



UNIVERSIDADE PRESBITERIANA MACKENZIE

MESTRADO EM ENGENHARIA ELÉTRICA

**PROGRAMAÇÃO EVOLUTIVA COM DISTRIBUIÇÃO
ESTÁVEL ADAPTATIVA**

Leopoldo Bulgarelli de Carvalho

Texto de dissertação apresentado ao Programa de Pós-Graduação em Engenharia Elétrica como requisito das exigências do exame para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Dr. Pedro Paulo Balbi de Oliveira

São Paulo

Julho/2007

Banca Examinadora

Dr. Pedro Paulo Balbi de Oliveira
Universidade Presbiteriana Mackenzie

Dr. Maurício Marengoni
Universidade Presbiteriana Mackenzie

Dr. Fernando Manuel Ramos
Lab. Associado de Computação e Matemática Aplicada - Instituto Nac. de Pesquisas Espaciais

Dedicatória

À minha companheira **Silvia**, pelo carinho, amor, incentivo, participação e compreensão pelos finais de semana de dedicação no desenvolvimento desse trabalho.

À minha mãe **Marlene**, pelo exemplo de perseverança, força, carinho e amor em todos os momentos da minha vida.

À minha querida irmã **Roberta** e família por todo apoio e companheirismo.

Agradecimentos

A **Deus**, pela oportunidade de mais uma conquista em minha vida.

À Universidade Presbiteriana Mackenzie, pelo idealismo e dedicação ao ensino na nossa sociedade.

Aos pesquisadores referenciados, suas obras, artigos e dedicação que serviram de base para esta pesquisa.

Aos familiares, amigos e todos aqueles que me acompanharam e ainda estão comigo nesta jornada da vida.

Em especial ao professor **Dr. Pedro Paulo Balbi de Oliveira**, que me conduziu neste trabalho e sem o qual não o teria realizado.

Ao MACKPESQUISA pelo incentivo financeiro no desenvolvimento de mais um projeto de pesquisa científica.

Este trabalho conta com o apoio financeiro da FAPESP, através do projeto de pesquisa Processo 2005/04696-3, e da Wolfram Research, através do Mathematica Academic Grant 1149. A ambas as instituições meus sinceros agradecimentos.

Sumário

Banca Examinadora	1
Dedicatória	2
Agradecimentos	3
Sumário	4
Resumo	8
Abstract	9
Capítulo 1: Introdução	10
Capítulo 2: Distribuições Estáveis de Probabilidade	12
2.1 Introdução	12
2.2 Definição de Distribuição de Probabilidade Estável	12
2.3 Caracterização de uma Estável	13
2.4 Parametrização da Estável	14
Figura 1. Funções de densidade para distribuições estáveis com parâmetros $S(\alpha, 0.5, 1, 0)$.	14
Figura 2. Funções de densidade para distribuições estáveis com parâmetros $S(1, \beta, 1, 0)$.	15
Figura 3. Funções de densidade para distribuições estáveis com parâmetros $S(2, 0, \gamma, 0)$.	15
Figura 4. Funções de densidade para distribuições estáveis com parâmetros $S(2, 0, 1, \delta)$.	16
Figura 5. Comparativo entre as funções densidade de probabilidade da Normal padrão, da distribuição de Cauchy e de Lévy com parâmetros de escala $\gamma = 0$ e dispersão $\delta = 1$.	17
Figura 6. Comparativo entre as distribuições de probabilidade acumuladas de uma Normal padrão, da distribuição de Cauchy e de Lévy com parâmetros de escala $\gamma = 0$ e dispersão $\delta = 1$.	18
Tabela 1. Comparativo entre as distribuições de probabilidade de uma Normal padrão, da distribuição de Cauchy e de Lévy com parâmetros de escala $\gamma = 0$ e dispersão $\delta = 1$.	18
Capítulo 3: Programação Evolutiva	19
3.1 Introdução: Computação Evolutiva	19
3.2 Mecanismos Adaptativos	20
3.3 Operadores Genéticos	21
3.3.2 Mutação de Cauchy	22
3.3.3 Mutação de Lévy	23
3.3.4 Operadores de Mutação Combinados	23
3.4 Métodos de Seleção	24

<u>Capítulo 4: Os Algoritmos Utilizados</u>	<u>26</u>
4.1 A Família de Algoritmos de Yao e Colaboradores	26
4.2 Novos Algoritmos Propostos	27
4.2.1 Auto-Adaptatação no Parâmetro α	28
Tabela 2. Definições dos algoritmos auto-adaptativos que utilizam distribuições de probabilidade estáveis.	31
4.2.2 Aplicação por Nível de Adaptação	31
Tabela 3. Definições de algoritmos por combinação de níveis de adaptação.	32
4.2.3 Auto-Adaptação Elistista	33
Figura 7. Exemplo do esquema auto-adaptativo elitista no parâmetro estratégico α^* .	34
4.2.4 Auto-Adaptação nos Parâmetros α e γ da Distribuição Estável	36
<u>Capítulo 5: Resultados</u>	<u>38</u>
5.1 Funções de Teste e Parametrizações dos Algoritmos	38
Tabela 4. Funções de teste.	39
5.2 Análise Gráfica	41
5.2.1 Resultados da Auto-Adaptação no Parâmetro α de uma Distribuição Estável	41
Figura 8. Comparativo de desempenho dos algoritmos para a função f_1 (eixo das ordenadas em logaritmo natural).	42
Figura 9. Comparativo de desempenho dos algoritmos para a função f_2 (eixo das ordenadas em logaritmo natural).	42
Figura 10. Comparativo de desempenho dos algoritmos para a função f_3 (eixo das ordenadas em logaritmo natural).	43
Figura 11. Comparativo de desempenho dos algoritmos para a função f_4 .	43
Figura 12. Comparativo de desempenho dos algoritmos para a função f_5 .	44
Figura 13. Comparativo de desempenho dos algoritmos para a função f_6 .	44
Figura 14. Comparativo de desempenho dos algoritmos para a função f_7 .	45
Figura 15. Comparativo de desempenho dos algoritmos para a função f_8 (eixo das ordenadas em logaritmo natural).	45
Figura 16. Comparativo de desempenho dos algoritmos para a função f_9 (eixo das ordenadas em logaritmo natural).	46
Figura 17. Comparativo de desempenho dos algoritmos para a função f_{10} (eixo das ordenadas em logaritmo natural).	46
Figura 18. Comparativo de desempenho dos algoritmos para a função f_{11} .	47
Figura 19. Comparativo de desempenho dos algoritmos para a função f_{12} .	47
5.2.2 Resultados dos Algoritmos com Variação por Nível de Adaptação	48
Figura 20. Comparativo de desempenho dos algoritmos para a função f_1 (eixo das ordenadas em logaritmo natural).	49

Figura 21. Comparativo de desempenho dos algoritmos para a função f_2 (eixo das ordenadas em logaritmo natural).	49
Figura 22. Comparativo de desempenho dos algoritmos para a função f_3 (eixo das ordenadas em logaritmo natural).	50
Figura 23. Comparativo de desempenho dos algoritmos para a função f_4 .	50
Figura 24. Comparativo de desempenho dos algoritmos para a função f_5 .	51
Figura 25. Comparativo de desempenho dos algoritmos para a função f_6 .	51
Figura 26. Comparativo de desempenho dos algoritmos para a função f_7 .	52
Figura 27. Comparativo de desempenho dos algoritmos para a função f_8 (eixo das ordenadas em logaritmo natural).	52
Figura 28. Comparativo de desempenho dos algoritmos para a função f_9 (eixo das ordenadas em logaritmo natural).	53
Figura 29. Comparativo de desempenho dos algoritmos para a função f_{10} (eixo das ordenadas em logaritmo natural).	53
Figura 30. Comparativo de desempenho dos algoritmos para a função f_{11} .	54
Figura 31. Comparativo de desempenho dos algoritmos para a função f_{12} .	54
5.2.3 Resultados da Auto-Adaptação Elitista e Auto-Adaptação nos Parâmetros α e γ da Distribuição Estável	55
Figura 32 Comparativo de desempenho dos algoritmos para a função f_1 (eixo das ordenadas em logaritmo natural).	56
Figura 33 Comparativo de desempenho dos algoritmos para a função f_2 (eixo das ordenadas em logaritmo natural).	56
Figura 34 Comparativo de desempenho dos algoritmos para a função f_3 (eixo das ordenadas em logaritmo natural).	57
Figura 35 Comparativo de desempenho dos algoritmos para a função f_4 .	57
Figura 36 Comparativo de desempenho dos algoritmos para a função f_5 .	58
Figura 37 Comparativo de desempenho dos algoritmos para a função f_6 .	58
Figura 38 Comparativo de desempenho dos algoritmos para a função f_7 .	59
Figura 39 Comparativo de desempenho dos algoritmos para a função f_8 (eixo das ordenadas em logaritmo natural).	59
Figura 40 Comparativo de desempenho dos algoritmos para a função f_9 (eixo das ordenadas em logaritmo natural).	60
Figura 41 Comparativo de desempenho dos algoritmos para a função f_{10} (eixo das ordenadas em logaritmo natural).	60
Figura 42 Comparativo de desempenho dos algoritmos para a função f_{11} .	61
Figura 43 Comparativo de desempenho dos algoritmos para a função f_{12} .	61
5.2.4 Resultados dos Algoritmos para o Dobro do Número de Gerações Original	62
Figura 44 Comparativo de desempenho dos algoritmos para a função f_1 (eixo das ordenadas em logaritmo natural).	63

Figura 45 Comparativo de desempenho dos algoritmos para a função f_2 (eixo das ordenadas em logaritmo natural).	63
Figura 46 Comparativo de desempenho dos algoritmos para a função f_3 (eixo das ordenadas em logaritmo natural).	64
Figura 47 Comparativo de desempenho dos algoritmos para a função f_4 .	64
Figura 48 Comparativo de desempenho dos algoritmos para a função f_5 .	65
Figura 49 Comparativo de desempenho dos algoritmos para a função f_6 .	65
Figura 50 Comparativo de desempenho dos algoritmos para a função f_7 .	66
Figura 51 Comparativo de desempenho dos algoritmos para a função f_8 (eixo das ordenadas em logaritmo natural).	66
Figura 52 Comparativo de desempenho dos algoritmos para a função f_9 (eixo das ordenadas em logaritmo natural).	67
Figura 53 Comparativo de desempenho dos algoritmos para a função f_{10} (eixo das ordenadas em logaritmo natural).	67
Figura 54 Comparativo de desempenho dos algoritmos para a função f_{11} .	68
Figura 55 Comparativo de desempenho dos algoritmos para a função f_{12} .	68
5.3 Análise Numérica	69
5.3.1 Análise Numérica da Auto-Adaptação no Parâmetro α de uma Distribuição Estável	69
Tabela 5 Resultados experimentais para SEP e GEP.	70
5.3.2 Análise Numérica da Aplicação por Nível de Adaptação	71
Tabela 6 Resultados experimentais do SEP por nível de adaptação.	72
5.3.3 Análise Numérica da Auto-Adaptação Elitista e Auto-Adaptação nos Parâmetros α e γ da Distribuição Estável	73
Tabela 7 Resultados experimentais dos algoritmos EAA, PM/ γ e VFF.	74
5.3.4 Resultados dos Algoritmos para o Dobro do Número de Gerações Original	75
Tabela 8 Resultados experimentais dos algoritmos PM/ γ , FFV, VFF e VFV.	75
Tabela 9 Tabela 9. Comparativo de resultados obtido em [Lee e Yao, 2004] e estudo atual.	76
5.3.5 Resultados dos Algoritmos VFV e VFV sem Restrições no Espaço de Busca	78
Tabela 10 Comparativo de resultados entre o algoritmo VFV e VFV-SR.	78
<u>Capítulo 6: Conclusão</u>	<u>80</u>
Referências Bibliográficas	83
Anexo A - Teste de Hipótese Estatístico	87

Resumo

Aplicações recentes em programação evolutiva tem sugerido a utilização de diferentes distribuições estáveis de probabilidade, tais como de Cauchy e de Lévy, no processo aleatório associado às mutações, como alternativa à tradicional (e também estável) distribuição Normal. A motivação para tanto é melhorar os resultados em algumas classes de problemas de otimização, com relação aos obtidos através da distribuição Normal.

Esse trabalho propõe uma nova classe de algoritmos auto-adaptativos com respeito à determinação dos parâmetros da distribuição estável mais adequada para cada problema de otimização. Tais algoritmos foram derivados de um existente na literatura, especialmente sua versão apresentada em [Lee e Yao, 2004].

Em um primeiro momento foram estudadas as principais características das distribuições estáveis que são, nesse trabalho, o foco dos processos aleatórios associados às mutações. Posteriormente, foram apresentadas as diferentes abordagens descritas pela literatura e as sugestões de algoritmos com características auto-adaptativas.

As avaliações dos algoritmos propostos utilizaram funções de teste padrão da literatura, e os resultados comparativos de desempenho foram realizados com relação a um algoritmo tradicional baseado na distribuição Normal. Posteriormente, foram aplicados novos comparativos entre as diversas abordagens auto-adaptativas definidas no presente estudo, e feito um comparativo do melhor algoritmo auto-adaptativo aqui proposto com o melhor algoritmo adaptativo obtido de [Lee e Yao, 2004].

Os resultados evidenciaram superioridade numérica e estatística da abordagem baseada em distribuições estáveis, sobre o método tradicional baseado na distribuição Normal. No entanto, o método proposto não se mostrou mais eficaz que o método adaptativo sugerido em [Lee e Yao, 2004], o que pode ter sido decorrente de decisões de implementação não explícitas naquele trabalho, que tiveram de ser tomadas no presente contexto.

Palavras-chave: Computação evolutiva; algoritmo genético; distribuição estável; distribuição de Lévy; mutação adaptativa; auto-adaptação.

Abstract

Recent applications in evolutionary programming have suggested the use of different stable probability distributions, such as Cauchy and Lévy, in the random process associated with the mutations, as an alternative to the traditional (and also stable) Normal distribution. The motivation for this is the attempt to improve the results in some classes of optimisation problems, over those obtained with Normal distribution.

Based upon an algorithm proposed in the literature, mostly its version in [Lee and Yao, 2004], that use non Normal stable distributions, we study herein the effect of turning it adaptive in respect to the determination of the more adequate stable distribution parameters for each problem.

The evaluations relied upon standard benchmarking functions of the literature, and the comparative performance tests were carried out in respect to the baseline defined by a standard algorithm using Normal distribution.

The results suggest numerical and statistical superiority of the stable distribution based approach, when compared with the baseline. However, they showed no improvement over the adaptive method of [Lee and Yao, 2004], possibly due to a consequence of implementation decisions that had to be made in the present implementation, that were not made explicit therein.

Keywords: Evolutionary computation; genetic algorithm; stable probability distribution; Lévy distribution; adaptive mutation; auto-adaptation.

Capítulo 1: Introdução

Fenômenos naturais tão diversos quanto a variação da altura de indivíduos, concentração de melanina na pele em um dado grupo de humanos ou até mesmo o movimento térmico dos átomos e moléculas, tem em comum o fato de poderem ser modelados a partir de uma mesma distribuição de probabilidade. É por esse motivo que a distribuição Normal é largamente estudada e aplicada na resolução de problemas, não pelo grau de importância dos fenômenos que ela é capaz de modelar mas sim pela frequência que eles ocorrem na natureza.

Porém, apesar de toda aplicabilidade, sua utilização não permite uma generalização para outras classes de problemas, não tão frequentes mas de suma importância para a evolução da ciência, tais como a flutuação de preços no mercado financeiro ou o ritmo cardíaco. Por exemplo, de acordo com [Tsallis, 2000], estudos conduzidos por pesquisadores de Harvard levaram à descoberta que padrões erráticos observados nos batimentos cardíacos de pessoas saudáveis podem ser modelados por uma distribuição de Lévy, enquanto que os dados de pacientes com falhas cardíacas severas estão muito mais próximos de uma distribuição Normal. Em 1995, pesquisadores usaram métodos numéricos para estudar uma torneira gotejante e descobriram que os intervalos de tempo entre as gotas flutuam segundo uma distribuição de Lévy. As constatações sugerem que um processo hidrodinâmico comum poderia estar por trás de ambos os fenômenos.

Assim como a maior parte dos fenômenos naturais, as primeiras aplicações de programação evolutiva foram inspiradas pelo uso da distribuição Normal no processo aleatório do operador de mutação. Mais recentemente, foram introduzidas algumas aplicações comparando o desempenho da distribuição Normal com diferentes tipos de distribuições, tais como a de Cauchy e a de Lévy. As primeiras aplicações que substituem a distribuição Normal no processo aleatório associado às mutações foram sugeridas por [Yao e Liu, 1996; Rudolph, 1998; Yao, Liu e Lin, 1999], onde a utilização da distribuição de Cauchy permitiu um melhor desempenho para otimização de funções multimodais. Ainda nessa linha, [Lee e Yao, 2001; Lee e Yao, 2004] propuseram um processo adaptativo com distribuição de Lévy e novamente os resultados obtidos demonstraram superioridade sobre a aplicação convencional que utiliza a distribuição Normal de probabilidade.

A proposta desse trabalho é estudar o efeito da modificação dos algoritmos sugeridos em [Yao e Liu, 1996; Lee e Yao, 2004], tornando-os auto-adaptativos na escolha dos parâmetros da distribuição estável mais adequada para cada problema.

A seguir, no Capítulo 2, trata-se de distribuições de probabilidade estáveis, através de sua definição, caracterização e parametrização. O Capítulo 3 faz então uma descrição de computação evolutiva e as principais famílias de interesse nesse estudo, bem como os principais mecanismos adaptativos provenientes desse tipo de abordagem referenciados na literatura. No Capítulo 4, inicialmente define-se um algoritmo utilizado na literatura e, baseado neste, propõem-se novos algoritmos adaptativos, que serão posteriormente testados. Na seqüência, o Capítulo 5 apresenta os resultados da utilização dos diferentes tipos de algoritmos definidos no capítulo anterior, no problema da busca pelo ponto de mínimo de várias funções de teste conhecidas, e os resultados são apresentados na forma de análise gráfica e numérica. Finalmente, o Capítulo 6 conclui o trabalho com uma série de comentários finais, onde são discutidos os experimentos e propostas aqui realizados, e avaliando-se conseqüências e próximos passos a serem dados em um prolongamento do presente estudo.

Capítulo 2: Distribuições Estáveis de Probabilidade

2.1 Introdução

Grande parte das aplicações analíticas com utilização de dados contínuos remetem ao uso de funções com distribuição Normal de probabilidade. A forma simplificada com que se apresenta a função de distribuição Normal, a facilidade na estimativa dos seus parâmetros característicos, e principalmente suas propriedades estatísticas são alguns dos principais motivos que tornam sua aplicação tão difundida.

Apesar de modelar a maior parte dos problemas, nem sempre é possível, entretanto, pressupor Normalidade, principalmente para os casos degenerados onde outras distribuições apresentam-se mais eficientes.

É nesse contexto que a aplicação de funções de probabilidade estáveis (que inclui as Normais, como será detalhado a seguir) apresentam-se como uma boa alternativa para solução de problemas que necessitam de uma parametrização específica, pois são representantes de classes genéricas de distribuição.

2.2 Definição de Distribuição de Probabilidade Estável

Uma variável aleatória X , que tome todos os valores reais $-\infty < x < \infty$, tem distribuição Normal (ou Gaussiana) [Meyer, 1976], se sua função distribuição de probabilidade apresenta-se da seguinte forma

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right], \quad (1)$$

onde os parâmetros x , μ e σ devem satisfazer as condições: $-\infty < x < \infty$, $-\infty < \mu < \infty$ e $\sigma > 0$. Se X tiver distribuição dada conforme a Equação 1, então a variável aleatória $Y = (X-\mu)/\sigma$ terá distribuição Normal reduzida ou padrão $N(0, 1)$.

Uma das principais características da distribuição Normal é a propriedade da soma de variáveis identicamente distribuídas [Gnedenko e Kolmogorov, 1954; Nolan, 2002]. Sejam X_1 e X_2 variáveis aleatórias independentes com distribuição Normal de probabilidade, tais que $X_1 \sim N(\mu_1, \sigma_1^2)$ e $X_2 \sim N(\mu_2, \sigma_2^2)$. Então a variável aleatória $X = X_1 + X_2$ tem distribuição Normal $N(\mu_1 + \mu_2, \sqrt{\sigma_1^2 + \sigma_2^2})$. É esta situação – em que a forma da distribuição de X se mantém preservada – que permite caracterizar a distribuição Normal como *estável*, e, como se verá a seguir, várias outras distribuições apresentam esta mesma propriedade.

2.3 Caracterização de uma Estável

A melhor maneira de descrever todas as possíveis distribuições estáveis é através da sua função característica ou transformada de Fourier. Para uma determinada variável X com função de distribuição $F(x)$, sua função característica é definida por

$$\phi(u) = E[\exp[iuX]] = \int_{-\infty}^{\infty} \exp[iuX] dF(x), \quad (2)$$

onde $E[.]$ é a esperança estatística. A função sinal será usada posteriormente e pode ser definida por

$$\text{sign } u = \begin{cases} -1 & u < 0, \\ 0 & u = 0, \\ 1 & u > 0. \end{cases} \quad (3)$$

Uma variável aleatória X é estável se, e somente se $X \stackrel{d}{=} aZ + b$ (o símbolo d significa identicamente distribuído), onde Z é uma variável aleatória com função característica igual a

$$\phi(x) = \begin{cases} \exp[-|u|^\alpha (1 + i\beta \tan[\pi\alpha/2](\text{sign } u))] & \alpha \neq 1, \\ \exp[-|u| (1 + i\beta 2/\pi(\text{sign } u) \ln[|u|])] & \alpha = 1. \end{cases} \quad (4)$$

com os parâmetros: $0 < \alpha \leq 2$, $-1 \leq \beta \leq 1$, $a > 0$, $b \in \Re$. Essas distribuições são simétricas em torno de zero quando $\beta = 0$ e $b = 0$; nesse caso a função característica de aZ tem a seguinte forma simplificada:

$$\phi(x) = \exp[-a^\alpha |u|^\alpha]. \quad (5)$$

Cinco parâmetros são necessários para caracterizar uma distribuição estável, sendo que os quatro primeiros configuram as curvas de distribuição, e o quinto parâmetro define a forma com que a função característica será construída $S(\alpha, \beta, \gamma, \delta; k)$. De acordo com [Nolan, 2005] a escolha de $k = 0$ é a melhor parametrização para situações que exijam aproximações numéricas e inferência estatística de distribuições estáveis. Isso acontece porque a forma da função característica permite que todos os parâmetros sejam contínuos nos seus intervalos de domínio.

Uma variável aleatória X apresenta distribuição estável $S(\alpha, \beta, \gamma, \delta; 0)$ se

$$X \stackrel{d}{=} \begin{cases} \gamma(Z - \beta \tan(\pi\alpha/2)) + \delta & \alpha \neq 1, \\ \gamma Z + \delta & \alpha = 1 \end{cases} \quad (6)$$

onde $Z = Z(\alpha, \beta)$ é dado conforme a Equação 4. Então X terá a seguinte função característica

$$\phi(x) = \begin{cases} \exp[-\gamma^\alpha |u|^\alpha (1 - i\beta \tan[\pi\alpha/2](\text{sign } u)) + i\delta u] & \alpha \neq 1, \\ \exp[-\gamma |u| (1 + i\beta 2/\pi(\text{sign } u) \ln[|u|]) + i\delta u] & \alpha = 1. \end{cases} \quad (7)$$

Devido a fixação do parâmetro $k = 0$ em todo o estudo, a notação que caracteriza a distribuição estável será reduzida para $S(\alpha, \beta, \gamma, \delta)$.

2.4 Parametrização da Estável

Para simular as diferentes classes de distribuições estáveis existentes o presente trabalho utilizou o pacote STABLE, implementado no software *Mathematica*, por [Rimmer e Nolan, 2005]. A caracterização de uma distribuição estável utilizando esse pacote depende apenas da configuração dos quatro parâmetros característicos das distribuições de probabilidade.

A Figura 1 ilustra três diferentes parametrizações para o *índice de estabilidade* ou *expoente característico* $\alpha \in (0, 2]$.

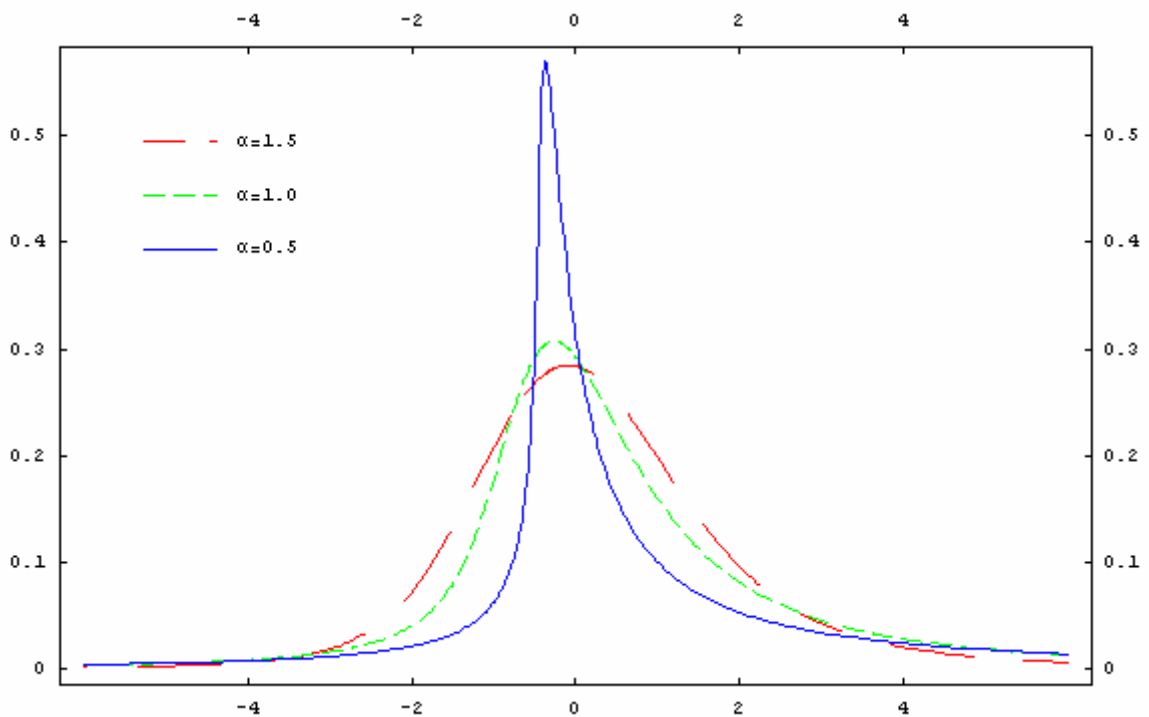


Figura 1. Funções de densidade para distribuições estáveis com parâmetros $S(\alpha, 0.5, 1, 0)$.

A Figura 2 ilustra alguns exemplos do *parâmetro de simetria* $\beta \in [-1, 1]$. Quando $\beta \in [-1, 0)$ a curva é assimétrica à direita, quando $\beta \in (0, 1]$ a curva é assimétrica à esquerda e se $\beta = 0$ a curva é estritamente simétrica.

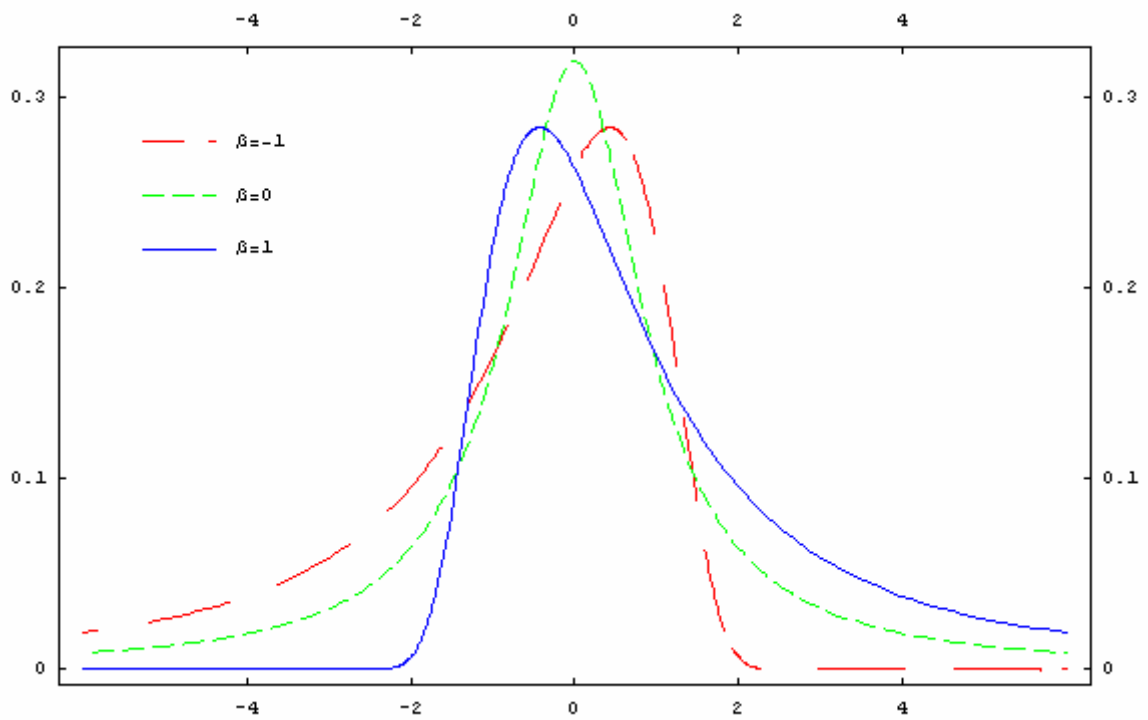


Figura 2. Funções de densidade para distribuições estáveis com parâmetros $S(1, \beta, 1, 0)$.

A Figura 3 ilustra alguns exemplos do *parâmetro de escala* $\gamma > 0$. Esse parâmetro define a intensidade da dispersão dos dados em torno de um valor médio, ou seja, quanto maior o valor de γ maior será a variância dos pontos.

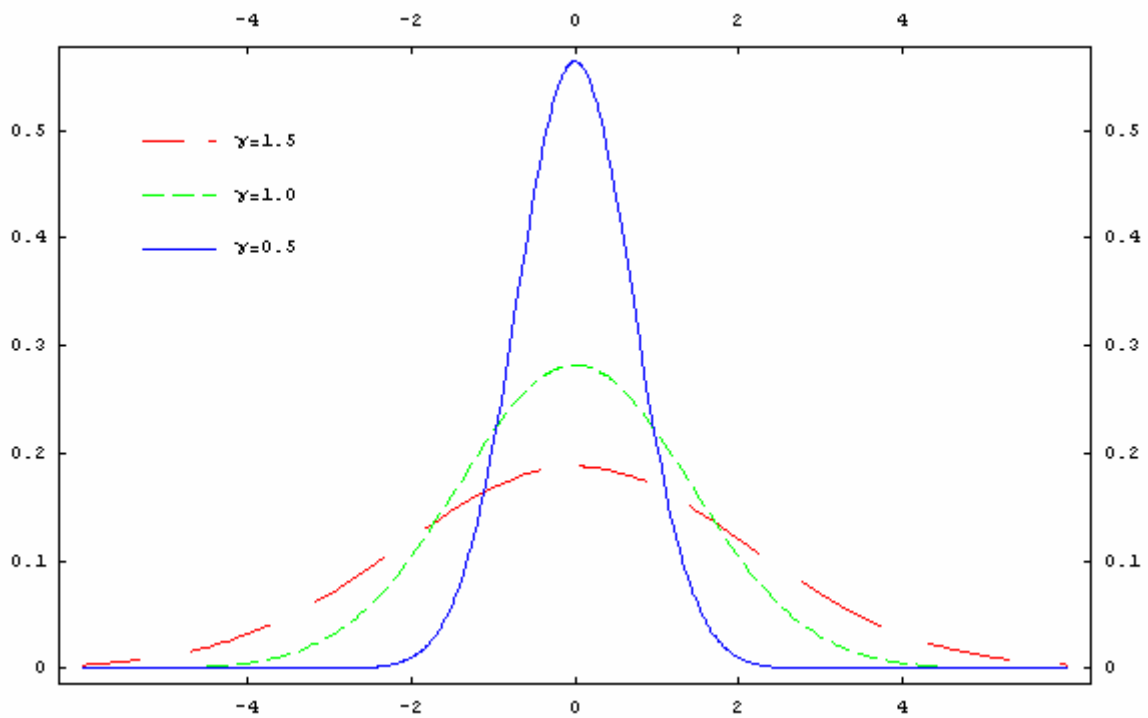


Figura 3. Funções de densidade para distribuições estáveis com parâmetros $S(2, 0, \gamma, 0)$.

A Figura 4 ilustra alguns exemplos do *parâmetro de locação* $\delta \in \mathfrak{R}$. Através desse parâmetro define-se o valor médio ou ponto de equilíbrio da função densidade de probabilidade. No exemplo a seguir, quando $\delta = 0$, a curva encontra-se centrada em zero.

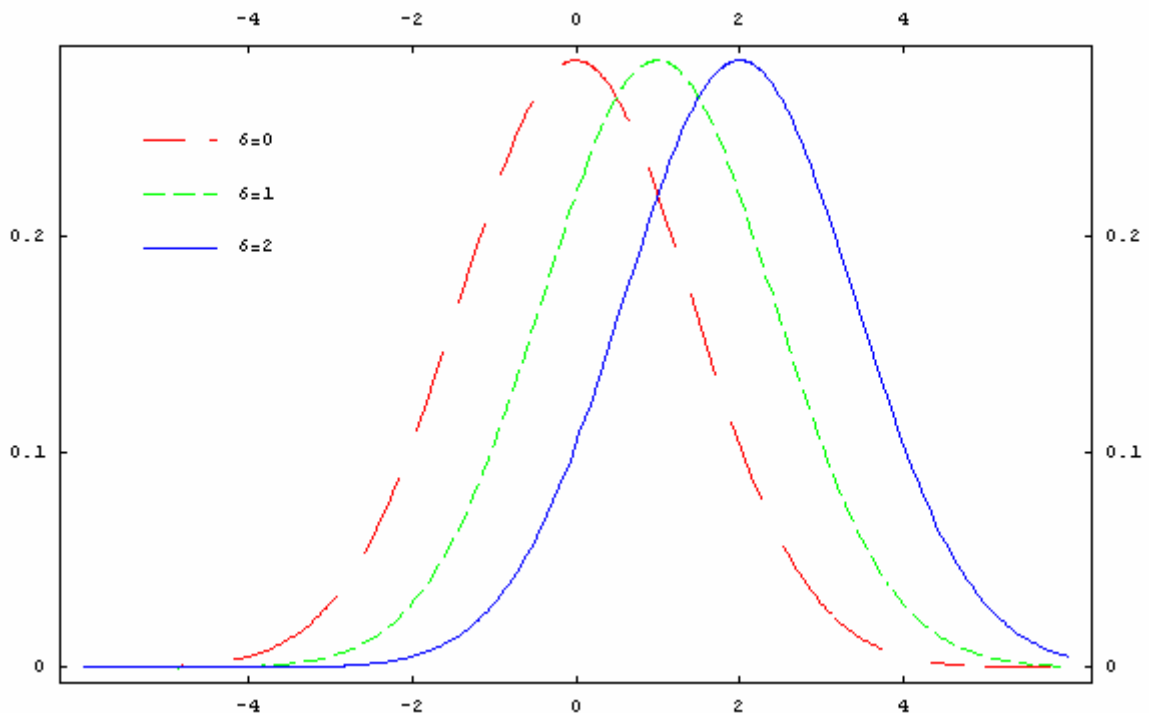


Figura 4. Funções de densidade para distribuições estáveis com parâmetros $S(2, 0, 1, \delta)$.

