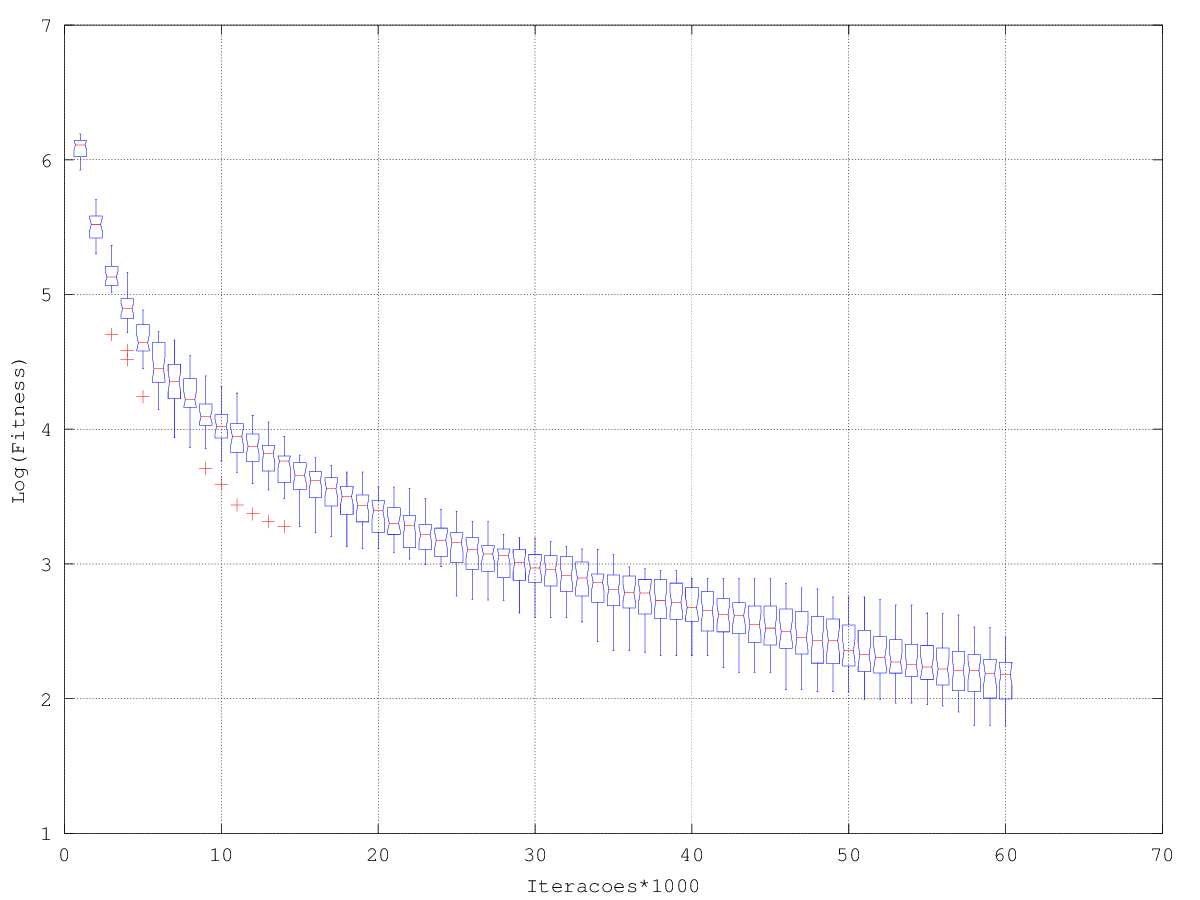
**Figura 2. Rastrigin com 2 variáveis**



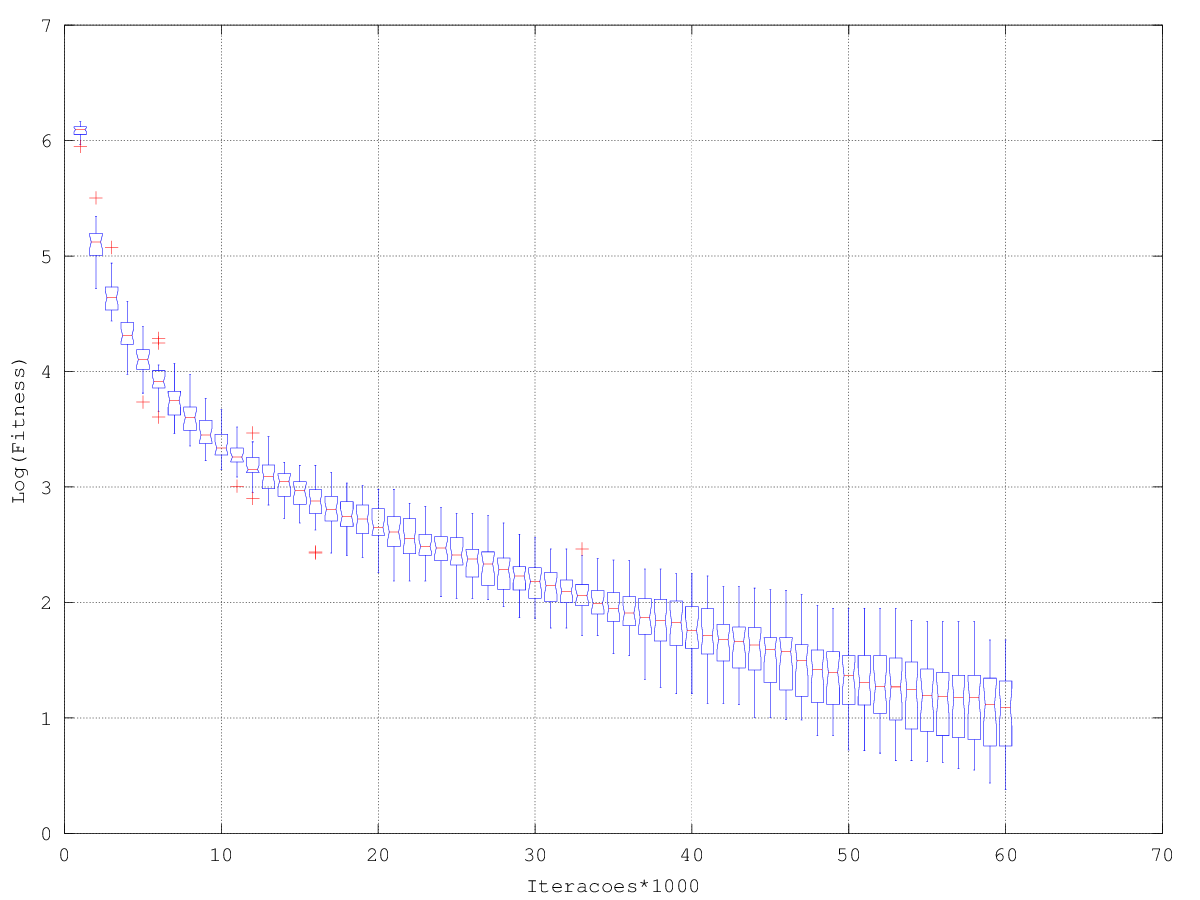
**Figura 3. Rastrigin com 2 variáveis**



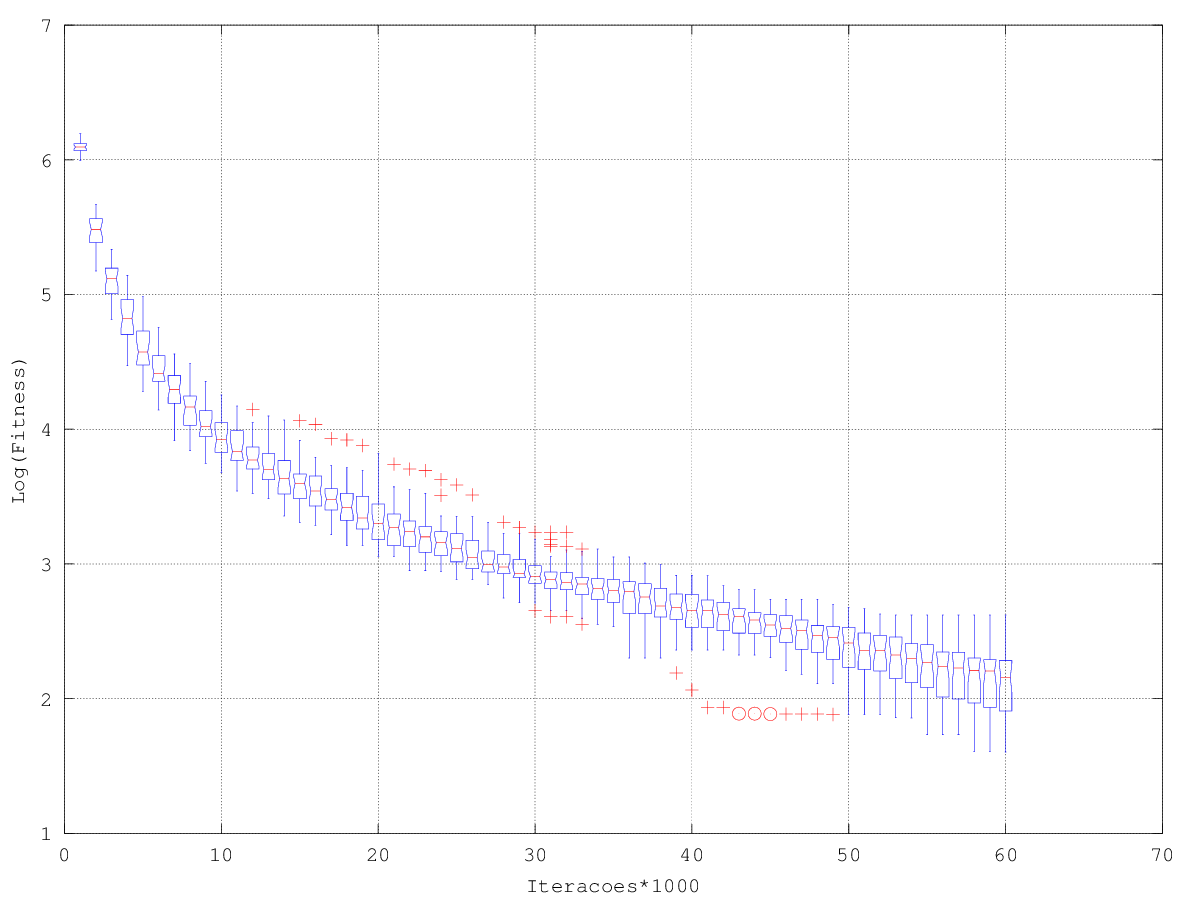
**Figura 4. Rastrigin com 2 variáveis**



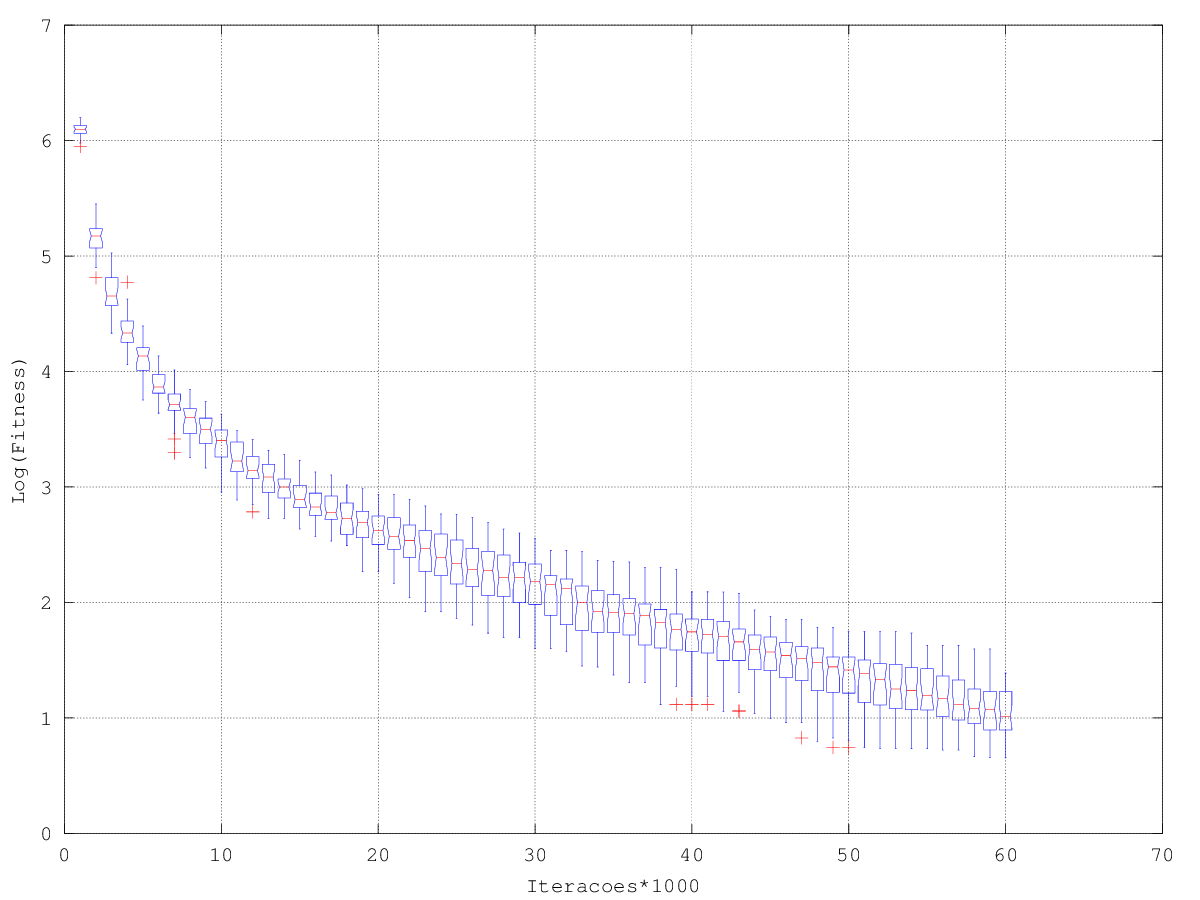
**Figura 5. Rastrigin com 2 variáveis**



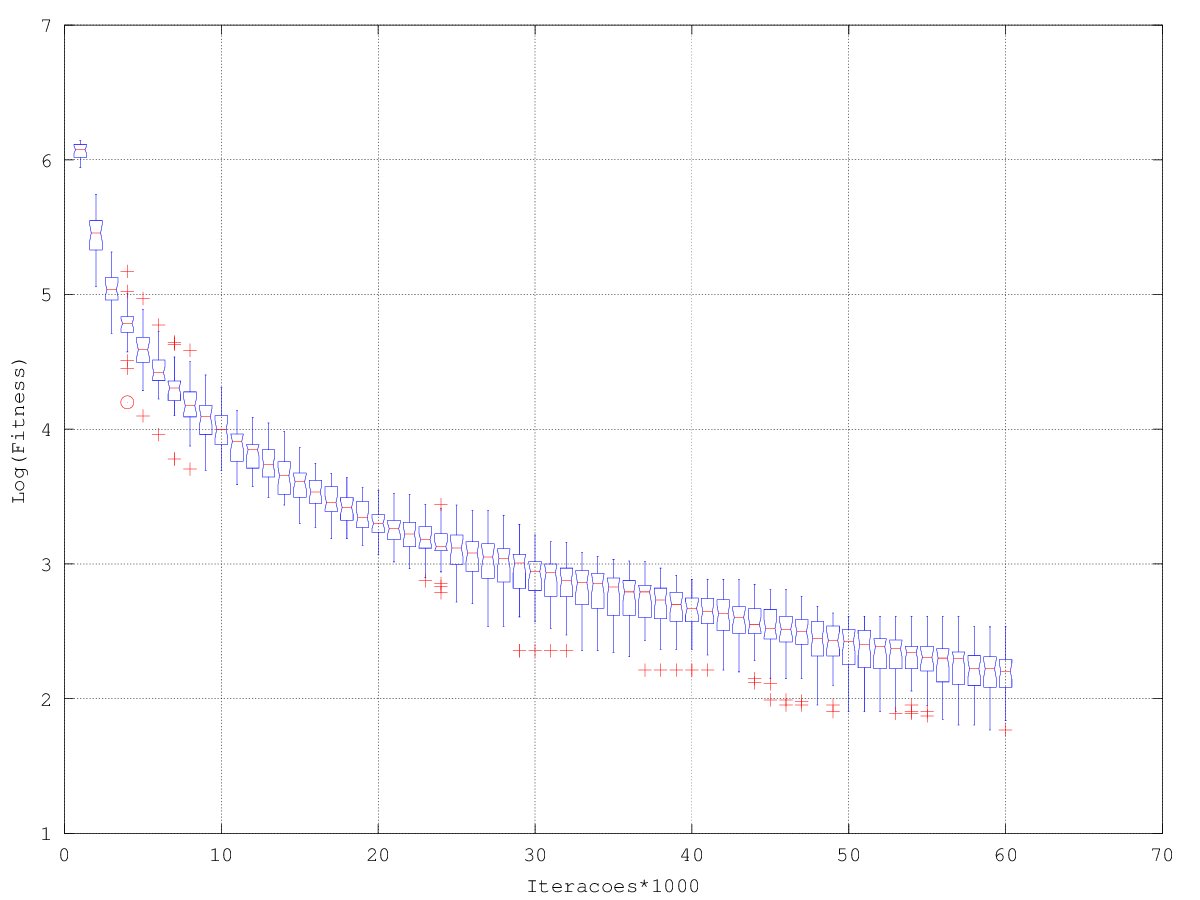
**Figura 6. Rastrigin com 2 variáveis**



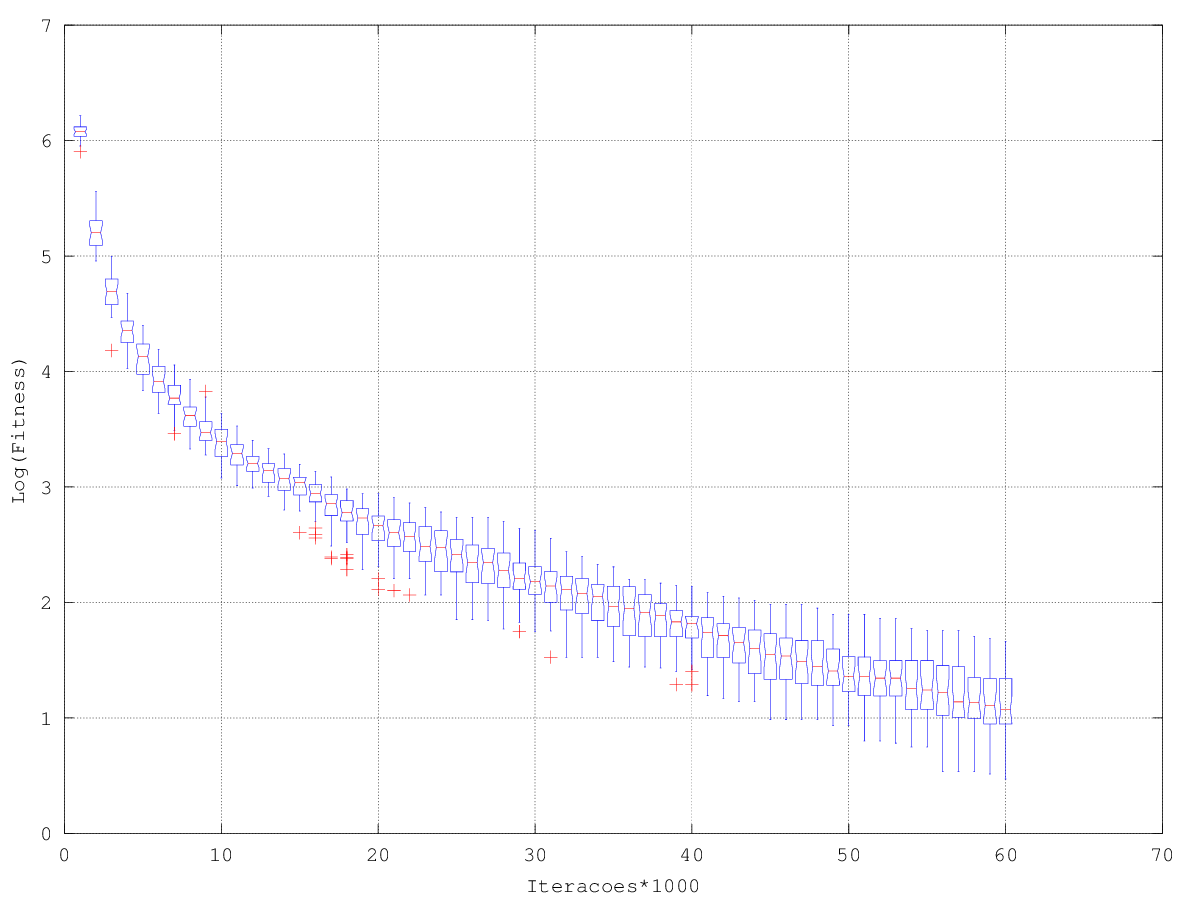