Através de uma distribuição estável é possível obter qualquer outra distribuição também estável, bastando para isso, fazer uma correta configuração de seus parâmetros característicos (α , β , γ e δ). Para exemplificar, considere as três distribuições de probabilidade estáveis mais conhecidas: Normal, de Cauchy e de Lévy. Uma distribuição estável com parâmetros $S(2, 0, \sigma^2/2, \mu)$ é equivalente a uma distribuição Normal $N(\mu, \sigma^2)$, enquanto que uma estável $S(1, 0, \gamma, \delta)$ corresponde a uma distribuição de Cauchy $C(\gamma, \delta)$ e finalmente uma estável $S(1/2, 1, \gamma, \delta + \gamma)$ é equivalente a uma distribuição de Lévy $L(\gamma, \delta)$.

A rigor, as distribuições de Lévy constituem uma classe de distribuições assimétricas à direita. No entanto, a distribuição de Lévy estudada por [Lee e Yao, 2004] que também é objeto de estudo nesse trabalho, refere-se a uma estável com parâmetros $S(\alpha, 0, 1, 0)$, que é simétrica e centrada em zero, ou seja, um caso específico de estável que pode ser denominada de ‘Lévy simétrica’. Por convenção, sempre que for utilizado o termo Lévy no presente estudo, refere-se a distribuição Lévy simétrica que é equivalente à uma estável $S(\alpha, 0, 1, 0)$.

A Figura 5 ilustra os três exemplos citados anteriormente com o parâmetro de escala γ fixado em 0 e o parâmetro de dispersão δ fixado em 1. Percebe-se que a função densidade de

probabilidade da Normal com seu formato característico de ‘sino’, concentra a maioria dos seus pontos nas redondezas em torno da média até dois desvios-padrão ($\mu \pm 2\sigma$), onde $\sigma = 1$ para uma Normal padrão. A distribuição de Cauchy, apesar da semelhança com a Normal, apresenta maior probabilidade de ocorrência de pontos além da fronteira de dois desvios-padrão, tanto para o lado negativo quanto para o lado positivo do eixo das abcissas. A distribuição de Lévy simétrica é a que concentra maior probabilidade bem próximo da região da média ($\mu = 0$).

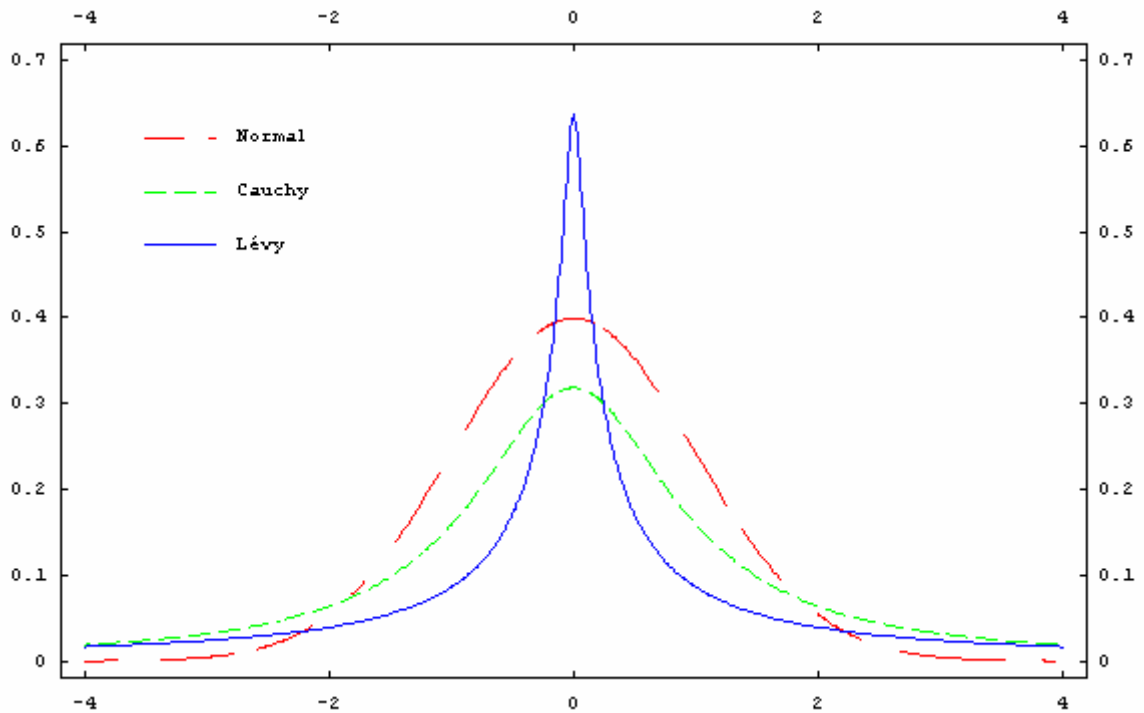


Figura 5. Comparativo entre as funções densidade de probabilidade da Normal padrão, da distribuição de Cauchy e de Lévy com parâmetros de escala $\gamma = 0$ e dispersão $\delta = 1$.

A Figura 6 compara as distribuições de probabilidade acumuladas de uma Normal, Cauchy e de Lévy. Analisando o eixo das abcissas no intervalo $x \notin (-2, +2)$, percebe-se que as distribuições de Cauchy e Lévy apresentam maior probabilidade de ocorrência quando comparadas à distribuição Normal. É por esse motivo que elas são denominadas como distribuições de cauda pesada (*heavy tails*) [Rimmer e Nolan, 2005], ou gorda, ou longa [Anderson, 2006].

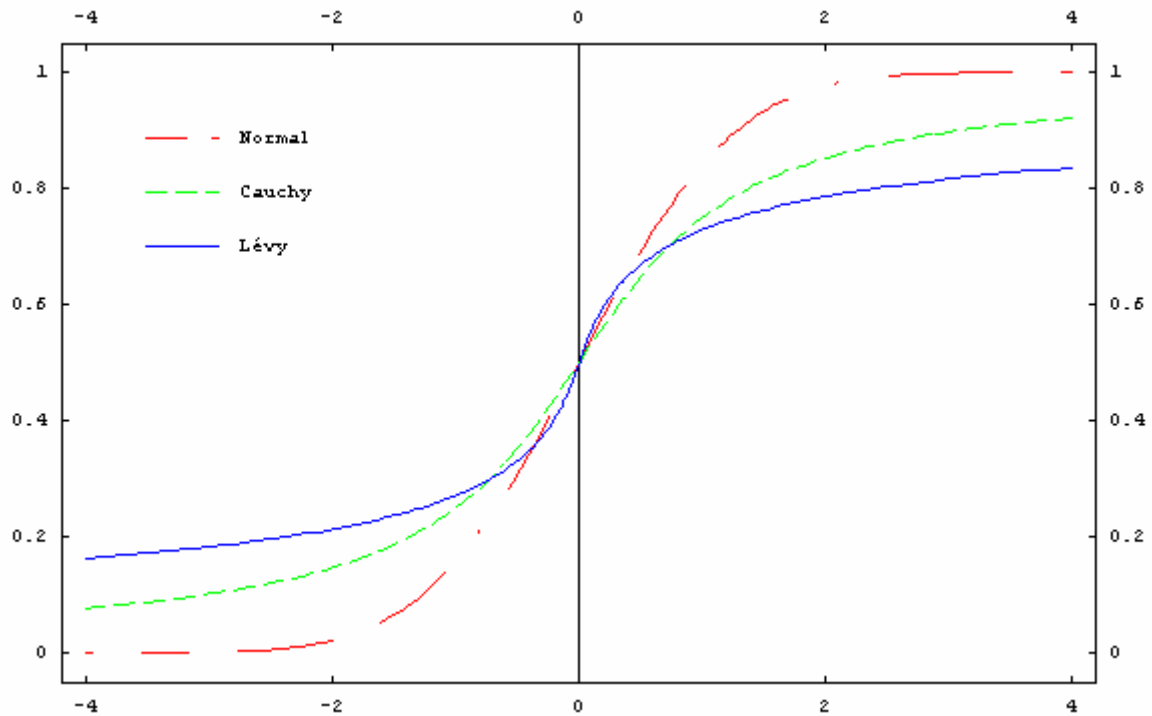


Figura 6. Comparativo entre as distribuições de probabilidade acumuladas de uma Normal padrão, da distribuição de Cauchy e de Lévy com parâmetros de escala $\gamma=0$ e dispersão $\delta=1$.

Outra forma de verificar a mesma informação é através da Tabela 1 que apresenta o comparativo numérico probabilístico das distribuições Normal, Cauchy e de Lévy para alguns valores de x , retirado de [Nolan, 2005]. Para exemplificar, considere as probabilidades na linha da tabela onde $x=3$. Enquanto que, para a distribuição Normal, a probabilidade de ocorrência de um número nessa faixa é um evento raro (0.1%), para a distribuição de Cauchy e de Lévy existe uma probabilidade estatisticamente significativa de ocorrência desse mesmo número, 10.2% e 18,3% respectivamente.

	P(X > x)		
x	Normal	Cauchy	Lévy
0	0.500000	0.500000	0.500000
1	0.158655	0.250000	0.271280
2	0.022750	0.147584	0.213928
3	0.001350	0.102416	0.183545
4	0.000032	0.077979	0.163773

Tabela 1. Comparativo entre as distribuições de probabilidade de uma Normal padrão, da distribuição de Cauchy e de Lévy com parâmetros de escala $\gamma=0$ e dispersão $\delta=1$.

Capítulo 3: Programação Evolutiva

3.1 Introdução: Computação Evolutiva

Evolução em sua essência é um processo composto de duas etapas, uma variação aleatória e uma subsequente seleção [Mayr, 1988]. Uma população de indivíduos exposta a um determinado ambiente produz uma série de comportamentos que estão pior ou melhor adaptados às demandas desse ambiente. Um processo seletivo é capaz de eliminar os indivíduos menos adaptados, permitindo que os indivíduos restantes reproduzam-se entre si, gerando novos indivíduos que carregam em seu genótipo as características impostas por esse meio.

Um processo evolutivo elementar pode ser modelado através de um algoritmo evolutivo e simulado em um computador [Fogel, 1994; Bäck, Hammel e Schwefel, 1997], na forma de uma equação de diferenças

$$x[t+1] = s(v(x[t])) \quad (8)$$

onde uma população no tempo t , denotada por $x[t]$, é operada por uma função de variação v seguida de uma função de seleção s , gerando uma nova população $x[t+1]$. Após sucessivas iterações de variação e seleção, o algoritmo evolui a população guiando-a em direção às soluções de maior valor da função objetivo.

Define-se função de avaliação uma entidade capaz de atribuir uma ‘nota’ ao desempenho de cada indivíduo da população, sem que ela precise deter conhecimento de como encontrar a solução para o problema em questão.

Assim como a função de avaliação está relacionada ao processo de aptidão dos indivíduos na busca da solução, o conceito de variabilidade está relacionado à criação dos operadores que devem ser aplicados à população para obter os indivíduos da próxima geração. Tais operadores costumam ser cópias conceituais de fenômenos genéticos existentes na natureza e são do tipo:

- *Seleção*: seleciona um ou mais ancestrais bem avaliados pela entidade de avaliação, tornando-os elementos da geração seguinte sem nenhuma mudança de suas características genéticas;
- *Recombinação*: operador que simula a criação de um novo indivíduo a partir da troca de material genético entre dois ancestrais, bem avaliados pela entidade de avaliação;
- *Mutação*: operador que simula a criação de um novo indivíduo a partir da introdução de modificações aleatórias no material genético de um ancestral.

Um outro conceito básico em computação evolutiva trata da necessidade de que a primeira geração de indivíduos seja gerada aleatoriamente e que os indivíduos contenham intrinsecamente os parâmetros necessários para obter a solução do problema. Esse conjunto de características genotípicas dos indivíduos de uma população são definidos como parâmetros objetivos, pois os diferenciam dentre as possíveis soluções de um problema.

Classicamente a computação evolutiva pode ser subdividida em quatro famílias principais de algoritmos: a programação evolutiva, a programação genética, os algoritmos genéticos e as estratégias evolutivas [Bittencourt, 2006; Eiben e Smith, 2003]. Essa divisão é feita de acordo com a importância e a operacionalização dos conceitos vistos acima, quando de sua implementação em modelos computacionais.

A programação evolutiva que foi desenvolvida pela primeira vez na atividade de predição do comportamento de máquinas de estado finito, e utilizavam apenas os operadores de seleção e mutação. Suas idéias iniciais não foram muito consideradas pela comunidade da computação evolutiva por desprezar o uso do operador genético de recombinação. Ainda que a proposta original tratasse de predição de comportamento de máquinas de estado finitos, o enfoque da programação evolutiva se adapta a qualquer estrutura de problema. Aqui, a representação sempre deriva do problema. Por exemplo, uma rede neural pode ser representada tal como é implementada, pois a mutação não exige uma codificação linear.

Outra família de interesse no presente estudo são as estratégias evolutivas, não pela sua abordagem, que diferentemente da programação evolutiva utiliza a recombinação como operador secundário, mas sim pela idéia de auto-adaptação. Suas primeiras aplicações, em meados dos anos 60, tiveram como adeptos engenheiros e grupos científicos da Alemanha [Goldberg, 1989].

3.2 Mecanismos Adaptativos

De acordo com sua taxonomia, os algoritmos adaptativos podem ser classificados primeiramente pelo tipo e, segundo, pelo nível de adaptação. O tipo de adaptação define como o parâmetro é modificado, enquanto que o nível define onde a mudança ocorre.

Quanto ao tipo de adaptação, os algoritmos são divididos em *estáticos*, quando não ocorre mudança nos parâmetros estratégicos durante o processo evolutivo, e *dinâmicos*, em caso contrário. O termo adaptação dinâmica é usado para classificar algoritmos onde os parâmetros estratégicos, responsáveis pela configuração ou controle do esquema adaptativo, mudam de acordo com algumas regras, sem a interferência de fatores externos. Quanto ao nível, as adaptações podem ocorrer no *componente*, no *indivíduo*, na *população*, ou até mesmo ao nível de *ambiente*, quando sua interferência no processo evolutivo é dinâmica [Nieberg e Beyer, 2006].

Denomina-se componente em computação evolutiva, o conjunto de características genéticas presentes em cada indivíduo.

Baseado no mecanismo de adaptação, os algoritmos podem ser subdivididos em três classes: *determinístico*, *adaptativo* e *auto-adaptativo*. Um mecanismo adaptativo pode ser considerado determinístico se os parâmetros estratégicos são modificados de acordo com uma regra determinística, sem obter nenhuma informação da população ao longo das gerações. Um exemplo dessa classe de adaptação é o controle temporal da taxa de mutação ao longo do processo evolutivo [Holland, 1992]. Já os algoritmos com regra de adaptação dinâmica obtêm as informações da população, e em cada nova geração, modificam apropriadamente os parâmetros estratégicos. Finalmente, os algoritmos auto-adaptativos realizam uma etapa de mutação dos próprios parâmetros estratégicos que controlam o operador de mutação. Naturalmente, como os esquemas adaptativo e auto-adaptativo dependem diretamente de informação advinda da própria evolução, cujo resultado é sempre de natureza estocástica, esses esquemas são não determinísticos.

O foco do presente trabalho é a utilização de programação evolutiva na implementação de um algoritmo com conceitos de auto-adaptação no operador de mutação. A escolha da mutação como único operador de variabilidade deve-se a necessidade de avaliar o efeito do uso da distribuição estável no processo evolutivo, independente de outras interferências.

3.3 Operadores Genéticos

3.3.1 Mutação Gaussiana

Estudos demonstram que a eficiência de um algoritmo evolutivo na resolução de um problema de otimização está diretamente relacionada com o processo de variabilidade do operador de mutação [Koenig, 2002; Lee e Yao, 2004; Yao e Liu, 1996].

Mais do simplesmente gerar uma variabilidade na etapa de mutação, esses estudos tratam da aplicação de diferentes distribuições de probabilidade, dentre aquelas que sejam mais adequadas à resolução de um problema específico.

Intuitivamente, uma escolha considerada tradicional, seria aquela cuja distribuição de probabilidade esteja mais relacionada aos fenômenos biológicos, dado que um algoritmo deva representar com a maior exatidão possível um processo evolutivo. Uma candidata natural para esse fim seria a distribuição Normal de probabilidade devido a seu amplo domínio de atuação na modelagem de fenômenos naturais.

Nessa linha, o método clássico de mutação adaptativa, sugerido inicialmente por Rechenberg e Schwefel [Beyer, 2001], utiliza como função geradora de variabilidade uma distribuição de probabilidade Normal padrão, definida conforme a Equação 1.

Considere σ um parâmetro estratégico ponderador de um processo aleatório associado à mutação do parâmetro objetivo. A mutação de σ' é dada por

$$\sigma' = \sigma \exp[\tau z], \quad (9)$$

onde σ' é o parâmetro estratégico descendente de σ , z é um número aleatório com distribuição Normal padrão $N(0, 1)$, e τ é um parâmetro estratégico conhecido como *taxa de aprendizado*. O valor de $\tau \propto \sqrt{n}$, onde n é a dimensão da função objetivo, é considerado ótimo para otimização da distribuição quadrática, porém sua recomendação pode ser diferente [Bäck e Schwefel, 1993]. A escolha de $z = N(0, 1)$ é referenciada como uma regra de mutação “log-normal” do parâmetro σ' , devido à semelhança da distribuição dos pontos gerados por essa transformação com a distribuição de probabilidade Log-Normal [Nieberg e Beyer, 2006].

Usando o novo valor σ' , ocorre a mutação do parâmetro objetivo x , conforme

$$x'_i = x_i + \sigma' N(0, 1). \quad (10)$$

Esse método é referenciado como *Classical Evolutionary Programming* (CEP) [Koenig, 2002; Lee e Yao, 2004; Yao e Liu, 1996].

3.3.2 Mutação de Cauchy

A primeira aplicação de uma distribuição de Cauchy em programação evolutiva foi introduzida por [Yao e Liu, 1996], para otimização de funções de teste com um grande número de mínimos locais, capazes de dificultar a convergência do algoritmo em busca do ótimo global. Esse método foi chamado de *Fast Evolutionary Programming* (FEP). O algoritmo apresenta as mesmas características da Equação 10, substituindo a função geradora de variabilidade definida através de uma Normal padrão $N(0, 1)$ por C_j , conforme

$$x'_i = x_i + \sigma' C_j \quad (11)$$

onde C_j é uma variável aleatória j dimensional, com distribuição de probabilidade de Cauchy dada por

$$f(x) = \frac{1}{\pi} \frac{\gamma}{\gamma^2 + (x - \delta)^2}, \quad (12)$$

com $x \in (-\infty, +\infty)$, $\gamma = 1$, e $\delta = 0$. Percebe-se através da Figura 5 que a distribuição de Cauchy padrão $C(0, 1)$ apresenta uma cauda mais ‘pesada’ que a distribuição Normal padrão, aumentando a probabilidade de ocorrência de números aleatórios no intervalo $x \notin (-2, +2)$.

3.3.3 Mutaç o de L vy

Uma terceira aplica o tamb m sugerida por [Lee e Yao, 2004], para otimiza o de fun es de teste com superf cies de busca com v rios  timos locais, foi denominada *L vy Evolutionary Programming* (LEP). O algoritmo apresenta as mesmas caracter sticas da Equa o 10, substituindo a fun o geradora de variabilidade de uma Normal padr o $N(0, 1)$ por uma vari vel aleat ria de L vy L_α conforme

$$x'_i = x_i + \sigma' L_j(\alpha), \quad (13)$$

onde $L_j(\alpha)$   uma vari vel aleat ria de L vy j dimensional, com fun o densidade de probabilidade dada por

$$f_{\alpha,\gamma}(x) = \frac{1}{\pi} \int_0^\infty \exp[-\gamma q^\alpha] \cos[qx] dq, \quad (14)$$

com $y \in \Re$, e $\gamma = 1$ e $\alpha \in (0, 2]$. Percebe-se que a distribui o de L vy padr o $L(0, 1)$, conforme apresentado pela Figura 5, tem uma cauda ainda mais ‘pesada’ que a distribui o de Cauchy padr o, quando comparada com a distribui o Normal padr o. Por esse motivo, essa distribui o possui uma capacidade ainda maior de busca em superf cies multimodais. Percebe-se tamb m que a distribui o de L vy apresenta uma grande probabilidade de ocorr ncia de n meros aleat rios nas proximidades de zero. Essa caracter stica permite que o resultado final de um processo de otimiza o utilizando L vy seja muito refinado, pois nas proximidades do  timo global o n vel de variabilidade deve ter baixa amplitude.

3.3.4 Operadores de Muta o Combinados

Outra forma de abordagem   a combina o dos m todos evolutivos apresentados nas se es anteriores. Em [Chellapilla, 1998] foi proposto a cria o de dois novos operadores denominados *Mean Mutation Operator* (MMO) e *Adaptative Mean Mutation Operator* (AMMO). Nos dois casos esses operadores utilizam uma combina o entre a gera o de variabilidade atrav s das distribui es de probabilidade Normal e de Cauchy.

Um algoritmo modelado a partir de um MMO pode ser descrito como

$$x'_i = x_i + 0.5 \sigma' (N(0, 1) + C(0, 1)), \quad (15)$$

enquanto que, um algoritmo modelado a partir de um AMMO, pode ser descrito como

$$x'_i = x_i + \sigma'_1 N(0, 1) + \sigma'_2 C(0, 1), \quad (16)$$

onde $N(0, 1)$   a distribui o de probabilidade Normal padr o e $C(0, 1)$   uma distribui o de Cauchy padr o. Os par metros σ'_1 e σ'_2 s o definidos da forma

$$\sigma'_i = \sigma_i \exp[\tau' N(0, 1) + \tau N_j(0, 1)], \quad (17)$$

onde os parâmetros estratégicos τ e τ' são definidos, segundo recomendação de [Bäck e Schwefel, 1993], como

$$\tau = \frac{1}{\sqrt{2\sqrt{\mu}}} \text{ e } \tau' = \frac{1}{\sqrt{2\mu}}, \quad (18)$$

onde μ é o número de indivíduos da população.

3.4 Métodos de Seleção

O conceito desse operador é simular o processo de reprodução assexuada (sem recombinação), permitindo que os indivíduos mais adaptados tornem-se elementos da geração seguinte enquanto que os demais sejam extintos. Diferentemente do elitismo que seleciona os n melhores indivíduos avaliados pela função de aptidão, um método de seleção probabilístico permite que os indivíduos melhor avaliados tenham maior chance de seleção, mas sem a garantia de que ela realmente aconteça. Por esse motivo o método de seleção probabilístico é considerado mais natural, pois, apesar de aumentar a chance de seleção dos melhores indivíduos, não elimina a possibilidade de seleção dos indivíduos menos aptos.

Dentre os operadores de seleção probabilísticos a seleção por *torneio* pode ser descrita da seguinte forma:

1. Sejam i e j elementos de uma população composta por m indivíduos;
2. Para cada indivíduo i da população, selecionar aleatoriamente um outro indivíduo j e comparar o resultado de suas funções de aptidão;
3. Caso o resultado da função de aptidão do indivíduo i seja melhor que o resultado da função de aptidão do indivíduo j , então o indivíduo i fará parte da geração seguinte enquanto que o indivíduo j será desprezado.

Na Etapa 2, um determinado indivíduo pode ser comparado com um ou mais indivíduos em uma mesma rodada, de acordo com o *tamanho* desse torneio (no caso acima, de tamanho 2). Quanto ao processo de seleção dos j indivíduos que farão parte do torneio do indivíduo i , este pode ser caracterizado como aleatório ou determinístico. O critério de seleção *determinístico* é composto por uma regra de seleção julgamental definida pelo pesquisador, enquanto que o critério *não determinístico* procura atribuir a mesma probabilidade de seleção para cada um dos indivíduos da população.

O critério de seleção *não determinístico* ainda pode ser subdividido em dois tipos: com ou sem repetição. Quando este for com repetição, um mesmo indivíduo j pode ser selecionado mais

de uma vez para participar do torneio classificatório com o indivíduo i ; caso contrário, o indivíduo j será selecionado aleatoriamente apenas uma vez.

Capítulo 4: Os Algoritmos Utilizados

Este capítulo trata da definição dos algoritmos referenciados pela literatura e sugestões de aplicações auto-adaptativas para o mesmo problema. Todos os experimentos e códigos de programação desenvolvidos utilizaram o software *Mathematica* (versão 5.2), da Wolfram Research.

4.1 A Família de Algoritmos de Yao e Colaboradores

O termo família descrito no título dessa seção, refere-se a três diferentes versões de algoritmos desenvolvidos em diferentes artigos pelo mesmo autor [Yao e Liu, 1996; Yao, Liu e Lin, 1999; Lee e Yao, 2004].

A descrição do algoritmo se dá contexto de um processo de minimização, cujo objetivo é encontrar um vetor \vec{x}_{\min} de tal forma que $\forall \vec{x}$, valha a seguinte desigualdade $f(\vec{x}_{\min}) \leq f(\vec{x})$ [Bäck, Hammel e Schwefel, 1997].

Assim, considere-se uma população com m indivíduos, representada pelo par ordenado $(\vec{x}_{ij}, \vec{\sigma}_{ij})$, onde o índice $i=1, 2, \dots, m$ representa cada um dos indivíduos, cada \vec{x}_i é um vetor de parâmetros objetivos, e $\vec{\sigma}_i$ um vetor de parâmetros estratégicos correspondente. Considere-se ainda que cada indivíduo possui uma quantidade n de componentes associados a seus parâmetros objetivos, valendo a seguinte representação:

$$\vec{x}_{ij} = \{x_{i1}, x_{i2}, \dots, x_{in}\}, \quad (19)$$

$$\vec{\sigma}_{ij} = \{\sigma_{i1}, \sigma_{i2}, \dots, \sigma_{in}\}. \quad (20)$$

Com base no exposto, as etapas de uma programação evolutiva são feitas da seguinte forma:

1. A partir de uma população inicial definida pelo par $(\vec{x}_{ij}, \vec{\sigma}_{ij})$, gerar filhos $(\vec{x}'_{ij}, \vec{\sigma}'_{ij})$, da seguinte maneira:

$$\sigma'_{ij} = \sigma_{ij} \exp[\tau' N(0, 1) + \tau N_j(0, 1)], \quad (21)$$

$$x'_{ij} = x_{ij} + \sigma'_{ij} N_j(0, 1), \quad (22)$$

onde $N(0, 1)$ é um número aleatório gerado de uma distribuição Normal com média $\mu=0$ e variância $\sigma^2=1$, $N_j(0, 1)$ é um vetor j dimensional de números com as mesmas propriedades estocásticas do anterior, e os parâmetros estratégicos τ e τ' são definidos conforme a Equação 18.

2. Para todos os indivíduos da população, incluindo pais e filhos, aplicar a função objetivo $f(\bar{x}_i)$, para $i=1, 2, \dots, 2m$.
3. Inicializar uma função de avaliação $w_i(q)$ para todos os indivíduos dessa população como $w_i(q) = 0$, para $i=1, 2, \dots, 2m$. Para cada indivíduo i , realizar uma quantidade q de torneios de tamanho 2, selecionando outro indivíduo j onde $i \neq j$, e comparar o resultado de suas funções objetivo. Se $f(\bar{x}_i) < f(\bar{x}_j)$, então a função de avaliação $w_i(q)$ original desse indivíduo é acrescida de uma unidade.
4. Selecionar, dentre pais e filhos, os m melhores indivíduos, através dos maiores valores obtidos em suas respectivas funções de avaliação $w_i(q)$, para obter a população seguinte.
5. Repetir as Etapas de 1 a 4 até que o critério de parada seja satisfeito.

A propósito da definição do algoritmo, existe três diferentes versões, cuja principal diferença está na forma de determinação do parâmetro objetivo dos descendentes x'_{ij} , que pode ser função

- Do parâmetro estratégico do pai σ_{ij} ou do próprio filho σ'_{ij} gerado, e
- Da distribuição geradora de variabilidade, que pode ser unidimensional $N(0, 1)$ ou multidimensional $N_j(0, 1)$.

Para se ter uma idéia das possíveis diferenças encontradas em suas definições, em duas das três proposições [Yao e Liu, 1996; Yao, Liu e Lin, 1999], o mesmo parâmetro estratégico do pai σ_{ij} é utilizado para compor o parâmetro objetivo dos descendentes, porém a distribuição geradora de probabilidade é unidimensional na primeira referência e multidimensional na segunda. A terceira definição citada em [Lee e Yao, 2004], apesar de possuir uma distribuição geradora de probabilidade multidimensional, semelhante a [Yao, Liu e Lin, 1999], utiliza o parâmetro estratégico do filho σ'_{ij} na mutação do parâmetro objetivo dos descendentes.

4.2 Novos Algoritmos Propostos

Uma questão que pode ser levantada a partir das diferentes versões de algoritmos citadas na Seção 4.1 é saber se elas atuam significativamente sobre os resultados dos experimentos. Para isso, considere três versões de operadores de mutação que combinam essas características:

$$x'_{ij} = x_{ij} + \sigma'_{ij} N(0, 1), \quad (23)$$

$$x'_{ij} = x_{ij} + \sigma_{ij} N_j(0, 1), \quad (24)$$

$$x'_{ij} = x_{ij} + \sigma_{ij} N(0, 1). \quad (25)$$

A Equação 23 traz as mesmas características que a Equação 22 quanto à utilização do parâmetro estratégico do filho σ'_{ij} para a determinação do parâmetro objetivo x'_{ij} , exceto no que se refere à dimensão da distribuição Normal $N(0, 1)$, que agora passa a ser unidimensional. Por outro lado, as Equações 24 e 25 utilizam o parâmetro estratégico σ_{ij} do pai na determinação do parâmetro objetivo x'_{ij} , mas diferem na dimensão da distribuição geradora de variabilidade, que na primeira é multidimensional – i.e., $N_f(0, 1)$ – e na segunda é unidimensional, i.e., $N(0, 1)$.

Sabendo-se que as definições tratadas nas Seções de 4.2.1 a 4.2.4 tem como base o algoritmo citado na Seção 4.1, acrescentando-se a cada uma delas suas respectivas particularidades, considere os seguintes detalhes:

1. Todo algoritmo de otimização deve respeitar o espaço de busca envolvido. No entanto, [Lee e Yao, 2004] não explicita como o algoritmo deve se comportar na situação em que o parâmetro objetivo x'_{ij} gerado para um filho ultrapasse os limites do espaço de busca. Assim, o presente estudo modificou a Etapa 1 no sentido de se utilizar a Equação 22 somente se o valor x'_{ij} gerado estiver dentro dos limites estabelecidos pela função de teste em questão; caso contrário, seu valor simplesmente mantém o do parâmetro objetivo x_{ij} do pai. Alternativas a essa poderiam ter sido utilizadas, como por exemplo, gerar um novo número aleatório, ou levar o parâmetro ao limite da fronteira atingida; no entanto, o impacto dessa alternativas não foi aqui verificado.
2. Como todas as aplicações envolvendo o algoritmo tratam da resolução de problemas de minimização, fica implícito na Etapa 2 do algoritmo citado na Seção 4.1, que o resultado esperado pela função objetivo é aquele que atingir o menor valor absoluto.
3. Na Etapa 3 do algoritmo citado na Seção 4.1, devido à ausência da definição do mecanismo de torneio citado em [Lee e Yao, 2004], quanto ao fato de ser determinístico ou aleatório (com ou sem reposição), o torneio implementado no presente estudo utiliza uma seleção aleatória sem repetição.

4.2.1 Auto-Adaptatação no Parâmetro α da Distribuição Estável

O conceito de auto-adaptatação em um processo evolutivo está diretamente relacionado à possibilidade de ajustes nos parâmetros que controlam o processo de mutação ao longo das gerações. Mais do que isso, as adaptações dos parâmetros estratégicos devem ser feitas com base nas informações existentes dentro da população que está sendo evoluída. Uma forma de promover o aumento no grau de liberdade dos parâmetros estratégicos seria utilizar no operador de mutação uma função geradora de variabilidade que assumisse diferentes formas ou que

simplesmente englobasse as principais classes de distribuições de probabilidade existentes, tais como a Normal, de Cauchy e de Lévy. Para isso o algoritmo poderia utilizar a distribuição de probabilidade estável $S(\alpha, \beta, \gamma, \delta)$ na geração de variabilidade do operador de mutação, conforme apresentada no Capítulo 2.

A idéia de auto-adaptação dos parâmetros estratégicos de uma distribuição estável que será apresentada neste trabalho foi motivada pelo esquema adaptativo proposto em [Lee e Yao, 2004], que utilizava o algoritmo definido na Seção 4.1 combinado com o operador de mutação de Lévy descrito na Seção 3.3.3. Nesse esquema adaptativo, para cada indivíduo da população inicial, gerava-se 4 novos descendentes utilizando valores fixos do parâmetro estratégico α da distribuição de Lévy no domínio $\{1.0, 1.3, 1.7, 2.0\}$, sendo que o descendente mais apto era selecionado para compor a geração seguinte, enquanto os demais eram extintos.

O caráter adaptativo do método sugerido em [Lee e Yao, 2004] representa um diferencial frente aos métodos determinísticos, pois considera as informações da população na definição do parâmetro estratégico α , fazendo com que o processo se aproxime dos fenômenos naturais, cuja intervenção do homem é a mínima possível.

Porém, como uma pequena variação do parâmetro α da Lévy – ou parâmetro α da estável $S(\alpha, 0, 1, 0)$ – pode representar classes completamente distintas de distribuições de probabilidades, torná-lo auto-adaptativo pode ser um diferencial ainda maior na determinação da curva mais adequada para a resolução de um problema específico, haja visto que o método sugerido em [Lee e Yao, 2004] utiliza um número finito de valores α .

Possivelmente inspirados por esse mesmo tipo de idéia, [de Oliveira e Gutierrez, 2006; Gutierrez e de Oliveira, 2007] trabalharam com aplicações auto-adaptativas dos parâmetros estratégicos de uma distribuição estável, com o objetivo de encontrar o algoritmo mais apto para a resolução de problemas de minimização como os aqui tratados. Utilizando estratégias evolutivas, o estudo implementou um algoritmo utilizando dois mecanismos de seleção (estratégias *soma* e *vírgula*) proposto por Rechenberg e Schwefel [Beyer, 2001]. Para permitir a aplicação auto-adaptativa, o genótipo do indivíduo assumiu a seguinte forma $[x_1, x_2, \{\alpha_{11}, \beta_{12}, \gamma_{13}, \delta_{14}\}, \{\alpha_{21}, \beta_{22}, \gamma_{23}, \delta_{24}\}]$, onde: x_1 e x_2 são os parâmetros objetivos cujo ponto mínimo deseja-se localizar; e $\{\alpha_{11}, \beta_{12}, \gamma_{13}, \delta_{14}\}$ e $\{\alpha_{21}, \beta_{22}, \gamma_{23}, \delta_{24}\}$ são os conjuntos de parâmetros estratégicos da distribuição estável que atuam no operador genético de mutação, respectivamente, das componentes x_1 e x_2 .

É nesse contexto que o presente trabalho encontra-se inserido, por um lado utilizando mutações com parâmetros (objetivos e estratégicos) auto-adaptativos associados a distribuições

de probabilidade estáveis, e por outro, fazendo o uso de conceitos de estratégias evolutivas no operador genético de mutação.

Quanto à seleção dos parâmetros da distribuição estável que devem fazer parte do processo auto-adaptativo, deve-se levar em consideração a natureza do problema que será resolvido e principalmente sua característica na determinação do formato da curva de probabilidade. O parâmetro α é responsável pela diferenciação das famílias de distribuições de probabilidade, portanto, é um parâmetro de interesse no processo adaptativo. Já os parâmetros β e δ são responsáveis pela definição do grau de simetria e locação da curva de probabilidade. Considerando a natureza do problema e, sabendo-se que o papel da distribuição de probabilidade é promover um acréscimo de variabilidade ao processo de mutação, tanto β quanto δ deveriam ser fixados em zero para que a aleatoriedade seja considerada homogênea. O conceito de homogeneidade refere-se a estabelecer uma mesma proporção na geração de números aleatórios, tanto para valores positivos quanto negativos. Outro parâmetro de interesse no processo auto-adaptativo é o parâmetro γ que é responsável pela determinação da intensidade da dispersão dos números aleatórios em torno de uma média.

A primeira proposta aqui sendo feita de criação de um método auto-adaptativo no mecanismo de mutação é através da evolução do parâmetro α ao longo das gerações, permitindo que seus valores variem livremente no intervalo $(0, 2]$. Para isso considere-se o par ordenado $(\vec{\alpha}_i, \vec{\sigma}_\alpha)$ onde $\vec{\alpha}_i$ é um vetor de parâmetros estratégicos para cada indivíduo i da população, com seus respectivos desvios-padrão associados $\vec{\sigma}_\alpha$, conforme:

$$\vec{\alpha}_i = \{\alpha_1, \alpha_2, \dots, \alpha_m\}, \quad (26)$$

$$\vec{\sigma}_\alpha = \{\sigma_1, \sigma_2, \dots, \sigma_m\}. \quad (27)$$

As equações para mutação dos parâmetros estratégicos têm a seguinte forma:

$$\sigma'_\alpha = \sigma_\alpha \exp[\tau' N(0, 1) + \tau N(0, 1)], \quad (28)$$

$$\alpha'_i = \alpha_i + \sigma_\alpha N(0, 0.1), \quad (29)$$

onde $N(0, 1)$ é um número escalar gerado a partir de uma distribuição Normal de média $\mu = 0$ e variância $\sigma^2 = 1$, $N(0, 0.1)$ é um número escalar gerado a partir de uma distribuição Normal de média $\mu = 0$ e variância $\sigma^2 = 0.1$. Os parâmetros estratégicos τ e τ' são definidos conforme a Equação 18. A escolha arbitrária de $\sigma^2 = 0.1$ na Equação 29 deve-se ao fato de que os valores gerados a partir da distribuição Normal $N(0, 1)$ mostraram-se inicialmente pouco eficazes no processo adaptativo do parâmetro α . Possivelmente esse efeito é causado pela pequena

variabilidade requerida pelo parâmetro α , que deve estar contido no intervalo $(0, 2]$, conforme sua definição.

Definido o processo auto-adaptativo do parâmetro estratégico α da distribuição estável, basta incluí-lo na definição do parâmetro objetivo representado pelas Equações de 22 a 25. Portanto, a primeira equação resultante substitui a Normal no processo aleatório associado às mutações por uma distribuição estável unidimensional $S(\alpha_i, \beta, \gamma, \delta)$ aplicada nas Equações 23 e 25, e uma estável $S_j(\alpha_i, \beta, \gamma, \delta)$ com j dimensões para os algoritmos citados nas Equações 22 e 24, resultando em:

Característica		Algoritmo		
Desvio-padrão	Dimensão	Sigla	Definição	Equação
Filho	Multidimensional	FM	$x'_{ij} = x_{ij} + \sigma'_{ij} S_j(\alpha_i, \beta, \gamma, \delta)$	(30)
Filho	Unidimensional	FU	$x'_{ij} = x_{ij} + \sigma'_{ij} S(\alpha_i, \beta, \gamma, \delta)$	(31)
Pai	Multidimensional	PM	$x'_{ij} = x_{ij} + \sigma_{ij} S_j(\alpha_i, \beta, \gamma, \delta)$	(32)
Pai	Unidimensional	PU	$x'_{ij} = x_{ij} + \sigma_{ij} S(\alpha_i, \beta, \gamma, \delta)$	(33)

Tabela 2. Definições dos algoritmos auto-adaptativos que utilizam distribuições de probabilidade estáveis.

Para facilitar a referência aos algoritmos aqui tratados, caracterizados pelas Equações de 30 a 33, foram criadas diferentes siglas mnemônicas para denominá-los. Para exemplificar, considere o primeiro algoritmo da Tabela 2, denominado FM, que contém um parâmetro estratégico do **f**ilho (F) para o desvio-padrão, e utiliza uma distribuição estável **m**ultidimensional (M) no processo aleatório associado às mutações. As demais siglas são definidas pelo uso do parâmetro estratégico do **p**ai (P) e uma distribuição estável **u**nidimensional (U) no processo aleatório associado às mutações.

Essa nova aplicação será referenciada no presente trabalho por SEP, sigla de *Stable Evolutionary Programming*. Apesar de representar uma extensão dos algoritmos GEP e LEP, quanto a generalização no uso da distribuição de probabilidade estável no processo aleatório associado a mutação, a principal diferença do algoritmo SEP incorre no seu aspecto auto-adaptativo.

4.2.2 Aplicação por Nível de Adaptação

Uma questão que surge quando da aplicação do algoritmo diz respeito ao nível de adaptação do parâmetro estratégico α . Note-se que, assim como na natureza, um algoritmo evolutivo é composto por uma seqüência de *gerações*, cada população é composta por vários *indivíduos*, e cada indivíduo apresenta um conjunto de características fenotípicas,

relativamente independentes umas das outras, as quais podem ser vistas como as *componentes* que o definem.

Assim, o *nível de adaptação* em questão refere-se basicamente à definição sobre onde as mudanças do mecanismo adaptativo devem ocorrer, sejam elas no nível de todas as *gerações* do processo evolutivo ou de apenas uma seqüência delas, seja no nível de todos os *indivíduos* da população ou de cada indivíduo isoladamente, ou seja no nível do conjunto integral de *componentes* que definem um indivíduo ou de cada componente individual.

A idéia de trabalhar com diferentes níveis de adaptação, surge do conceito de graus de liberdade que um determinado parâmetro pode adquirir frente à resolução de um problema. De alguma maneira, ao se testar a possibilidade de serem unidimensionais (U) ou multidimensionais (M), assume-se a possibilidade de que existe um nível de adaptação que possa trazer diferentes resultados no processo adaptativo. Alguns estudos comprovam que tornar fixo ou variável um parâmetro por um determinado número de gerações pode ser uma boa estratégia, pois permite um ajuste gradativo da aptidão dos indivíduos da população no processo iterativo, evitando convergência prematura do processo na busca pela solução ótima [Lima e de Oliveira, 2006].

Com o intuito de testar o resultado gerado pelos diferentes níveis de adaptação do parâmetro estratégico α , quanto à possibilidade de ser fixo ou variável, foram geradas novas definições que complementam a lista de algoritmos existentes.

Nível de Adaptação			Algoritmo		
População	Indivíduo	Geração	Sigla	Definição	Equação
Fixo	Fixo	Fixo	FFF	$x'_{ijk} = x_{ijk} + \sigma_{ijk} S_j(\alpha, \beta, \gamma, \delta)$	(34)
Fixo	Fixo	Variável	FFV	$x'_{ijk} = x_{ijk} + \sigma_{ijk} S_j(\alpha_k, \beta, \gamma, \delta)$	(35)
Fixo	Variável	Fixo	FVF	$x'_{ijk} = x_{ijk} + \sigma_{ijk} S_j(\alpha_j, \beta, \gamma, \delta)$	(36)
Fixo	Variável	Variável	FVV	$x'_{ijk} = x_{ijk} + \sigma_{ijk} S_j(\alpha_{jk}, \beta, \gamma, \delta)$	(37)
Variável	Fixo	Fixo	VFF	$x'_{ijk} = x_{ijk} + \sigma_{ijk} S_j(\alpha_i, \beta, \gamma, \delta)$	(38)
Variável	Fixo	Variável	VFV	$x'_{ijk} = x_{ijk} + \sigma_{ijk} S_j(\alpha_{ik}, \beta, \gamma, \delta)$	(39)
Variável	Variável	Fixo	VVF	$x'_{ijk} = x_{ijk} + \sigma_{ijk} S_j(\alpha_j, \beta, \gamma, \delta)$	(40)
Variável	Variável	Variável	VVV	$x'_{ijk} = x_{ijk} + \sigma_{ijk} S_j(\alpha_{ijk}, \beta, \gamma, \delta)$	(41)

Tabela 3. Definições de algoritmos por combinação de níveis de adaptação.

A Tabela 3 apresenta os diferentes algoritmos trabalhados por combinação dos níveis de adaptação. Para exemplificar o forma de leitura da tabela, considere o primeiro algoritmo definido pela Equação 34, que é fixo na população, fixo no indivíduo e fixo na geração. Ser fixo por

população, significa que o processo aleatório associado a todos os indivíduos da população está submetido ao mesmo parâmetro estratégico α . Por outro lado, ser fixo no indivíduo significa que a variação aleatória dos seus componentes é regida pelo mesmo parâmetro estratégico α . Por último, ser fixo por geração significa que o processo aleatório de uma seqüência (ou bloco) de populações está submetida ao mesmo parâmetro estratégico α ; neste contexto, o fato de ser variável por geração implica na manutenção do valor de α por apenas uma geração.

Outro detalhe é que todos os algoritmos citados na Tabela 3 foram formulados de acordo com o algoritmo PM, que utiliza o desvio-padrão do pai σ_{ijk} para mutação do parâmetro objetivo do filho x'_{ijk} e são multidimensionais no processo aleatório associados a distribuição de probabilidade estável. Para acrescentar o nível de adaptação ao longo das gerações foi necessária a inclusão de um índice k , para referenciar os parâmetros objetivos ou estratégicos ao longo das gerações; naturalmente, quando o índice não é utilizado, o parâmetro em questão se mantém inalterado ao longo das gerações. Como o conceito de variabilidade por gerações foi introduzido nesta seção, todos os algoritmos citados nas demais seções, apesar de não apresentar o índice k , são por definição variáveis no nível de geração.

Portanto, uma forma de distinguir os diferentes níveis de adaptação e suas possíveis combinações é através do índice que estiver acompanhando o parâmetro α da distribuição estável. Quando o índice for igual a i , refere-se ao nível adaptação por indivíduo da população, quando for igual a j , refere-se ao nível de adaptação por componente do indivíduo e quando for igual a k , refere-se ao nível de adaptação por geração de populações.

Para facilitar a referência aos algoritmos aqui tratados, foram criadas diferentes siglas mnemônicas para denominá-los. Para exemplificá-las, o primeiro algoritmo da Tabela 3, denominado FFF, é o que contém um parâmetro estratégico F (**f**ixo) por população, um parâmetro F (**f**ixo) por componente, e um parâmetro F (**f**ixo) por geração.

Para fins de comparação, os algoritmos do tipo SEP citados na seção anterior, tratam-se de uma adaptação ao nível de indivíduo e geração. Observe-se que o algoritmo referenciado pela sigla VFV que contém um parâmetro estratégico variável por população, um parâmetro fixo por componente e um parâmetro variável por geração é o próprio algoritmo PM citado na seção anterior.

4.2.3 Auto-Adaptação Elistista

Uma maneira intuitiva de gerar novos descendentes é utilizar as características genéticas do indivíduo que apresenta o melhor desempenho na resolução de determinado problema. Com

isso, o parâmetro estratégico α buscado para a distribuição estável é herdado de pai para filho. Esse tipo de técnica busca reproduzir para os demais indivíduos, as mesmas condições paramétricas que permitiram a criação do indivíduo mais apto.

Para exemplificar, considere a Figura 7 que apresenta um esquema simplificado do mecanismo citado no parágrafo anterior. Dado uma população de indivíduos, seleciona-se através de elitismo o parâmetro estratégico α do indivíduo mais apto (α^*). Para cada novo descendente, define-se o seu parâmetro estratégico α' através da mutação do parâmetro estratégico α^* selecionado. Repare que a mutação do parâmetro estratégico α^* , representado na Figura 7 pela função $f(\cdot)$, é de caráter não determinístico, pois, utiliza-se de uma função geradora de números aleatórios para compor o parâmetro de cada indivíduo.

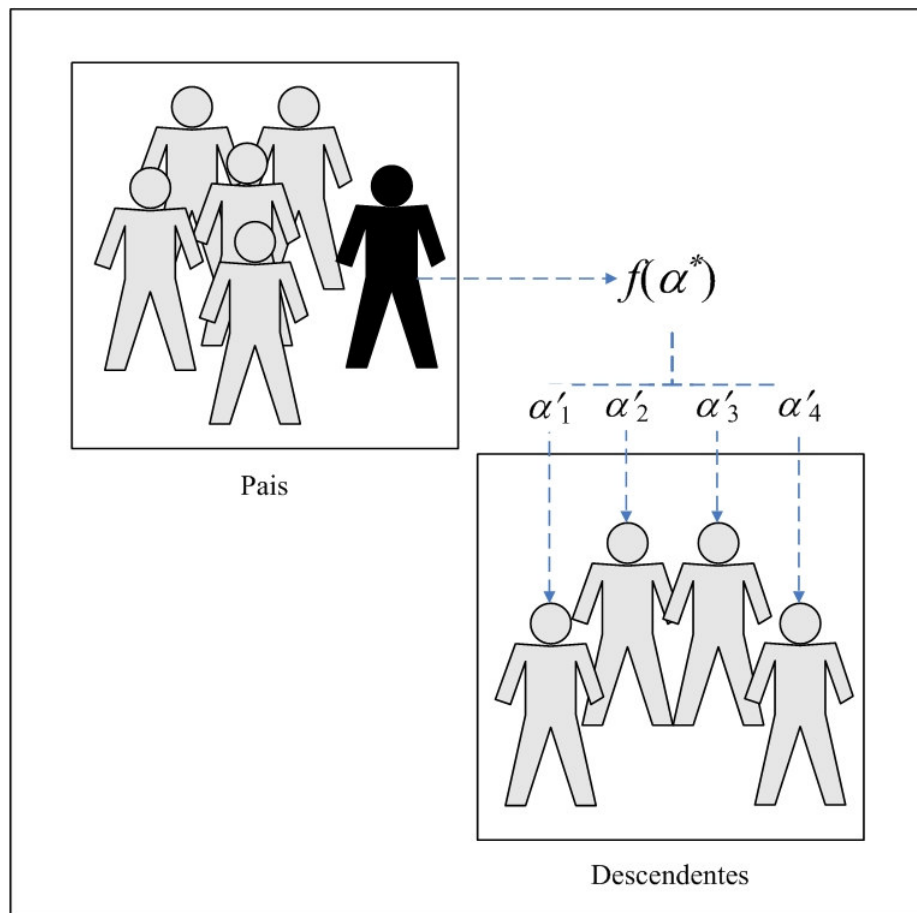


Figura 7. Exemplo do esquema auto-adaptativo elitista no parâmetro estratégico α^* .

A fim de acomodar essa idéia, torna-se necessária uma alteração no algoritmo descrito na Seção 4.1, o que leva à seguinte formulação:

1. A partir de uma população inicial definida por $\{(\bar{x}_{ij}, \bar{\sigma}_{ij}), (\bar{\alpha}_i, \bar{\sigma}_\alpha)\}$, gerar filhos $\{(\bar{x}'_{ij}, \bar{\sigma}'_{ij}), (\bar{\alpha}'_i, \bar{\sigma}'_\alpha)\}$ da seguinte maneira:

$$\begin{aligned}\sigma'_{ij} &= \sigma_{ij} \exp[\tau' N(0, 1) + \tau N_j(0, 1)], \\ x'_{ij} &= x_{ij} + \sigma_{ij} S_j(\alpha_i, \beta, \gamma, \delta), \\ \sigma'_\alpha &= \sigma_\alpha \exp[\tau' N(0, 1) + \tau N(0, 1)],\end{aligned}\tag{42}$$

$$\alpha'_i = \alpha^* + \sigma_\alpha N(0, 0.1),\tag{43}$$

onde σ'_{ij} é definido conforme a Equação 21, x'_{ij} é definido conforme o algoritmo PM, $N(0, 1)$ é um escalar gerado a partir de uma distribuição Normal de média $\mu = 0$ e variância $\sigma^2 = 1$, α^* é o parâmetro estratégico selecionado do indivíduo mais apto, $N(0, 0.1)$ é outro escalar, gerado a partir de uma distribuição Normal de média $\mu = 0$ e variância $\sigma^2 = 0.1$, e os parâmetros estratégicos τ e τ' são definidos conforme a Equação 18.

2. Para todos os indivíduos da população, incluindo pais e filhos, aplicar a função objetivo $f(\bar{x}_i)$, para $i = 1, 2, \dots, 2m$.
3. Inicializar uma função de avaliação $w_i(q)$ para todos os indivíduos dessa população como $w_i(q) = 0$, onde $i = 1, 2, \dots, 2m$. Para cada indivíduo i dessa população, realizar uma quantidade q de torneios de tamanho um, selecionando outro indivíduo j onde $i \neq j$, e compare o resultado de suas funções objetivo. Se $f(\bar{x}_i) < f(\bar{x}_j)$, então a função de avaliação desse indivíduo é acrescida de uma unidade, conforme $w_i(q) = w_i(q) + 1$.
4. Selecione, dentre pais e filhos, os m melhores indivíduos através dos maiores valores obtidos em suas respectivas funções de avaliação $w_i(q)$ para obter a população seguinte.
5. Utilizando a função de avaliação $w_i(q)$, selecione do indivíduo mais apto o parâmetro estratégico α^* de sua distribuição estável $S_j(\alpha_i, \beta, \gamma, \delta)$.
6. Repita as Etapas de 1 a 5 até que o critério de parada seja satisfeito.

Essa nova definição não altera significativamente o algoritmo descrito na Seção 4.1, alterando-o nos seguintes aspectos:

- Substitui a distribuição geradora de variabilidade de uma distribuição Normal $N(0, 1)$ por uma estável $S_j(\alpha_i, \beta, \gamma, \delta)$;
- Acrescenta na Etapa 1 as Equações 42 e 43 para mutação dos parâmetros estratégicos α_i e σ_α , respectivamente; e

- Acrescenta a Etapa 5 para seleção do parâmetro α^* do indivíduo mais apto.

Naturalmente, o parâmetro α^* utilizado pela Equação 43 só passa a ser utilizado a partir da segunda geração de indivíduos, dado que na primeira ele não interfere na definição do gerador de variabilidade $S_j(\alpha_i, \beta, \gamma, \delta)$, presente na equação do parâmetro objetivo x'_{ij} . Na primeira geração, os parâmetros estratégicos α_i dos indivíduos foram inicializados com seus respectivos valores aleatórios.

Para facilitar a citação ao novo algoritmo proposto utilizaremos a sigla EAA, iniciais de *Elitist Auto-Adaptation*.

4.2.4 Auto-Adaptação nos Parâmetros α e γ da Distribuição Estável

Uma outra proposta de método auto-adaptativo no mecanismo de mutação seria permitir a variação do parâmetro γ responsável pelo nível de dispersão dos números aleatórios em torno de uma média. Sabe-se que uma redução ou aumento do parâmetro γ pode influenciar diretamente no processo aleatório associado à distribuição estável, ocasionando a compressão ou achatamento da curva de probabilidade, conforme já ilustrado na Figura 3. Devido à essa propriedade, a auto-adaptação desse parâmetro pode permitir um melhor resultado no processo de otimização, tendo em vista que cada problema a ser resolvido requer um tipo de distribuição de probabilidade, cujo parâmetro permite ajustar.

Então, considere $\vec{\gamma}_i$ um vetor de parâmetros estratégicos para cada indivíduo i da população, e $\vec{\sigma}_\gamma$ seus respectivos desvios-padrão associados, isto é,

$$\vec{\gamma}_i = \{\gamma_1, \gamma_2, \dots, \gamma_j\}, \text{ e} \quad (44)$$

$$\vec{\sigma}_\gamma = \{\sigma_1, \sigma_2, \dots, \sigma_j\}, \quad (45)$$

as equações para mutação dos parâmetros estratégicos são equivalentes às Equações 28 e 29 apresentadas na Seção 4.2.1, tendo a seguinte forma:

$$\sigma'_\gamma = \sigma_\gamma \exp[\tau' N(0, 1) + \tau N(0, 1)], \quad (46)$$

$$\gamma'_i = \gamma_i + \sigma_\gamma N(0, 0.1), \quad (47)$$

onde $N(0, 1)$ é um número escalar gerado a partir de uma distribuição Normal de média $\mu = 0$ e variância $\sigma^2 = 1$, e $N(0, 0.1)$ é um outro escalar gerado a partir de uma distribuição Normal de média $\mu = 0$ e variância $\sigma^2 = 0.1$. Os parâmetros estratégicos τ e τ' são definidos conforme a Equação 18.

Baseado no algoritmo PM, que utiliza o desvio-padrão $\sigma_{i,j}$ do pai para mutação do parâmetro objetivo $x_{i,j}$, e que também é multidimensional para o número de componentes dos indivíduos, a nova equação de mutação do parâmetro objetivo assume a seguinte forma

$$x'_{ij} = x_{ij} + \sigma_{ij} S_j(\alpha_i, \beta, \gamma, \delta). \quad (48)$$

Devido ao fato de representar uma extensão do algoritmo PM, agregando a este a variabilidade do parâmetro γ , seu referenciamento nesse trabalho será feita pela sigla PM/ γ .

Capítulo 5: Resultados

Os experimentos realizados consistiram na aplicação dos diferentes algoritmos propostos, que foram implementados e executados nas funções de teste relacionadas na Tabela 4 (abaixo). Neste capítulo, são apresentados os resultados das análises gráficas e numéricas para comparar o desempenho desses algoritmos, apresentando-se ainda o teste estatístico T-Student para comprovação das hipóteses assumidas em cada um dos comparativos.

5.1 Funções de Teste e Parametrizações dos Algoritmos

Para a realização de uma tábua de testes dos algoritmos implementados, foram selecionadas 12 funções objetivo utilizadas na literatura [Digalakis e Margaritis, 2000; GEATbx, 2007; Hedar, 2007; Lee e Yao, 2004; Yao e Liu, 1996].

As três primeiras funções são unimodais, e, portanto, com apenas um ponto ótimo, enquanto que as funções de f_4 a f_{12} são não-lineares multimodais, contendo vários pontos de mínimos locais e apenas um mínimo global, exceto a f_{11} que contém dois mínimos globais.

A Tabela 4 a seguir apresenta as funções utilizadas no estudo, o número n de dimensões do espaço de busca, seus intervalos de domínio $x \in \mathfrak{R}^n$ válidos, conjunto solução \vec{x}_{\min} e valores dos mínimos globais $f(\vec{x}_{\min})$. As funções f_8 e f_9 utilizam uma função degrau u definida da seguinte forma

$$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a, \\ 0 & -a \leq x_i \leq a, \\ k(-x_i - a)^m & x_i < -a. \end{cases} \quad (49)$$

Nome	Função	n	Intervalo	Conjunto Solução	Mínimo Global
<i>De Jong</i>	$f_1 = \sum_{i=1}^n x_i^2$	30	$-100 \leq x_i \leq 100$	0	0
<i>Rotated hyper-ellipsoid</i>	$f_2 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	30	$-100 \leq x_i \leq 100$	0	0
<i>Rosenbrock</i>	$f_3 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$-30 \leq x_i \leq 30$	1	0
<i>Schwefel</i>	$f_4 = -\sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	30	$-500 \leq x_i \leq 500$	420.9687	-12569.5
<i>Rastrigin</i>	$f_5 = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$-5.12 \leq x_i \leq 5.12$	0	0
<i>Ackley</i>	$f_6 = -20 \exp[-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}] - \exp[\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)] + 20 + e$	30	$-32 \leq x_i \leq 32$	0	0
<i>Griewangk</i>	$f_7 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$-600 \leq x_i \leq 600$	0	0
<i>Penalized (P8)</i>	$f_8 = \frac{\pi}{n} \{10 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2\}$ $+ \sum_{i=1}^n u(x_i, 10, 100, 4), y_i = 1 + \frac{x_i + 1}{4}$	30	$-50 \leq x_i \leq 50$	-1	0
<i>Penalized (P16)</i>	$f_9 = 0.1 \{ \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2 \times$ $[1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	30	$-50 \leq x_i \leq 50$	1	0
<i>Levy</i>	$f_{10} = \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_i + 1)] + (y_n - 1)^2 \times$ $[1 + 10 \sin^2(2\pi y_n)], y_i = 1 + \frac{x_i - 1}{4}$	30	$-10 \leq x_i \leq 10$	1	0
<i>Six-Hump Camel Back</i>	$f_{11} = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$-5 \leq x_i \leq 5$	(-0.0898, 0.7126) (0.0898, -0.7126)	-1.0316
<i>Goldstein-Price</i>	$f_{12} = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times$ $[30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$-2 \leq x_i \leq 2$	(0, -1)	3.0000

Tabela 4. Funções de teste.

Tendo em vista a intenção de se estabelecer um comparativo de resultados, deve-se salientar que as funções de f_1 a f_9 acima são exatamente as utilizadas em [Lee e Yao, 2004], de onde foram feitas duas correções:

- Na função f_4 , a variável x_i que aparece dentro da função raiz quadrada deve estar antecedida pela função de valor absoluto;
- Na função f_8 , a primeira ocorrência da variável y deve receber o índice 1.

Quanto às funções f_{11} e f_{12} presentes aqui, elas também foram utilizadas em [Lee e Yao, 2004] porém estão numeradas de maneira diferente (f_{10} e f_{11}). Já a função f_{10} aqui utilizada encontra-se exclusivamente em [Hedar, 2007].

Quanto à parametrização dos algoritmos, foram fixados os valores a seguir, para permitir uma comparação dos resultados com os apresentados em [Lee e Yao, 2004]:

- Tamanho populacional: $\mu = 100$ indivíduos
- Quantidade de torneios: $q = 10$
- Desvio-padrão inicial: $\sigma = 3$
- Número de gerações para as funções de f_1 a f_{10} : 1500
- Número de gerações para as funções de f_{11} a f_{12} : 30
- Quantidade de execuções: 50

No que se refere aos parâmetros, valem algumas considerações:

- A quantidade q de torneios para cada um dos indivíduos é 10, porém, o tamanho do torneio que define quantos serão os participantes de cada um dos q torneios está determinado em 2;
- Todos os parâmetros estratégicos σ utilizados no algoritmo compartilham do mesmo desvio-padrão inicial;
- O número de gerações para as funções f_{11} e f_{12} é menor devido ao fato de apresentarem uma convergência rápida para o ótimo global, ocasionada pelo menor número de dimensões ($n = 2$) que as caracterizam;
- O número de gerações máximo de 1500 para as funções de f_1 a f_{10} , e máximo de 30 para as funções de f_{11} a f_{12} , correspondem ao critério de parada descrito na Etapa 5 da Seção 4.2.

Quanto aos algoritmos que utilizam o nível de adaptação por gerações de populações, quando não apresentam o índice k , qualificador desse tipo de abordagem, apresentam um número fixo de 10 gerações que repetem o parâmetro estratégico α .

Conforme já foi descrito anteriormente (Seção 4.2.1), todos os algoritmos adaptativos que utilizam parâmetros estratégicos fixos na distribuição estável foram definidos da seguinte forma:

- Parâmetro de simetria: $\beta = 0$;
- Parâmetro de escala: $\gamma = 1$;
- Parâmetro de locação: $\delta = 0$.

A quantidade de 50 execuções refere-se ao número de populações iniciais independentes geradas a partir de um processo aleatório, que simula uma distribuição de probabilidade uniforme no intervalo de domínio para cada uma das funções presentes na Tabela 4. Todos os parâmetros objetivos e estratégicos não mencionados anteriormente, foram inicializados por um processo aleatório uniforme, respeitando seus respectivos intervalos de domínio. A simulação de números aleatórios de uma distribuição de probabilidade uniforme é um procedimento padrão do software *Mathematica*.

5.2 Análise Gráfica

Uma forma de comparar o desempenho dos algoritmos ao longo das gerações é através de uma análise gráfica dos seus resultados observados. Portanto, cada uma das curvas representa o algoritmo que está sendo analisado, e cada ponto dessa curva representa a média aritmética do valor da avaliação (no caso, o próprio valor da função objetivo) dos melhores indivíduos em 50 execuções.

Em todas as funções a seguir, o eixo das abcissas representa o número de gerações, porém, exclusivamente para as funções f_1 - f_3 e f_8 - f_{10} , o eixo das ordenadas apresenta o logaritmo da média aritmética do valor da função objetivo dos melhores indivíduos em 50 execuções. Essa alteração serve para facilitar a visualização da diferença dos resultados obtidos entre os algoritmos para essas funções.

5.2.1 Resultados da Auto-Adaptação no Parâmetro α de uma Distribuição Estável

Os gráficos a seguir representam o resultado dos algoritmos citados na Seção 4.2.1, para cada uma das funções de teste apresentadas na Tabela 4.

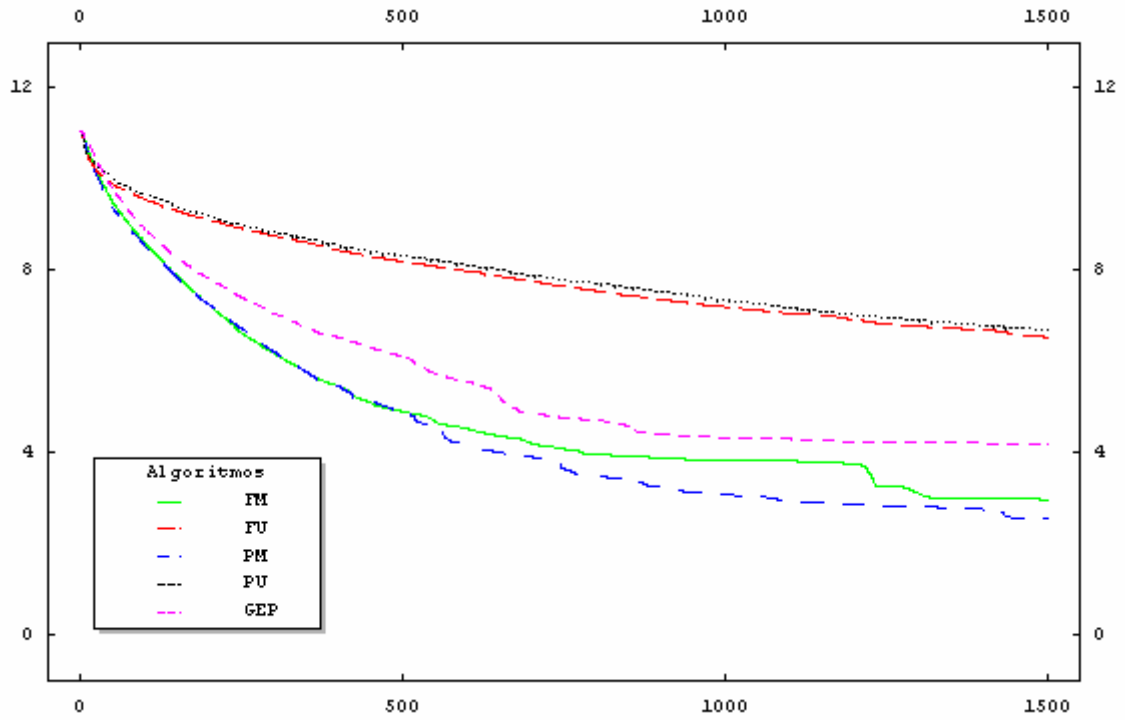


Figura 8. Comparativo de desempenho dos algoritmos para a função f_1 (eixo das ordenadas em logaritmo natural).

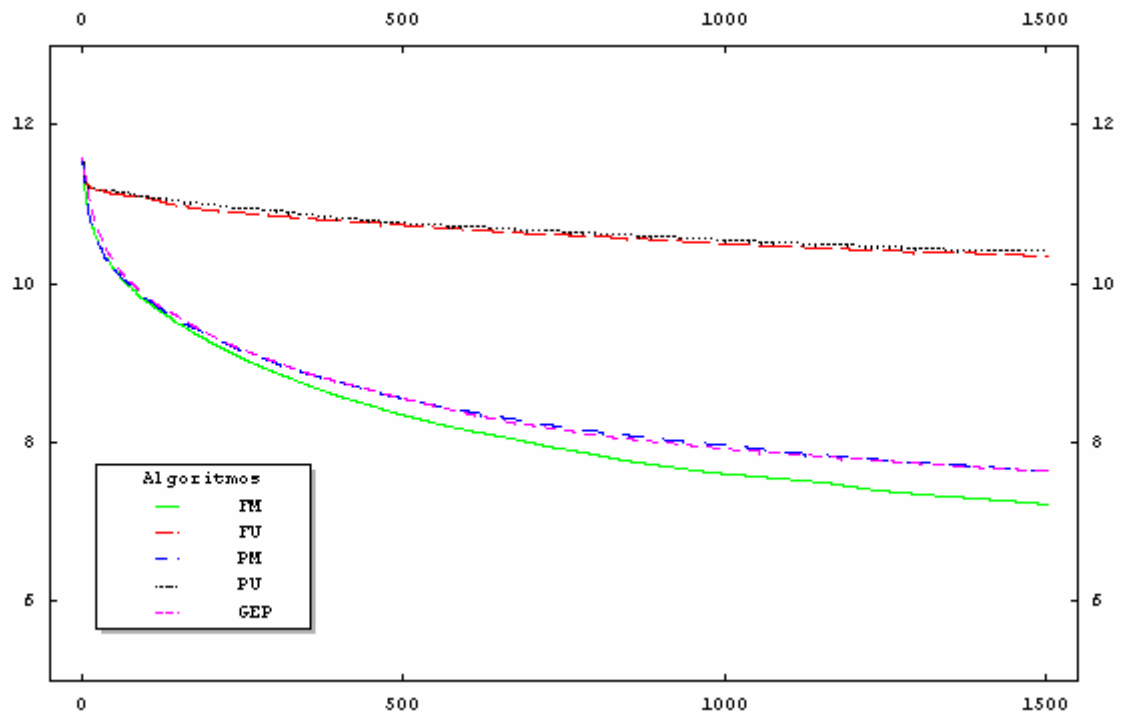


Figura 9. Comparativo de desempenho dos algoritmos para a função f_2 (eixo das ordenadas em logaritmo natural).