**Figura 7. Rastrigin com 2 variáveis**



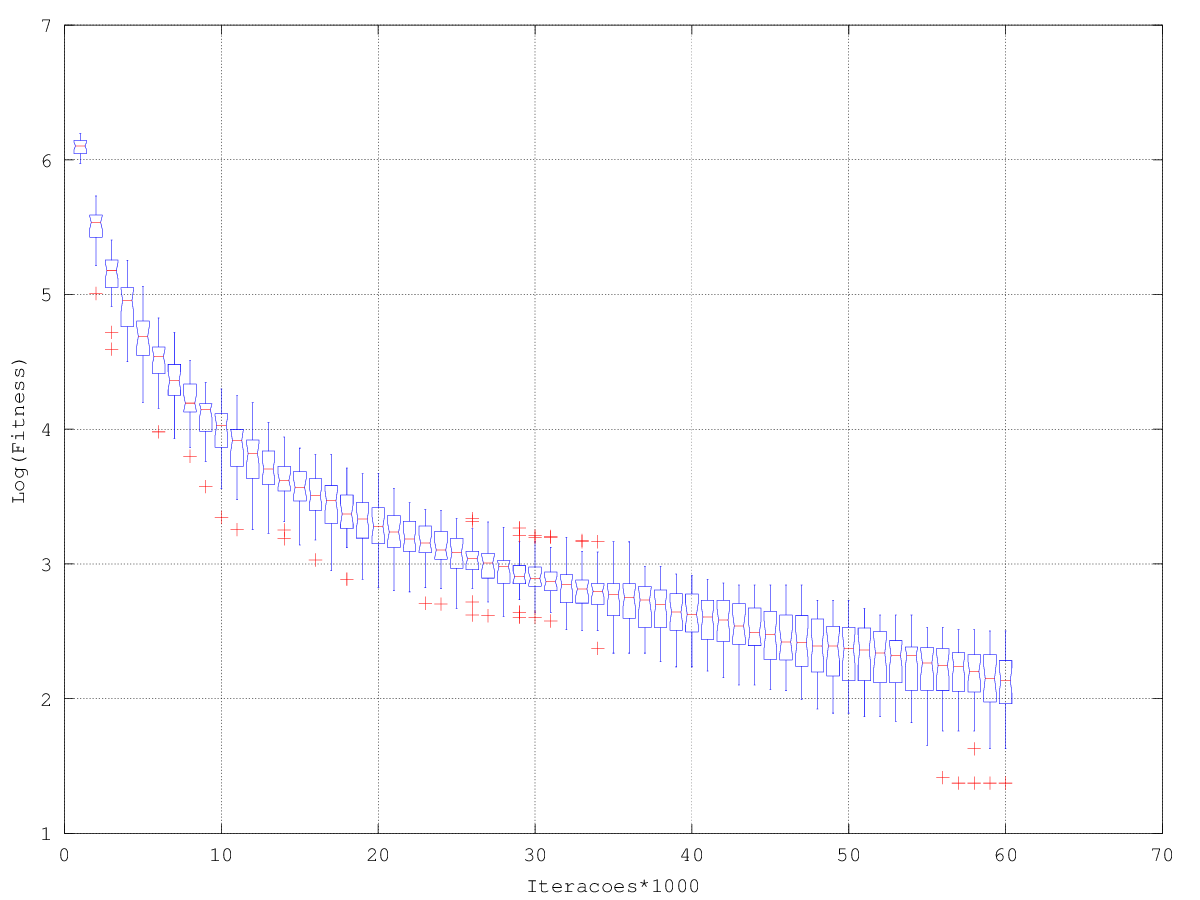
**Figura 8. Rastrigin com 2 variáveis**



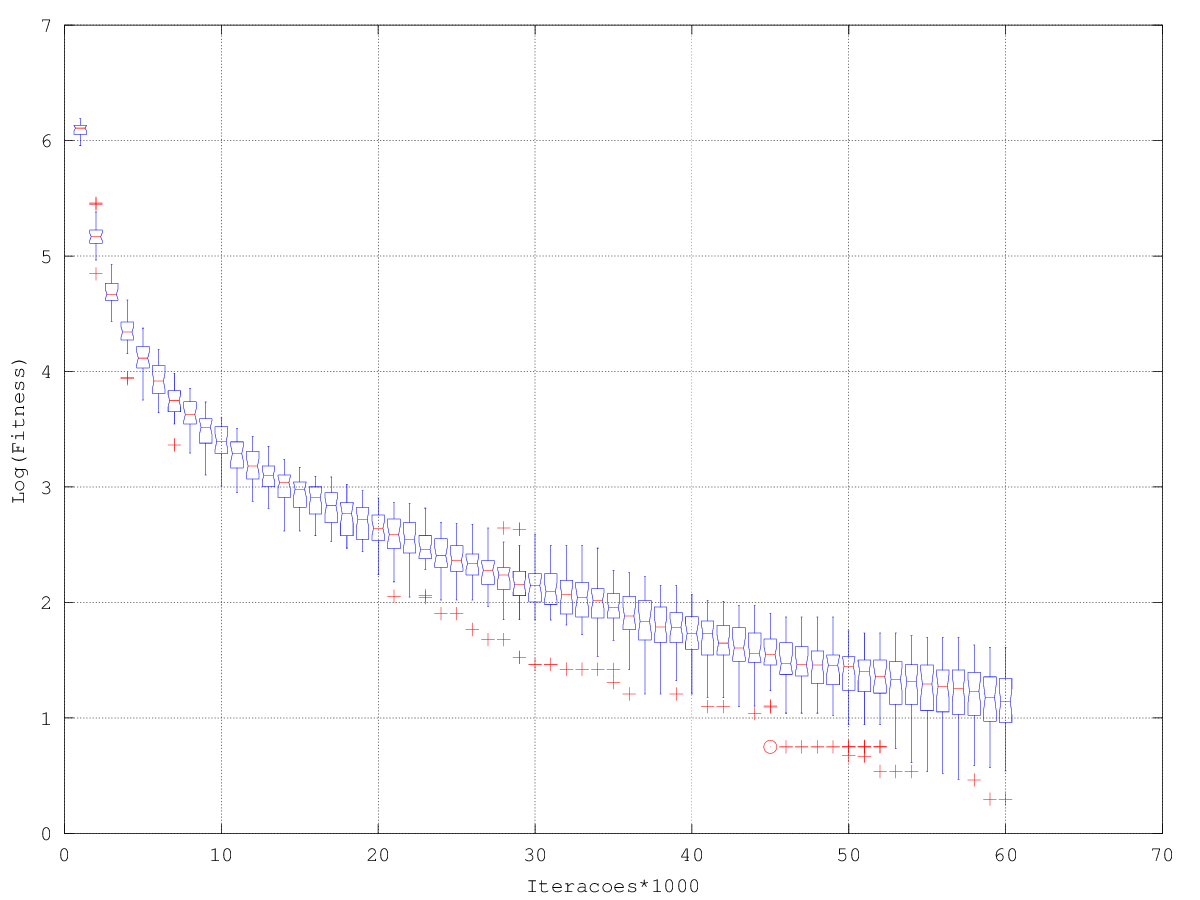
**Figura 9. Rastrigin com 2 variáveis**



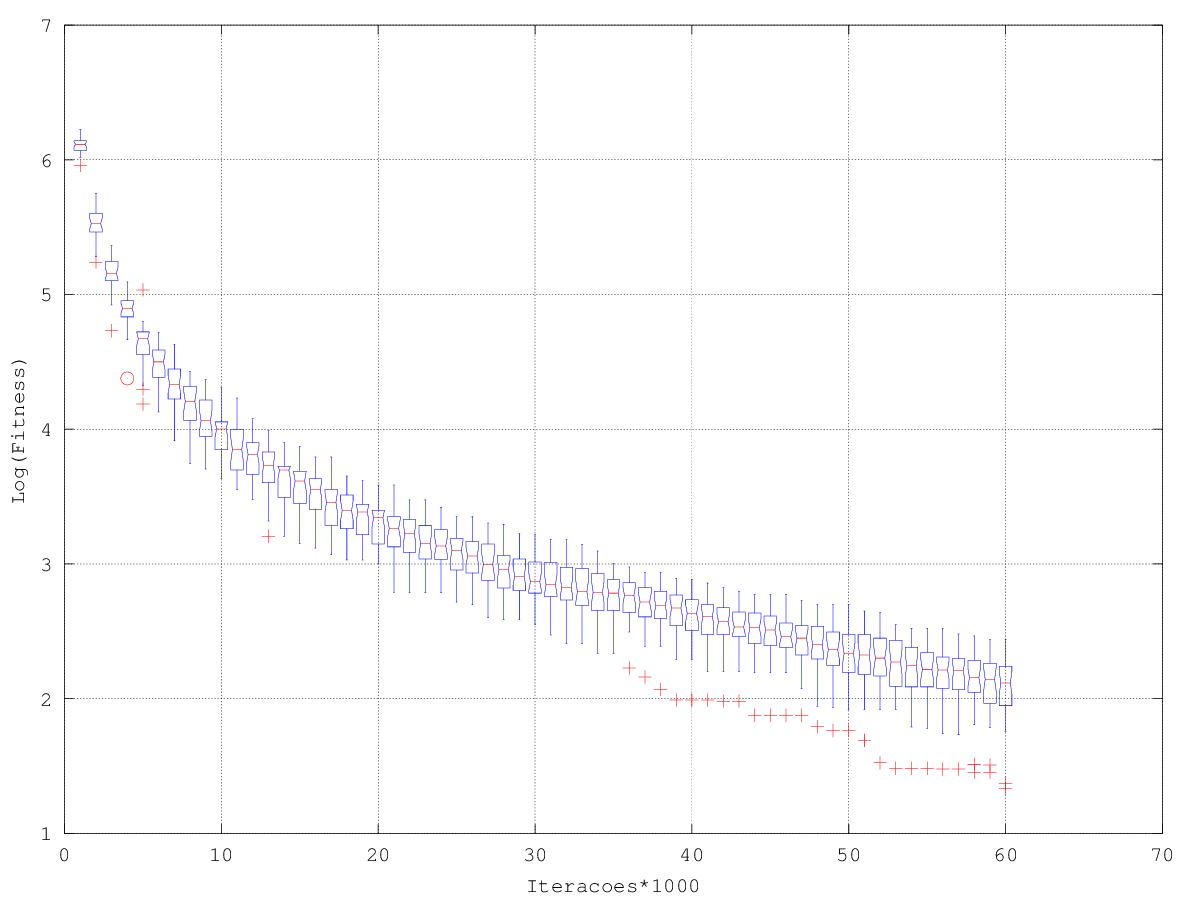
**Figura 10. Rastrigin com 2 variáveis**



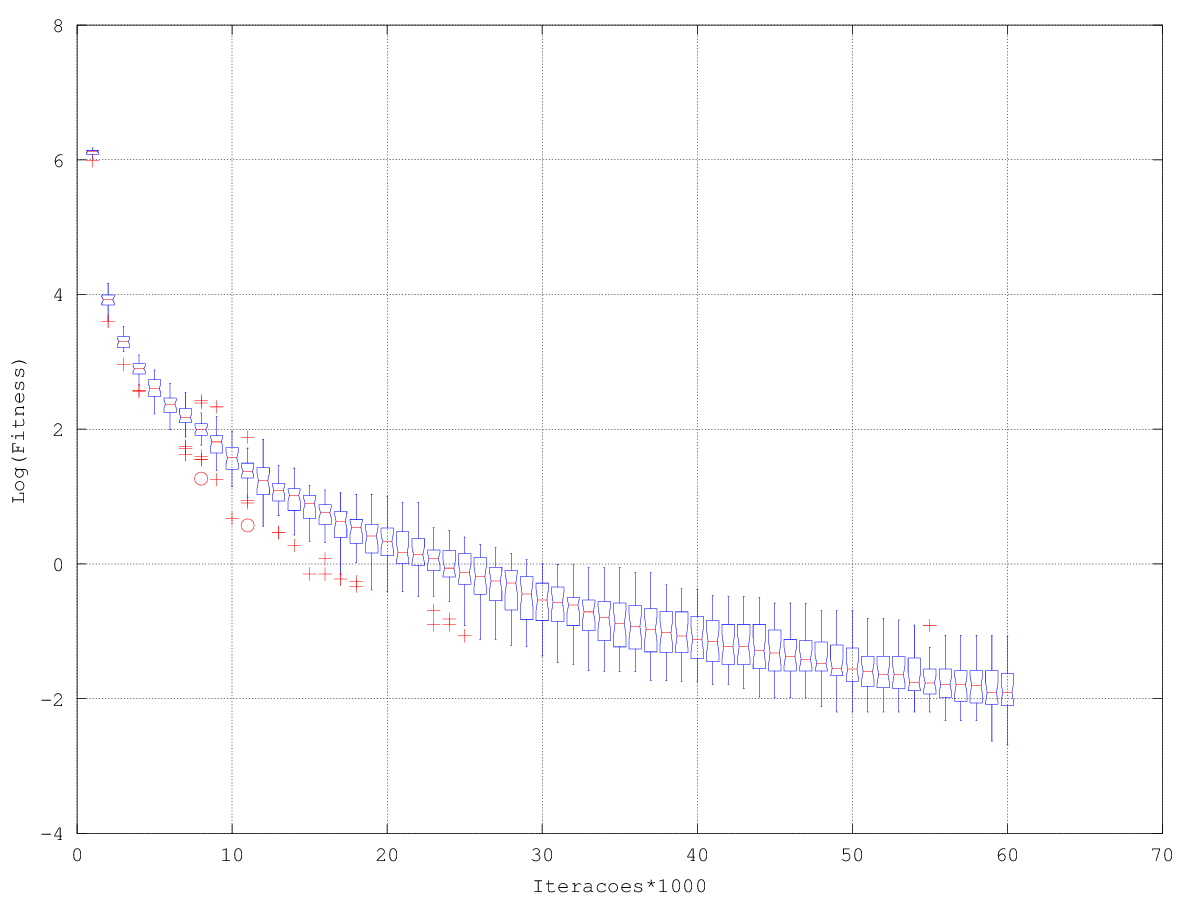
**Figura 11. Rastrigin com 2 variáveis**



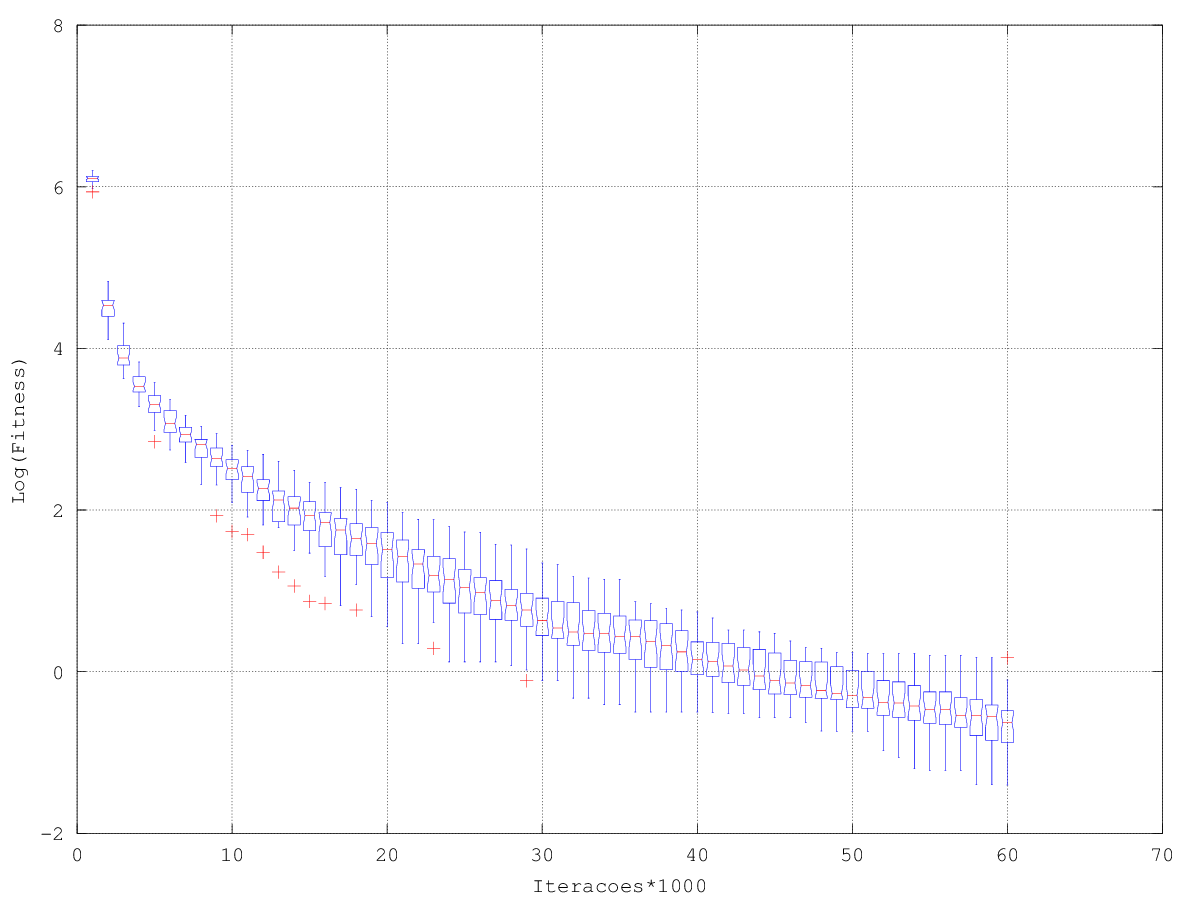
**Figura 12. Rastrigin com 2 variáveis**



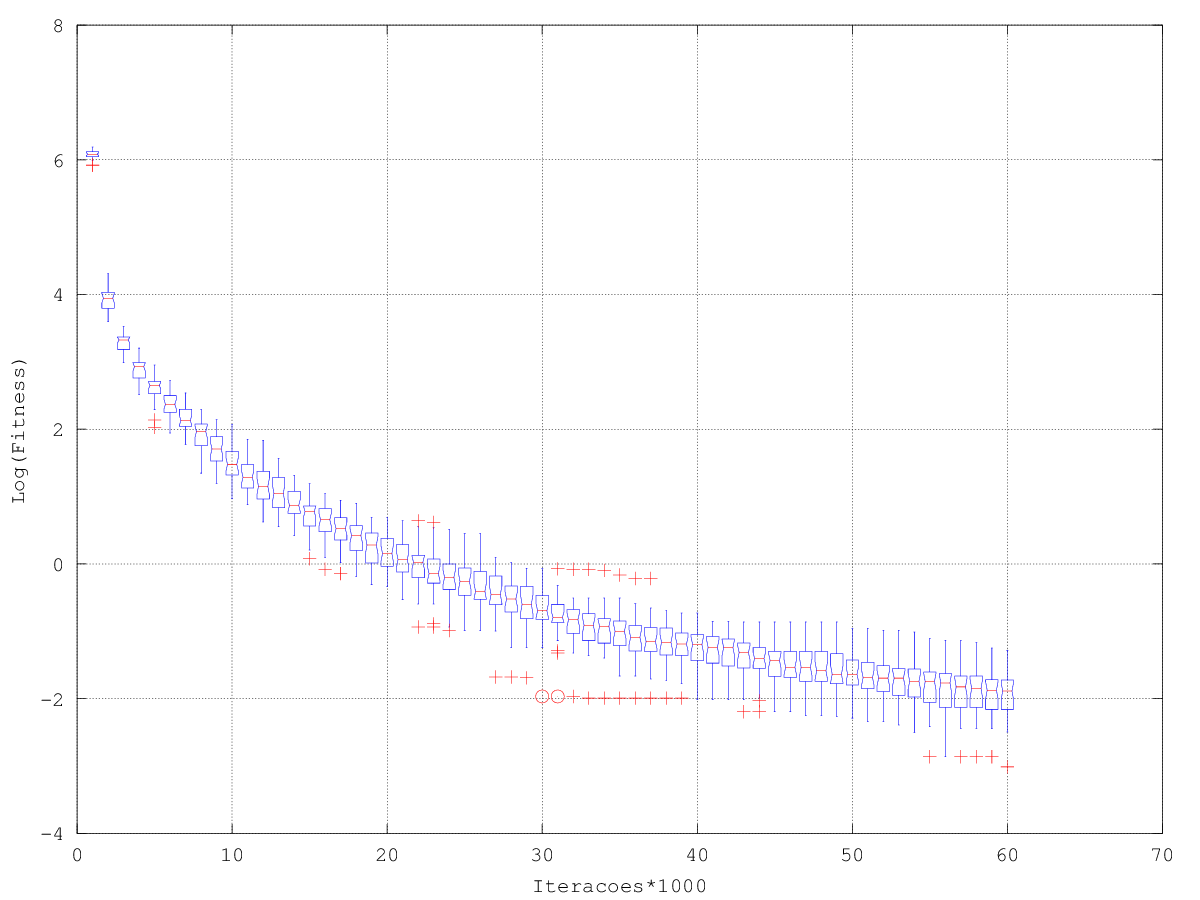
**Figura 13. Rastrigin com 2 variáveis**



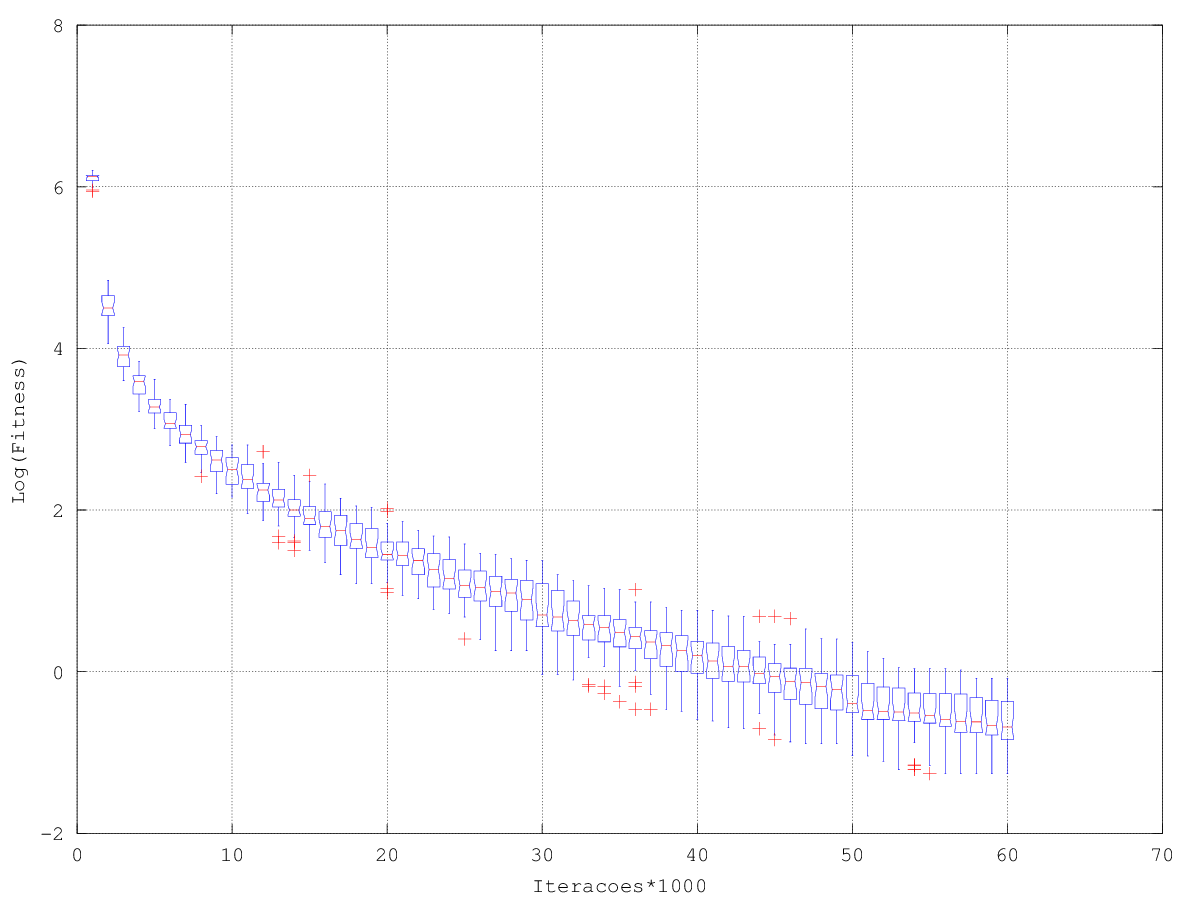
**Figura 14. Rastrigin com 2 variáveis**



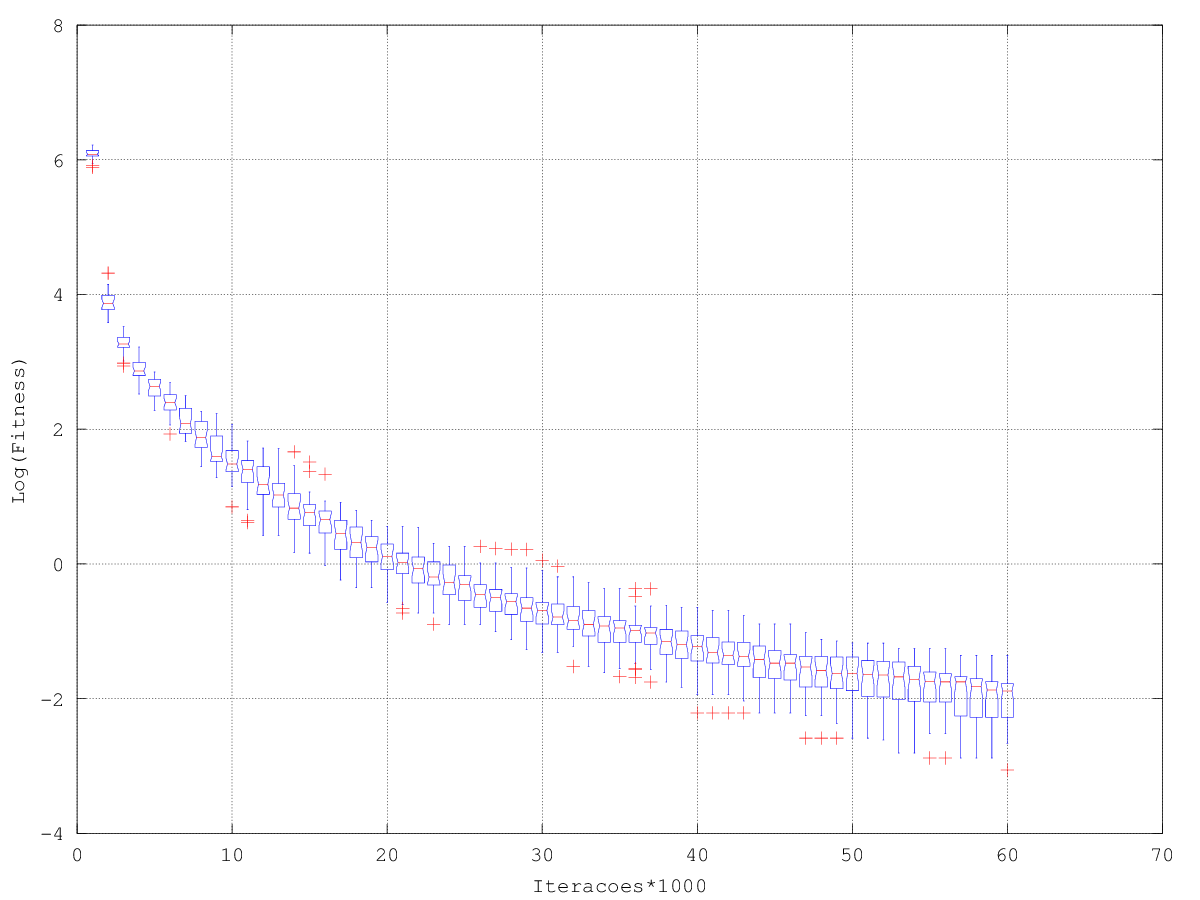
**Figura 15. Rastrigin com 2 variáveis**



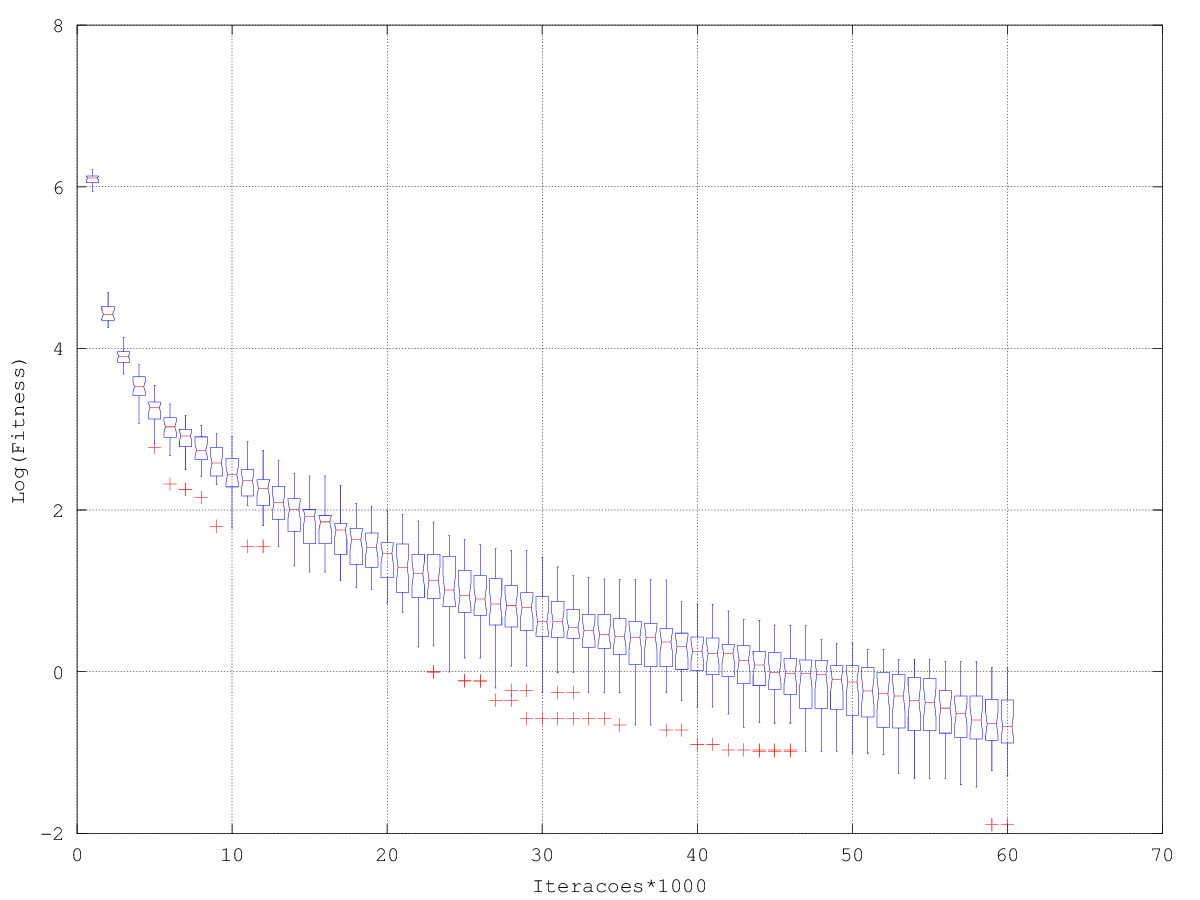
**Figura 16. Rastrigin com 2 variáveis**



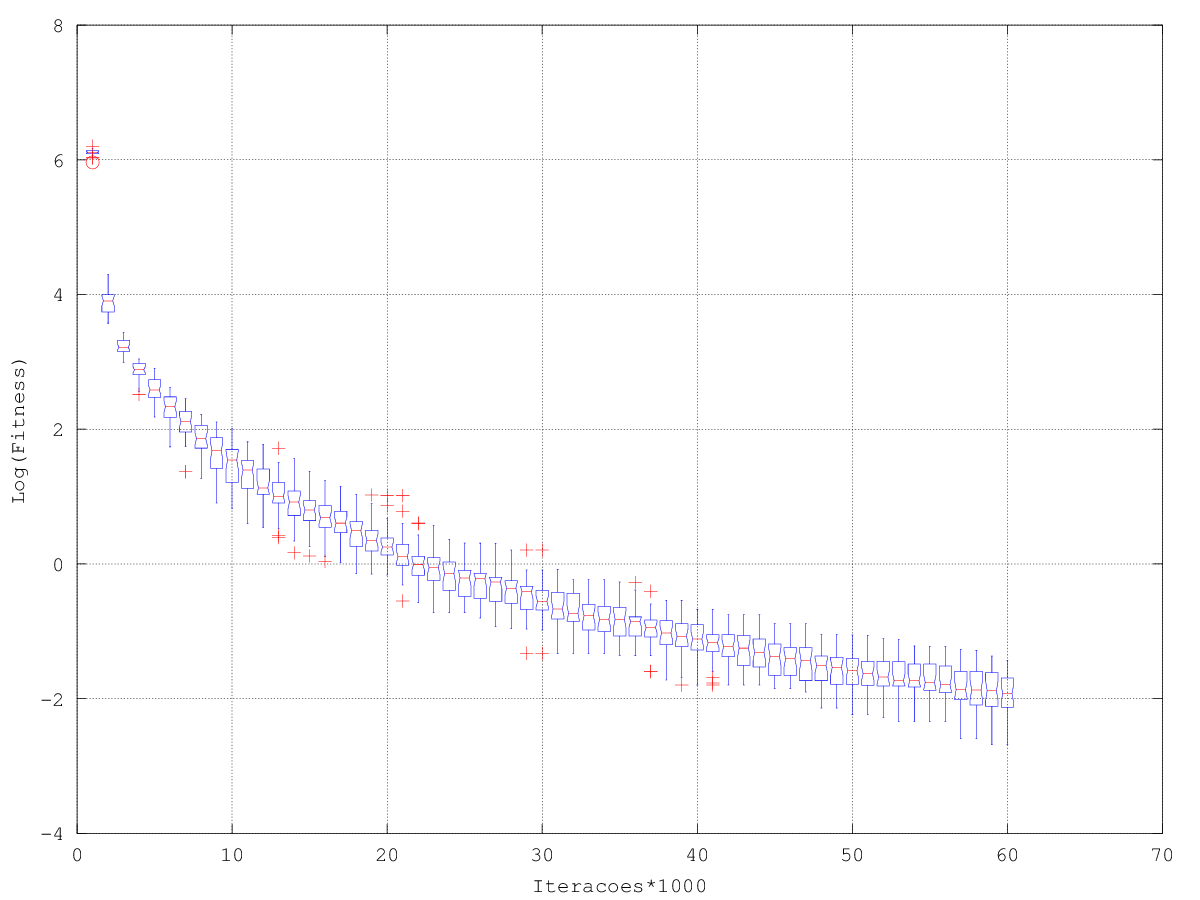
**Figura 17. Rastrigin com 2 variáveis**



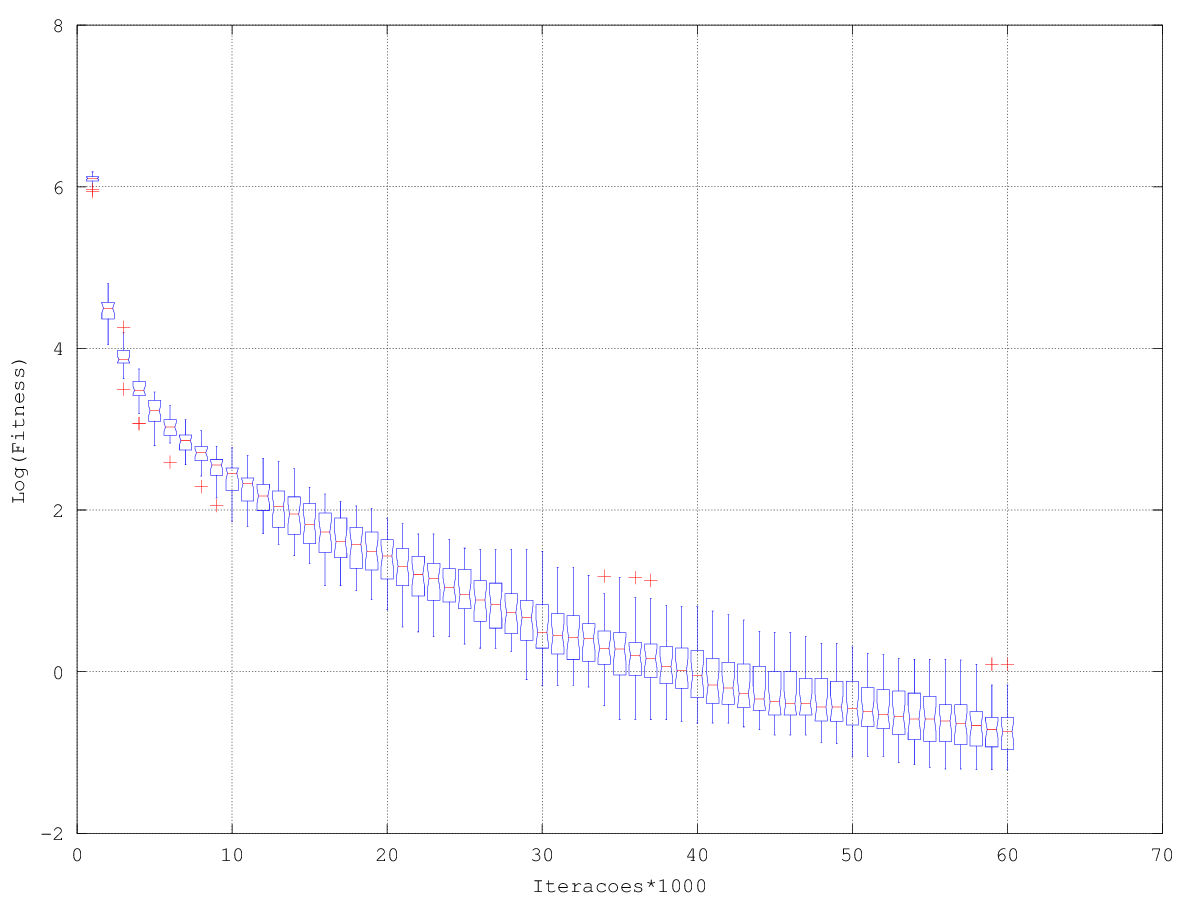
**Figura 18. Rastrigin com 2 variáveis**