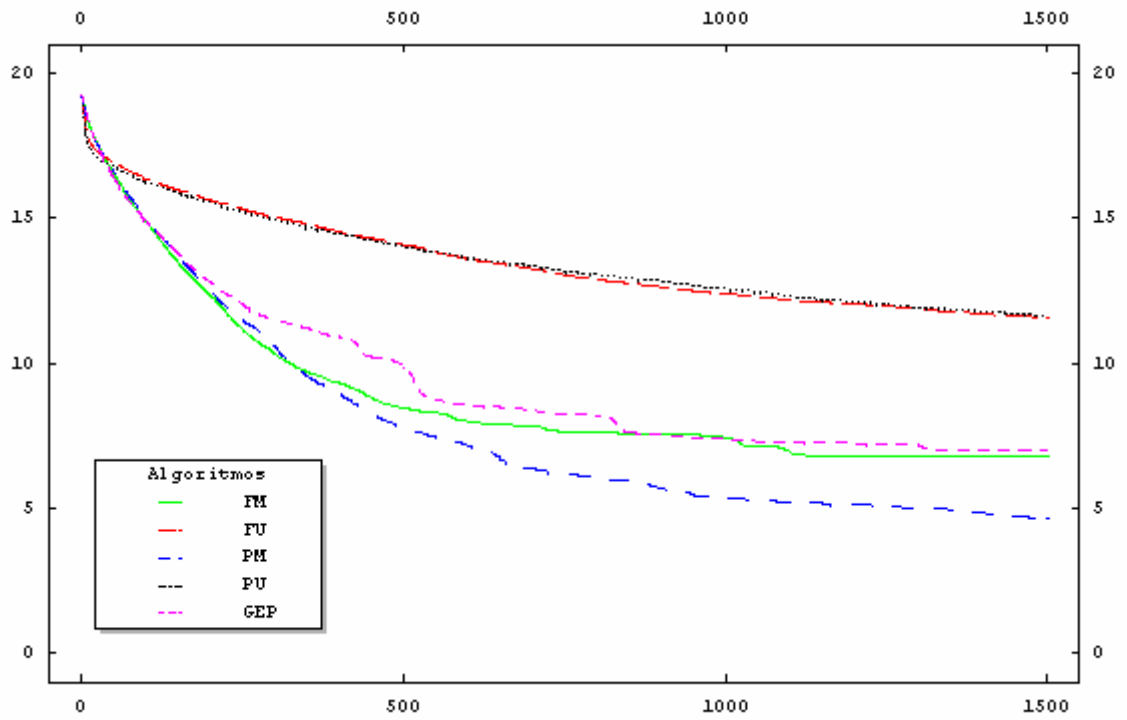


Figura 10. Comparativo de desempenho dos algoritmos para a função f_3 (eixo das ordenadas em logaritmo natural).

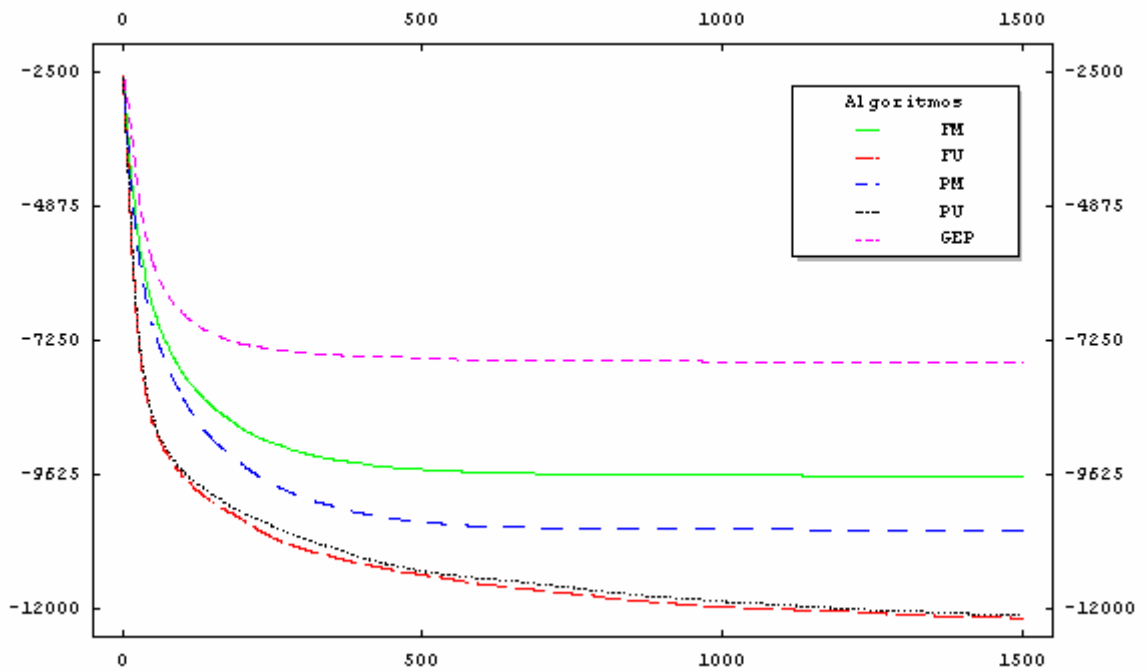


Figura 11. Comparativo de desempenho dos algoritmos para a função f_4 .

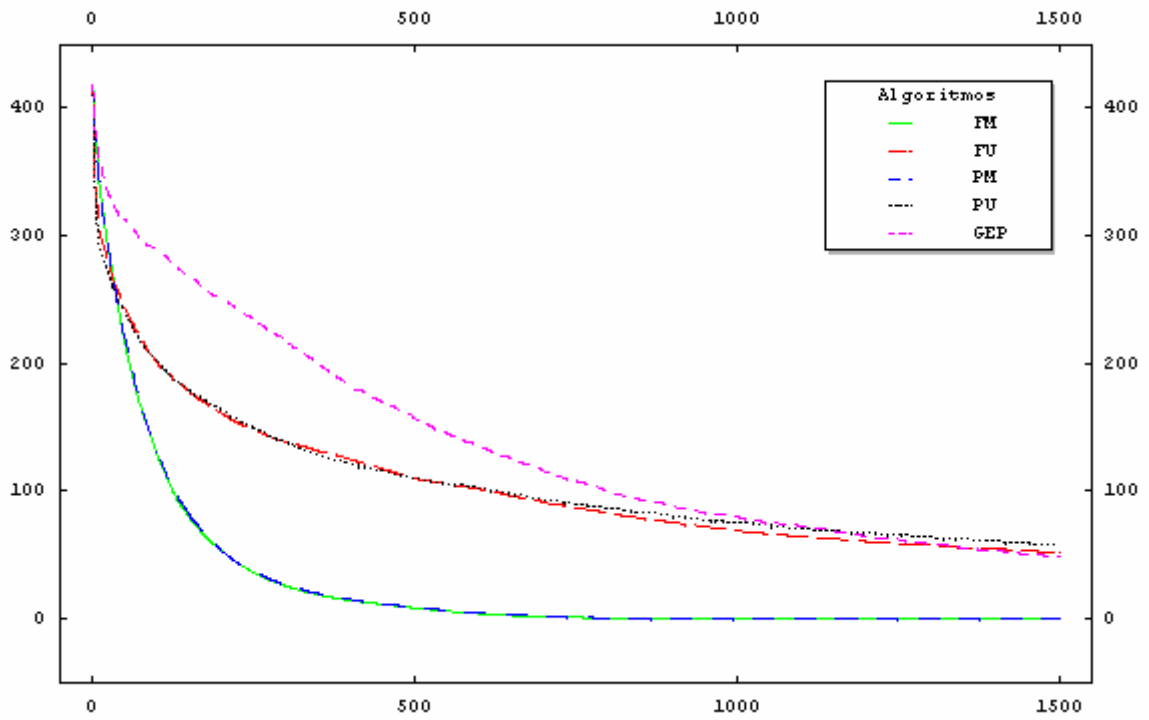


Figura 12. Comparativo de desempenho dos algoritmos para a função f_5 .

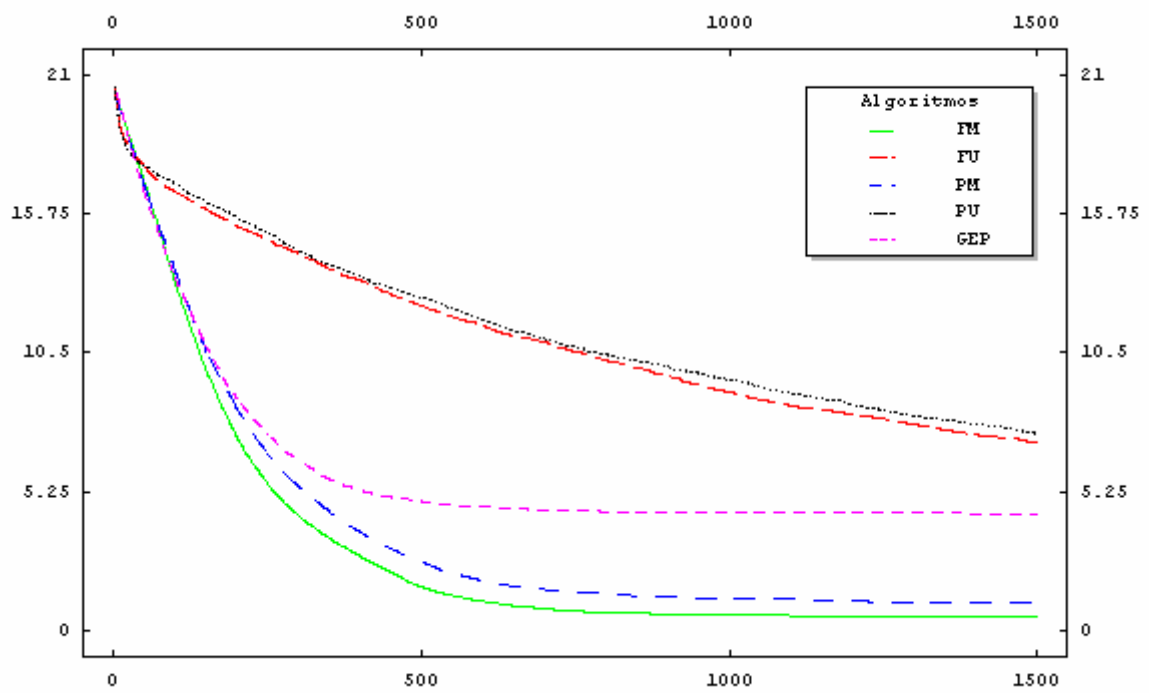


Figura 13. Comparativo de desempenho dos algoritmos para a função f_6 .

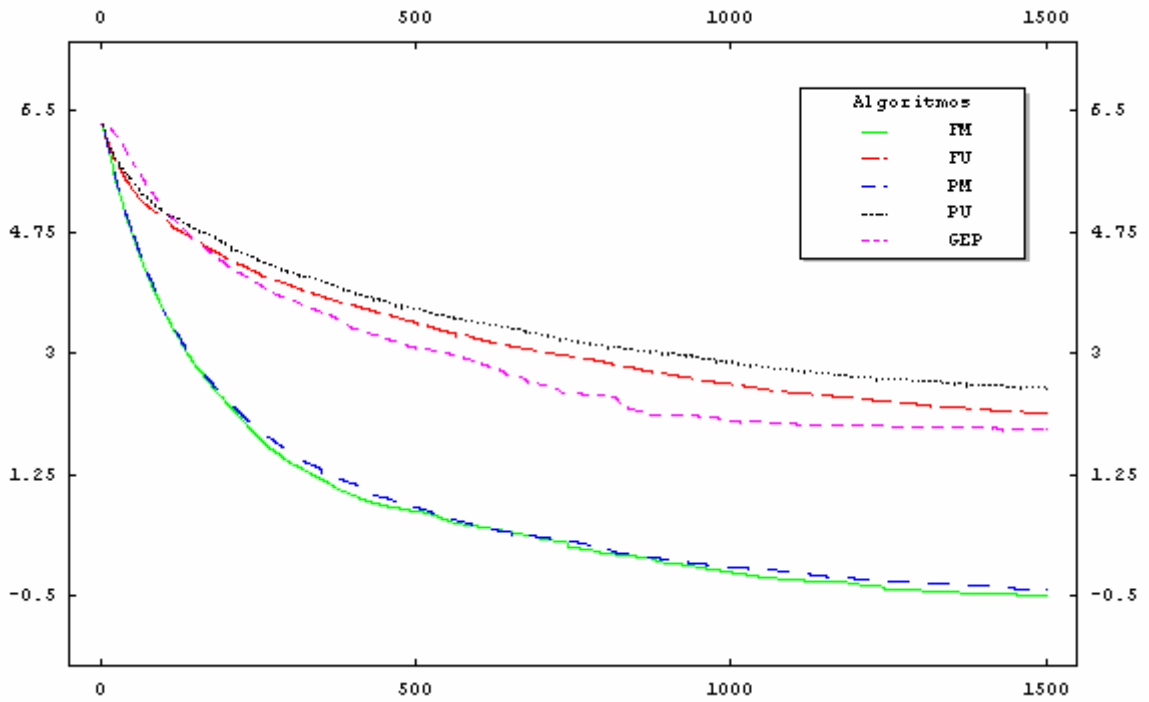


Figura 14. Comparativo de desempenho dos algoritmos para a função f_7 .

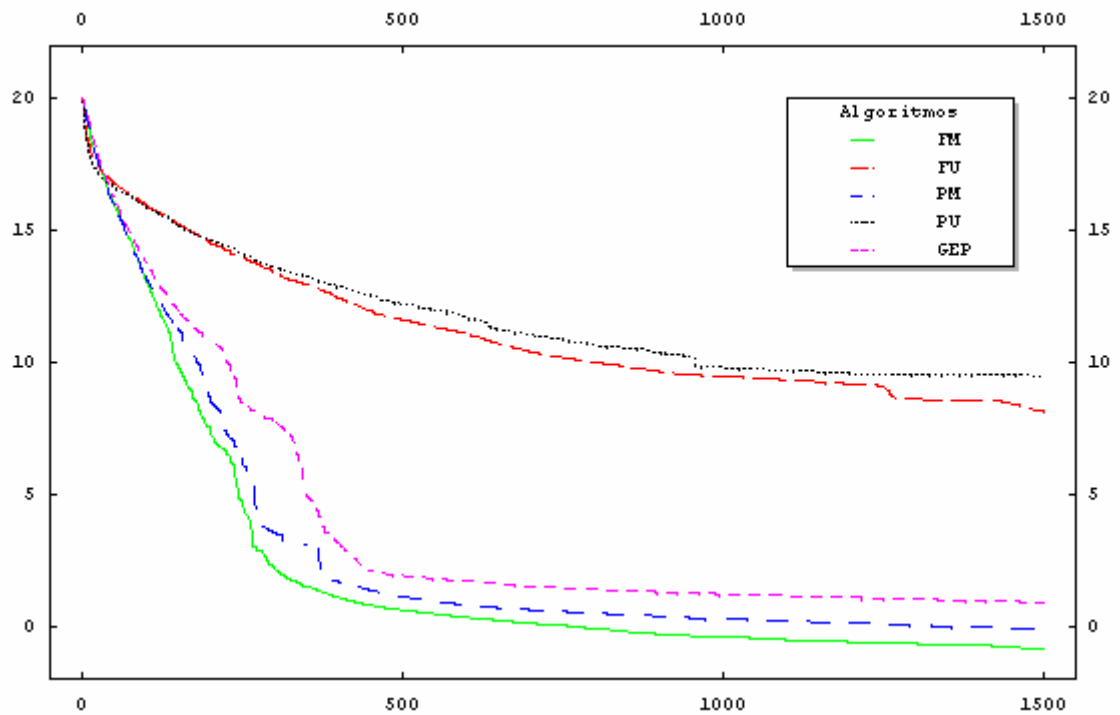


Figura 15. Comparativo de desempenho dos algoritmos para a função f_8 (eixo das ordenadas em logaritmo natural).

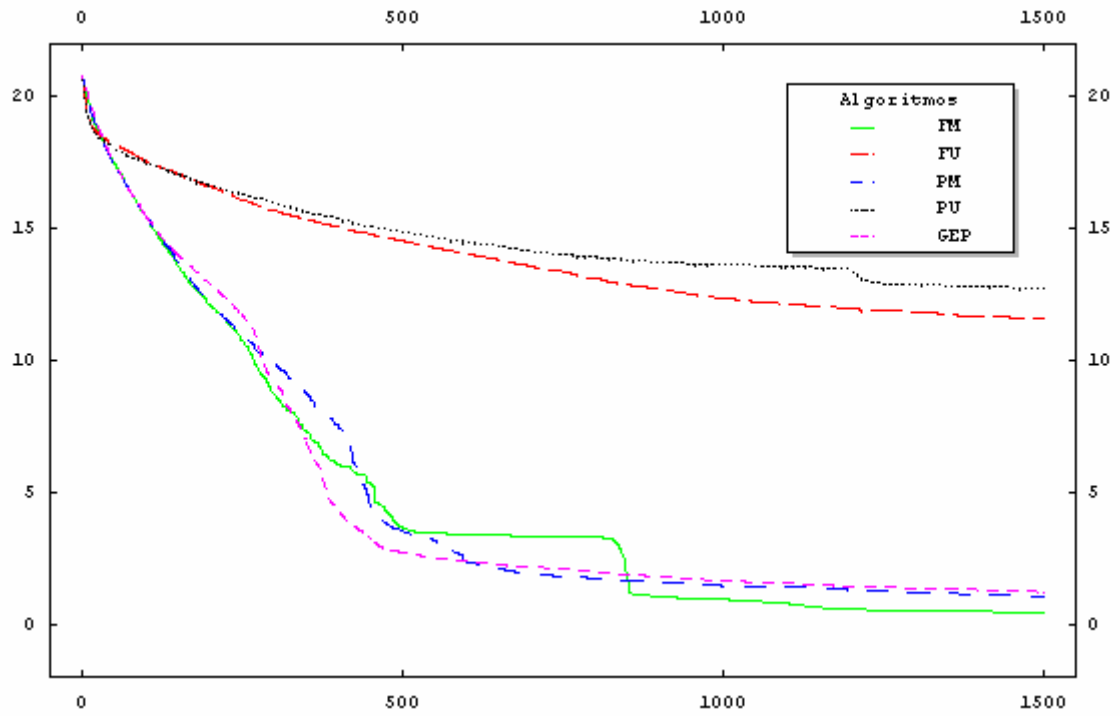


Figura 16. Comparativo de desempenho dos algoritmos para a função f_9 (eixo das ordenadas em logaritmo natural).

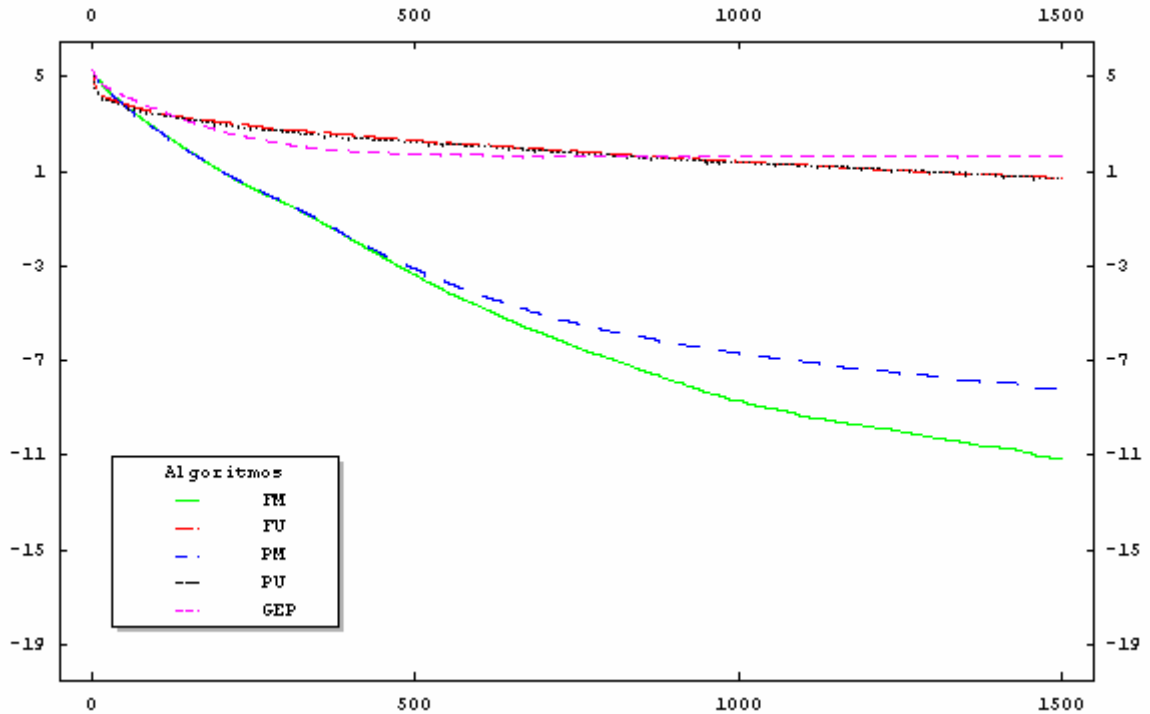


Figura 17. Comparativo de desempenho dos algoritmos para a função f_{10} (eixo das ordenadas em logaritmo natural).

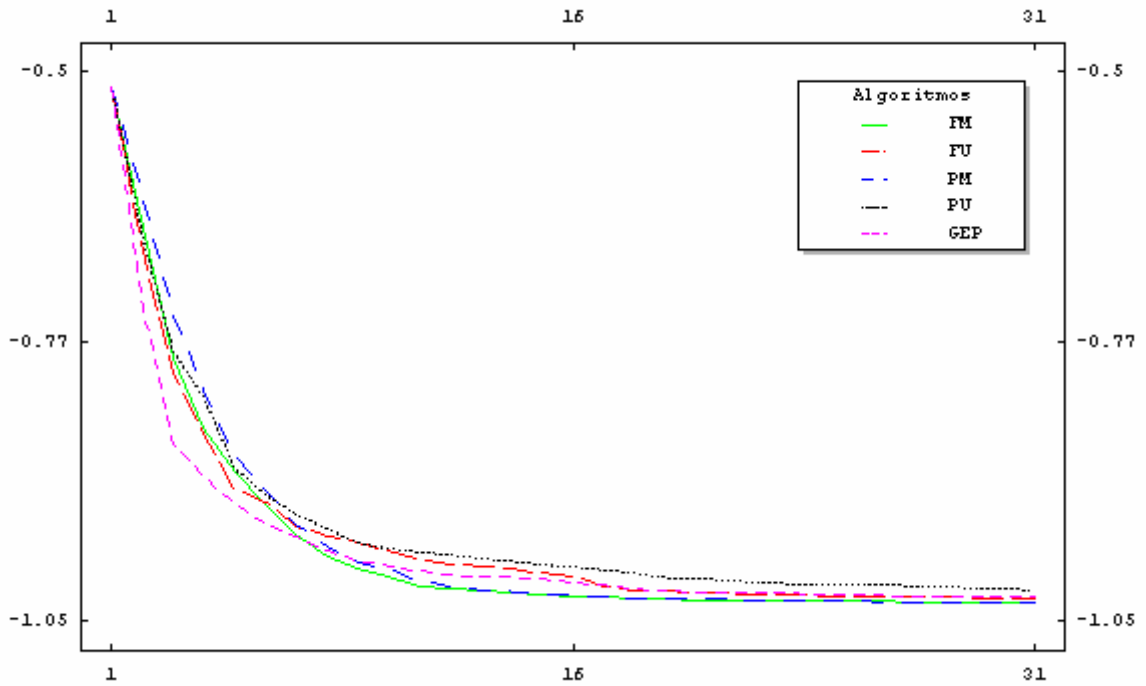


Figura 18. Comparativo de desempenho dos algoritmos para a função f_{11} .

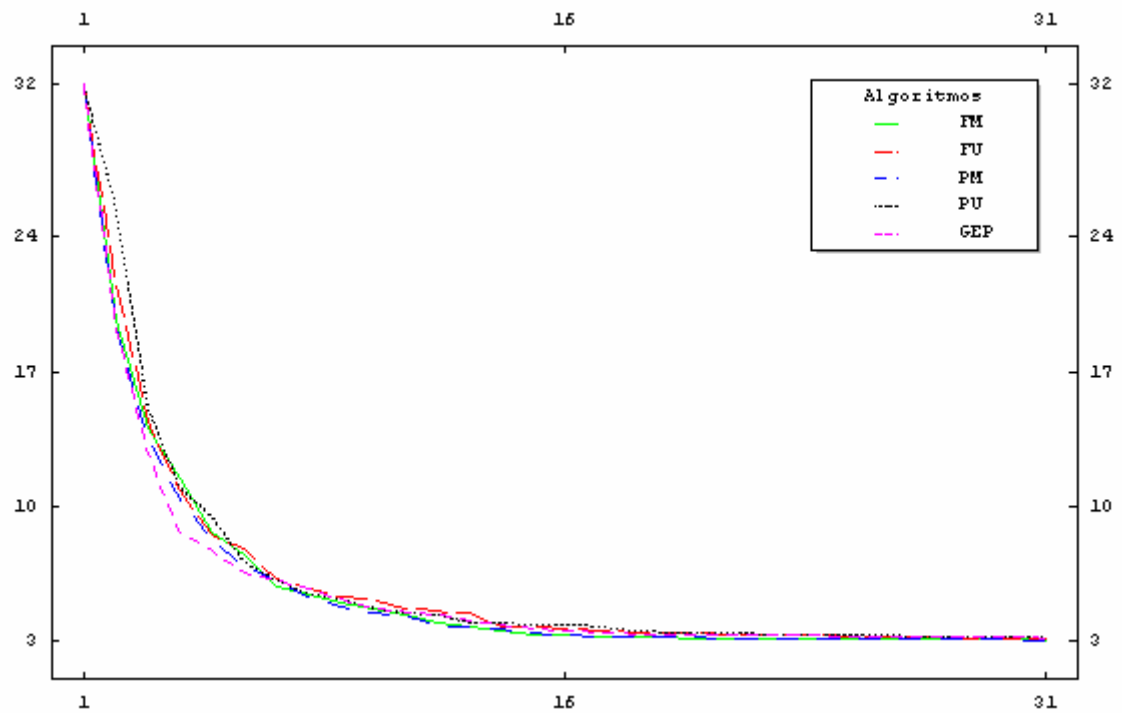


Figura 19. Comparativo de desempenho dos algoritmos para a função f_{12} .

A evolução do resultado ao longo das gerações para as funções f_1, f_2, f_3, f_8 e f_9 discrimina significativamente os algoritmos U em comparação aos algoritmos M e GEP, sempre em favor

destes últimos. Observe-se, entretanto, que não é possível fazer uma diferenciação visual significativa nos algoritmos M e GEP entre si.

Semelhantemente aos resultados apresentados nas funções anteriores, na função f_6 os algoritmos M destacam-se dos demais, porém, o GEP apresenta um resultado estabilizado em um nível inferior aos algoritmos M, enquanto que os algoritmos do tipo U apresentam-se em franco decréscimo.

Para a função f_4 , permanece a distinção dos resultados dos algoritmos U em comparação aos M, porém ocorre uma inversão quanto à qualidade do resultado. Enquanto para as funções anteriores o resultado favorável é verificado nos algoritmos M, para a função f_4 o resultado favorável apresenta-se nos algoritmos U. Além disso, existe uma clara distinção (para pior) entre o desempenho da curva do GEP e os algoritmos PM e FM, fato não verificado nas funções anteriores. Os motivos que levaram a inversão dos resultados nesta função não foram estudados no presente trabalho.

Nas funções f_5 , f_7 e f_{10} , permanece a discriminação favorável aos algoritmos M em detrimento ao resultado apresentado pelos algoritmos U, com a diferença que nestes casos, o resultado do GEP encontra-se próximo aos algoritmos U. Na função f_5 , o resultado final da curva do GEP assemelhou-se aos algoritmo do tipo U, porém sua curva inicial está em um patamar acima dos demais, sinalizando convergência lenta do resultado.

A análise gráfica das funções f_{11} e f_{12} não permite uma distinção significativa entre o quaisquer dos algoritmos em 30 execuções. Para as duas funções existe uma leve diferenciação do algoritmo GEP, cujo resultado é favorável já nas primeiras 7 execuções. Uma outra diferenciação acontece no algoritmo PU, que também apresenta-se levemente inferior aos demais nas duas funções.

De maneira geral, o resultado dos algoritmos M, seja ele P ou F, é bastante semelhante. Isso também acontece com o resultado dos algoritmos U, que independente de serem P ou F, apresentam resultados sem uma diferenciação visual significativa.

Vale ressaltar o resultado do algoritmo GEP que ora assemelha-se aos algoritmos U (f_5 , f_7 e f_{10}), ora aos algoritmos M (f_1 - f_3 e f_8 - f_9), sendo que na função f_6 a semelhança aos algoritmos M se faz através da tendência.

5.2.2 Resultados dos Algoritmos com Variação por Nível de Adaptação

Os gráficos a seguir representam o resultado dos algoritmos citados na Seção 4.2.2, para cada uma das funções de teste apresentadas na Tabela 4.

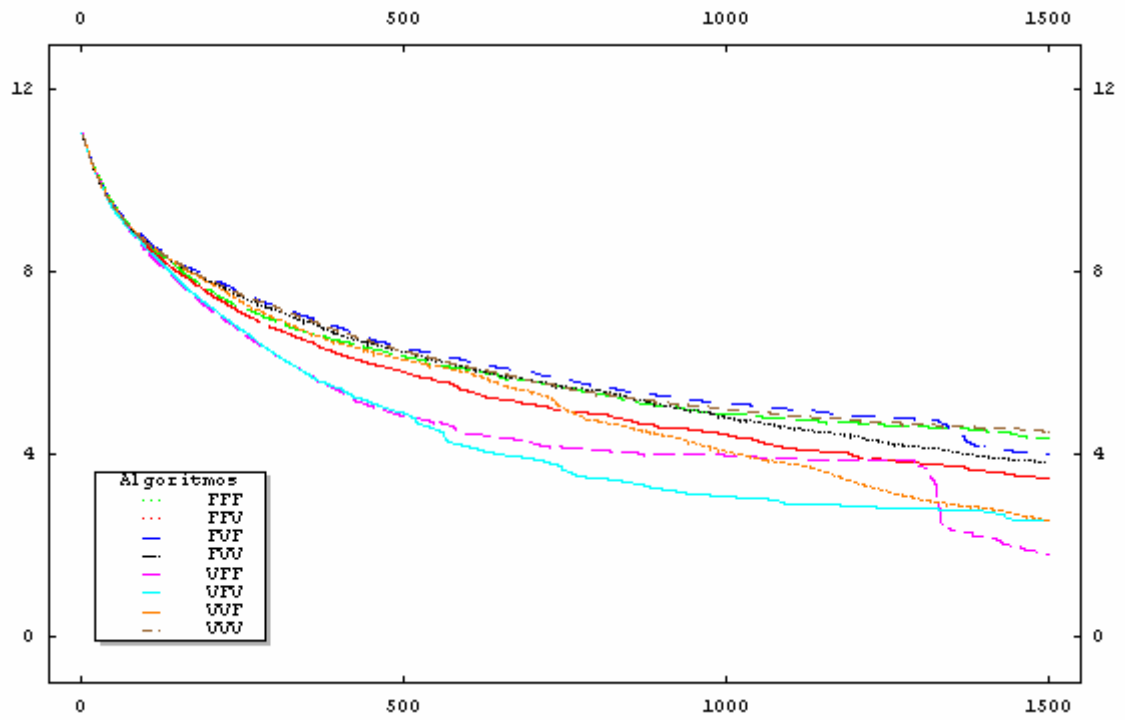


Figura 20. Comparativo de desempenho dos algoritmos para a função f_1 (eixo das ordenadas em logaritmo natural).

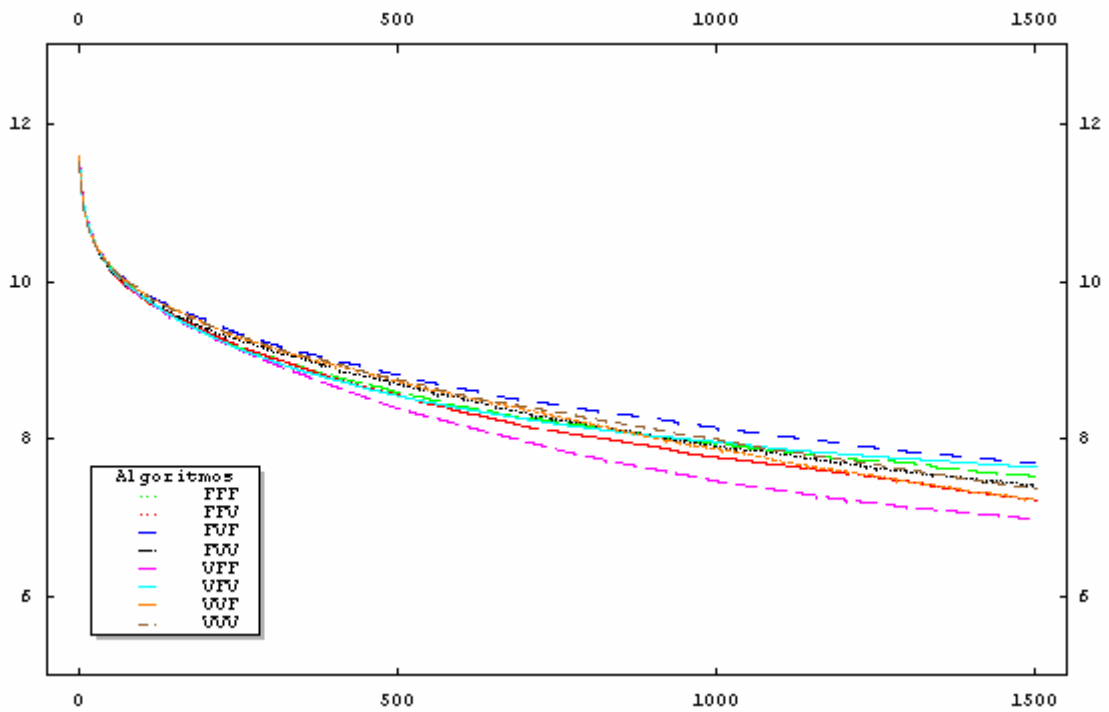


Figura 21. Comparativo de desempenho dos algoritmos para a função f_2 (eixo das ordenadas em logaritmo natural).