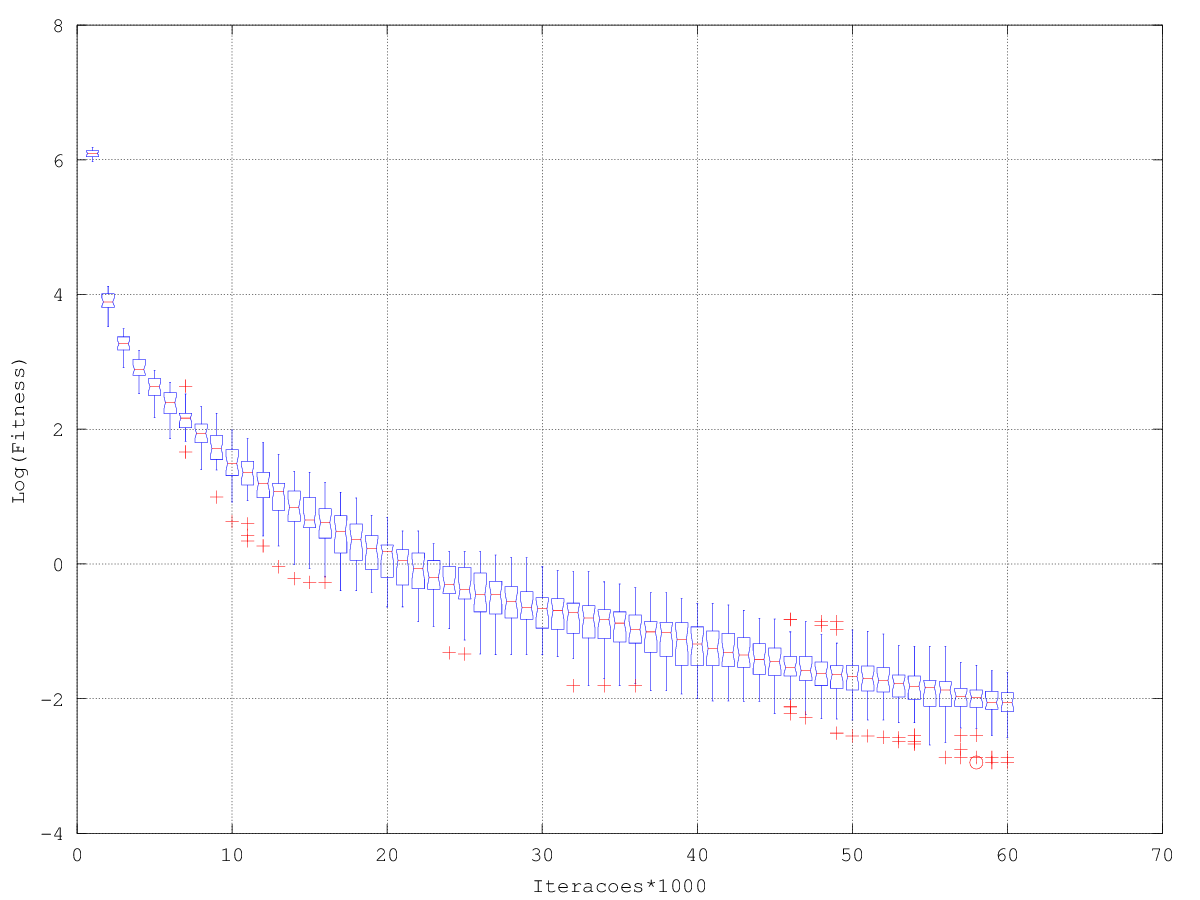
**Figura 19. Rastrigin com 2 variáveis**



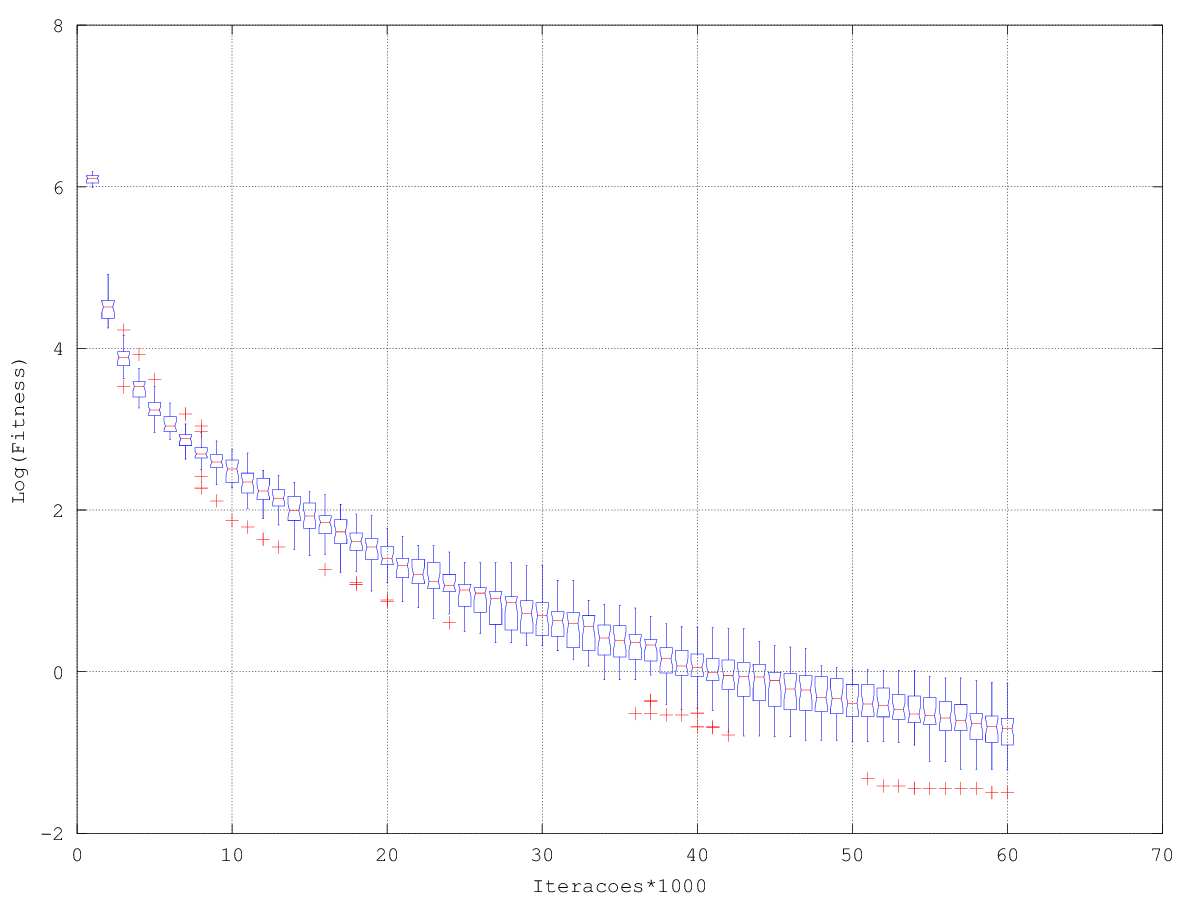
**Figura 20. Rastrigin com 2 variáveis**



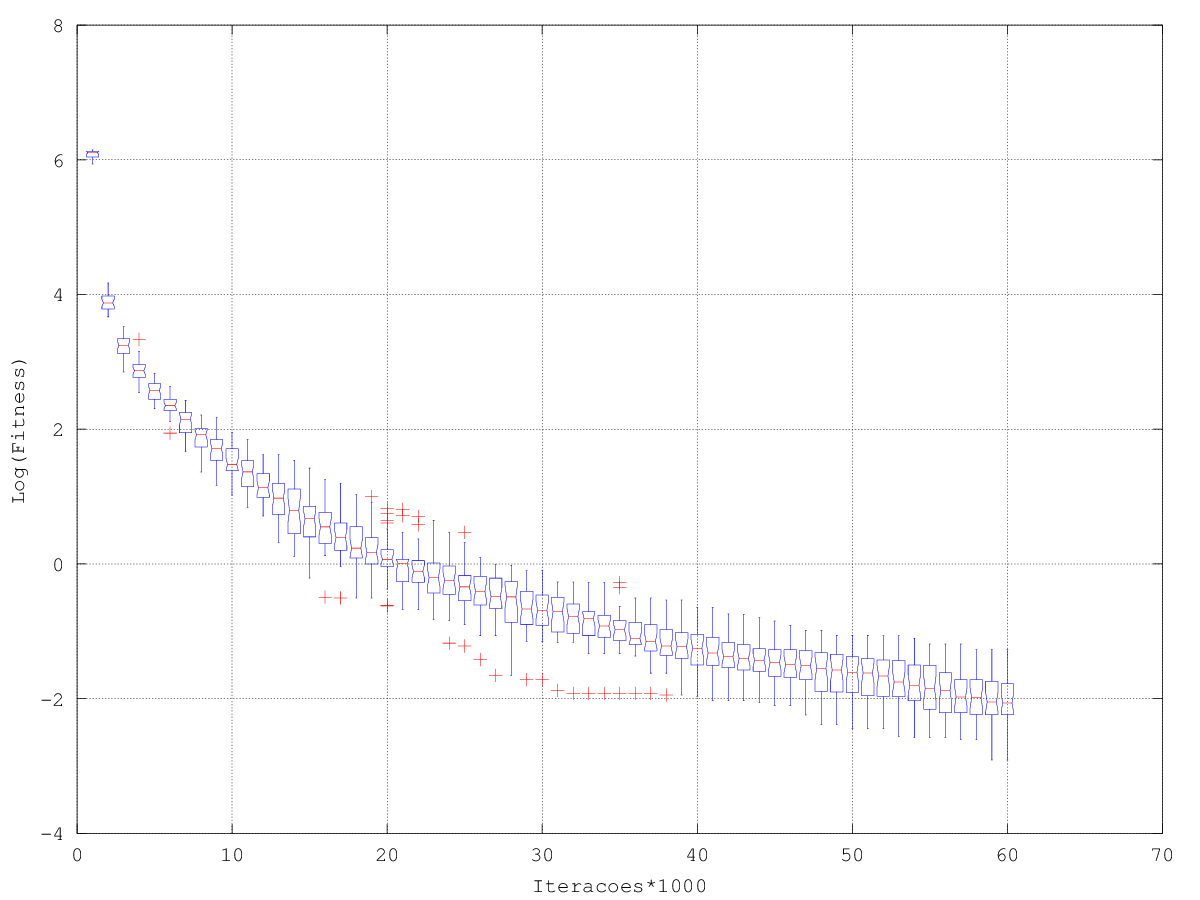
**Figura 21. Rastrigin com 2 variáveis**



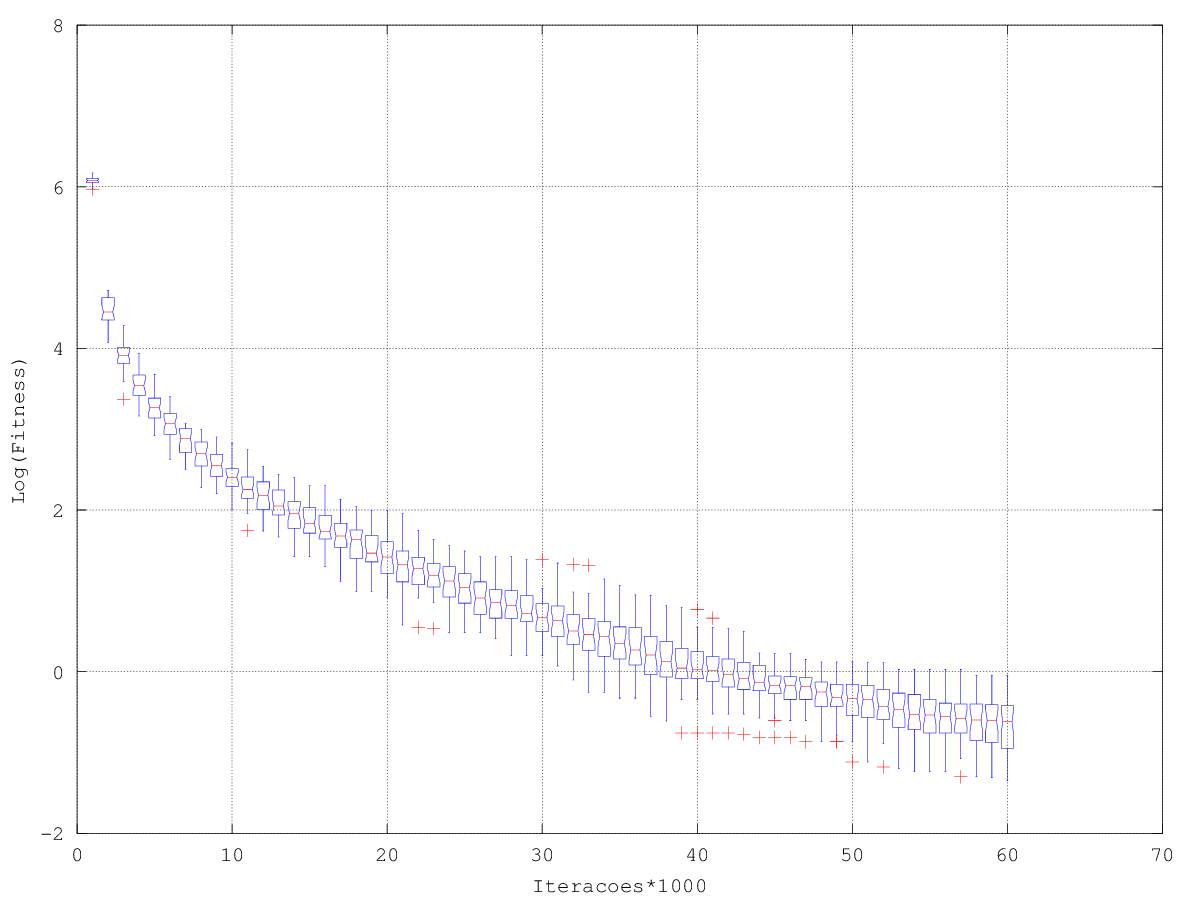
**Figura 22. Rastrigin com 2 variáveis**



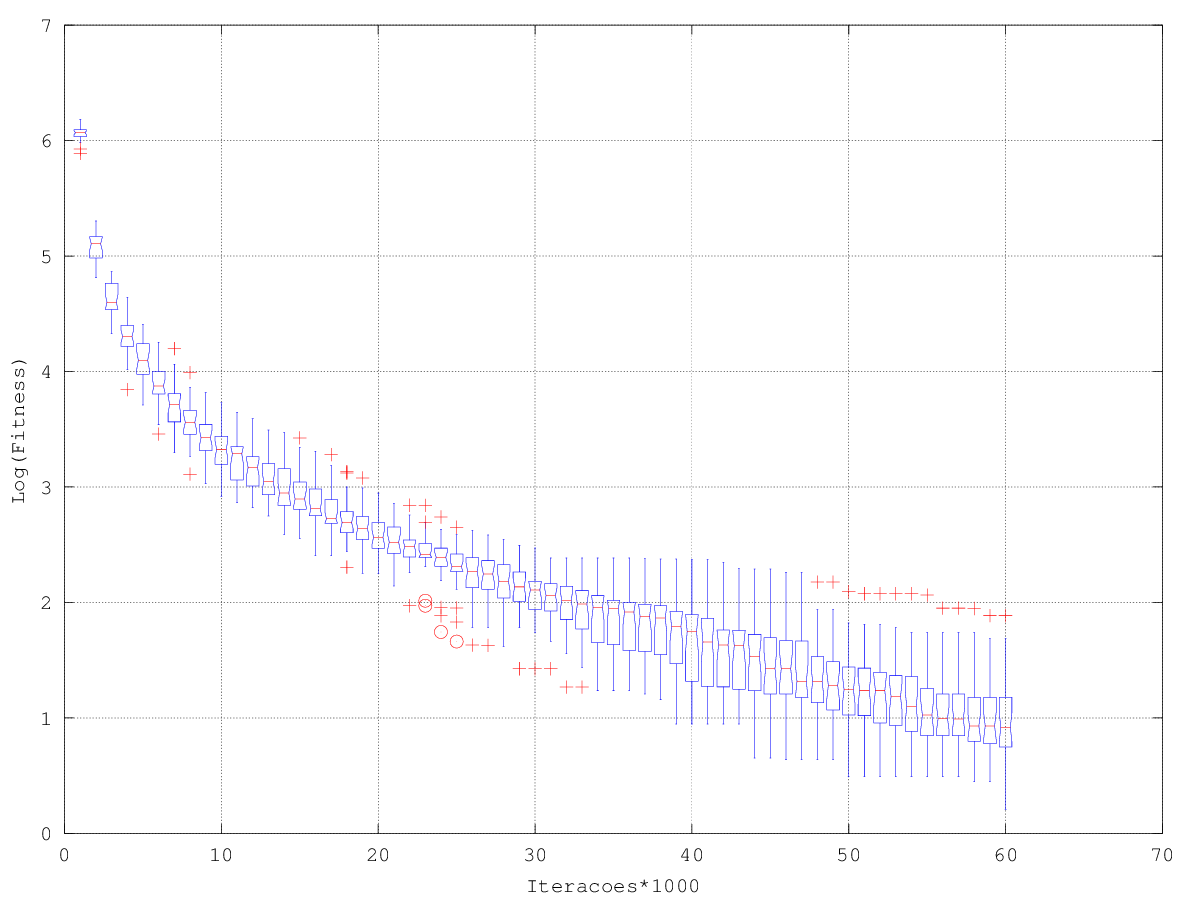
**Figura 23. Rastrigin com 2 variáveis**



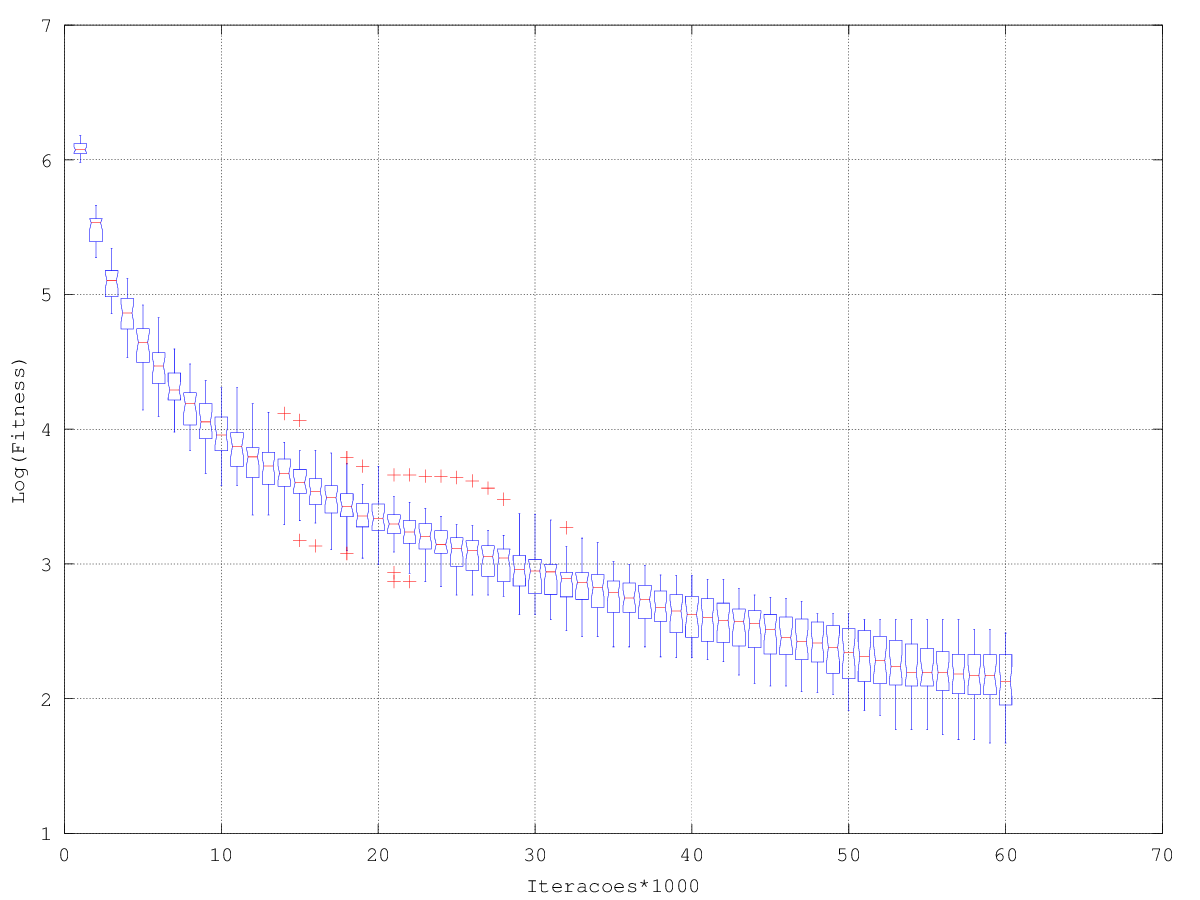
**Figura 24. Rastrigin com 2 variáveis**