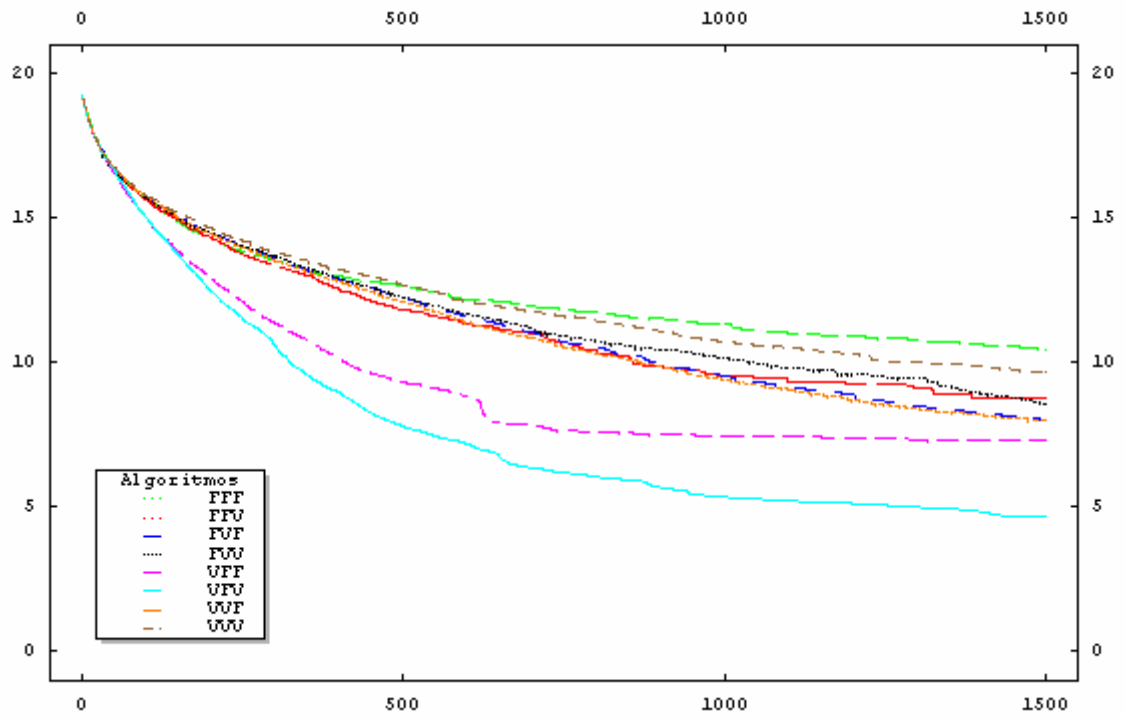


Figura 22. Comparativo de desempenho dos algoritmos para a função f_3 (eixo das ordenadas em logaritmo natural).

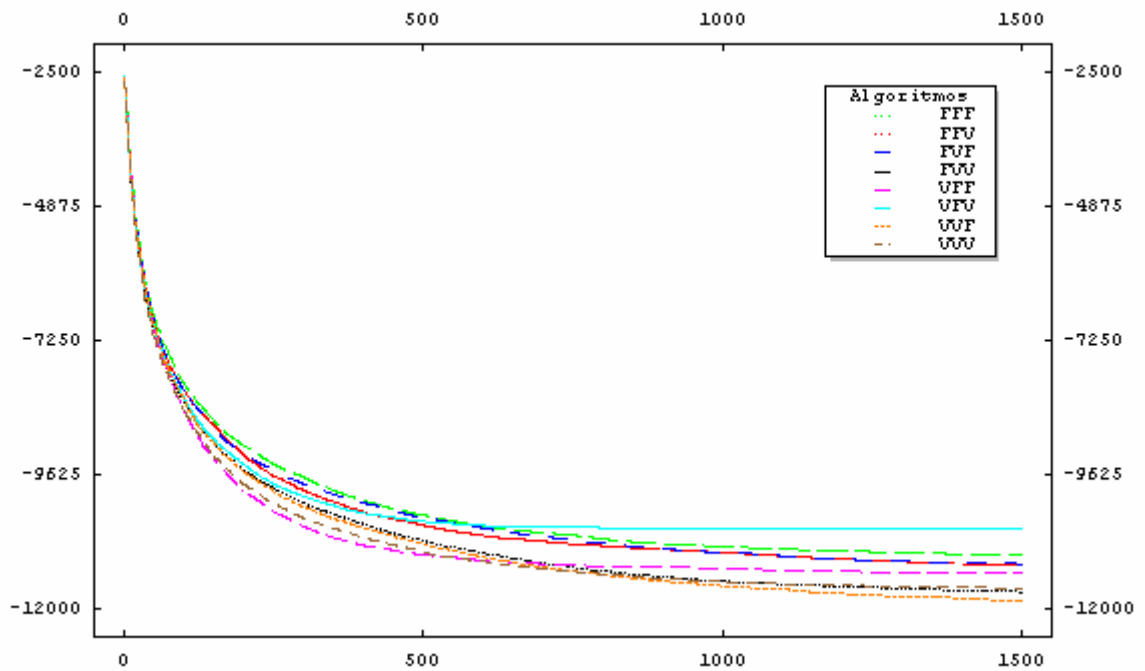


Figura 23. Comparativo de desempenho dos algoritmos para a função f_4 .

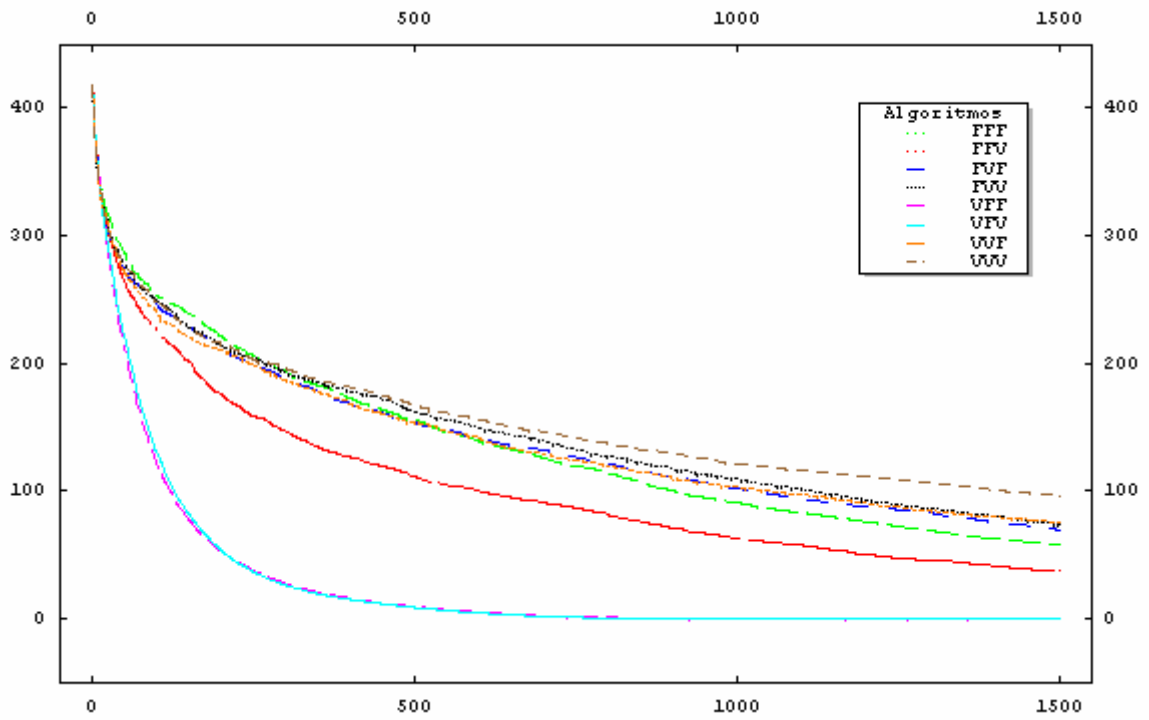


Figura 24. Comparativo de desempenho dos algoritmos para a função f_5 .

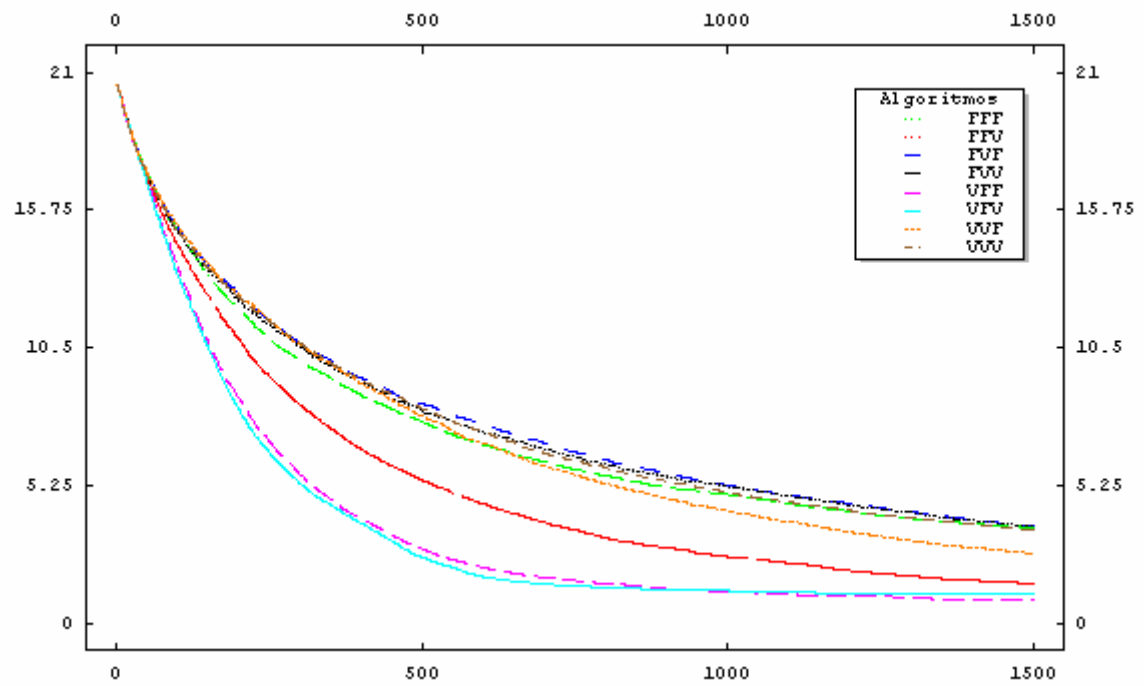


Figura 25. Comparativo de desempenho dos algoritmos para a função f_6 .

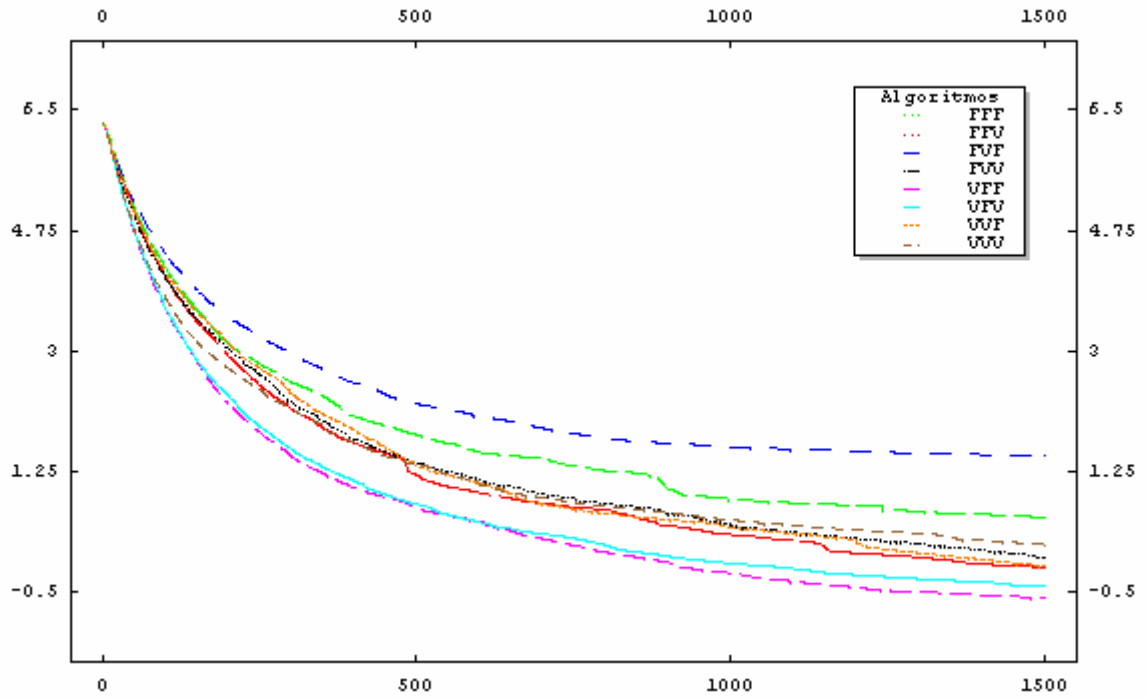


Figura 26. Comparativo de desempenho dos algoritmos para a função f_7 .

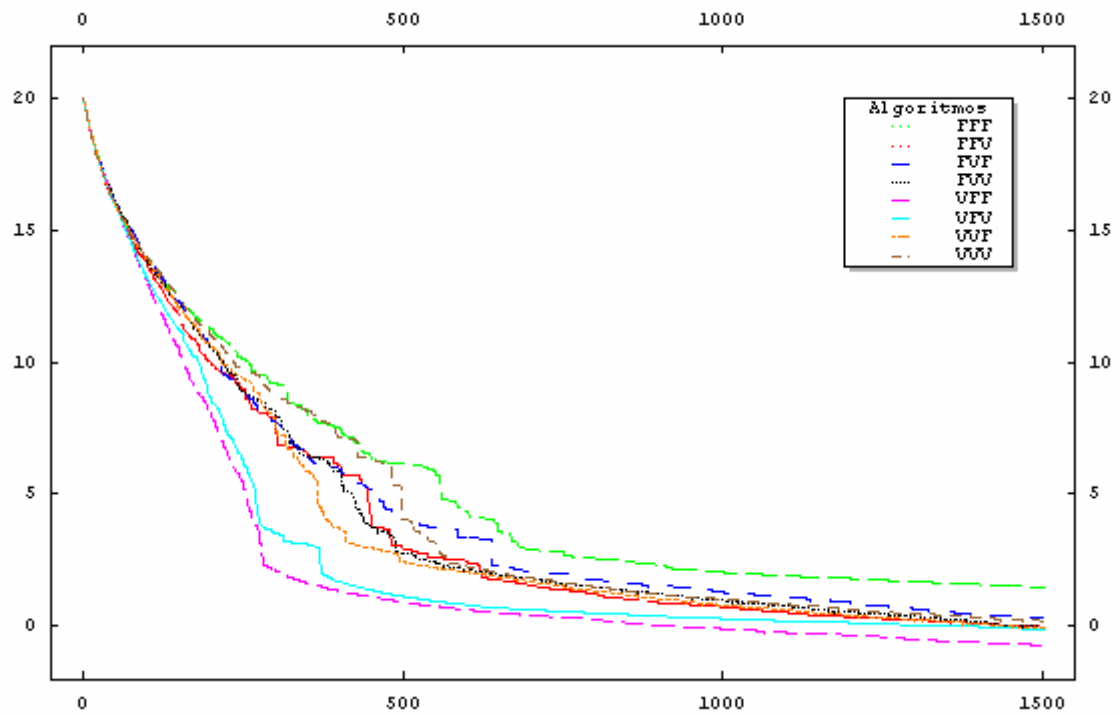


Figura 27. Comparativo de desempenho dos algoritmos para a função f_8
(eixo das ordenadas em logaritmo natural).

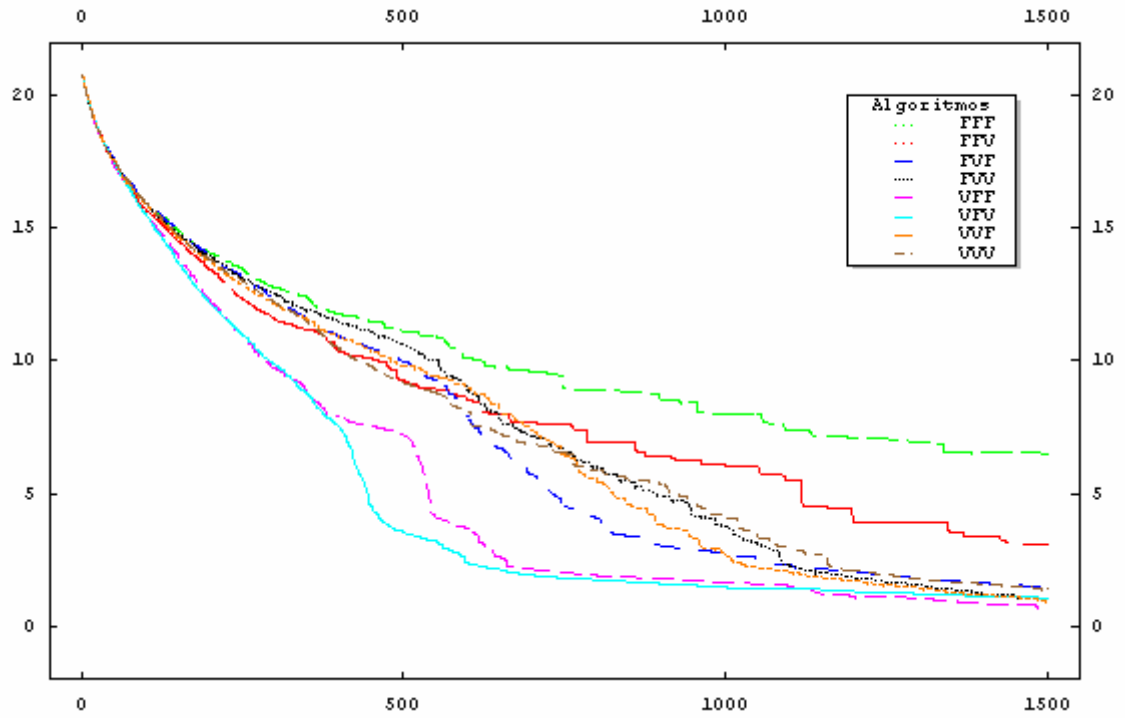


Figura 28. Comparativo de desempenho dos algoritmos para a função f_9 (eixo das ordenadas em logaritmo natural).

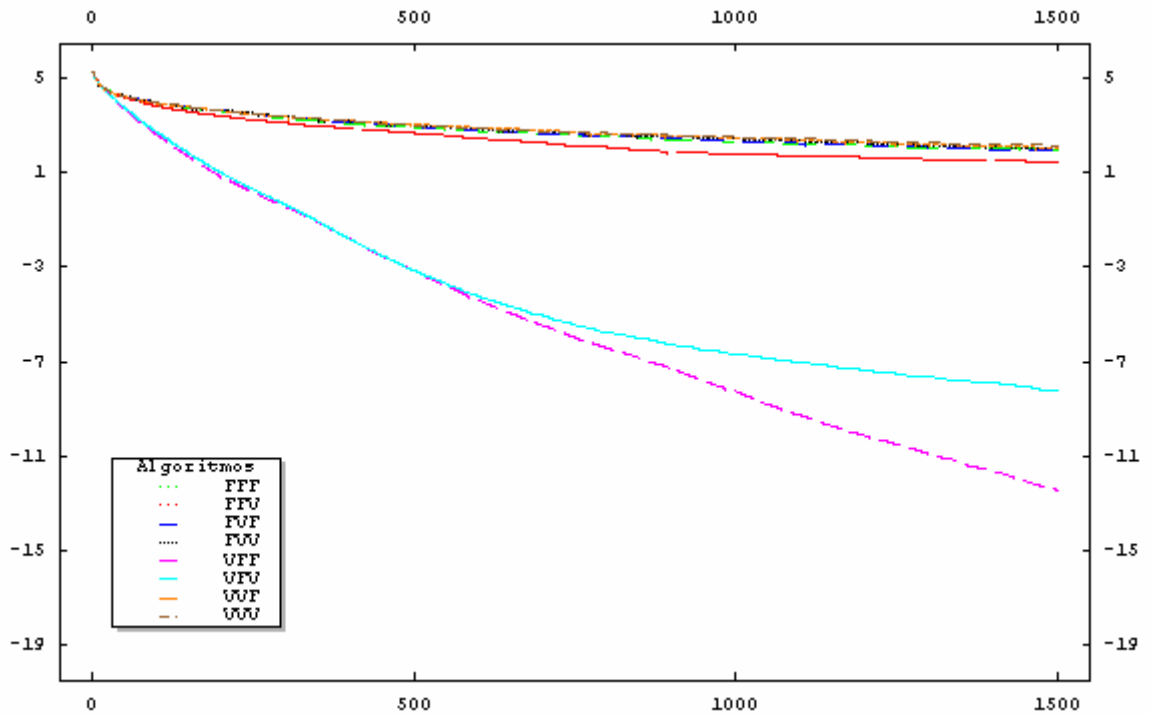


Figura 29. Comparativo de desempenho dos algoritmos para a função f_{10} (eixo das ordenadas em logaritmo natural).

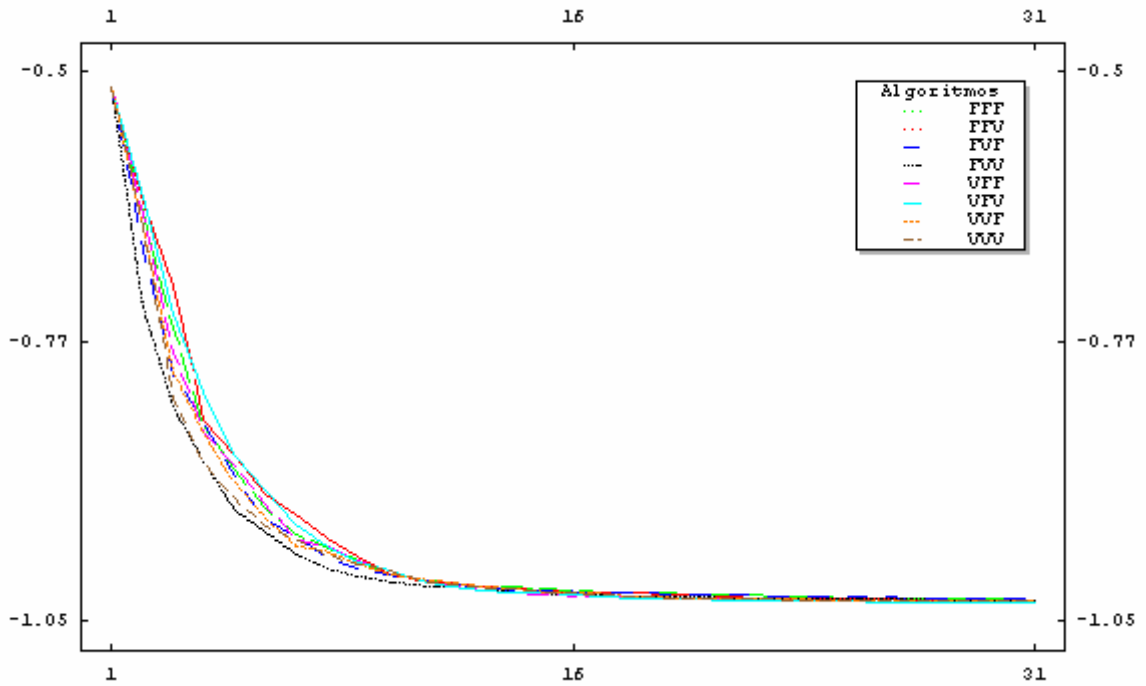


Figura 30. Comparativo de desempenho dos algoritmos para a função f_{11} .

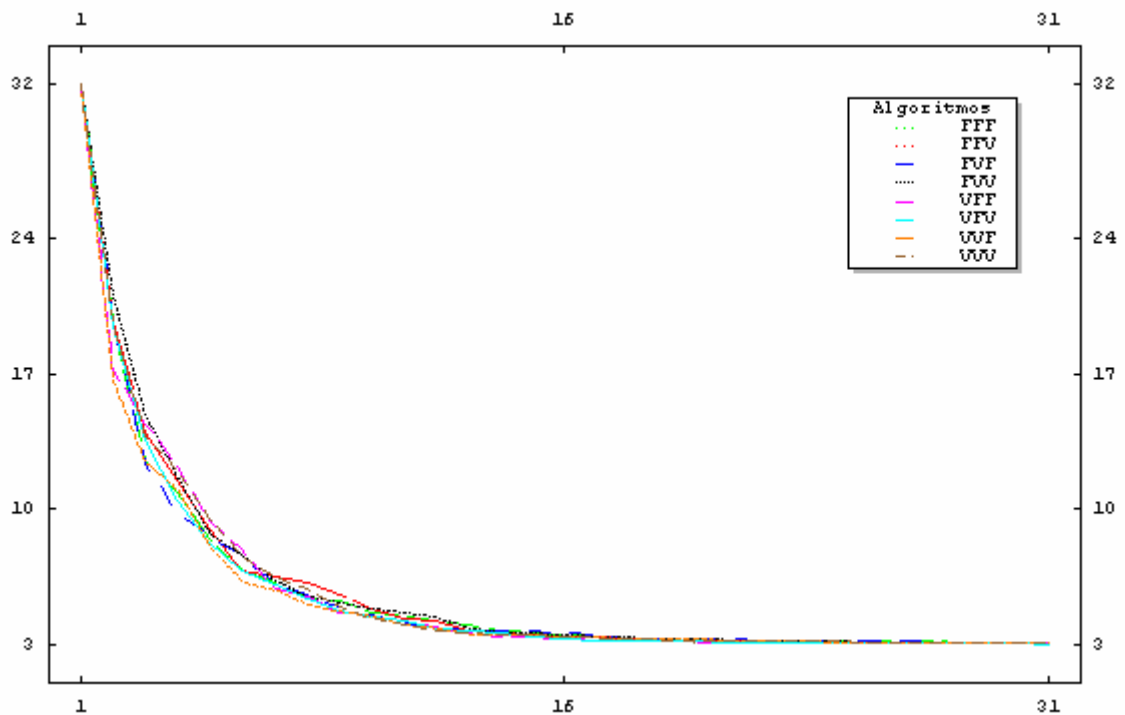


Figura 31. Comparativo de desempenho dos algoritmos para a função f_{12} .

O resultado do algoritmo VFV é bastante similar nas funções f_1 e f_3 , pois, apesar de apresentar rápida convergência inicial, permanece estabilizado ao final das 1500 gerações. Ainda

nestas duas funções, o algoritmo VFF apresenta forte tendência decrescente a partir de 1300 gerações (f_1), enquanto que na função f_3 , estabiliza-se em patamar inferior ao algoritmo VFV a partir de 600 gerações.

Nas funções f_2 , f_4 , f_7 e f_8 não existe clara distinção no resultado dos algoritmos, prevalecendo ligeira vantagem em favor do algoritmo VFF (f_2 , f_8 e f_9), sendo que nas últimas duas funções, ele é acompanhado pelo algoritmo VFV. Os piores resultados foram obtidos pelo algoritmo VFV na função f_4 , pelo algoritmo VFV na função f_7 , e pelo algoritmo FFF na função f_8 .

Na função f_9 , os algoritmos FFF e FFV apresentam os piores resultados, sendo que os demais permanecem empatados ao final de 1500 gerações. Nesta mesma função, a dupla de algoritmos VFV e VFF apresenta rápida convergência inicial, estabilizando-se em 700 gerações, enquanto que os demais algoritmos apresentam tendência decrescente mais acentuada que os anteriores.

Nas funções f_5 , f_6 , f_7 e f_{10} prevalece a vantagem dos algoritmos VFV e VFF sobre os demais. Nas funções f_6 e f_7 essa diferenciação não é tão significativa, porém, nas demais funções, ela aparece com maior evidência. Nas funções f_5 e f_6 , apesar de prevalecer a vantagem dos algoritmos VFV e VFF, nota-se ainda resultado mediano obtido pelo algoritmo FFV que, ao final das 1500 execuções, ainda apresenta tendência decrescente. Na função f_{10} , apesar dos algoritmos VFV e VFF apresentarem resultados destacados dos demais, o resultado do último prevalece sobre o anterior a partir de 600 gerações.

A análise gráfica das funções f_{11} e f_{12} não permite distinção significativa entre quaisquer dos algoritmos em 30 execuções.

5.2.3 Resultados da Auto-Adaptação Elitista e Auto-Adaptação nos Parâmetros α e γ da Distribuição Estável

Os gráficos a seguir representam o resultado dos algoritmos citados na Seções 4.2.3 e 4.2.4, para cada uma das funções de teste apresentadas na Tabela 4.

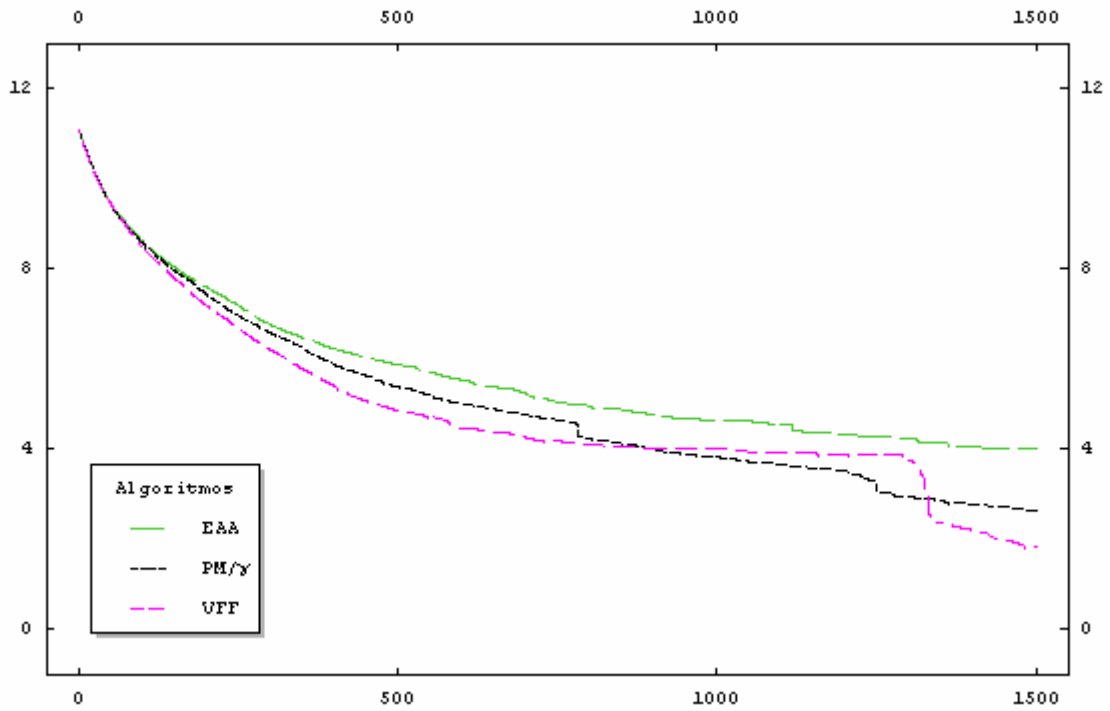


Figura 32. Comparativo de desempenho dos algoritmos para a função f_1 (eixo das ordenadas em logaritmo natural).

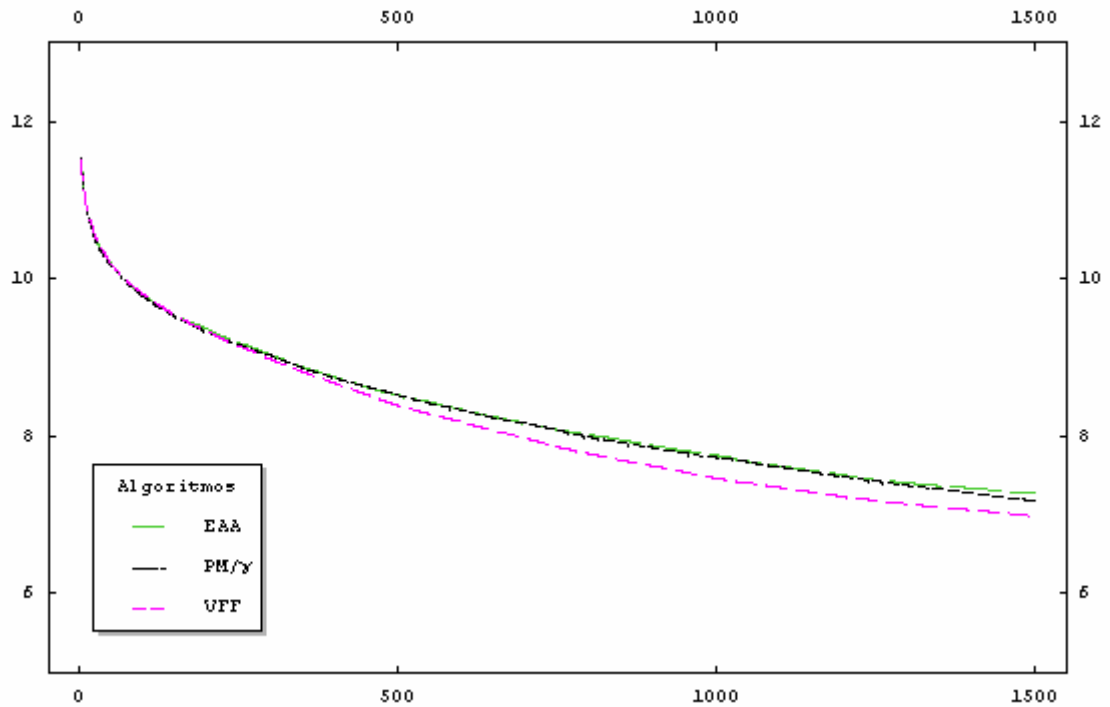


Figura 33. Comparativo de desempenho dos algoritmos para a função f_2 (eixo das ordenadas em logaritmo natural).