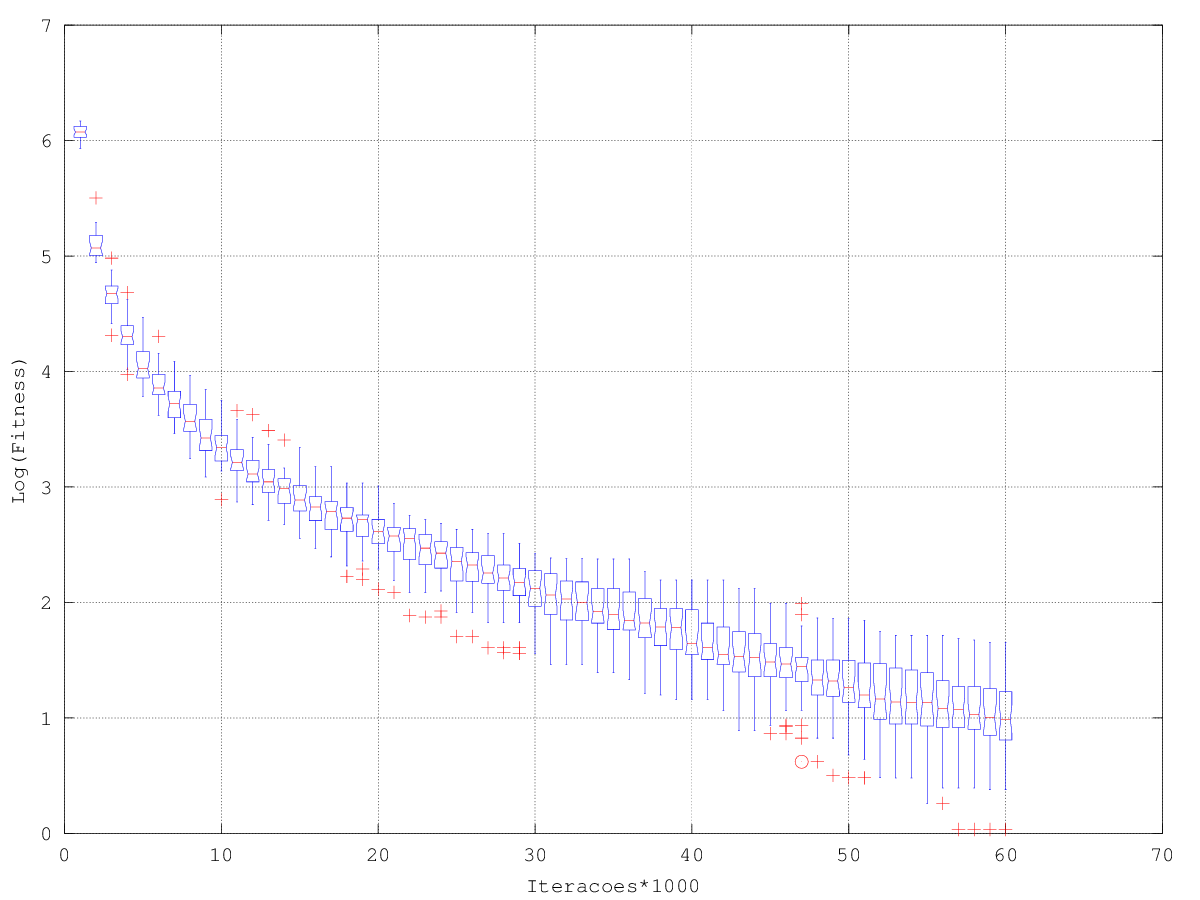
**Figura 25. Rastrigin com 2 variáveis**



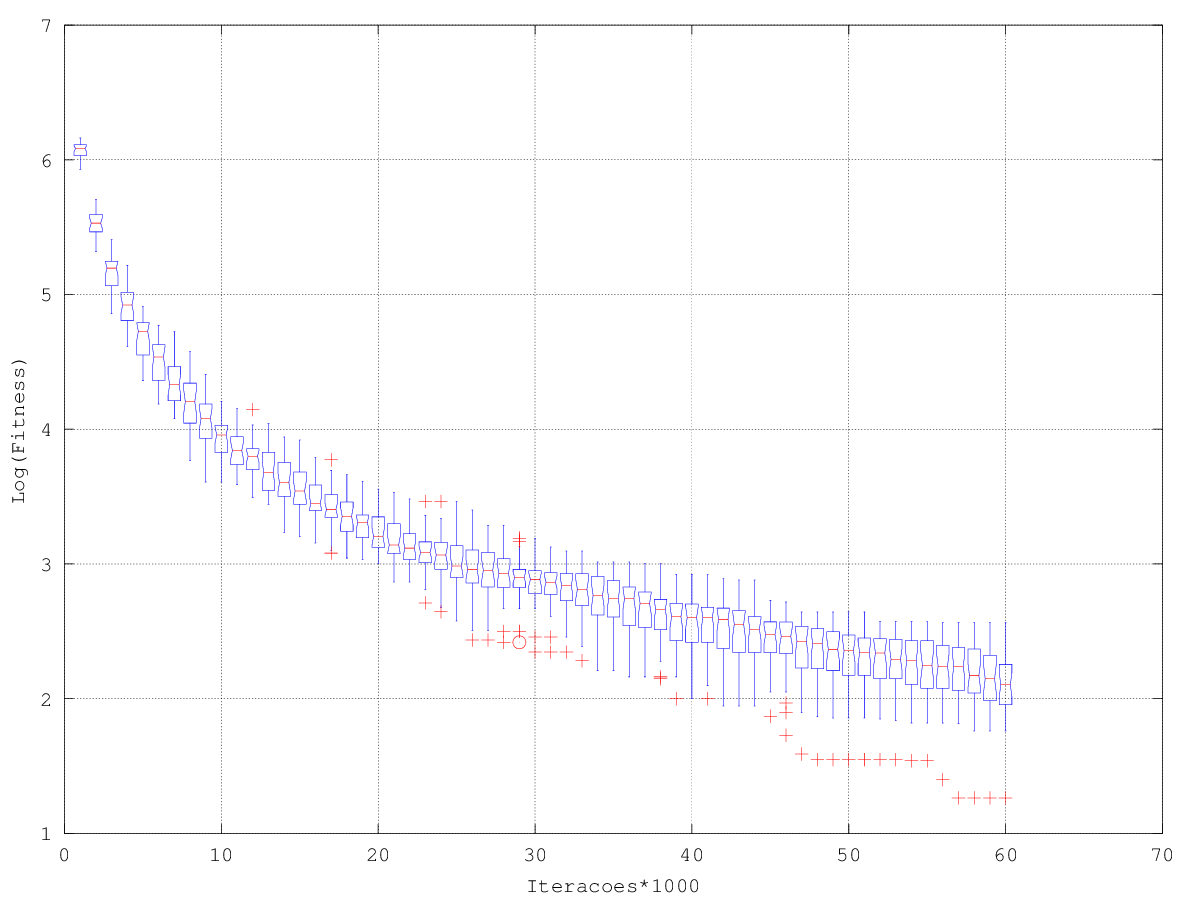
**Figura 26. Rastrigin com 2 variáveis**



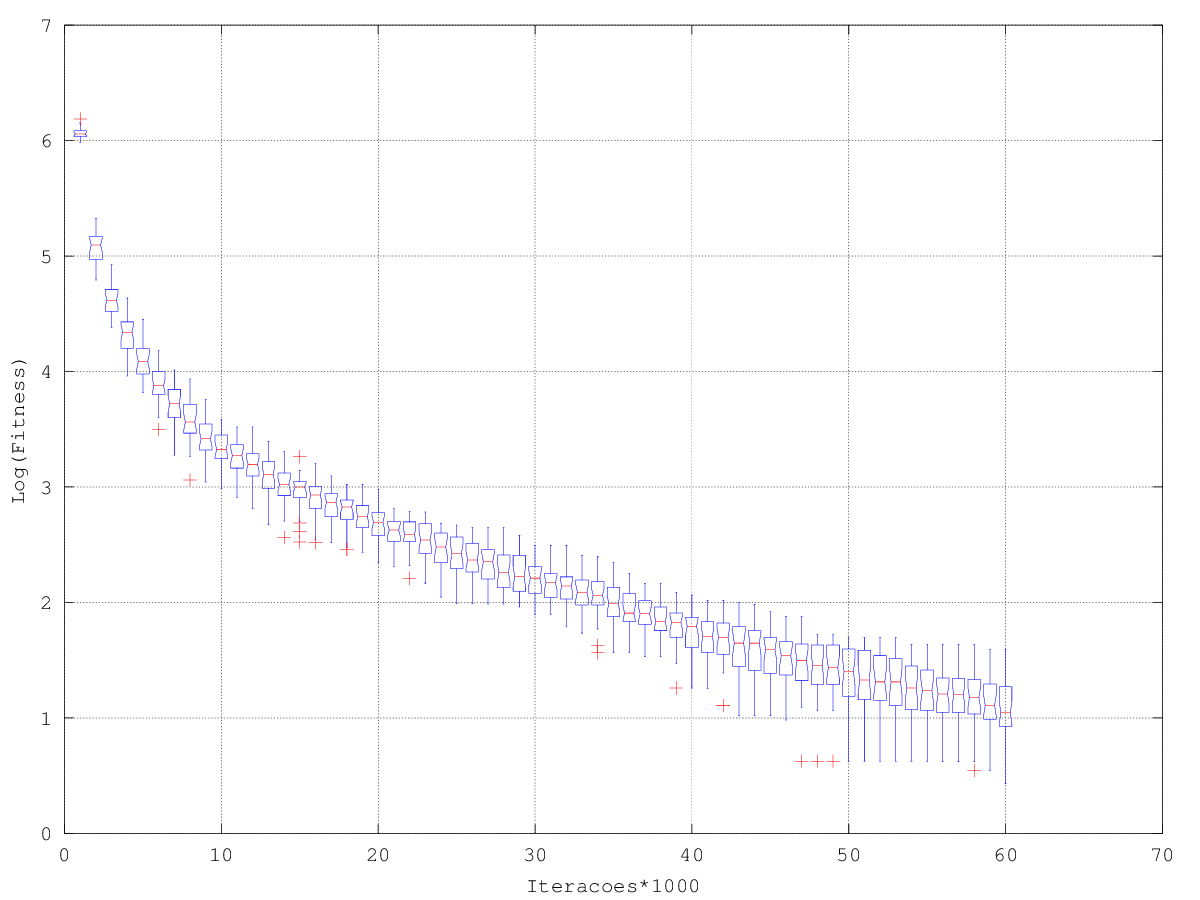
**Figura 27. Rastrigin com 2 variáveis**



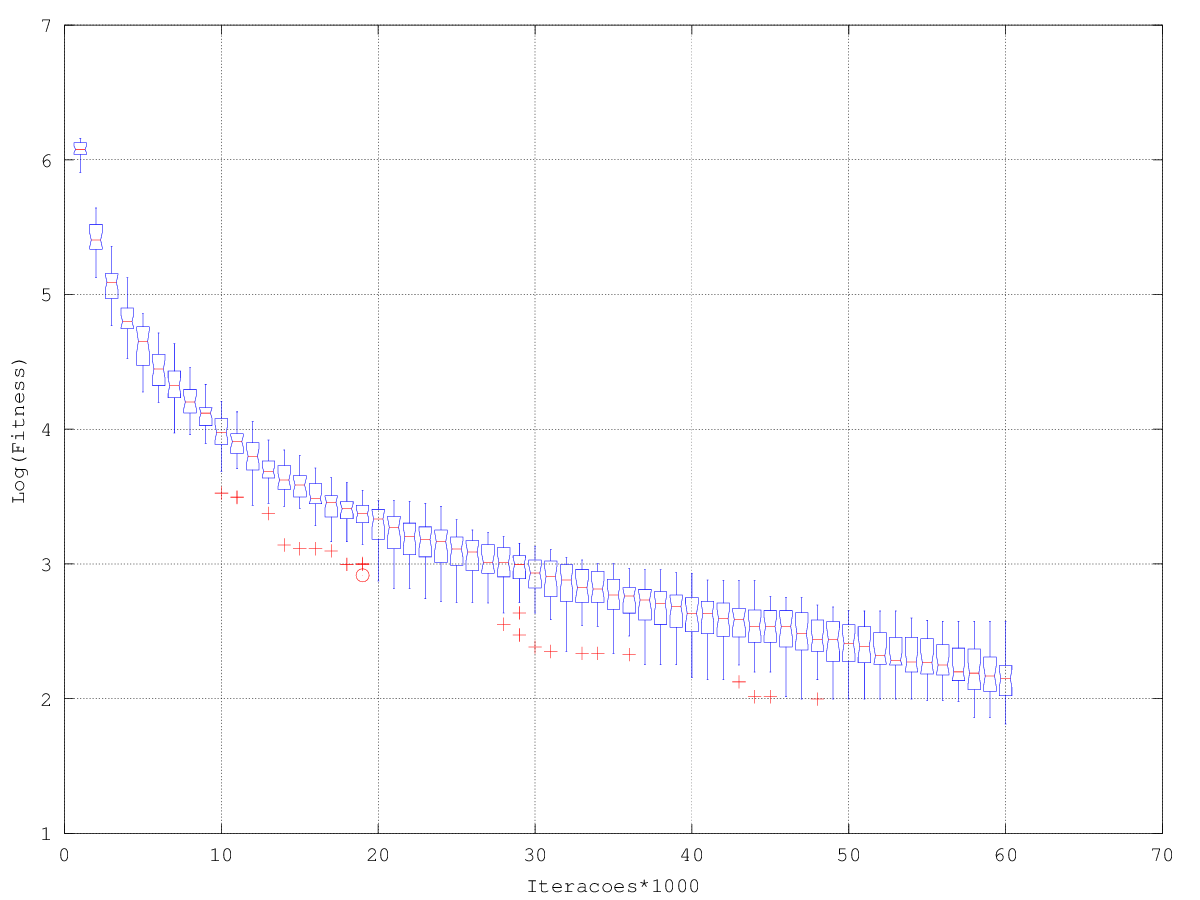
**Figura 28. Rastrigin com 2 variáveis**



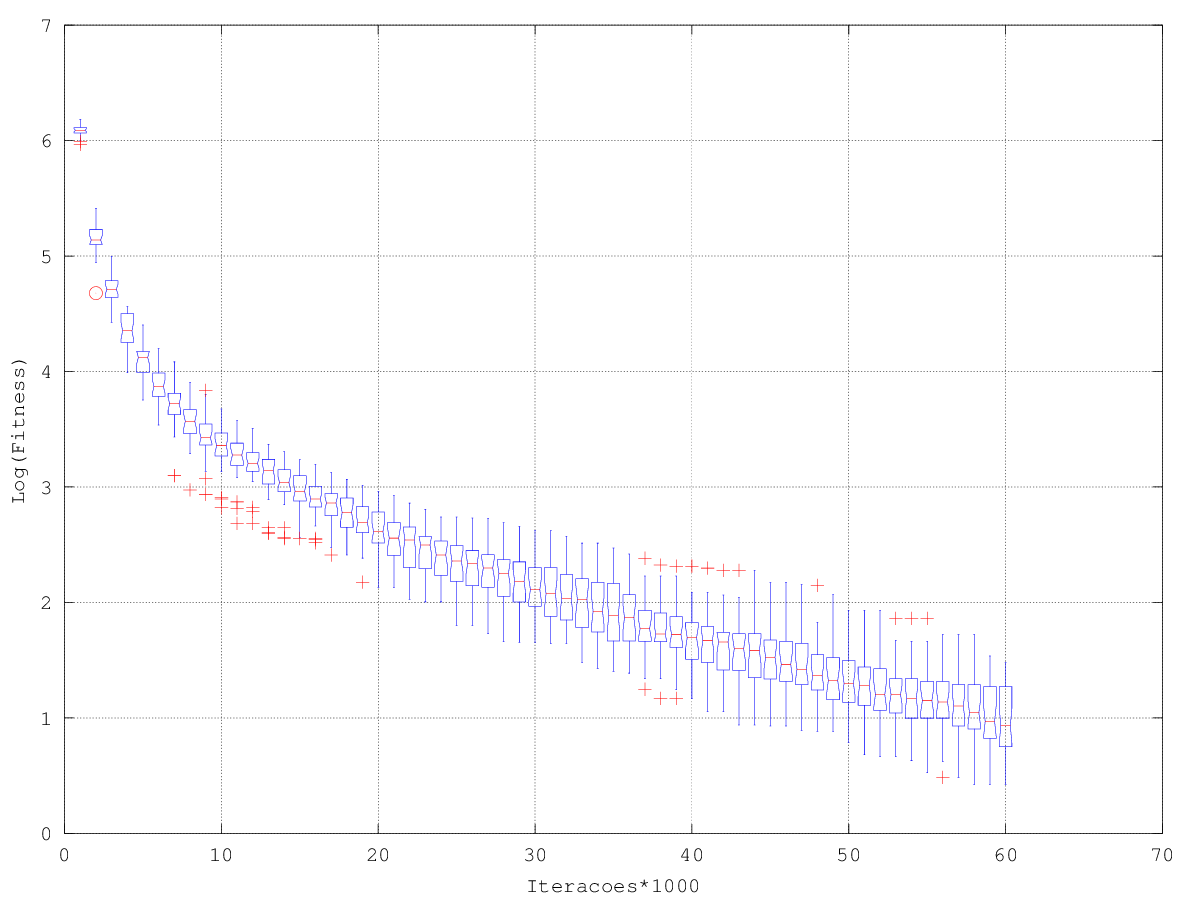
**Figura 29. Rastrigin com 2 variáveis**



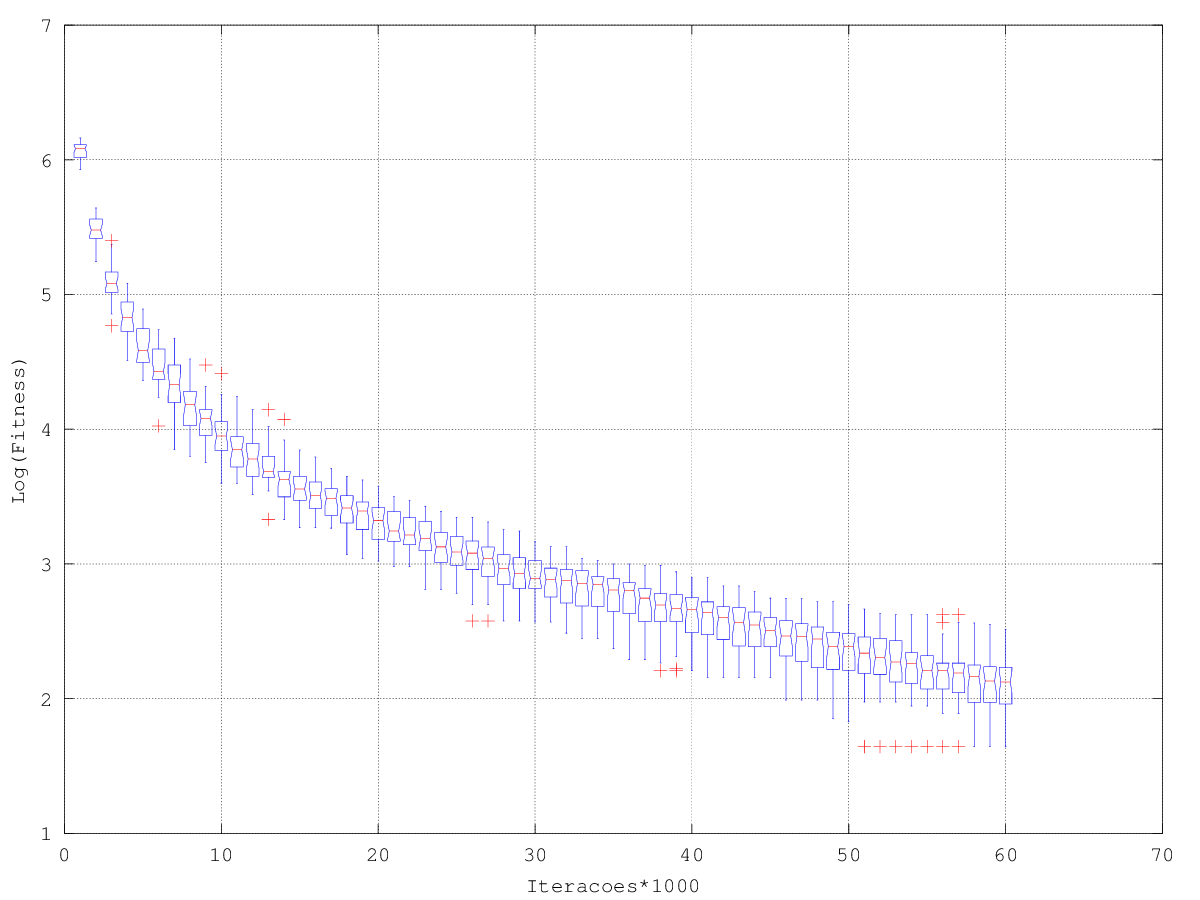
**Figura 30. Rastrigin com 2 variáveis**