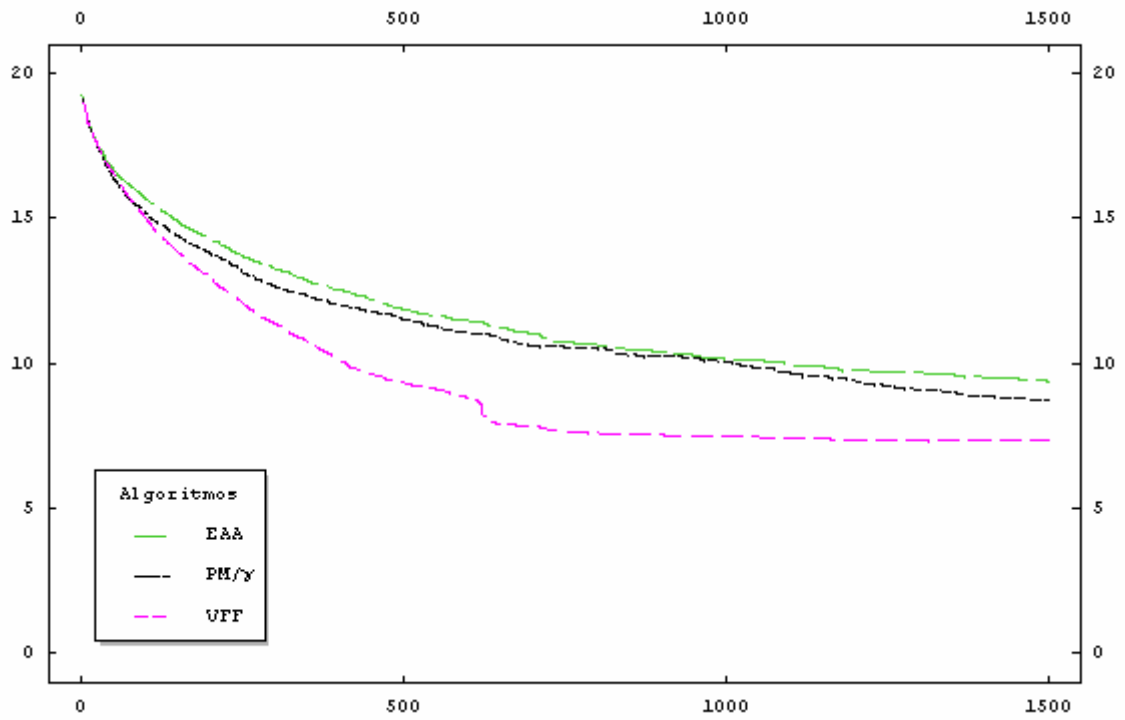


Figura 34. Comparativo de desempenho dos algoritmos para a função f_3 (eixo das ordenadas em logaritmo natural).

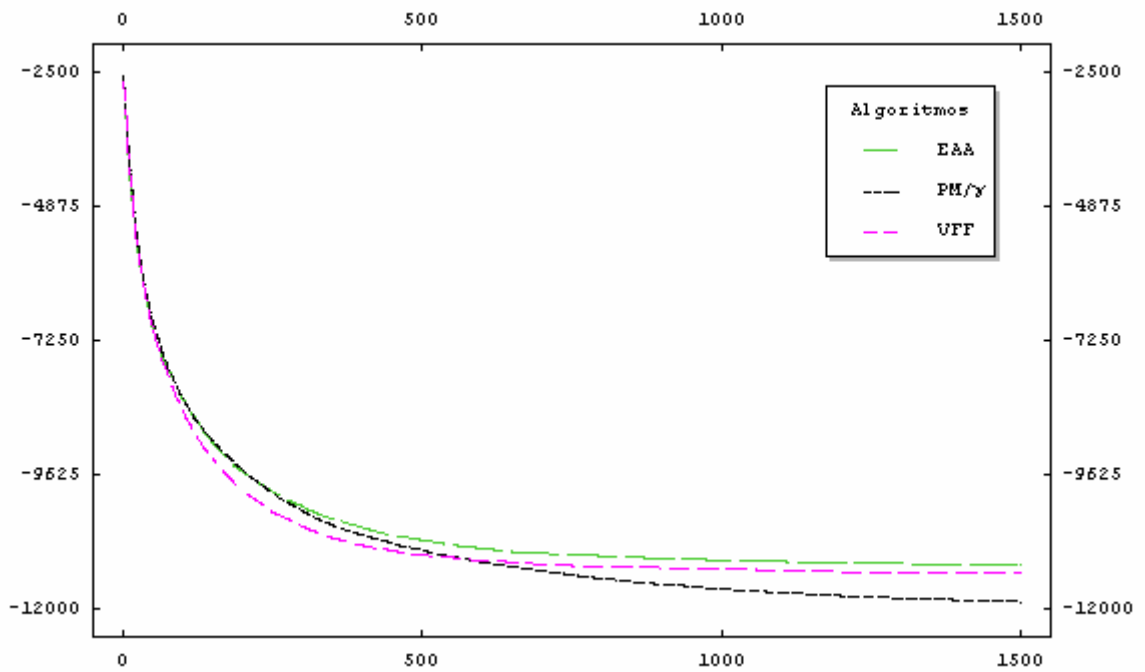


Figura 35. Comparativo de desempenho dos algoritmos para a função f_4 .

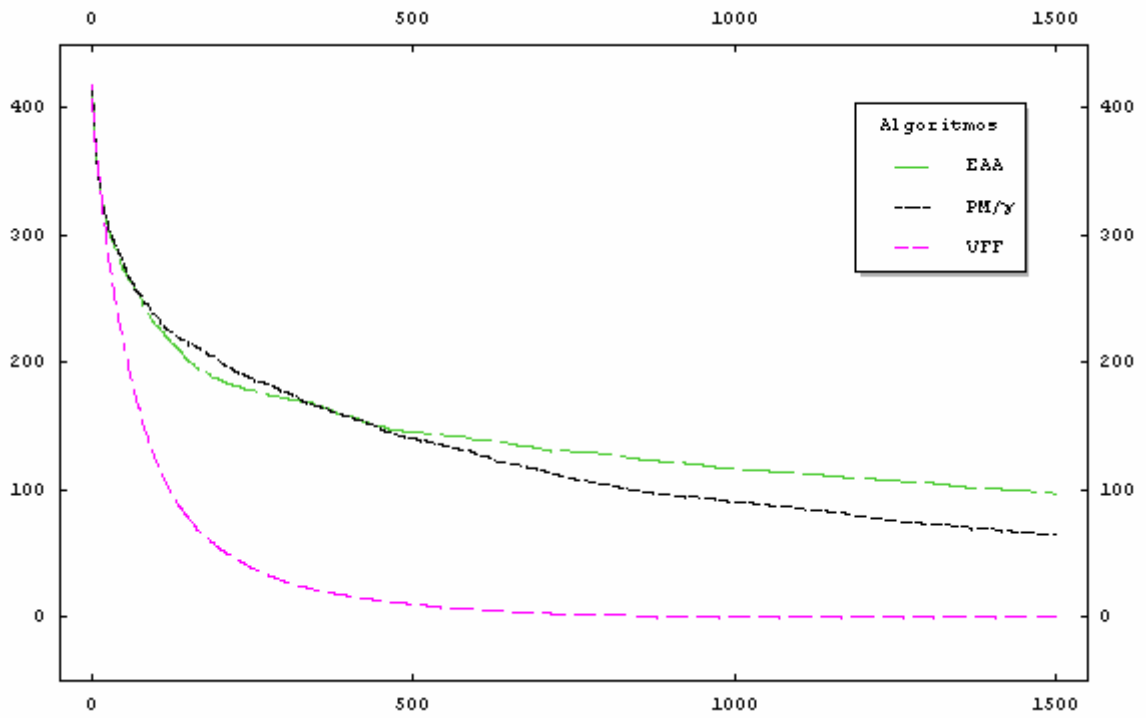


Figura 36. Comparativo de desempenho dos algoritmos para a função f_5 .

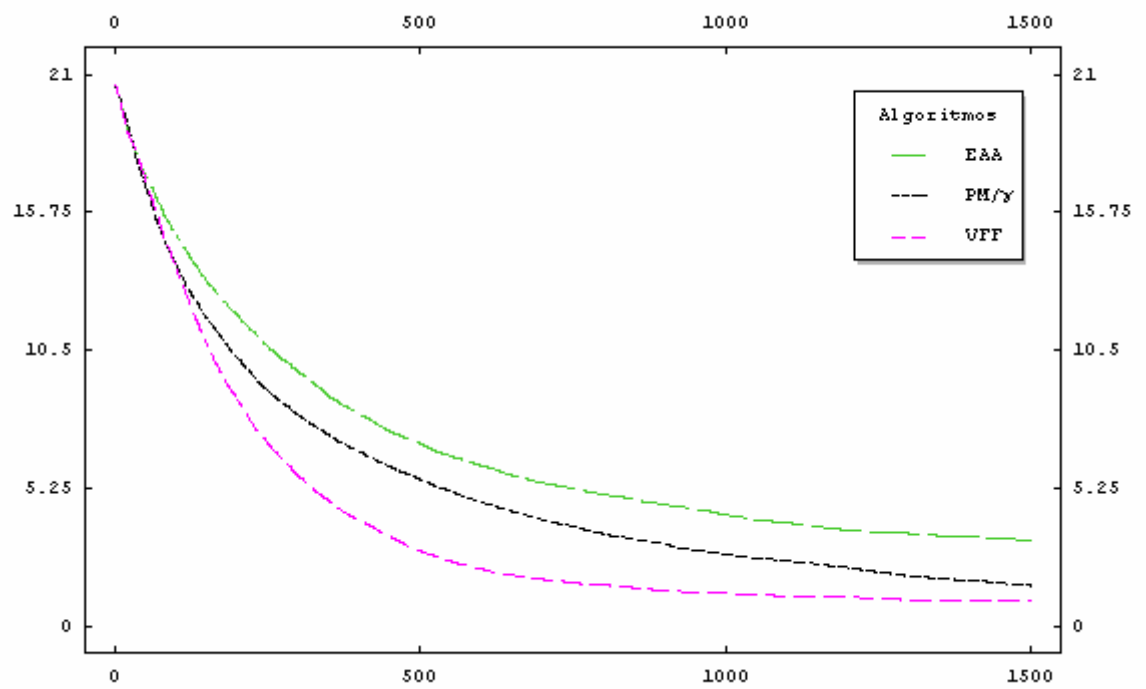


Figura 37. Comparativo de desempenho dos algoritmos para a função f_6 .

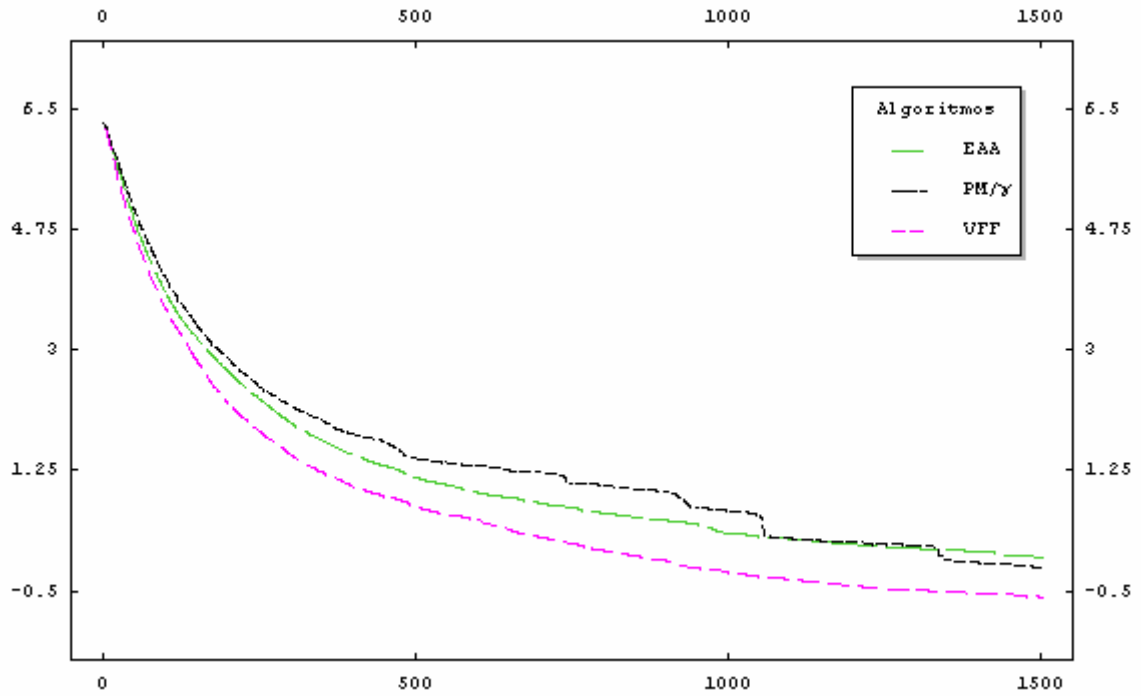


Figura 38. Comparativo de desempenho dos algoritmos para a função f_7 .

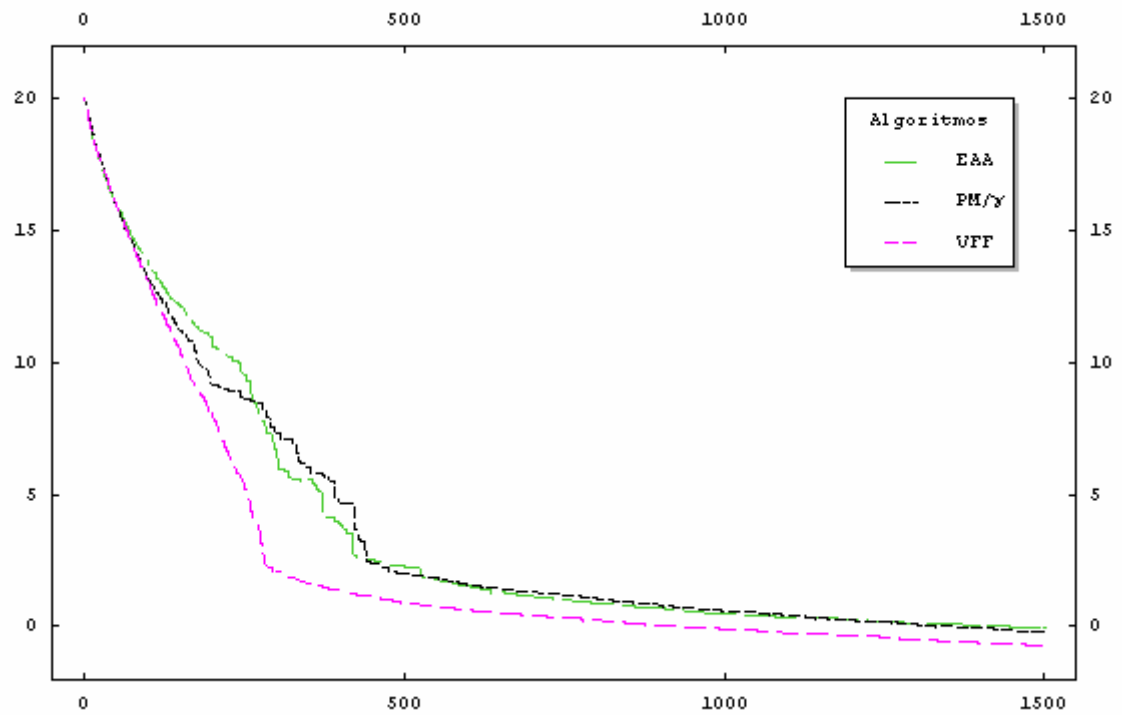


Figura 39. Comparativo de desempenho dos algoritmos para a função f_8 (eixo das ordenadas em logaritmo natural).

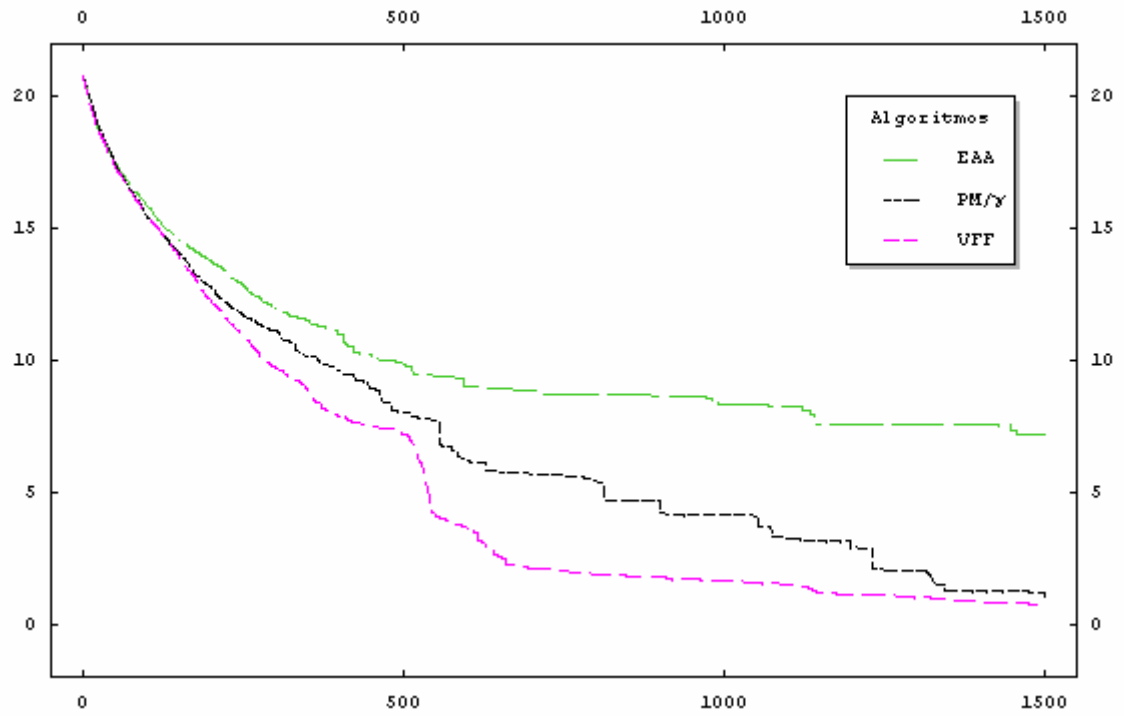


Figura 40. Comparativo de desempenho dos algoritmos para a função f_9 (eixo das ordenadas em logaritmo natural).

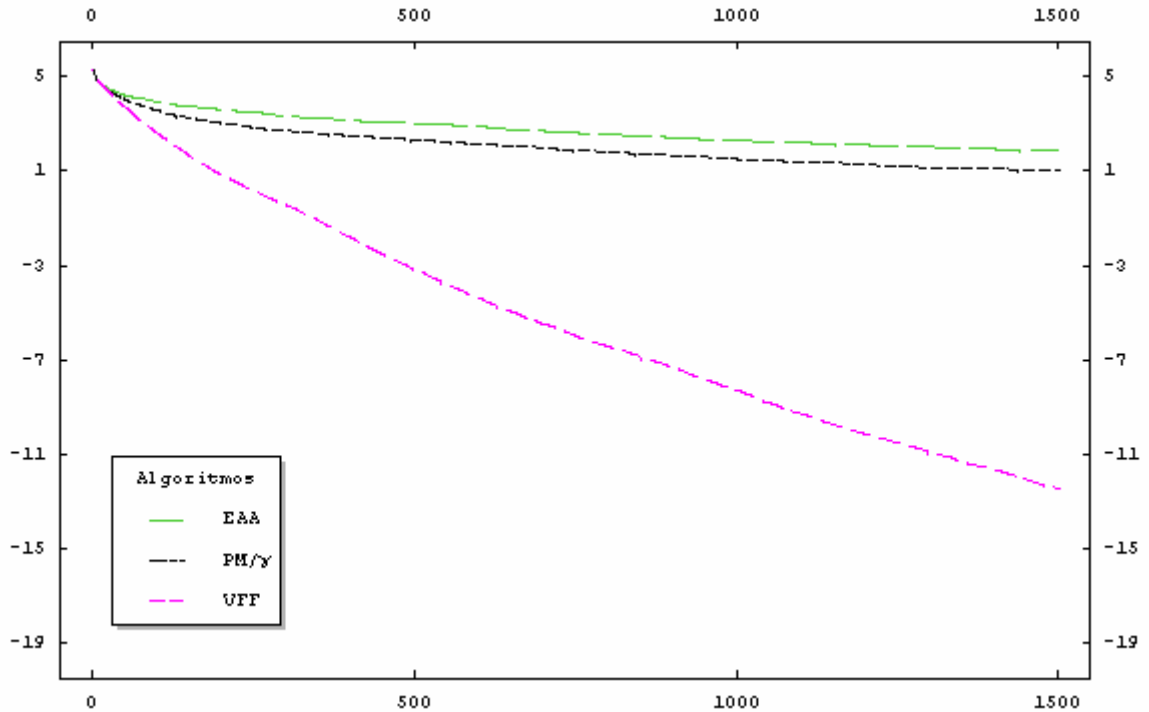


Figura 41. Comparativo de desempenho dos algoritmos para a função f_{10} (eixo das ordenadas em logaritmo natural).

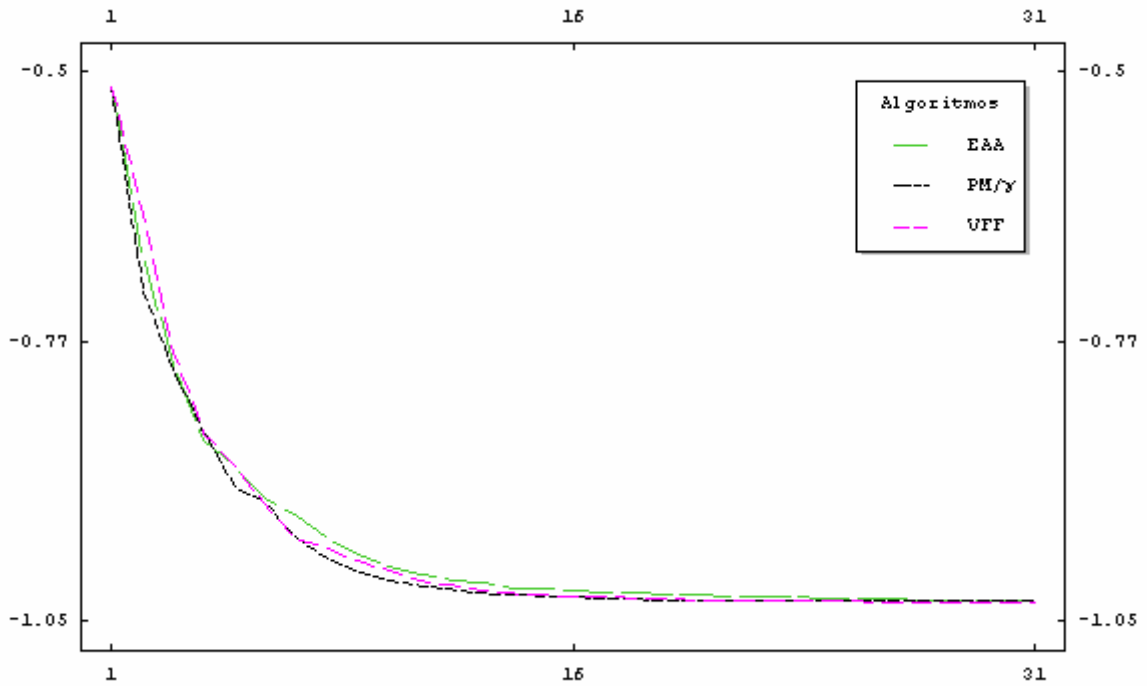


Figura 42. Comparativo de desempenho dos algoritmos para a função f_{11} .

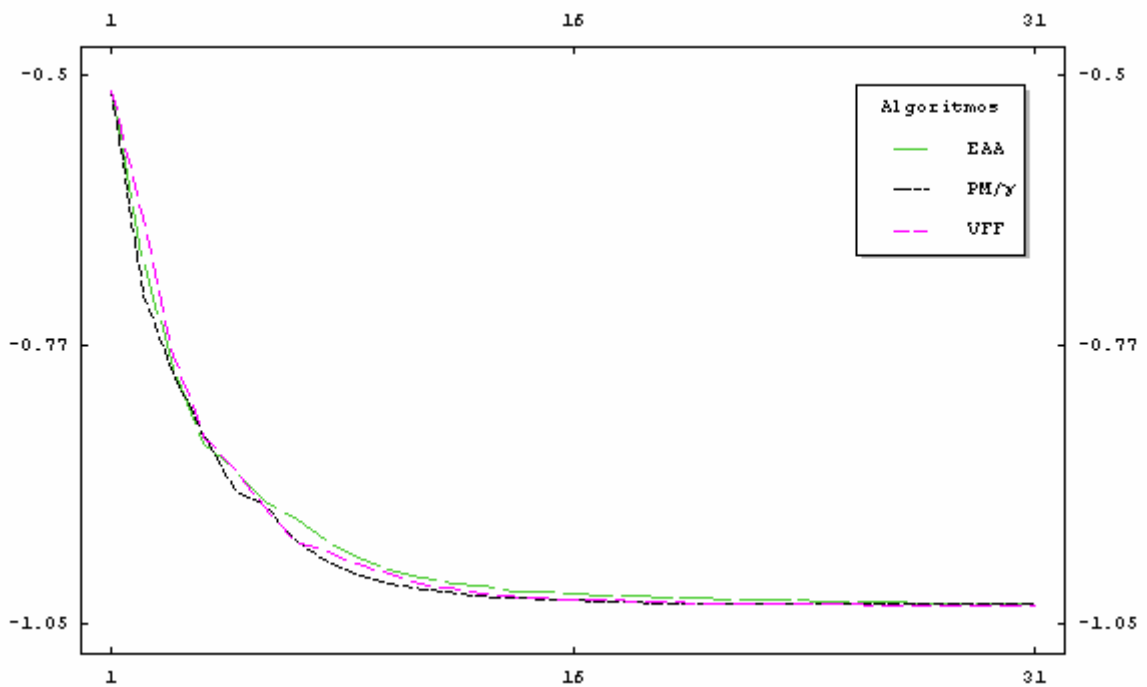


Figura 43. Comparativo de desempenho dos algoritmos para a função f_{12} .

Nas funções f_2 , f_4 , f_7 , e f_8 a análise gráfica não permite distinção significativa entre os resultados dos algoritmos ao final das 1500 execuções. Enquanto que na primeira e nas duas

últimas funções o algoritmo VFF apresenta resultado levemente melhor que os demais, na função f_4 o algoritmo PM/ γ é quem se destaca.

Merece algum destaque o resultado apresentado pelo algoritmo VFF na função f_1 , que apresentava sinais de estabilidade no intervalo de 700 a 1300 gerações e, logo em seguida, apresentou forte decréscimo em sua curva. Nessa alteração, cruzou a curva do algoritmo PM/ γ atingindo o melhor resultado ao final de 1500 gerações, sendo que sua tendência continua decrescente.

Nas funções f_3 , f_5 e f_{10} os algoritmos EAA e PM/ γ apresentam resultados similares entre si, porém, sempre muito aquém do resultado apresentado pelo algoritmo VFF. Nas duas primeiras funções, enquanto o algoritmo VFF demonstra sinais de estabilidade, o PM/ γ apresenta decréscimo na sua tendência.

Na função f_6 os algoritmos PM/ γ e VFF apresentam-se levemente melhores que o algoritmo EAA. Apesar de não existir distinção significativa entre os algoritmos PM/ γ e VFF ao final das execuções, verifica-se tendência decrescente acentuada do penúltimo algoritmo quando comparado ao último.

Nas funções f_{11} e f_{12} não é possível estabelecer distinção significativa entre os resultados dos algoritmos ao final de 30 gerações.

5.2.4 Resultados dos Algoritmos para o Dobro do Número de Gerações Original

Esta seção apresenta comparativo de resultados entre os melhores algoritmos selecionados nas seções anteriores, para o dobro do número de gerações original.

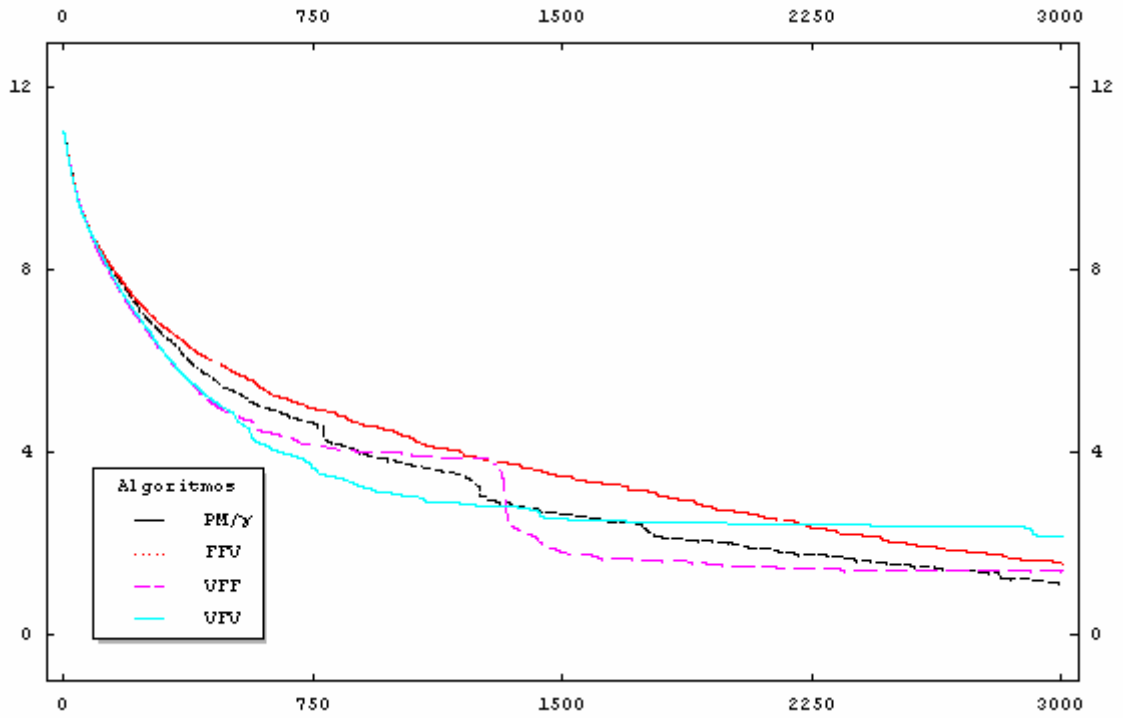


Figura 44. Comparativo de desempenho dos algoritmos para a função f_1 (eixo das ordenadas em logaritmo natural).

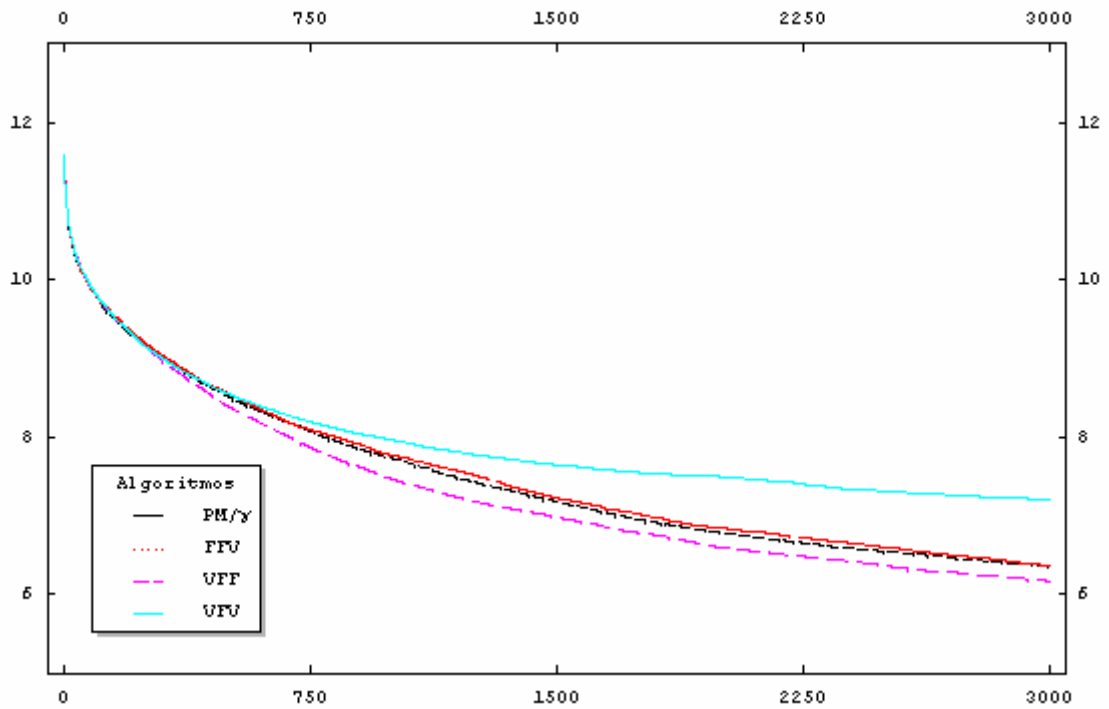


Figura 45. Comparativo de desempenho dos algoritmos para a função f_2 (eixo das ordenadas em logaritmo natural).

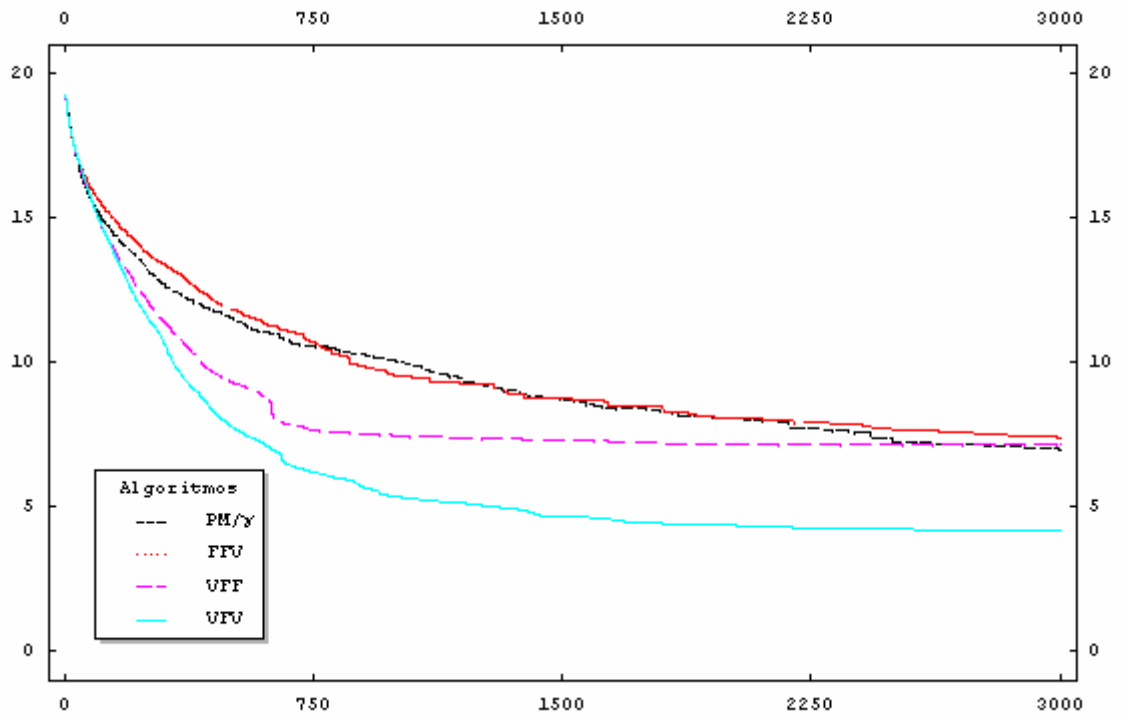


Figura 46. Comparativo de desempenho dos algoritmos para a função f_3 (eixo das ordenadas em logaritmo natural).