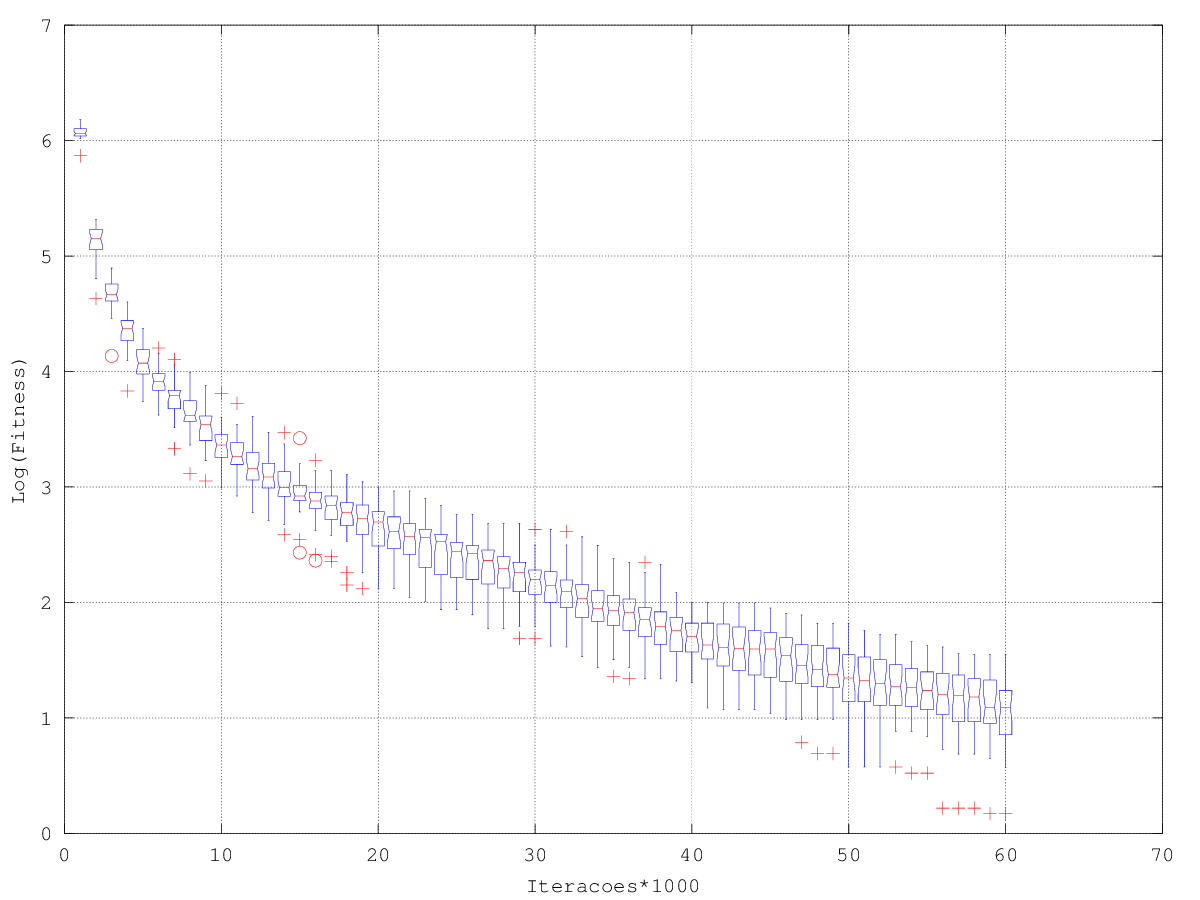
**Figura 31. Rastrigin com 2 variáveis**



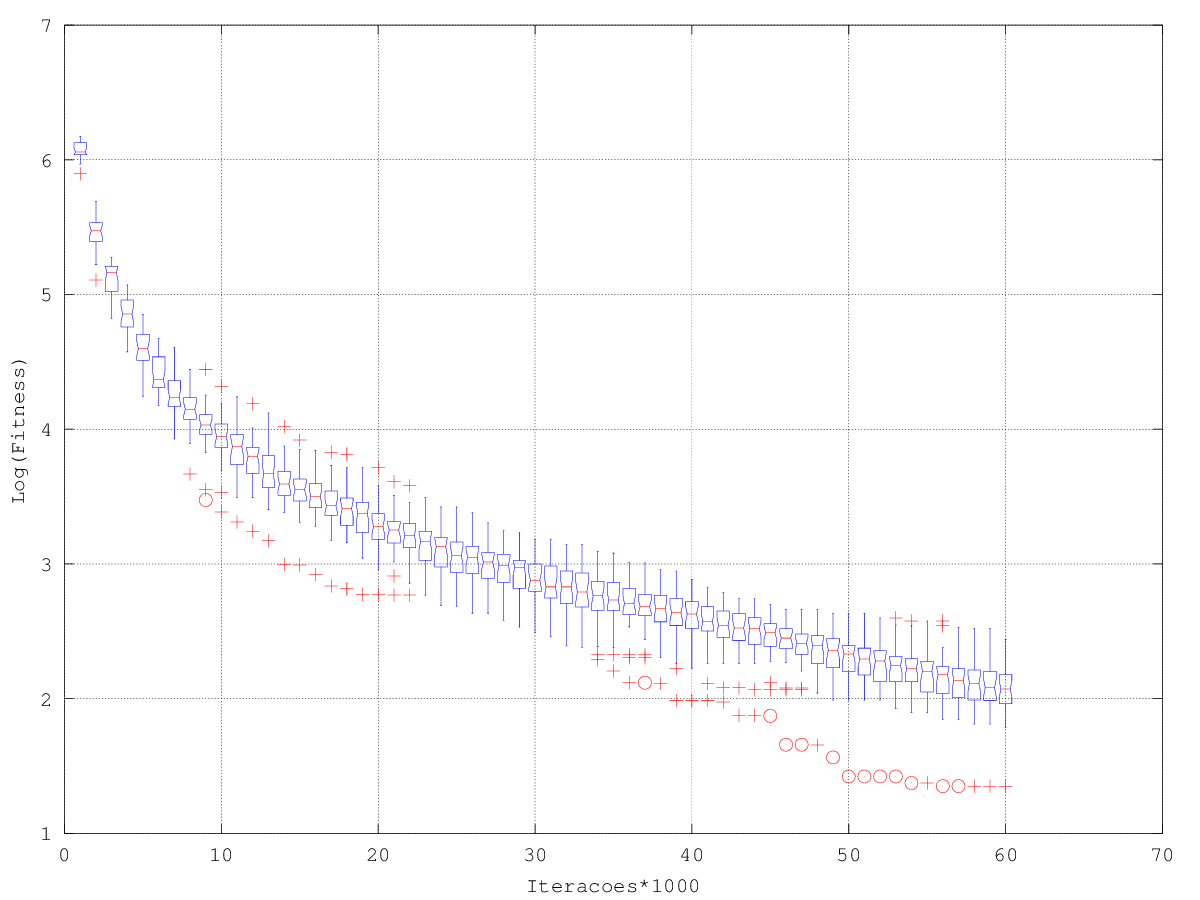
**Figura 32. Rastrigin com 2 variáveis**



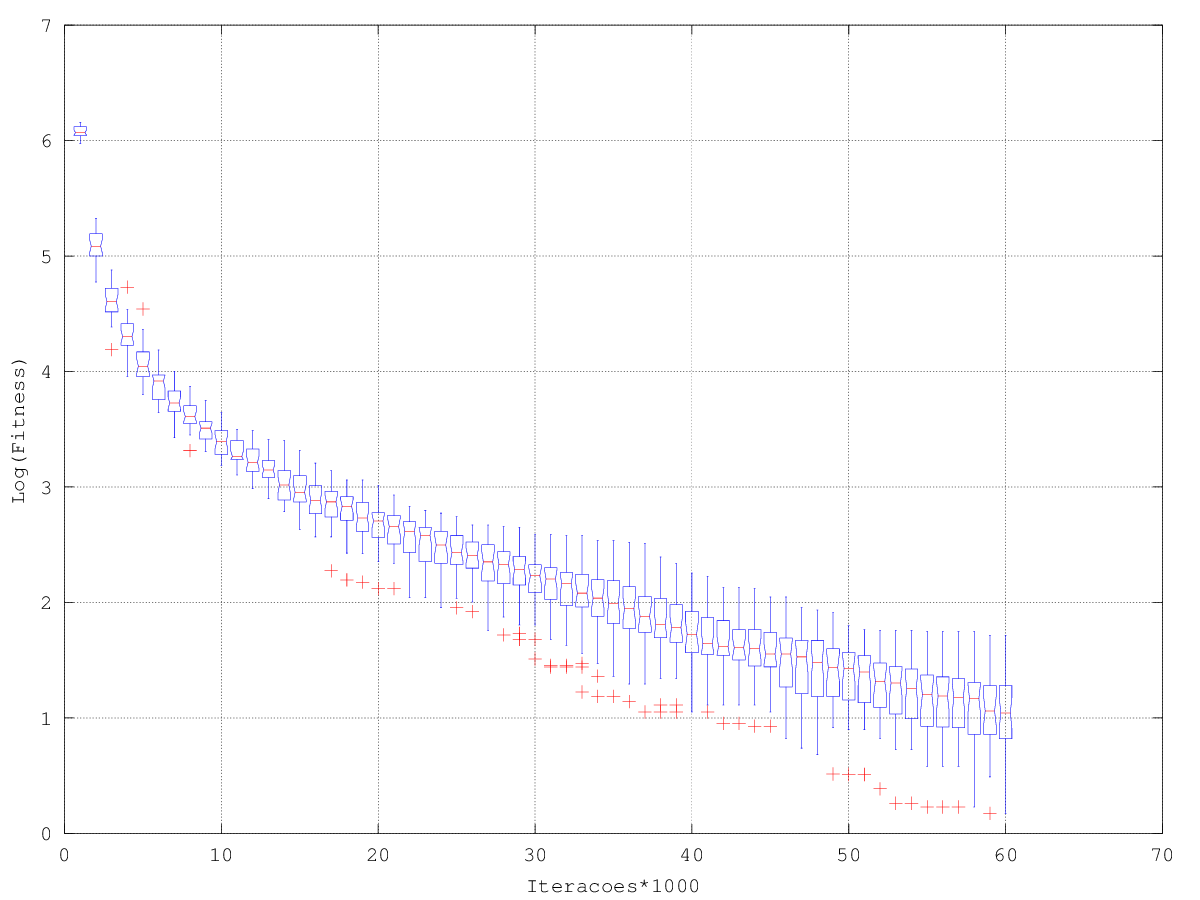
**Figura 33. Rastrigin com 2 variáveis**



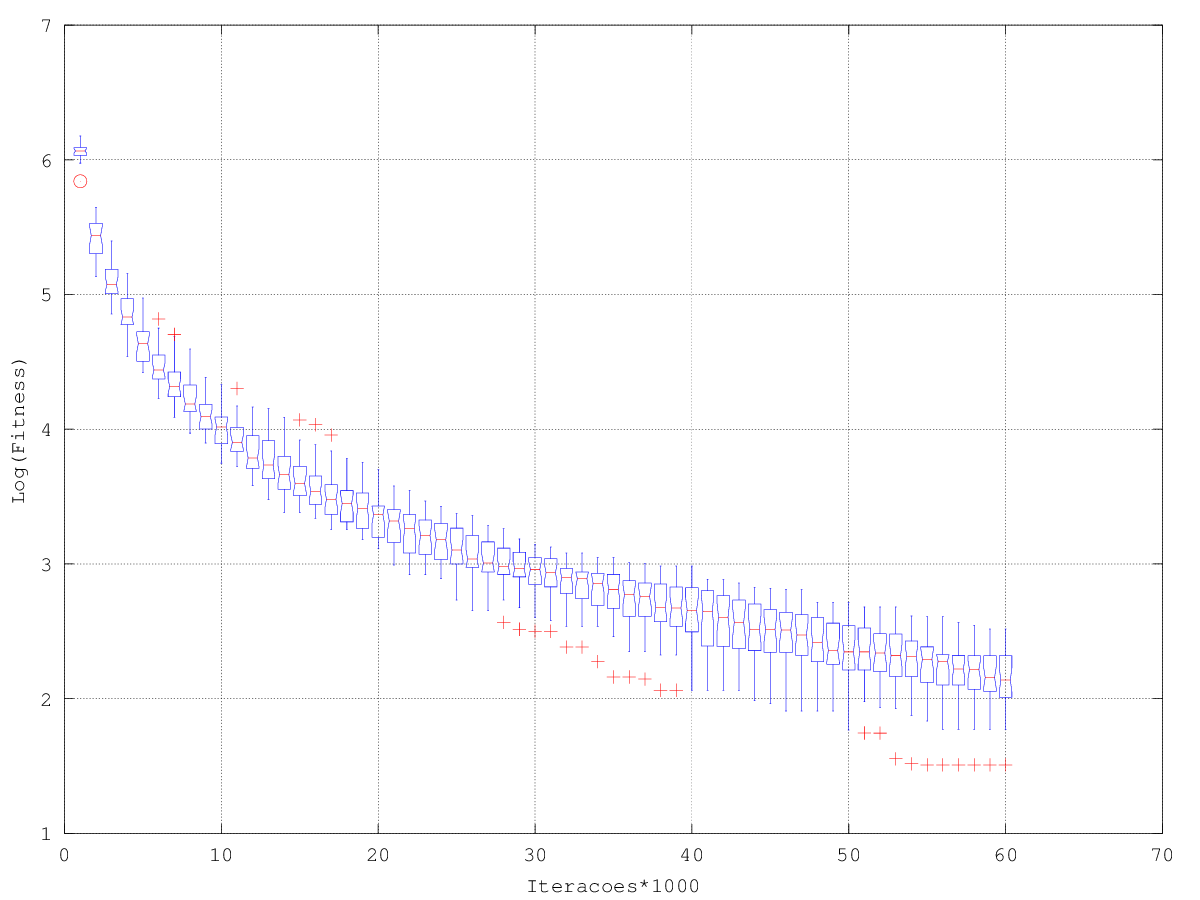
**Figura 34. Rastrigin com 2 variáveis**



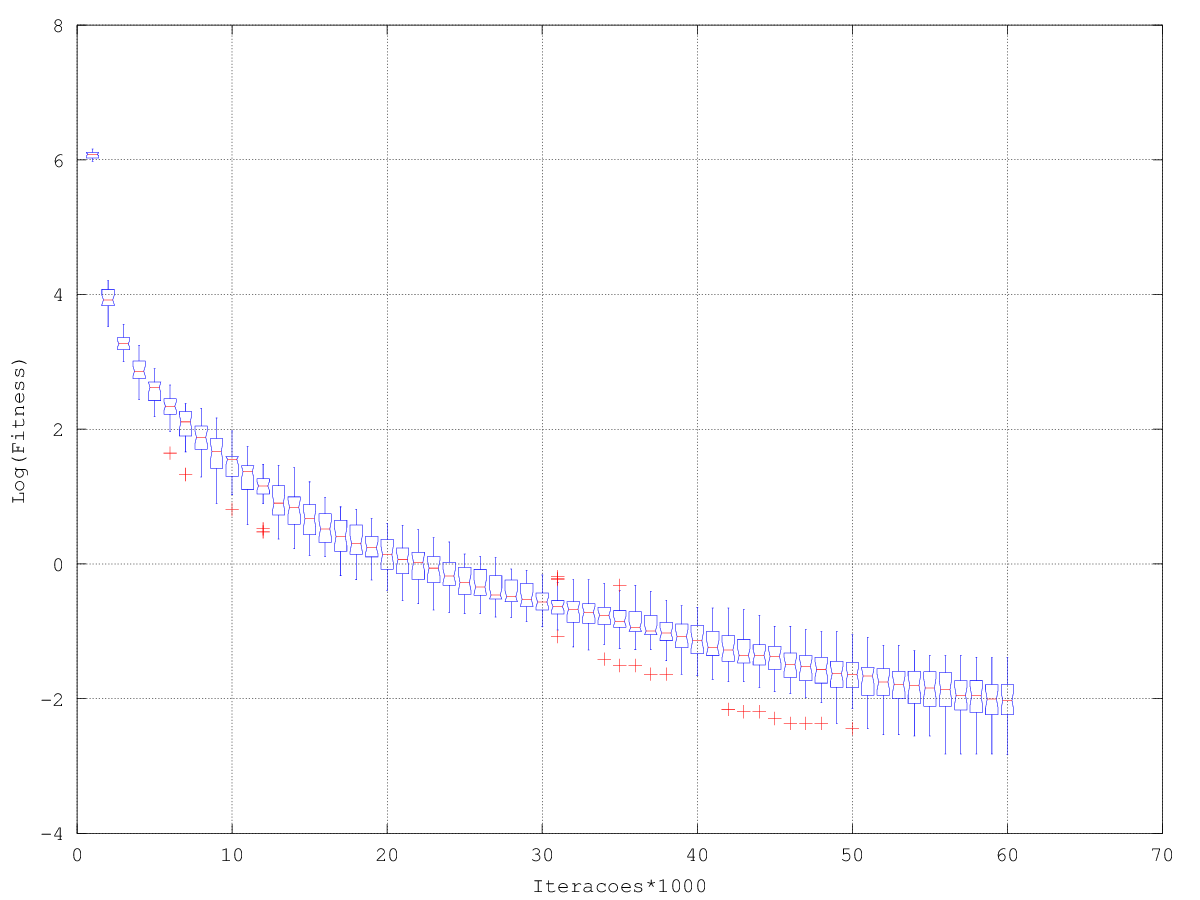
**Figura 35. Rastrigin com 2 variáveis**