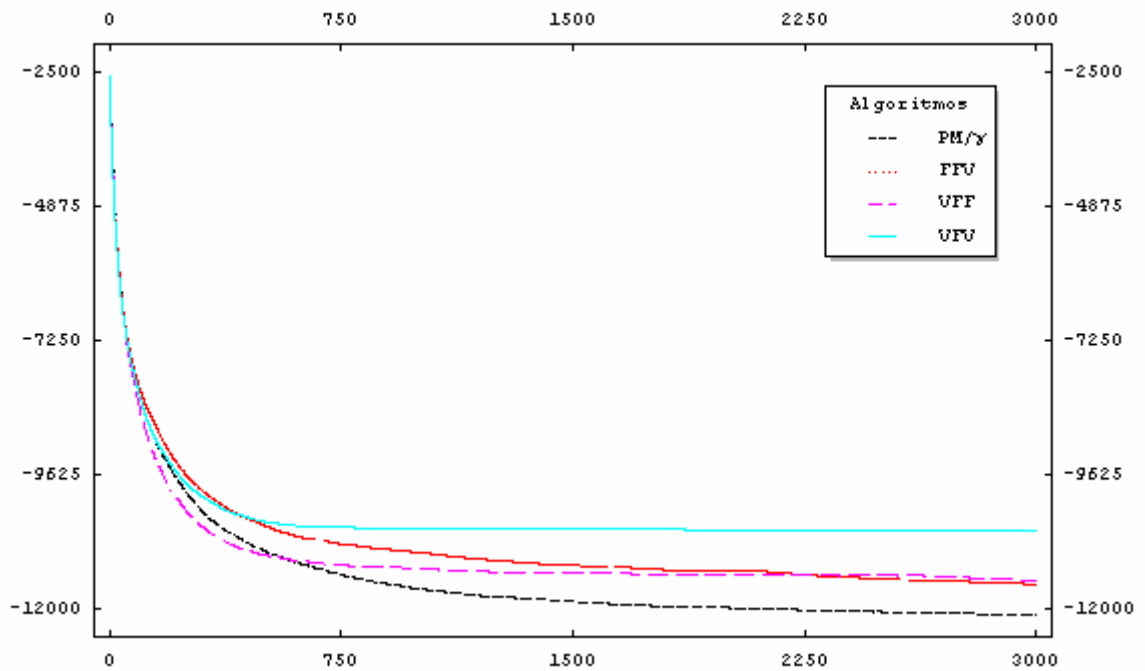


Figura 47. Comparativo de desempenho dos algoritmos para a função f_4 .

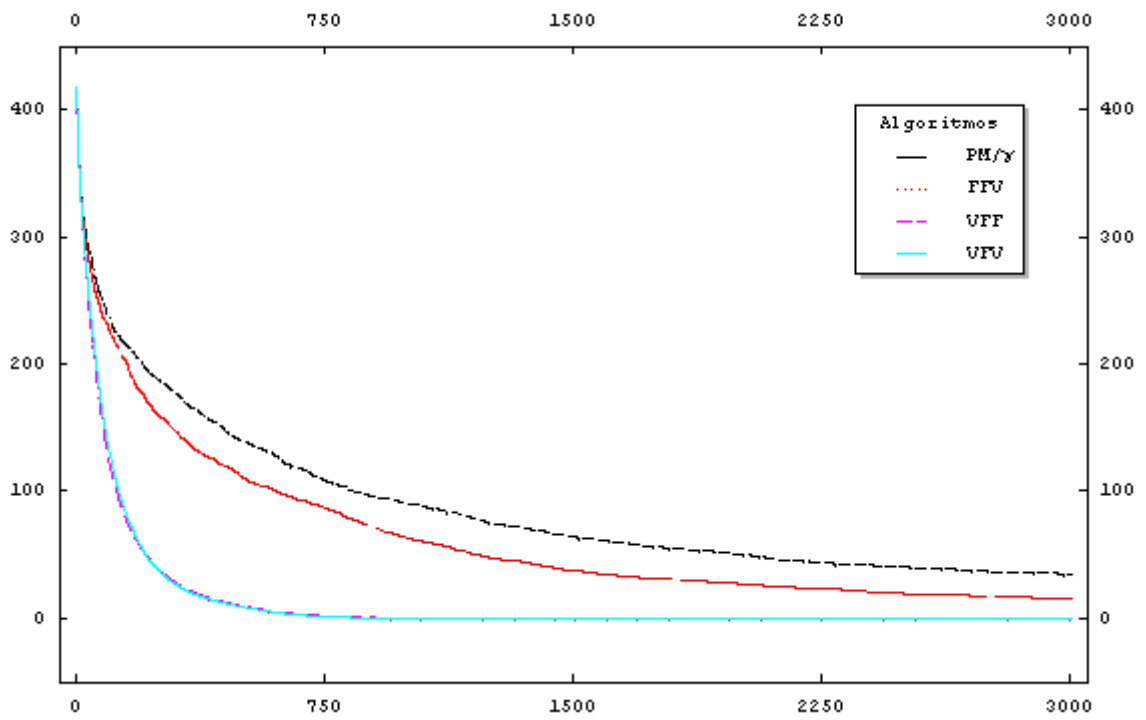


Figura 48. Comparativo de desempenho dos algoritmos para a função f_5 .

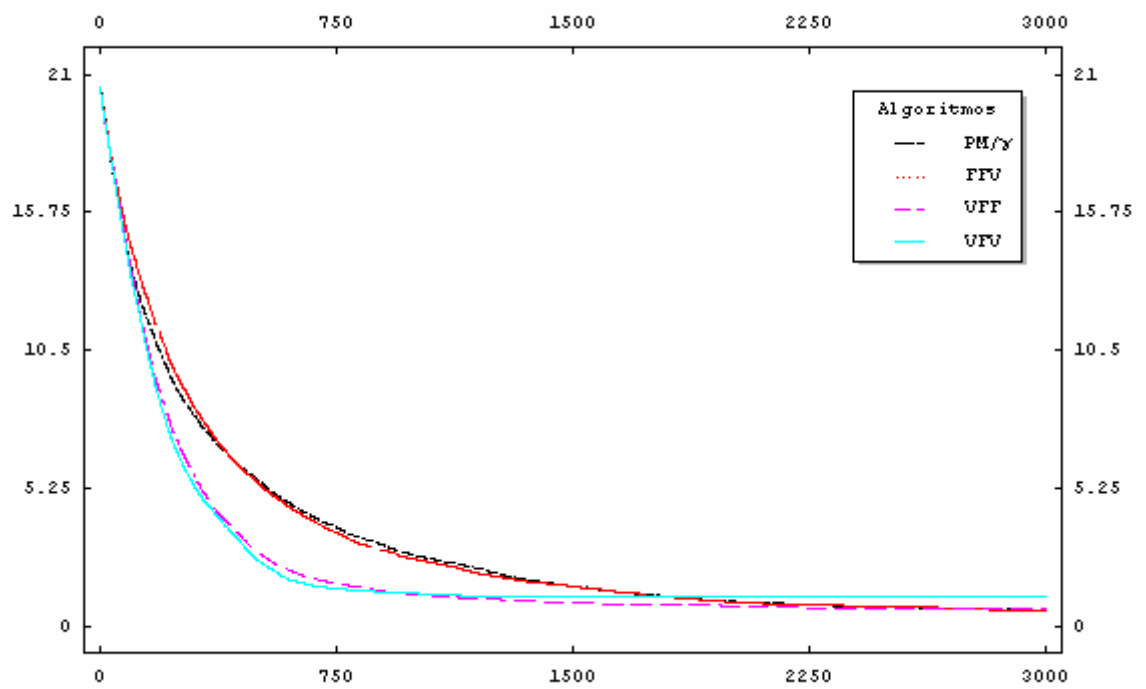


Figura 49. Comparativo de desempenho dos algoritmos para a função f_6 .

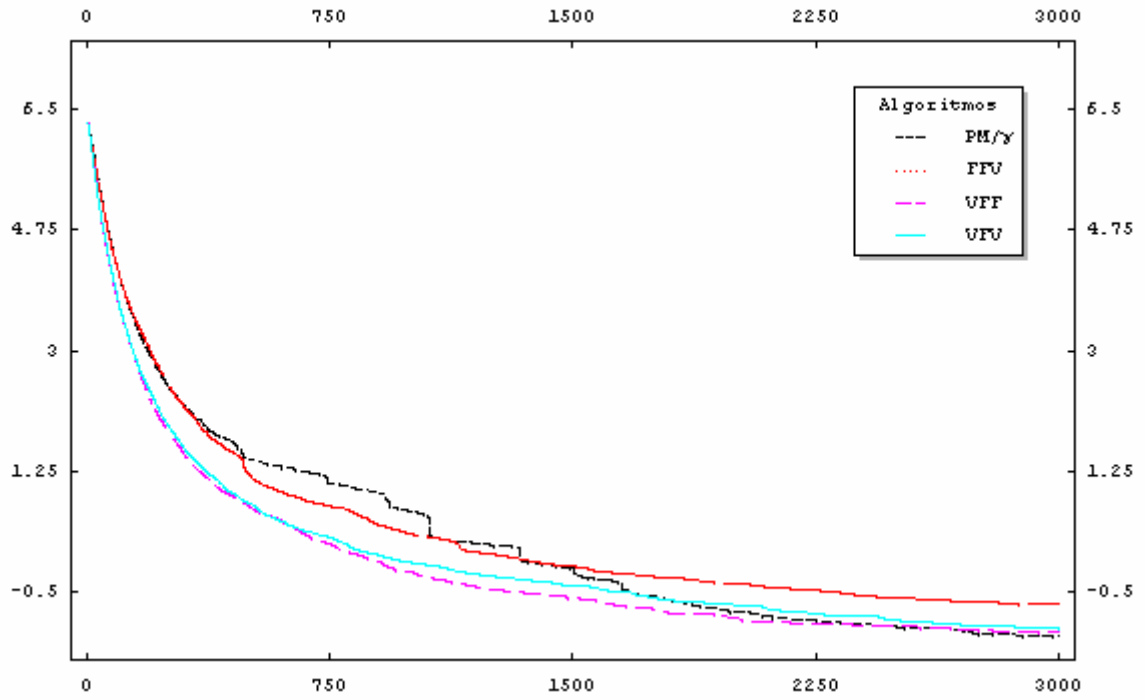


Figura 50. Comparativo de desempenho dos algoritmos para a função f_7 .

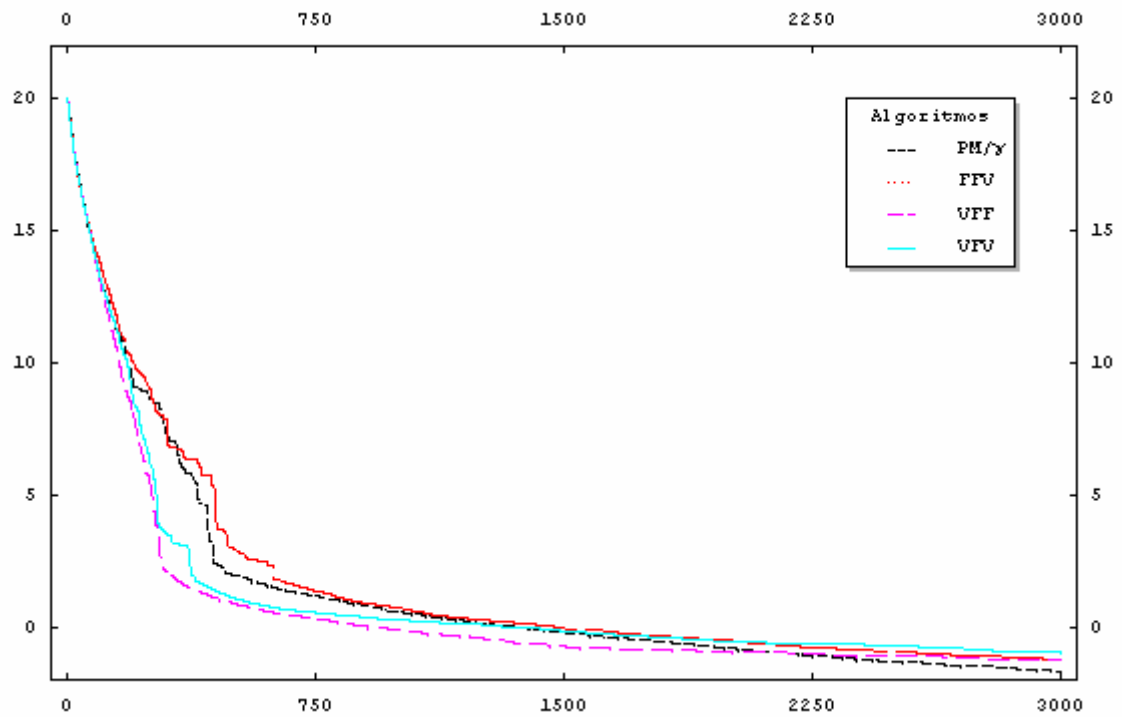


Figura 51. Comparativo de desempenho dos algoritmos para a função f_8
(eixo das ordenadas em logaritmo natural).

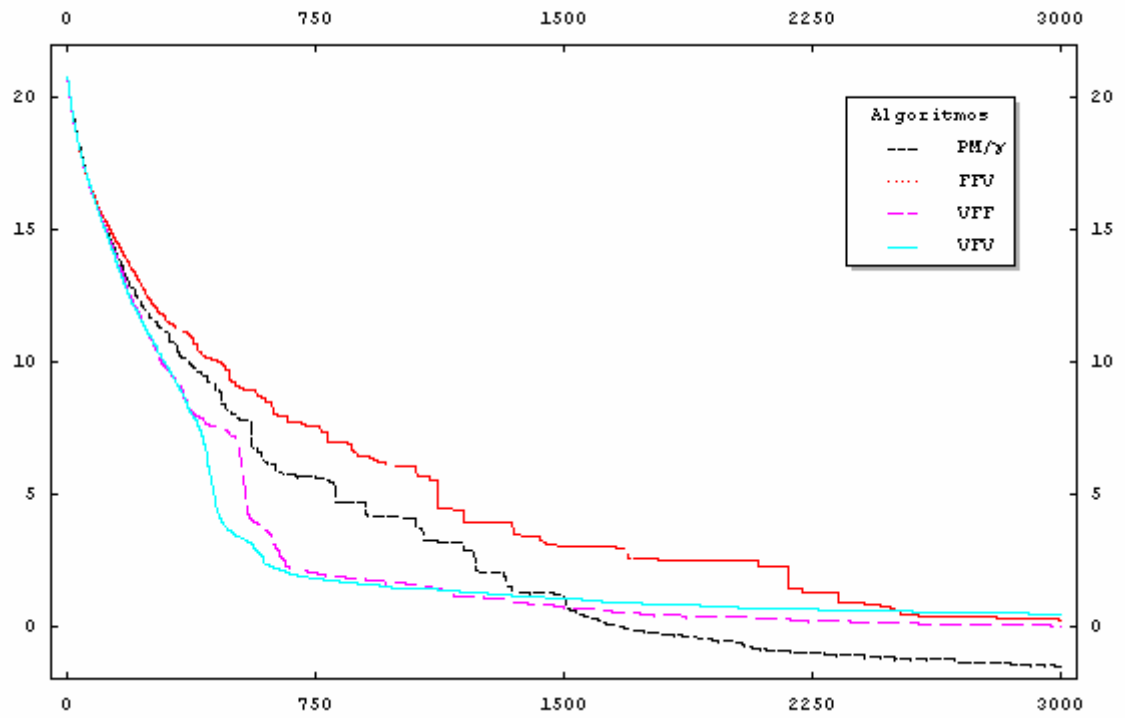


Figura 52. Comparativo de desempenho dos algoritmos para a função f_9 (eixo das ordenadas em logaritmo natural).

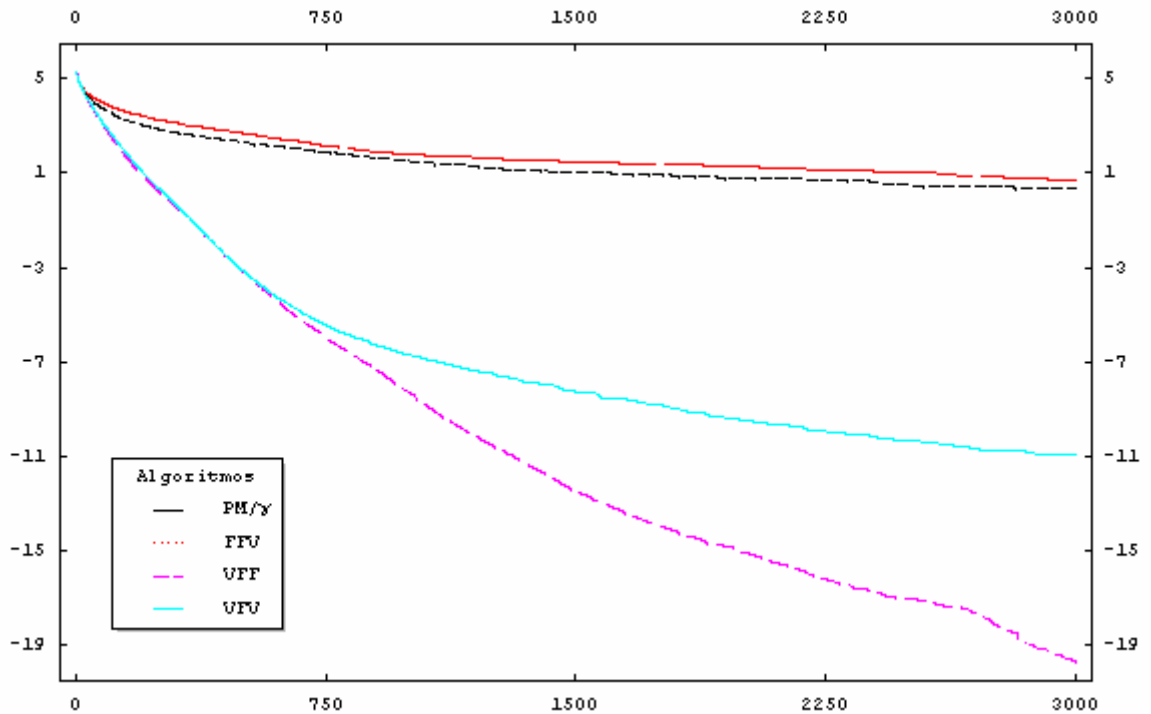


Figura 53. Comparativo de desempenho dos algoritmos para a função f_{10} (eixo das ordenadas em logaritmo natural).

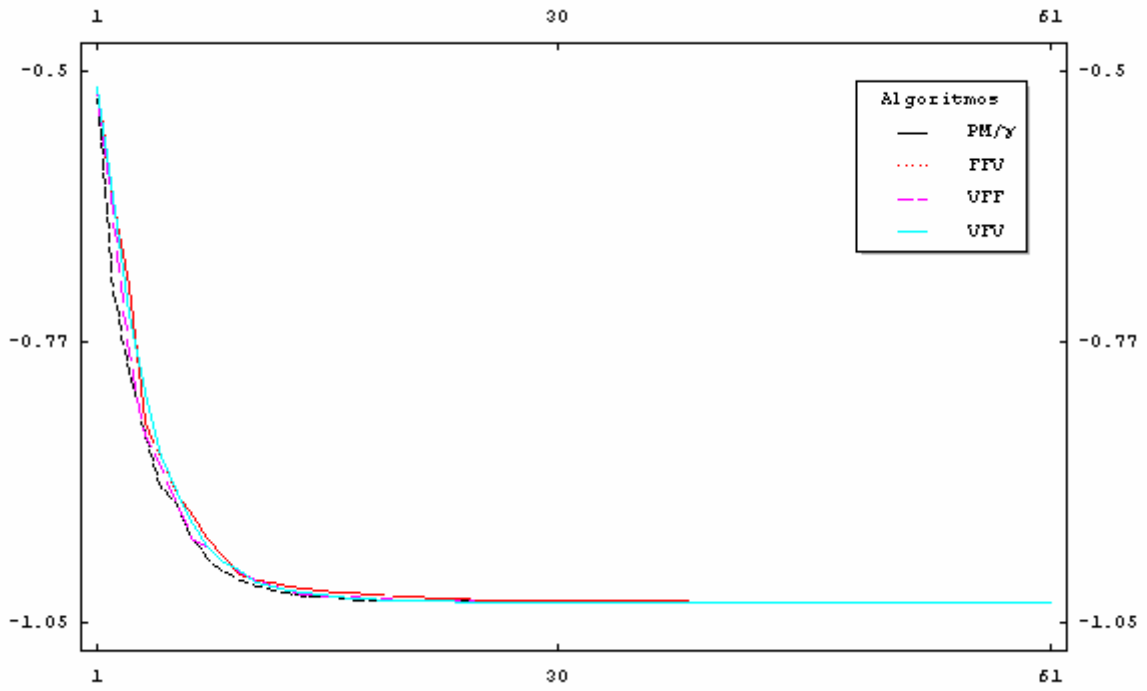


Figura 54. Comparativo de desempenho dos algoritmos para a função f_{11} .

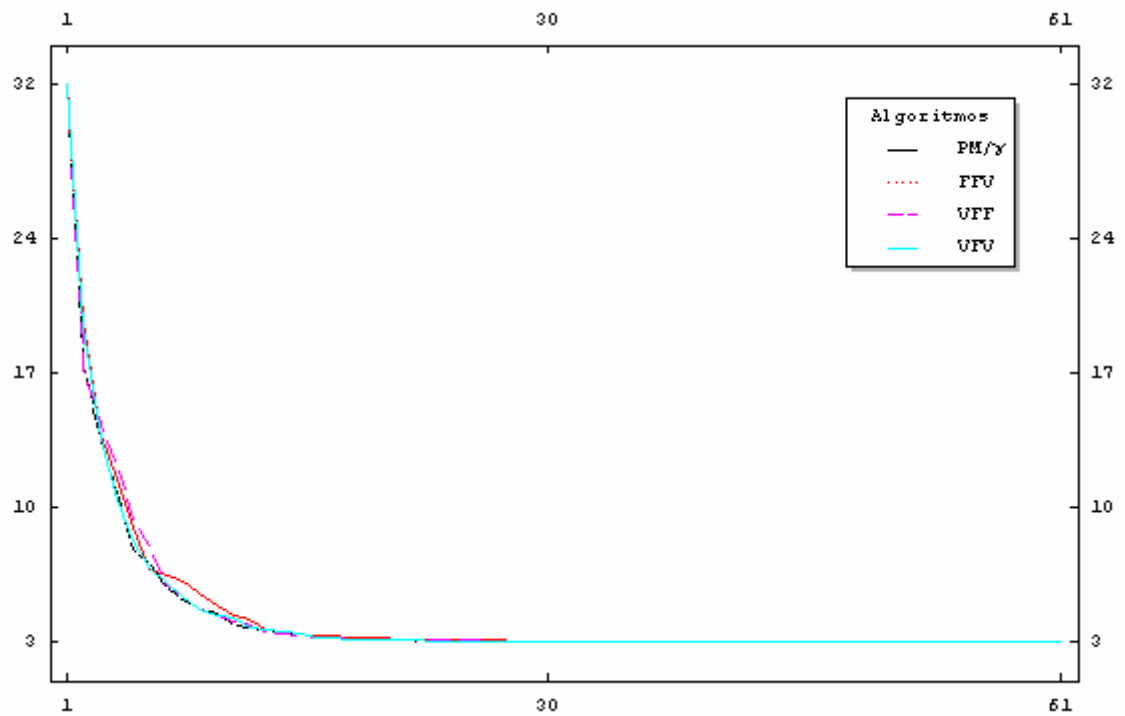


Figura 55. Comparativo de desempenho dos algoritmos para a função f_{12} .

Nas funções f_1, f_6, f_7, f_8 e f_9 os algoritmos VFF e VFV apresentaram convergência inicial rápida, seguida de estabilidade ao final do processo. Essa convergência lenta ao final do processo

permitiu que seus resultados fossem alcançados pelo algoritmo PM/ γ antes do final das 3000 gerações. Fato semelhante ocorreu com o algoritmo VFF na função f_3 , que foi ultrapassado pelo PM/ γ nas proximidades das últimas gerações. Nesta mesma função, o algoritmo VFV apresentou o melhor e mais destacado resultado.

Na função f_4 prevalece o resultado favorável do algoritmo PM/ γ frente aos demais, sendo que nessa mesma função, o pior resultado foi obtido pelo algoritmo VFV, que manteve-se constante a partir da geração 500.

Nas funções f_5 e f_{10} os algoritmos PM/ γ e VFV apresentam resultados similares entre si, porém, sempre muito aquém do resultado apresentado pelos algoritmos VFF e VFV. Nas duas funções, enquanto os algoritmos VFF e VFV demonstram sinais de estabilidade ao final do processo, os demais aparentam decrescimento na sua tendência. Considerando os melhores resultados obtidos na função f_{10} , enquanto o algoritmo VFV estabiliza-se em 750 gerações, o algoritmo VFF permanece com forte decrescimento na sua tendência.

As análises gráficas das funções f_{11} e f_{12} não permitem estabelecer distinção dos resultados dos algoritmos ao final de 60 gerações.

5.3 Análise Numérica

De posse dos resultados de cada um dos algoritmos, a análise numérica buscou estabelecer hipóteses estatísticas que permitissem diferenciá-los ou classificá-los com respeito às suas aptidões em resolver o problema de achar os mínimos das funções testadas.

5.3.1 Análise Numérica da Auto-Adaptação no Parâmetro α de uma Distribuição Estável

O primeiro comparativo estabelece a hipótese de que um determinado algoritmo do tipo SEP apresenta resultado superior ao obtido pelo GEP. Para isso, considere-se a Tabela 5, que contém os resultados experimentais das 50 execuções independentes para os quatro algoritmos SEP e o GEP, aplicados a cada uma das funções de teste presentes na Tabela 4.

Para todos os algoritmos, a primeira linha apresenta a média aritmética e a segunda o desvio-padrão de 50 execuções, do resultado da última geração. As médias aritméticas e desvios-padrão referem-se aos valores da função objetivo calculados para os melhores indivíduos. Portanto, os menores valores observados da segunda à sexta coluna da Tabela 5, representam os melhores resultados obtidos pelos algoritmos para cada uma das funções. A penúltima coluna da tabela apresenta o resultado do teste estatístico T-Student, bilateral com 49 graus de liberdade.

Portanto, o valor da estatística do teste deve ser maior ou igual a ± 2.0 , de acordo com a Equação 53 do Anexo A, para garantir que o nível mínimo (5%) de significância seja atingido, a fim de comprovar a diferença de médias entre o algoritmos PM e GEP (a razão para essa escolha será discutida mais abaixo). O nível de significância atingido pelo teste pode ser verificado através do p-valor, que encontra-se na segunda linha da penúltima coluna. A última coluna da tabela apresenta a diferença absoluta de médias entre os valores obtidos pelo algoritmos PM e GEP.

Função	Algoritmo					T-Student	Diferença de Média
	FM	FU	PM	PU	GEP		
f_1	19.7	687.6	12.7	800.5	66.0	-1.6803	-53.5
	51.4	736.9	45.6	736.1	220.3	0.0988	
f_2	1366.1	31169.9	2096.3	33146.7	2084.7	0.0371	11.6
	1004.1	8305.7	1707.8	8484.0	1398.9	0.9704	
f_3	876.7	1.0×10^5	107.0	1.1×10^5	1092.7	-1.8860	-985.6
	4002.1	2.0×10^5	117.1	2.6×10^5	3693.6	0.0652	
f_4	-9658.1	-12171.0	-10605.8	-12127.8	-7626.8	-13.7125	-2979.0
	1736.4	231.5	1418.3	228.6	590.1	< 0.0001	
f_5	4.2×10^{-3}	51.8	2.0×10^{-3}	57.4	48.5	-7.3976	-48.5
	2.6×10^{-2}	11.5	1.0×10^{-2}	11.4	46.3	< 0.0001	
f_6	0.5	7.1	1.1	7.4	4.4	-8.3992	-3.3
	1.2	2.3	1.8	2.0	2.1	< 0.0001	
f_7	0.6	8.4	0.7	12.2	6.8	-4.6414	-6.1
	0.8	7.1	0.9	14.4	9.3	< 0.0001	
f_8	0.4	3.4×10^3	0.9	1.3×10^4	2.5	-5.2325	-1.6
	0.6	1.4×10^4	1.3	6.9×10^4	1.8	< 0.0001	
f_9	1.5	1.0×10^5	2.9	3.3×10^5	3.4	-0.5365	-0.5
	2.3	4.3×10^5	5.5	1.1×10^6	3.3	0.5931	
f_{10}	1.5×10^{-5}	2.1	2.6×10^{-4}	2.0	5.1	-12.0323	-5.1416
	8.1×10^{-5}	2.4	1.2×10^{-3}	0.9	3.0	< 0.0000	
f_{11}	-1.0312	-1.0270	-1.0313	-1.0189	-1.0263	-6.5057	-4.9×10^{-3}
	0.0010	0.0074	0.0006	0.0255	0.0053	< 0.0001	
f_{12}	3.0229	3.0885	3.0198	3.1823	3.1238	-4.8611	-0.1
	0.04206	0.1758	0.0351	0.4158	0.1472	< 0.0001	

Tabela 5. Resultados experimentais para SEP e GEP.

Os valores assinalados em negrito na Tabela 5 correspondem aos melhores resultados obtidos pelos algoritmos do tipo SEP para uma determinada função, sem considerar o resultado do GEP. Assim, o algoritmo PM apresentou o melhor desempenho em 5 das 12 funções (f_1 , f_3 , f_5 , f_{11} e f_{12}), o algoritmo FM apresentou o melhor desempenho em 6 das 12 funções (f_2 , f_6 , f_7 , f_8 , f_9 e f_{10}), enquanto que o algoritmo FU apresentou apenas um melhor resultado obtido na função f_4 .

Apesar do algoritmo PM obter uma pequena desvantagem numérica em relação ao algoritmo FM, pela diferença de 1 melhor resultado em favor deste, o teste estatístico foi feito utilizando o algoritmo PM na comparação de resultados com o GEP. A primeira justificativa dessa iniciativa deve-se ao fato de que a quantidade de resultados favoráveis, considerando apenas os dois algoritmos, é exatamente o mesmo, dado que o resultado da função f_4 é favorável ao PM. A segunda justificativa deve-se ao fato de que o algoritmo PM é conceitualmente mais natural, pois utiliza-se das características genéticas do pai para posteriormente fazer a mutação do parâmetro objetivo do filho, sem contar que ambos utilizam a mesma definição de serem multidimensionais (M).

Agora, considerando o resultado apenas dos algoritmos GEP e PM e sua diferença de médias, este último apresentou o melhor desempenho em 11 das 12 vezes, excetuando-se na função f_2 , cujo resultado foi favorável ao GEP. Apenas 4 das 12 funções (f_1 , f_2 , f_3 e f_9) não atingiram o nível de significância mínimo para comprovar a diferença de médias entre os algoritmos PM e GEP, enquanto para as demais funções deve-se rejeitar a hipótese nula de igualdade de médias. Esse resultado qualifica a superioridade numérica do algoritmo PM sobre o GEP, pois em aproximadamente 67% das vezes (em 8 das 12 funções) seu resultado é numericamente superior.

5.3.2 Análise Numérica da Aplicação por Nível de Adaptação

O próximo comparativo estabelece a hipótese de que pode existir um algoritmo com diferentes níveis de adaptação que seja melhor que o algoritmo VFV (ou PM). Para isso, considere-se a Tabela 6, que contém os resultados experimentais das 50 execuções independentes para os oito novos algoritmos do tipo SEP aplicados a cada uma das funções de teste da Tabela 4.

Para todos os algoritmos, a primeira linha apresenta a média aritmética e a segunda o desvio-padrão das 50 execuções, do resultado da última geração. As médias aritméticas e desvios-padrão referem-se aos valores da função objetivo calculados para os melhores indivíduos. Portanto, os menores valores observados da segunda à nona coluna da tabela representam os melhores resultados obtidos pelos algoritmos para cada uma das funções. A penúltima coluna apresenta o resultado do teste estatístico T-Student, bilateral com 49 graus de liberdade, para a diferença de médias entre o algoritmos VFF e VFV (a razão para essa escolha será discutida mais abaixo). Ainda nessa coluna, a primeira linha apresenta o valor da estatística de teste e a segunda linha o seu correspondente p-valor. A última coluna apresenta a diferença absoluta de médias entre o resultado obtido pelos algoritmos VFF e VFV.