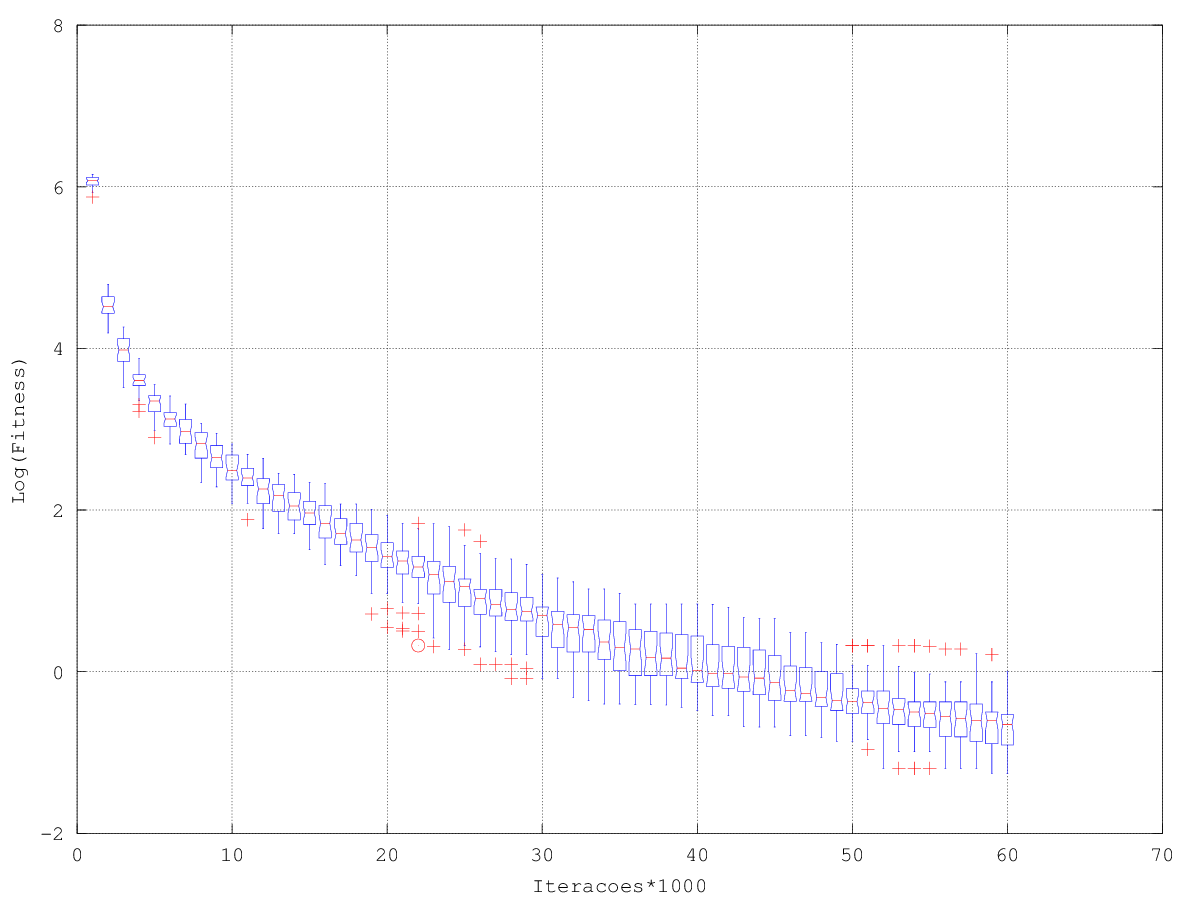
**Figura 36. Rastrigin com 2 variáveis**



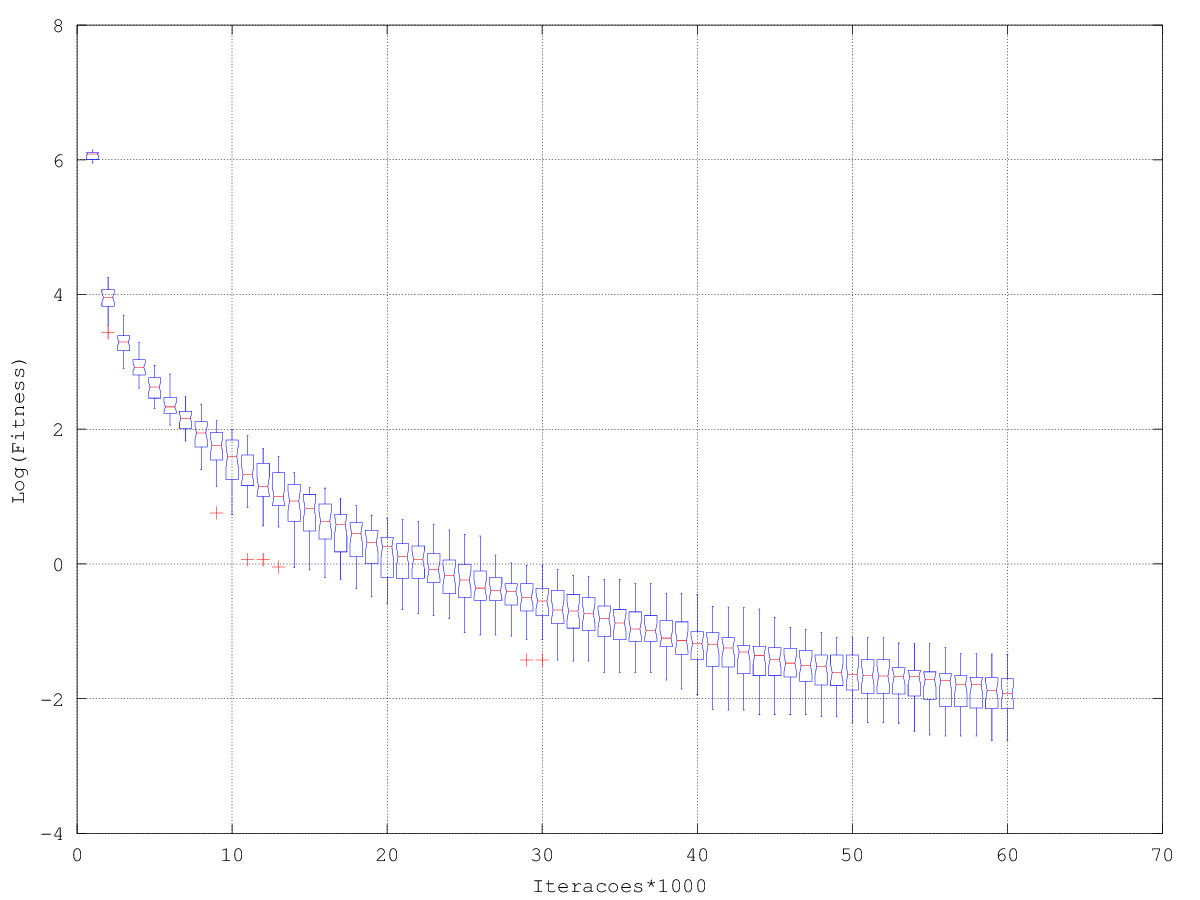
**Figura 37. Rastrigin com 2 variáveis**



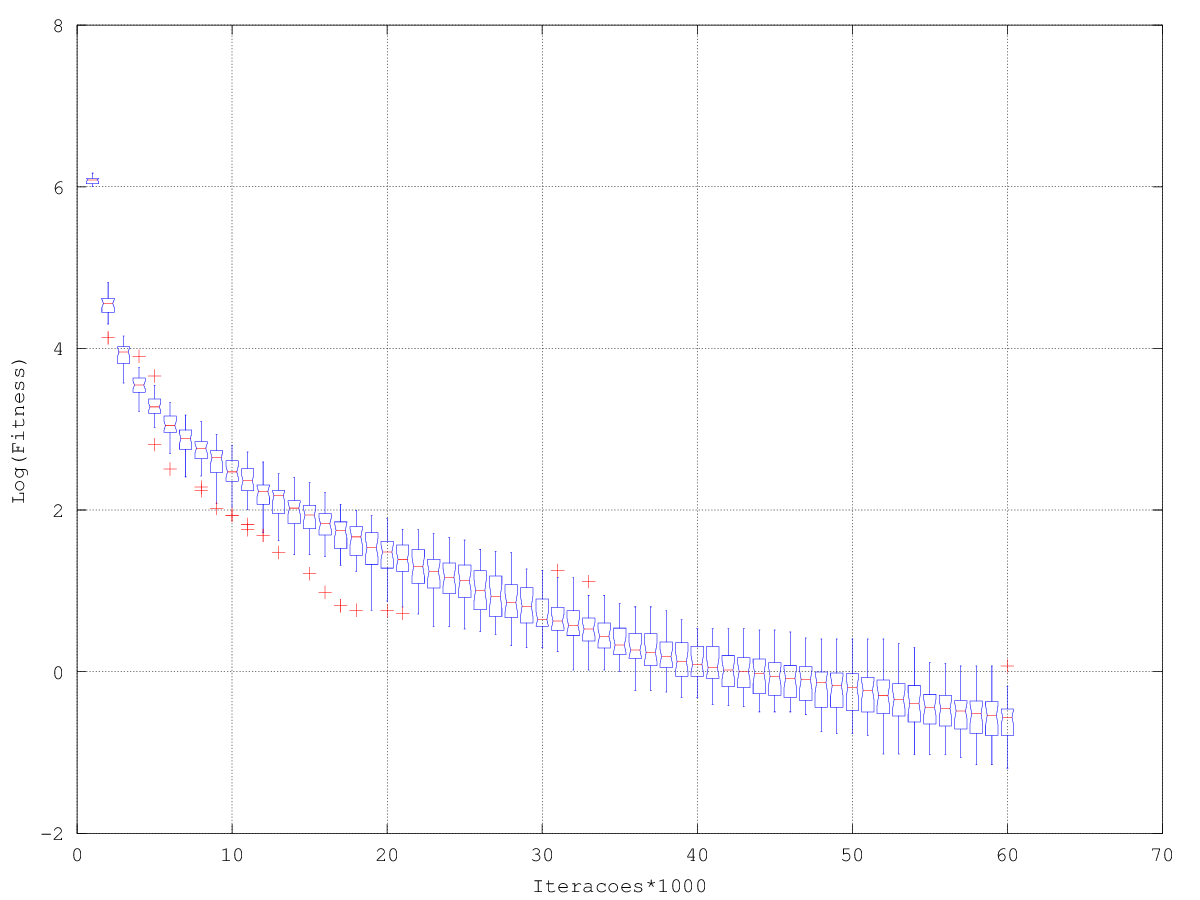
**Figura 38. Rastrigin com 2 variáveis**



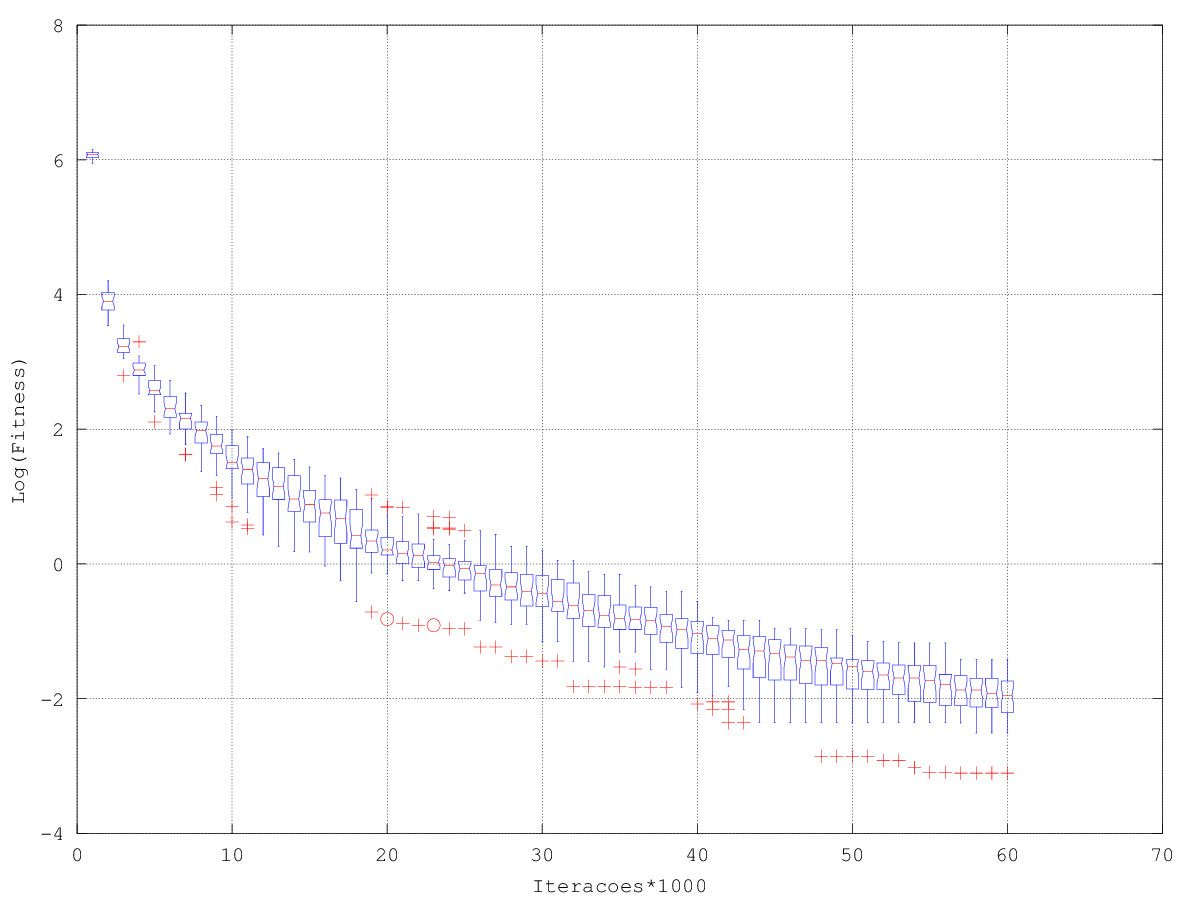
**Figura 39. Rastrigin com 2 variáveis**



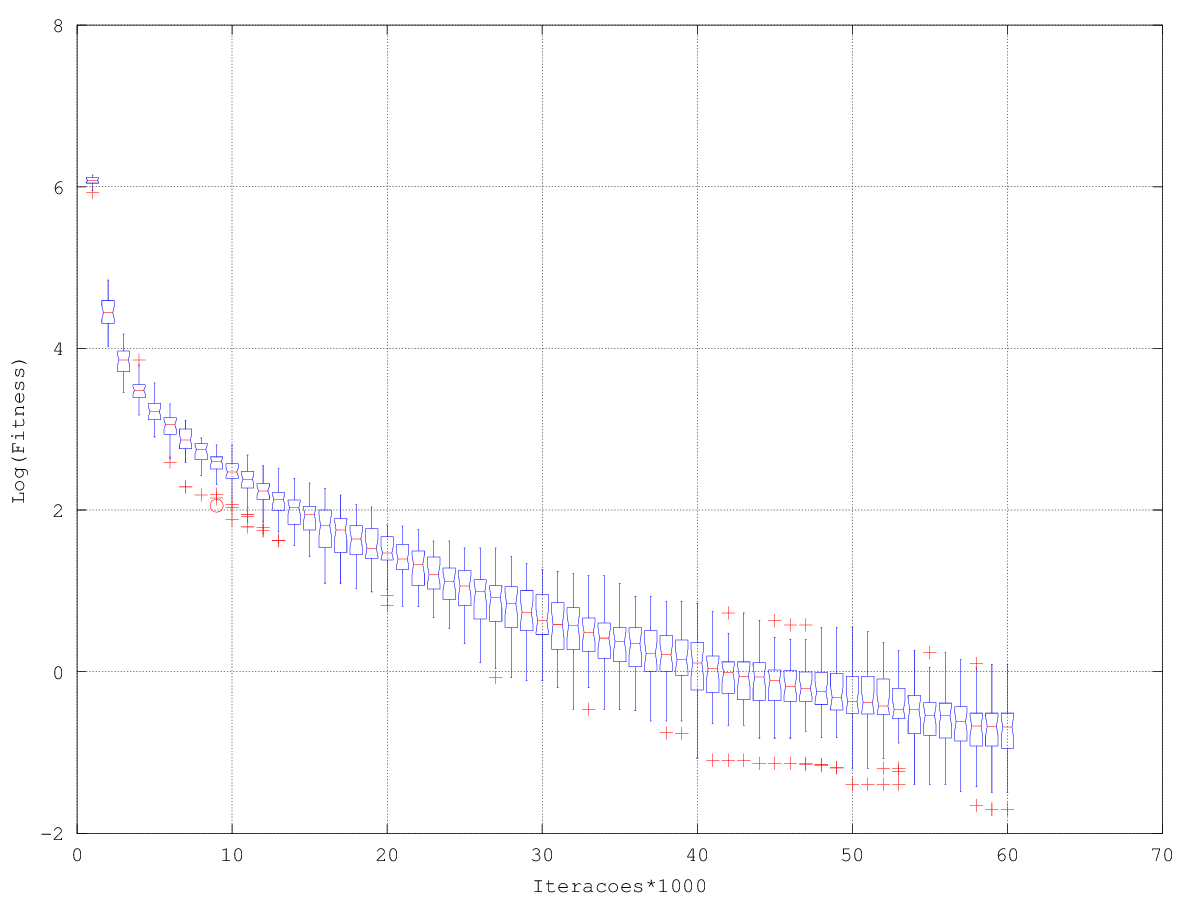
**Figura 40. Rastrigin com 2 variáveis**



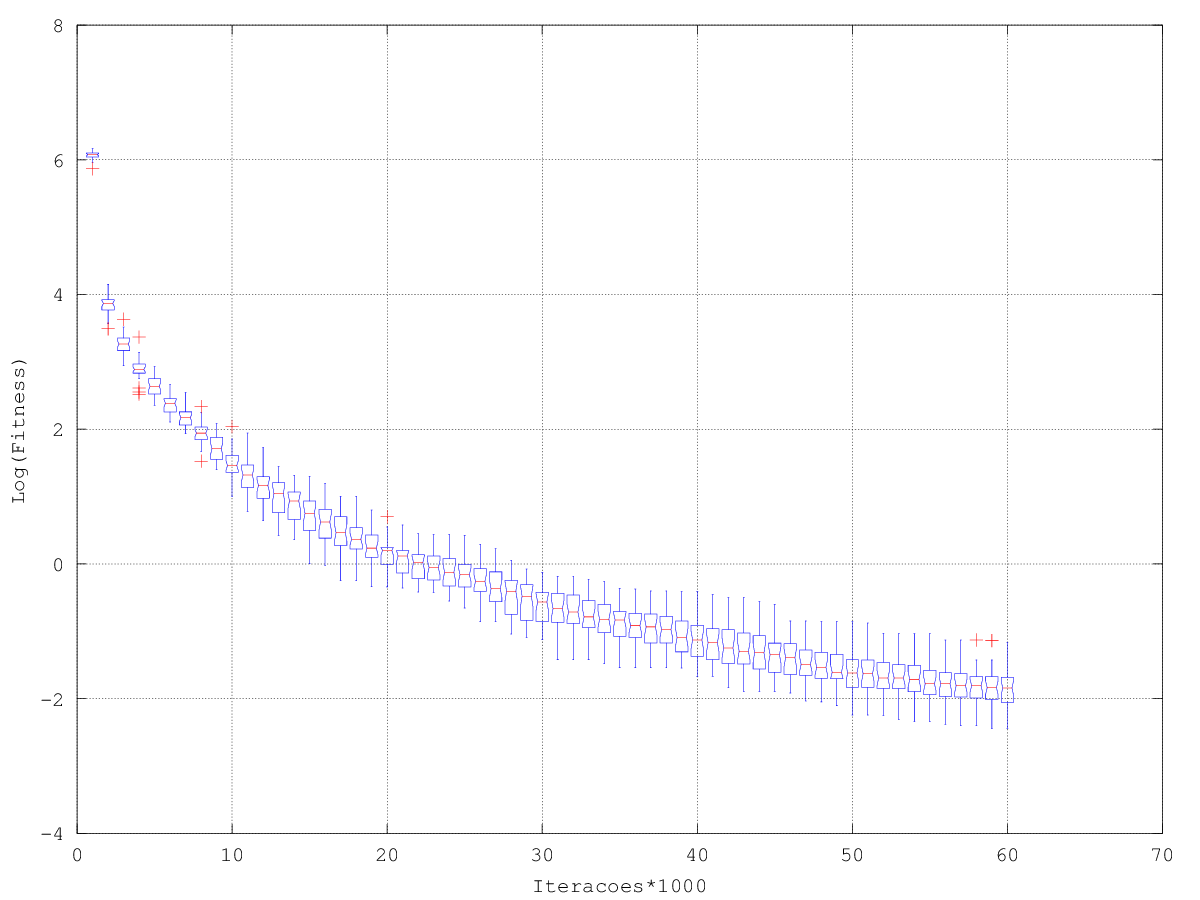
**Figura 41. Rastrigin com 2 variáveis**



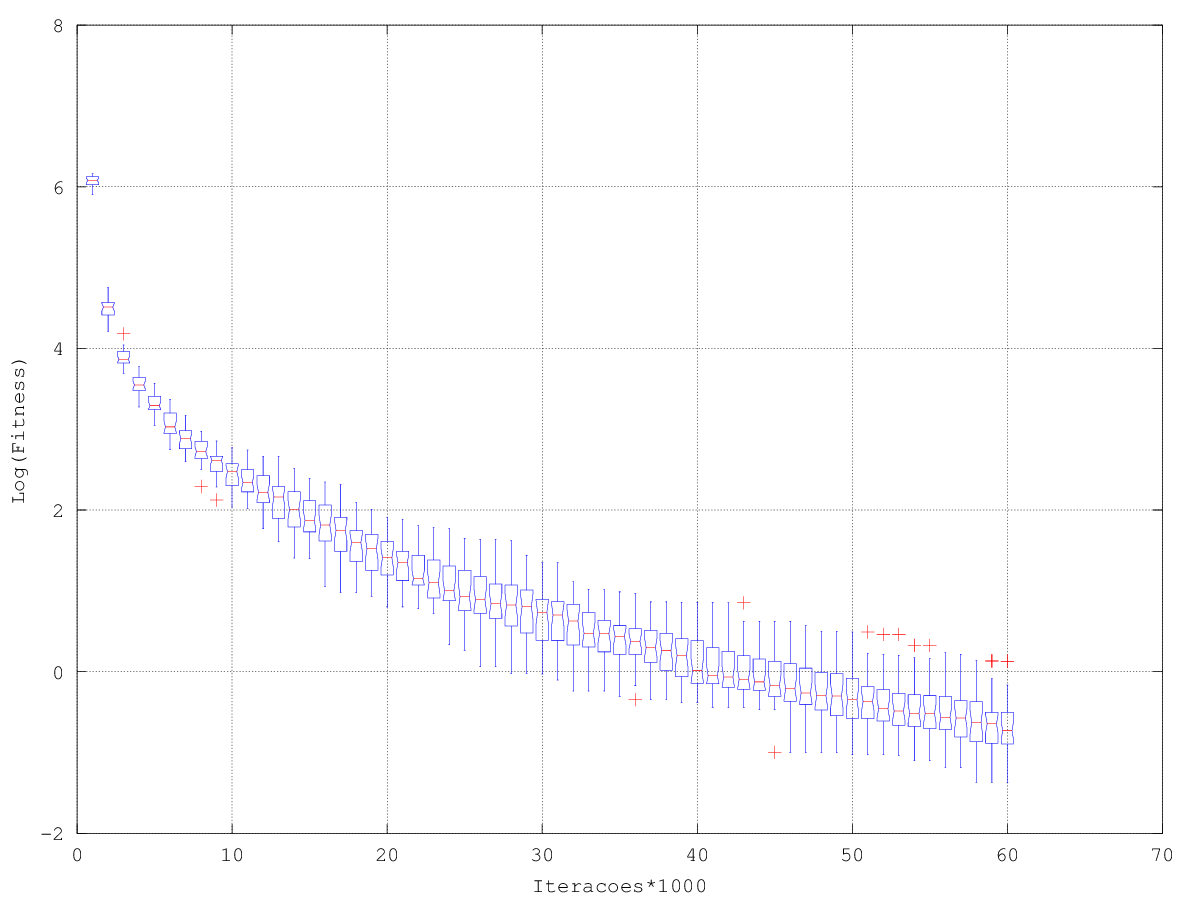
**Figura 42. Rastrigin com 2 variáveis**



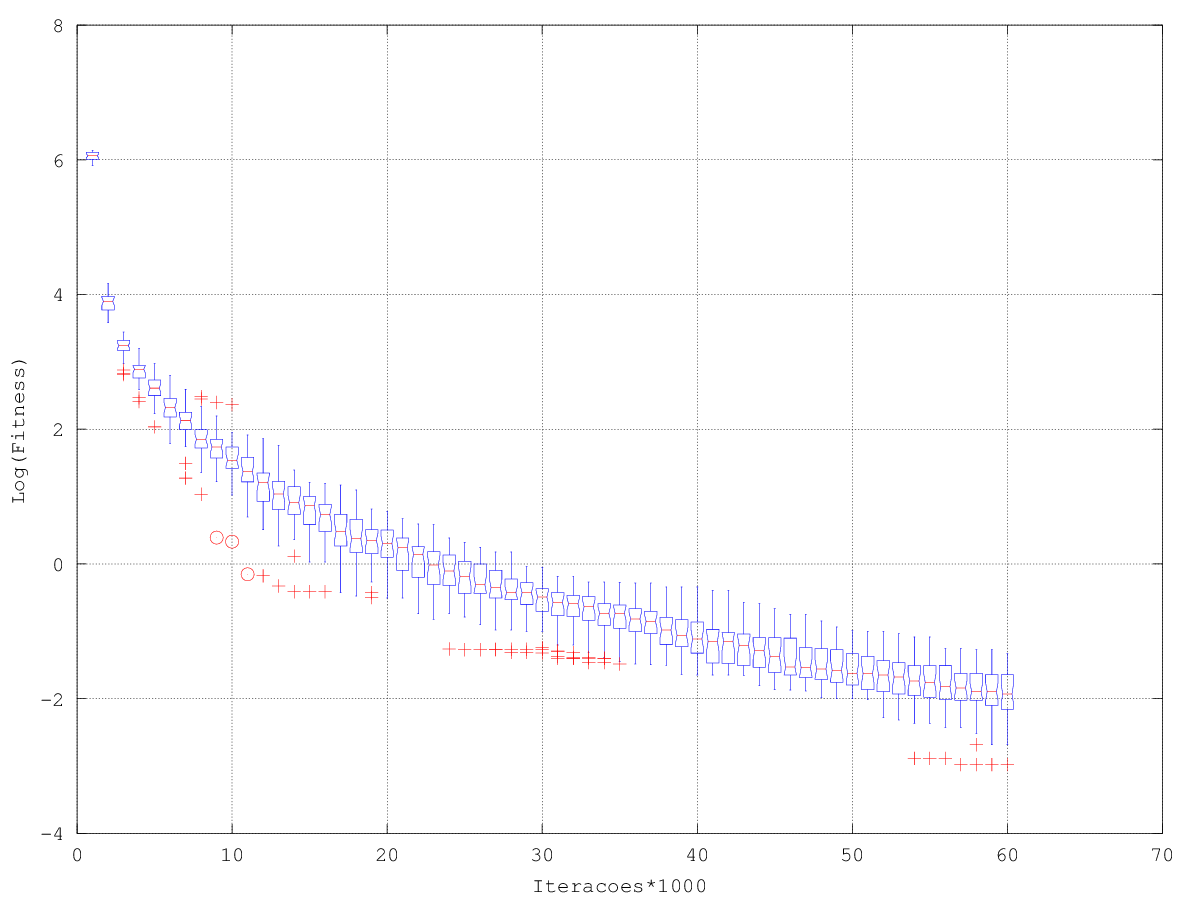
**Figura 43. Rastrigin com 2 variáveis**



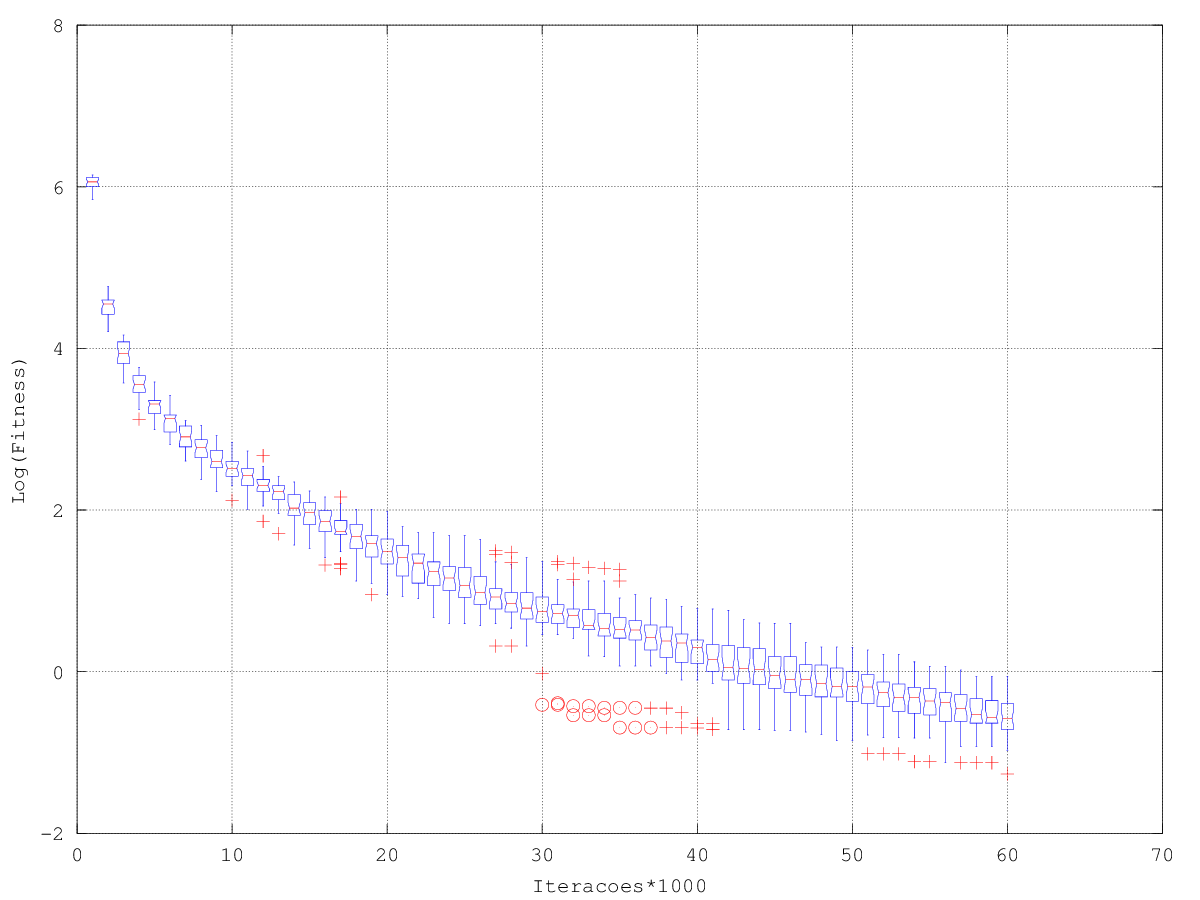
**Figura 44. Rastrigin com 2 variáveis**



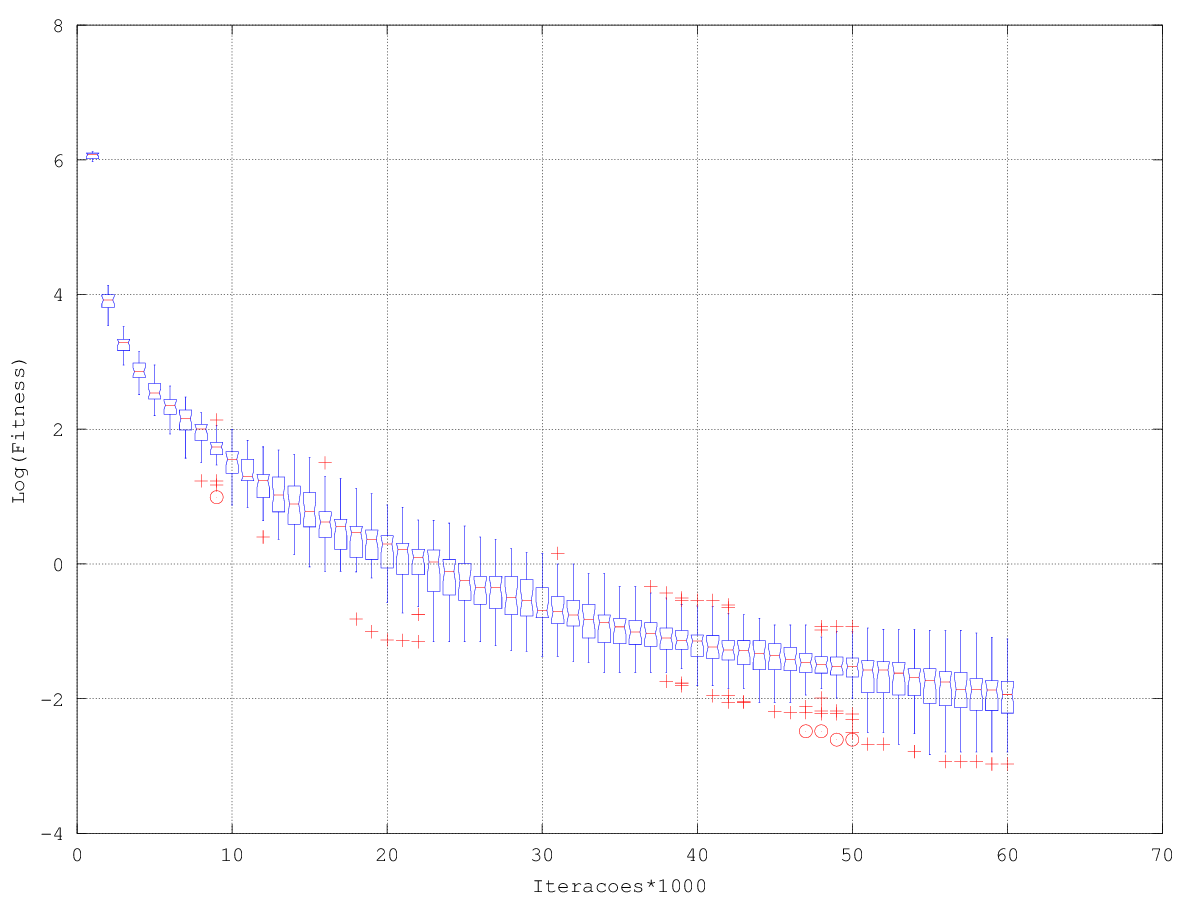
**Figura 45. Rastrigin com 2 variáveis**



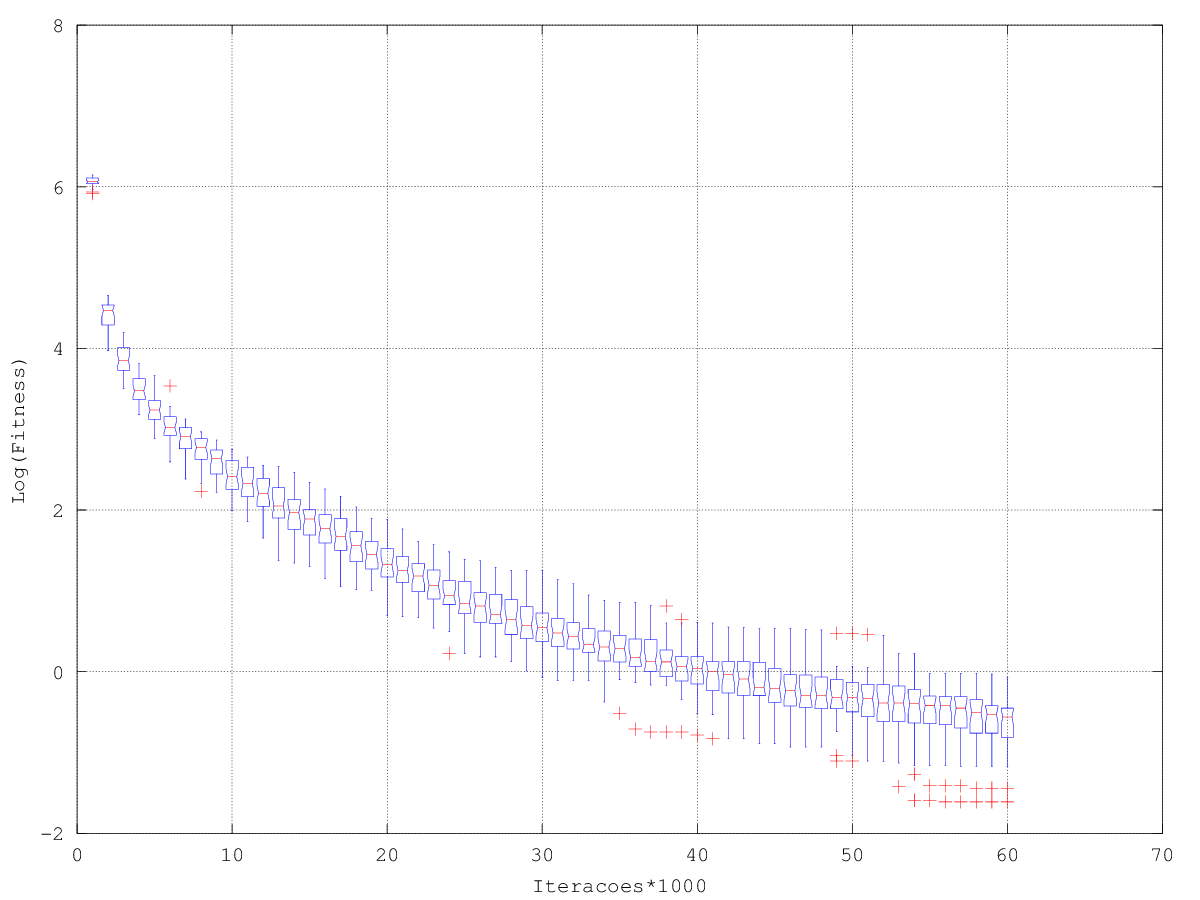
**Figura 46. Rastrigin com 2 variáveis**



**Figura 47. Rastrigin com 2 variáveis**



**Figura 48. Rastrigin com 2 variáveis**



**Figura 49. Rastrigin com 2 variáveis**

Teste Não Paramétrico Pareado de Wilcoxon

Os Testes Não Paramétricos Pareados de Wilcoxon foram realizados para cada par de experimentos. Os resultados são apresentados na Tabela 4. “P” significa que o experimento da linha é pior que o da coluna. “M” significa que o experimento da linha é melhor que o da coluna.

**Tabela 4. Classificação dos 15 experimento**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Teste Não Paramétrico Pareado de Wilcoxon** | | | | | | | | | | | | | | | |
|  | E1 | E2 | E3 | E4 | E5 | E6 | E7 | E8 | E9 | E10 | E11 | E12 | E13 | E14 | E15 |
| E1 | - | P | P | M | P | P | P | P | P | P | P | M | P | P | P |
| E2 | M | - | M | M | M | M | P | P | P | M | P | M | M | P | P |
| E3 | M | P | - | M | P | M | P | P | P | P | P | M | M | P | P |
| E4 | P | P | P | - | P | P | P | P | P | P | P | M | P | P | P |
| E5 | M | P | M | M | - | M | P | P | P | P | P | M | M | P | P |
| E6 | M | P | P | M | P | - | P | P | P | P | P | M | P | P | P |
| E7 | M | M | M | M | M | M | - | P | P | M | P | M | M | P | P |
| E8 | M | M | M | M | M | M | M | - | P | M | P | M | M | P | P |
| E9 | M | M | M | M | M | M | M | M | - | M | M | M | M | M | M |
| E10 | M | P | M | M | M | M | P | P | P | - | P | M | M | P | P |
| E11 | M | M | M | M | M | M | M | M | P | M | - | M | M | P | P |
| E12 | P | P | P | P | P | P | P | P | P | P | P | - | P | P | P |
| E13 | M | P | P | M | P | M | P | P | P | P | P | M | - | P | P |
| E14 | M | M | M | M | M | M | M | M | P | M | M | M | M | - | P |
| E15 | P | P | P | P | P | P | P | M | P | M | M | M | M | M | - |

Referências

[1] David E. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional, 1 edition, January 1989.

[2] A. Törn and A. Zilinskas. Global Optimization. Lecture Notes in Computer Science, Nº 350, Springer-Verlag, Berlin, 1989.