Função	Algoritmo								T-Student	Diferença de Média
	FFF	FFV	FVF	FVV	VFF	VFV	VVF	VVV		
f_1	78.5 179.4	32.6 81.3	54.7 75.4	45.7 83.0	6.2 22.7	12.7 45.6	12.9 33.8	90.6 275.9	-0.9124 0.3646	-6.6
f_2	1866.0 1394.0	1380.3 930.1	2194.2 1241.5	1655.0 921.1	1077.6 941.4	2096.3 1707.8	1390.7 1142.3	1588.7 903.5	-3.6937 0.0004	-1018.7
f_3	34353.7 105315.0	6227.9 23297.7	3049.1 5925.6	5131.1 13706.9	1511.1 8049.0	107.0 117.1	2978.2 7837.3	15604.3 41475.4	1.2333 0.2233	1404.0
f_4	-11071.9 1303.3	-11245.9 1149.9	-11221.3 398.3	-11715.8 402.0	-11381.6 1308.5	-10605.8 1418.3	-11871.2 397.0	-11647.4 345.6	-2.8430 0.0054	-775.8
f_5	57.7 48.5	37.7 48.6	70.0 43.9	73.9 46.2	5.5×10^{-3} 3.2×10^{-2}	2.0×10^{-3} 1.0×10^{-2}	75.7 45.1	96.1 46.9	0.7499 0.4562	-3.5×10^{-3}
f_6	3.7 3.4	1.5 1.7	3.7 1.9	3.7 1.5	0.9 1.3	1.1 1.8	2.7 1.5	3.6 1.7	-0.6330 0.5283	-0.2
f_7	1.8 6.1	0.9 1.1	4.4 9.3	1.0 1.4	0.5 0.7	0.7 0.9	0.9 0.9	1.2 1.1	-0.6789 0.4988	-0.1
f_8	4.4 6.1	0.9 1.9	1.4 1.5	1.0 1.1	0.5 0.8	0.9 1.3	0.9 1.7	1.2 1.1	-1.7986 0.0758	-0.4
f_9	670.1 3985.6	21.1 125.2	4.3 4.9	2.8 3.0	2.1 4.2	2.9 5.5	2.7 2.8	4.1 4.3	-0.8474 0.3990	-0.8
f_{10}	7.3 7.5	4.3 6.4	6.9 4.4	7.5 5.6	3.8×10^{-6} 8.6×10^{-6}	2.6×10^{-4} 1.2×10^{-3}	7.9 5.1	8.7 4.6	-1.5665 0.1237	-2.6×10^{-4}
f_{11}	-1.0288 0.0046	-1.0293 0.0032	-1.0277 0.0064	-1.0295 0.0024	-1.0311 0.0010	-1.0313 0.0006	-1.0305 0.0018	-1.0305 0.0023	-0.9487 0.3457	-1.5×10^{-4}
f_{12}	3.0766 0.0939	3.0497 0.0826	3.0799 0.1154	3.0690 0.1208	3.0253 0.0674	3.0198 0.0351	3.0603 0.0724	3.0314 0.0468	0.5117 0.6103	-5.5×10^{-3}

Tabela 6. Resultados experimentais do SEP por nível de adaptação.

Os valores assinalados em negrito na Tabela 6 correspondem ao melhor resultado obtido pelos algoritmos do tipo SEP para uma dada função de teste. Assim, pode-se observar que o algoritmo VFV apresentou o melhor desempenho em 4 das 12 funções (f_3, f_5, f_{11} e f_{12}), o VFF foi o melhor em 7 das funções ($f_1, f_2, f_6, f_7, f_8, f_9$ e f_{10}), e, finalmente, o algoritmo VVF teve o melhor desempenho apenas na função f_4 .

Devido ao número de ocorrências de melhores resultados, foram comparados os algoritmos VFV e VFF para uma comprovação de resultados através da análise de hipótese estatística, utilizando o teste T-Student. Considerando apenas os resultados dos dois algoritmos, o VFV apresentou o melhor desempenho em 4 das 12 funções (f_3, f_5, f_{11} e f_{12}), enquanto que o algoritmo VFF foi o melhor em 8 das 12 funções ($f_1, f_2, f_4, f_6, f_7, f_8, f_9$ e f_{10}).

Lembrando que o valor da estatística de teste deve ser maior ou igual a ± 2.0 para garantir que o nível mínimo requerido (5%) de significância, e que este pode ser verificado através do p-valor (na segunda linha da penúltima coluna), observa-se que apenas 2 das 12 funções (f_2 e f_4) atingiram o nível de significância mínimo para comprovar a diferença de médias entre os algoritmos VFV e VFF. Como os dois resultados são favoráveis ao algoritmo VFF, fica comprovado sua superioridade numérica sobre o algoritmo anterior.

5.3.3 Análise Numérica da Auto-Adaptação Elitista e Auto-Adaptação nos Parâmetros α e γ da Distribuição Estável

O penúltimo comparativo estabelece a hipótese de que uma das duas novas proposições de algoritmos pode apresentar resultado superior ao obtido pelo algoritmo VFF. Para isso, considere-se a Tabela 7, que contém os resultados experimentais das 50 execuções independentes para dois novos algoritmos (EAA e PM/ γ) e o algoritmo VFF, aplicados às funções de teste já mencionadas. Valem para a tabela, todas as convenções já apresentadas para as tabelas anteriores desta Seção 5.3.3. Lembre-se apenas que a última coluna da tabela apresenta a diferença absoluta de médias entre o resultado obtido pelo PM/ γ e o VFF (a razão para essa escolha será discutida mais abaixo).

Função	Algoritmo			T-Student	Diferença de Média
	EAA	PM/ γ	VFF		
f_1	54.4 164.8	13.9 35.8	6.2 22.7	1.2988 0.1976	7.8
f_2	1439.8 1089.1	1312.5 912.8	1077.6 941.4	1.2665 0.2083	234.9
f_3	11913.4 50397.5	6048.3 21203.7	1511.1 8049.0	1.4146 0.1621	4537.3
f_4	-11236.9 727.9	-11888.2 725.7	-11381.6 1308.5	-2.3937 0.0191	-506.5
f_5	96.9 35.1	64.5 57.7	5.5×10^{-3} 3.2×10^{-2}	7.8979 < 0.0001	64.5
f_6	3.3 2.4	1.5 1.9	0.9 1.3	1.8088 0.0739	0.6
f_7	1.0 1.5	0.8 1.2	0.5 0.7	1.4796 0.1428	0.3
f_8	1.0 1.6	0.8 1.8	0.5 0.8	1.2278 0.2238	0.3
f_9	1348.4 7137.2	2.8 12.8	2.1 4.2	0.4024 0.6888	0.8
f_{10}	6.4 6.5	2.9 4.7	3.8×10^{-6} 8.6×10^{-6}	4.2654 < 0.0001	2.9
f_{11}	-1.0300 0.0021	-1.0303 0.0021	-1.0311 0.0010	2.5349 0.0135	8.5×10^{-4}
f_{12}	3.0808 0.1518	3.0395 0.0558	3.0253 0.0674	1.1485 0.2536	1.4×10^{-2}

Tabela 7. Resultados experimentais dos algoritmos EAA, PM/ γ e VFF.

Os valores assinalados em negrito na Tabela 7 correspondem aos melhores resultados obtidos pelos dois novos algoritmos para a função correspondente. Note-se que o algoritmo VFF não apresentou o melhor desempenho em apenas uma função (f_4), sendo que nesta, o algoritmo PM/ γ foi que obteve o melhor resultado.

Devido à frequência de vitórias verificada pelos algoritmos VFF e PM/ γ , eles serão comparado entre si para uma comprovação de resultados através da análise de hipótese estatística, utilizando o teste T-Student.

Considerando o valor requerido da estatística de teste tal qual os casos anteriores já discutidos, observa-se que apenas 4 das 12 funções (f_4 , f_5 , f_{10} e f_{11}) atingiram o nível de significância mínimo que comprova a diferença de médias entre os algoritmos VFF e PM/ γ .

sendo que em apenas 1 delas (f_4) o resultado é desfavorável ao algoritmo VFF; ou seja, do ponto de vista estatístico o algoritmo VFF foi o que apresentou a melhor resultado.

5.3.4 Resultados dos Algoritmos para o Dobro do Número de Gerações Original

O último comparativo procura explorar os resultados apresentados nas análises gráficas e numéricas das seções anteriores, selecionando os melhores algoritmos em uma rodada com o dobro do número de gerações original. Para isso, considere-se a Tabela 8, que contém os resultados experimentais das 50 execuções independentes para os algoritmos PM/ γ , FFV, VFF e VFV, aplicados às funções de teste já mencionadas. Valem para a tabela todas as convenções já apresentadas para as tabelas anteriores da Seção 5.3.3. Lembre-se apenas que a última coluna da tabela apresenta a diferença absoluta de médias entre o resultado obtido pelo PM/ γ e o VFF (a razão para essa escolha será discutida mais abaixo).

Função	Algoritmo				T-Student	Diferença de Média
	PM/ γ	FFV	VFF	VFV		
f_1	3.2	4.7	4.1	8.6	-0.2506	-0.8850
	11.0	17.0	22.4	41.9	0.8028	
f_2	580.0	583.1	479.1	1349.0	0.7352	100.9
	617.3	567.3	748.4	1458.4	0.4640	
f_3	1047.3	1605.0	1308.6	65.7	-0.2108	-261.3
	3571.9	8611.2	8005.9	52.8	0.8337	
f_4	-12133.8	-11585.8	-11508.6	-10617.7	-3.0611	-625.1
	625.8	861.5	1301.4	1412.5	0.0031	
f_5	34.9	15.7	7.9×10^{-7}	1.7×10^{-5}	4.7554	34.9
	51.9	31.4	3.9×10^{-6}	1.2×10^{-4}	< 0.0001	
f_6	0.6	0.6	0.7	1.1	-0.1209	-3.0×10^{-2}
	1.3	0.9	1.2	1.8	0.9040	
f_7	0.3	0.5	0.3	0.3	-0.1614	-2.0×10^{-2}
	0.6	1.0	0.7	0.7	0.8721	
f_8	0.2	0.3	0.3	0.4	-0.9303	-0.1
	0.5	0.7	0.6	0.6	0.3545	
f_9	0.2	1.3	1.0	1.6	-2.1155	-0.8
	0.8	5.3	2.5	4.2	0.0387	
f_{10}	1.4	2.0	2.9×10^{-9}	1.8×10^{-5}	3.2134	1.4
	3.2	4.2	1.4×10^{-8}	1.1×10^{-4}	0.0023	
f_{11}	-1.0313	-1.0311	-1.0316	-1.0316	2.6524	2.5×10^{-4}
	0.0006	0.0014	0.0003	0.0000	0.0098	
f_{12}	3.0055	3.0063	3.0022	3.0004	1.7159	3.4×10^{-3}
	0.0099	0.0089	0.0097	0.0013	0.0893	

Tabela 8. Resultados experimentais dos algoritmos PM/ γ , FFV, VFF e VFV.

Os valores assinalados em negrito na Tabela 8 correspondem aos melhores resultados obtidos pelos dois novos algoritmos para a função correspondente. Assim, pode-se observar que o algoritmo PM/ γ apresentou o melhor desempenho em 5 das 12 funções (f_1 , f_4 , f_7 , f_8 e f_9), o algoritmo VFF foi o melhor em 3 das funções (f_2 , f_5 e f_{10}), o algoritmo VFV foi o melhor em 3 das funções (f_3 , f_{11} e f_{12}), e finalmente o algoritmo FFV obteve o melhor resultado na função f_6 .

Devido ao empate na segunda posição entre os algoritmos VFF e VFV, o primeiro foi selecionado para ser comparado com o algoritmo PM/ γ em uma comprovação de resultados através da análise de hipótese estatística, utilizando o teste T-Student. A seleção do algoritmo VFF para fazer parte do teste estatístico, em detrimento ao VFV, deve-se fato de que este apresentou-se melhor em funções relativamente mais difíceis.

Considerando o valor requerido da estatística de teste tal qual nos casos anteriores já discutidos, observa-se que apenas 5 das 12 funções (f_4 , f_5 , f_9 , f_{10} , e f_{11}) atingiram o nível de significância mínimo que comprova a diferença de médias entre os algoritmos PM/ γ e VFF, sendo que em 3 delas (f_5 , f_{10} e f_{11}) o resultado é favorável ao algoritmo VFF; ou seja, do ponto de vista estatístico o algoritmo VFF foi o que apresentou a melhor resultado.

Considere-se agora a Tabela 9, que compara os resultados obtidos em [Lee e Yao, 2004] e o resultado do estudo atual. A segunda coluna da tabela apresenta o resultado do melhor LEP através da aplicação de diferentes valores fixos de α ao longo das gerações, e a terceira coluna apresenta o seu GEP correspondente. A quinta coluna contém o melhor algoritmo SEP do presente estudo (VFF), e a sexta o resultado do GEP correspondente. A quarta e sétima colunas da tabela mostram o rendimento em percentual do melhor LEP e SEP em relação aos seus respectivos GEP.

Função	Literatura			Estudo Atual		
	Melhor LEP	GEP	Rendimento	VFF	GEP	Rendimento
f_1	0.000632	0.000950	50%	4.062080	66.213900	1530%
f_2	0.041850	497.120615	1187763%	479.144000	2084.730000	335%
f_3	43.400000	98.673779	127%	1308.610000	1092.660000	-17%
f_4	1100.300000	4592.823300	317%	1060.900000	4942.720000	366%
f_5	5.850000	63.415617	984%	0.000001	48.485900	6216140926%
f_6	0.019000	8.882487	46650%	0.667130	4.429730	564%
f_7	0.024000	0.058204	143%	0.337786	6.815870	1918%
f_8	0.000006	0.620834	10347133%	0.284514	2.479950	772%
f_9	0.000098	0.094838	96673%	1.015620	3.394570	234%
f_{11}	0.000630	0.001195	90%	0.000050	0.005300	10500%
f_{12}	0.000548	0.211548	38504%	0.002190	0.123830	5554%

Tabela 9. Comparativo de resultados obtido em [Lee e Yao, 2004] e estudo atual.

Mencione-se que os valores do algoritmo VFF foram retirados da Tabela 8, enquanto que os valores do algoritmo GEP foram retirados da Tabela 5, com uma diferença para as funções f_4 , f_{11} e f_{12} , que tiveram seus valores absolutos subtraídos do mínimo global para que o mínimo global final de todas as funções fosse equalizado em zero. Além disso, enquanto o resultado obtido em [Lee e Yao, 2004] considera 1500 gerações, o resultado comparativo do presente estudo utiliza-se de 3000 gerações.

Um detalhe importante que deve ser considerado é que, apesar da semelhança na implementação do algoritmo, os resultados numéricos obtidos especificamente pelo GEP são, em alguns casos, significativamente diferentes daqueles observados em [Lee e Yao, 2004]. Apenas 3 das 11 funções (f_5 , f_6 e f_{12}) apresentaram resultados favoráveis ao presente trabalho, sendo que nas funções unimodais (f_1 - f_3) o nível de precisão obtido em [Lee e Yao, 2004] é bastante superior ao obtido no presente trabalho. Apesar da semelhança nas parametrizações dos algoritmos, é possível que os parâmetros objetivos das populações iniciais tenham sido inicializados de maneira diferente em cada um dos estudos, pois esta condição não foi explicitada em [Lee e Yao, 2004]. Caso essa diferença se confirme, certamente isso seria um fator de contribuição para as diferenças verificadas na Tabela 9. Para dar uma idéia do efeito dessa proposição, foi realizado um teste utilizando o algoritmo GEP aplicado na função f_1 , onde os parâmetros objetivos da primeira população foram inicializados utilizando distribuição de probabilidade Normal $N(0, 1)$. O resultado observado nesse experimento foi uma média igual a 3.3 e desvio-padrão 5.2, frente a uma média de 66.2 e desvio-padrão igual a 220.3 obtida no presente estudo. Apesar de uma redução significativa na média e desvio-padrão, este resultado continua muito inferior aquele apresentado em [Lee e Yao, 2004]. Vale ressaltar que esse último resultado corresponde ao critério de parametrização utilizado no presente estudo, onde a primeira população foi inicializada utilizando distribuição de probabilidade uniforme, conforme pode ser verificado na Seção 5.1.

Em cada um dos estudos, comparando o resultado dos algoritmos adaptativos (LEP ou SEP) com seus respectivos GEP, obtem-se os valores relativos observados nas colunas ‘Rendimento’ da Tabela 9. Em 6 das 11 funções (f_2 , f_3 , f_6 , f_8 , f_9 e f_{12}), o algoritmo LEP apresentou melhor rendimento que o SEP, enquanto que em 5 das 11 funções (f_1 , f_4 , f_5 , f_7 e f_{11}), o algoritmo SEP apresentou melhor rendimento que o LEP, atestando a qualidade do SEP em um comparativo direto de resultados.

5.3.5 Resultados dos Algoritmos VFV e VFV sem Restrições no Espaço de Busca

Esta seção apresenta os resultados obtidos na primeira implementação dos algoritmos baseados na definição descrita na Seção 4.1, conforme sugerido em [Lee e Yao, 2004]. Devido a uma ausência na definição do comportamento do parâmetro objetivo x'_{ij} gerado para um filho que ultrapasse os limites do espaço de busca, a implementação inicial do algoritmo foi feita sem atender para esse detalhe. O controle desse tipo de restrição é particularmente importante com o uso de distribuições de probabilidade com cauda longa porque, por sua própria natureza, constantemente ocorre um valor aleatório que desloca o parâmetro objetivo x'_{ij} do filho para as extremidades do domínio da função.

A detecção desse problema ocorreu a partir dos resultados apresentados pela função f_4 , que em várias tentativas, não convergia para o seu ponto ótimo. Mesmo não havendo uma regra para restrição do espaço de busca, o algoritmo desenvolvido inicialmente apresentou bons resultados de otimização, conforme pode ser visto na Tabela 10. A segunda coluna apresenta os resultados do algoritmo VFV retirados da Tabela 6, que considera as restrições no espaço de busca definidas no item 1 da Seção 4.2, enquanto que a terceira coluna da tabela apresenta os resultados do algoritmo VFV sem nenhuma restrição no espaço de busca. Para facilitar sua referência, o algoritmo VFV sem restrição utiliza a sigla VFV-SR.

Função	Algoritmo	
	VFV	VFV-SR
f_1	12.7	7.6
	45.6	15.7
f_2	2096.3	1872.7
	1707.8	1518.7
f_3	107.0	427.0
	117.1	1063.8
f_5	2.0×10^{-3}	74.3
	1.0×10^{-2}	27.3
f_6	1.1	3.5
	1.8	2.1
f_7	0.7	1.6
	0.9	3.3
f_{11}	-1.0313	-1.0121
	0.0006	0.0289
f_{12}	3.0198	5.3700
	0.0351	2.9300

Tabela 10. Comparativo de resultados entre o algoritmo VFV e VFV-SR.

O resultado do algoritmo VFV-SR mostra-se superior ao VFV nas funções f_1 e f_2 . Nas funções f_3 , f_6 e f_7 , apesar de não apresentar resultado superior ao algoritmo VFV, o VFV-SR apresentou bom desempenho, tendo em vista o grau dificuldade destas funções. Estes resultados permitem concluir que os algoritmos do tipo SEP tem boa performance na resolução de problemas que apresentam condições adversas.

Capítulo 6: Conclusão

Recentes trabalhos em computação evolutiva tem explorado a utilização de diferentes distribuições de probabilidade no processo aleatório associado ao operador de mutação. Cronologicamente, os primeiros trabalhos relatam a utilização da distribuição Normal, seguidos pelo uso da distribuição de Cauchy, e, finalmente pela distribuição de probabilidade de Lévy.

Um dos principais aspectos que impulsionou essa transição está relacionado à forma característica da curva de probabilidade, que assume cada vez mais importância nas suas caudas, devido ao aumento da necessidade de geração de valores extremos, principalmente na otimização de funções multimodais. Além disso, as funções anteriores têm em comum a características de serem estáveis (propriedade da soma de variáveis aleatórias identicamente distribuídas), aspecto não explorado nesse trabalho mas que interfere no processo de otimização, pois, como a variabilidade na mutação representa um processo aditivo ao parâmetro objetivo, este último é identicamente distribuído a sua geradora de variabilidade.

Como uma representante das estáveis, a distribuição de Lévy vem assumindo cada vez mais importância nas ciências exatas devido à comprovação de sua capacidade frente a resolução de problemas de alta complexidade. Soma-se a esse resultado, o efeito causado pela adaptabilidade no parâmetro estratégico da distribuição de Lévy, que permite obter refinamento ainda mais significativo no resultado de um experimento.

Assim como na Lévy, o uso de distribuições de probabilidade estáveis constitui importante elemento no quesito adaptabilidade, devido a sua característica genérica na determinação de distribuições de probabilidade, associado à definição de uma quádrupla de parâmetros. Mais do que adaptabilidade, o presente estudo explorou a auto-adaptação dos parâmetros estratégicos de uma distribuição estável na resolução de problemas de minimização semelhante àqueles encontrados em [Lee e Yao, 2004].

O início desse trabalho foi marcado pela definição de um algoritmo do tipo GEP, semelhante ao descrito em [Lee e Yao, 2004], na tentativa de reprodução de seus resultados. De certa forma a reprodução não atingiu plenamente seu objetivo, pois os resultados observados nos dois estudos são claramente distintos. Dentre os fatores limitantes para a reprodução dos resultados apresentados em [Lee e Yao, 2004], destacam-se:

- Ausência na definição do processo de variabilidade responsável pela inicialização do parâmetro objetivo da população;
- Erros nas formulações das funções de teste;

- Ausência na definição do processo seletivo associado ao torneio (aleatório com ou sem repetição);
- Ausência na definição do critério para tratamento dos parâmetros objetivos quando gerados além dos domínios definidos pelas funções de teste; e
- Diferenças nas definições de algoritmos e equações de mutação, entre as publicações do mesmo autor;

O último item citado foi motivador da primeira proposta de abordagem adaptativa aqui sugerida. Nesse comparativo, os resultados gráficos e numéricos evidenciam que os algoritmos adaptativos que utilizam abordagem multidimensional (M) são mais eficientes que os algoritmos que utilizam abordagem unidimensional (U). Além disso, o fato de o parâmetro estratégico σ guardar as características do pai (P) ou utilizar a característica do filho (F) não influencia diretamente no seu resultado. As análises gráficas e numéricas bem como o fato de ser um algoritmo definido conforme os conceitos genéticos, levaram o algoritmo PM para uma comprovação de resultados frente ao algoritmo GEP, e ele se mostrou superior ao último.

Uma segunda proposição baseada na combinação de alguns níveis de adaptação sugeriu a criação de 7 novos algoritmos auto-adaptativos. Apesar de utilizar adaptação por indivíduo, população ou bloco de gerações, apenas se destacarem os algoritmos variáveis na população, fixo no indivíduo e variável ou fixo por blocos de gerações, respectivamente, os algoritmos VFV e VFF. Nesse contexto, realizou-se teste de comparação numérica entre esses algoritmos e o resultado final foi favorável ao VFF. Ao contrário do que pode se imaginar em princípio, tornar variáveis os parâmetros dos componentes do indivíduo não se mostrou boa estratégia. Por outro lado, permitir que cada indivíduo da população tenha o seu próprio parâmetro estratégico, e, controlar o nível de adaptação por blocos de gerações mostra-se como a estratégia mais adequada para a resolução da maioria das funções de teste listadas na Tabela 4.

Duas novas alternativas de métodos auto-adaptativos foram sugeridas na Seção 4.2.3, e dentre estas, aquela que apresentou melhor resultado (PM/ γ), foi levada a um confronto numérico com o algoritmo VFF. Apesar de perder no confronto numérico, as análises gráficas do algoritmo PM/ γ permitiram identificar sua característica de convergência lenta.

De posse dessa informação, foram selecionados alguns algoritmos que participaram das análises anteriores para uma nova rodada de testes considerando uma quantidade de gerações duas vezes maior que a original. A seleção dos algoritmos levou em consideração a característica de convergência lenta ao longo das gerações, haja vista que o critério de parada definido por [Lee e Yao, 2004] pode estar em desacordo com as novas definições de algoritmos utilizados no

presente estudo. Nessa última comparação, os algoritmos VFF e VFV foram algumas vezes ultrapassados pelo algoritmo PM/ γ próximo ao final das 3000 gerações, comprovando sua característica de convergência lenta. Porém, apesar do algoritmo PM/ γ apresentar o maior número de resultados favoráveis, quando levado a um confronto numérico com o segundo melhor algoritmo (VFF), seus resultados foram significativamente inferiores.

A qualidade do resultado apresentado pelo algoritmo auto-adaptativo VFF versus o resultado do algoritmo adaptativo verificado em [Lee e Yao, 2004] é o último comparativo do presente estudo (Tabela 9), onde estão presentes os resultados absolutos obtidos em [Lee e Yao, 2004] para o LEP e GEP, mas que, devido à ausência de algumas definições, não são considerados comparáveis ao presente nesse estudo. Para tornar válido o comparativo foi necessária a inclusão da informação adicional dos algoritmos LEP e SEP, relativa a seus respectivos GEP. Nesse comparativo fica evidenciada a superioridade numérica do LEP sobre SEP em 6 das 11 funções, e lembrando-se que o SEP utiliza o dobro de gerações para atingir seus resultados. No entanto, deve-se lembrar que a versão (pseudo-)adaptativa do algoritmo LEP utiliza-se, de fato, de 4 população de indivíduos para cada geração processada.

Apesar do efeito comparativo das últimas análises, o presente trabalho questiona a replicabilidade dos resultados apresentados em [Lee e Yao, 2004], fato que invalidaria a qualidade numérica apresentada na Tabela 9.

Uma outra questão que pode ser levantada é quanto ao poder do teste estatístico aplicado nos comparativos, tendo em vista a necessidade de aproximação por uma distribuição de probabilidade Normal. Neste sentido, seria interessante testar o uso de testes não paramétricos, que, apesar de terem menos poder, não necessitam de pressuposição paramétrica para sua aplicação.

Trabalhos futuros sobre o mesmo assunto poderiam considerar a possibilidade de trabalhar com distribuições de probabilidade diferentes da Normal no operador estratégico σ . Além disso, seria interessante explorar a possibilidade de trabalhar com outras formas de restrição no operador objetivo x'_{ij} , seja através da geração de um novo número aleatório ou levar o parâmetro objetivo x'_{ij} até a fronteira de domínio da função.

De forma geral, os resultados apresentados nesse estudo encorajam a aplicação do SEP em outros estudos, devido à sua praticidade na implementação, e qualidade dos resultados atingidos.

Referências Bibliográficas

- [Anderson, 2006] C. Anderson. “The Long Tail”. Disponível em http://en.wikipedia.org/wiki/The_Long_Tail. Último acesso em julho de 2007.
- [Bäck e Schwefel, 1993] T. Bäck e H.-P. Schwefel. “An overview of evolutionary algorithms for parameter optimization”, *Evolutionary Computation*, 1:1-23, 1993.
- [Bäck, Hammel e Schwefel, 1997] T. Bäck, U. Hammel e H.-P. Schwefel. “Evolutionary computational: comments on the history and current state”, *IEEE Transactions on Evolutionary Computation*, 1(1):3-17, 1997.
- [Beyer, 2001] H.-G. Beyer. *The theory of evolution strategies*. Springer-Verlag, 2001.
- [Bittencourt, 2006] G. Bittencourt. *Inteligência Artificial, ferramentas e teorias*. Editora da Universidade Federal de Santa Catarina, 3ª edição, 2006.
- [Chellapilla, 1998] K. Chellapilla. “Combining Mutation Operators in Evolutionary Programming”, *IEEE Transactions on Evolutionary Computation*, 3(2):91-96, 1998.
- [de Oliveira e Gutierrez, 2006] P.P.B. de Oliveira e A.B.M. Gutierrez. “Auto-Adaptação em Estratégias Evolutivas com Distribuições Estáveis: Resultados Preliminares”, In: *Anais do I Workshop on Computational Intelligence - IX Simpósio Brasileiro de Redes Neurais*. Ribeirão Preto, SP: Universidade de São Paulo, 2006.
- [Digalakis e Margaritis, 2000] J.G. Digalakis e K.G. Margaritis. “An experimental study of benchmarking functions for Genetic Algorithms”, *IEEE Transactions on Evolutionary Computation*, 2000.
- [Eiben e Smith, 2003] A.E. Eiben e J.E. Smith. *Introduction to evolutionary computing*. Springer-Verlag, 2003.

- [Eiben, Hinterding e Michalewicz, 1999] A.E. Eiben, R. Hinterding e Z. Michalewicz. “Parameter control in evolutionary algorithms”, *IEEE Transactions on Evolutionary Computation*, 3(2):124–141, 1999.
- [Fogel, 1994] D.B. Fogel. “An Introduction to Simulated Evolutionary Optimization”, *IEEE Trans. on Neural Networks. An Introduction to Evolutionary Computation*. 1(5): 3-14, 1994.
- [Fogel, 2000] D.B. Fogel. “Principles of evolutionary processes”, In: T. Bäck. D.B. Fogel e Z. Michalewicz, eds. *Evolutionary Computation I*. Institute of Physics Publishing, 3:23-26, 2000.
- [Gnedenko e Kolmogorov, 1954] B. Gnedenko e A. Kolmogorov. *Limit Distribution for Sums of Independent Independent Random Variables*. Cambridge, MA: Addition-Wesley, 1954.
- [GEATbx, 2007] “GEATbx: Example Functions (single and multi-objective functions)”. Disponível em http://www.geatbx.com/docu/fcnindex-01.html#P82_2611. Último acesso em julho de 2007.
- [Goldberg, 1989] D.E. Goldberg. *Genetic algorithms in search. optimization and machine learning*. Addison-Wesley. 1989. *Apud* [Eiben e Smith, 2003].
- [Gutierrez e de Oliveira, 2007] A.B.M. Gutierrez e P.P.B. de Oliveira. “Estratégias Evolutivas com Mutações Governadas por Distribuições Estáveis”, submetido ao *Six Congress of Logic Applied to Technology (Laptec)*, 2007.
- [Hedar, 2007] A.-R. Hedar. “Global Optimization Test Problems”. Disponível em http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm. Último acesso em julho de 2007.
- [Holland, 1992] J.-H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor. MI: University of Michigan Press, 2nd Edition,, 1992.

- [Koenig, 2002] A.C. Koenig. *A Study of Mutation Methods for Evolutionary Algorithms*. Advanced Topics in Artificial Intelligence, p.1-8, 2002. Disponível em <http://web.umr.edu/~tauritzd/courses/cs401/fs2002/project/Koenig.pdf>. Último acesso em julho de 2007. Material a ser publicado ou trata-se apenas de notas de aulas.
- [Lee e Yao, 2001] C.-Y. Lee e X. Yao. “Evolutionary algorithms with adaptive Lévy mutations”, *Proc. of the 2001 Congress on Evolutionary Computation*, IEEE Press, Piscataway, NJ, USA, p.568-575, 2001.
- [Lee e Yao, 2004] C.-Y. Lee e X. Yao. “Evolutionary Programming Using Mutations Based on the Lévy Probability Distribution”, *IEEE Transactions on Evolutionary Computation*, 8(1):1-13, 2004.
- [Lima e de Oliveira, 2006] H.S. Lima e P.P.B. de Oliveira. “Estratégias de variação automática da pressão seletiva em uma classe de algoritmos evolutivos”. *Seleção Documental*, 1(3):5-12, 2006.
- [Meyer, 1976] P.L. Meyer. *Probabilidade – Aplicações à Estatística*, Departamento de Matemática, Washington State University, 1976.
- [Mayr, 1988] E. Mayr. *Toward a New Philosophy of Biology: Observation of an Evolutionist*, Belknap, Harvard, 1988.
- [Nieberg e Beyer, 2006] S.M.-Nieberg e H.-G. Beyer. “Self-Adaptation in Evolutionary Algorithms”, In F. Lobo, C. Lima, and Z. Michalewicz, eds. *Parameter Setting in Evolutionary Algorithms*, Springer-Verlag, 2006. Disponível em <http://www2.staff.fh-vorarlberg.ac.at/~hgb/New-Papers/self-adaptation.pdf>. Último acesso em julho de 2007. Material a ser publicado ou trata-se apenas de notas de aulas.
- [Nolan, 2005] J-P. Nolan. *Stable Distributions. Models for Heavy Tailed Data*. American University. Math/Stat Department, 2005. Disponível em

<http://academic2.american.edu/~jpnolan/stable/chap1.pdf>. Último acesso em julho de 2007.

[Nolan, 2004] J.P. Nolan. STABLE Program for Windows. Pacote versão de 19/04/2004. Disponível em <http://academic2.american.edu/~jpnolan/stable/stable.html>. Último acesso em janeiro de 2007.

[Rimmer e Nolan, 2005] R.H Rimmer e J.P. Nolan. “Stable distributions in Mathematica”. *The Mathematica Journal*, 9(4):776-788, 2005.

[Rudolph, 1998] G. Rudolph. “Local convergence rates of simple evolutionary algorithms with Cauchy mutations”. Universität Dortmund. Fachbereich Informatik, Technical Report, LS XI, p.1-20, 1998.

[Tsallis, 2000] C. Tsallis. “As Distribuições de Lévy”, *Revista Brasileira de Ensino de Física*, 22(2):156-162, 2000.

[Yao e Liu, 1996] X. Yao e Y. Liu. “Fast evolutionary programming”, In *Evolutionary Programming V: Proceedings of the 5th Annual Conference on Evolutionary Programming*, p.451-460, Cambridge, MA: MIT Press, 1996.

[Yao, Liu e Lin, 1999] X. Yao, Y. Liu e G. Lin. “Evolutionary programming made faster”, *IEEE Transactions on Evolutionary Computation*, 3(2):82-102, 1999.

Anexo A - Teste de Hipótese Estatístico

Um teste de hipótese estatístico é aquele que estabelece suposições sobre a distribuição de uma variável aleatória. Uma das pressuposições para aplicar um teste assintótico de diferença de médias é que a variável aleatória de interesse seja normalmente distribuída. Porém, como na prática nem sempre é possível estabelecer normalidade, uma consequência do Teorema do Limite Central [Meyer, 1976] é que o teste de médias pode ser aplicado quando o tamanho da amostra (n) for suficiente grande e os dados forem comprovadamente unimodais.

Considere dois processos distintos, onde o primeiro pode ser caracterizado pela variável aleatória X com distribuição Normal $N(\mu_x, \sigma_x^2)$, enquanto que o segundo processo é caracterizado pela variável aleatória Y com distribuição Normal $N(\mu_y, \sigma_y^2)$. Com o objetivo de compará-los, deseja-se testar a hipótese nula $H_0: \mu_x = \mu_y$ contra a hipótese alternativa $H_1: \mu_x - \mu_y > 0$. Como nesse caso a variância é desconhecida, σ^2 pode ser estimado por

$$\hat{\sigma} = [1/(n-1)] \sum_{i=1}^n (X_i - \bar{X})^2. \quad (50)$$

Utilizando a estatística $(\bar{X} - \mu)\sqrt{n}/\hat{\sigma}$ que possui distribuição T-Student com $(n - 1)$ graus de liberdade, deve-se rejeitar a hipótese nula H_0 quando

$$(\bar{X} - \mu)\sqrt{n}/\hat{\sigma} > C, \quad (51)$$

onde C representa a região crítica ou improvável da distribuição de probabilidade sobre a hipótese nula. A definição da região crítica C depende do nível de significância do teste α preestabelecido, que normalmente é situado em 5%, mas que também pode ser igual a 1% ou 10%, de acordo com o julgamento do pesquisador. Seja α o nível de significância preestabelecido, então

$$\alpha = P[(\bar{X} - \mu)\sqrt{n}/\hat{\sigma} > C]. \quad (52)$$

Em um teste bilateral, a região crítica C pode ser calculada pela estatística $t_{n-1, 1-\alpha/2}$, obtida através de uma tábua de distribuição da T-Student. Nesse teste deseja-se rejeitar a hipótese H_0 sempre que

$$\bar{X} > \pm \hat{\sigma} t_{n-1, 1-\alpha/2} n^{-1/2} + \mu_0. \quad (53)$$