ESTRATÉGIAS PARA MELHORIA DO DESEMPENHO DO ALGORITMO DE PROGRAMAÇÃO DE EXPRESSÃO GENÉTICA NA TAREFA DE REGRESSÃO SIMBÓLICA

WAGNER R. WEINERT, HEITOR S. LOPES

Laboratório de Bioinformática, Programa de Pós-graduação em Engenharia e Informática,
Universidade Técnológica Federal do Paraná – UTFPR
Av. 7 de setembro, 3165 80230-901 Curitiba, Paraná, BRASIL
E-mails: weinert@cpgei.cefetpr.br, hslopes@pesquisador.cnpq.br

Abstract— Gene Expression Programming (GEP) is an evolutionary computation-based technique that uses concepts from both genetic algorithms and genetic programming. The basic difference of GEP with the others is the biologically-inspired concept of Open Read Frame, where the genotypical representation may not be fully transcribed into a phenotypical representation. This work proposes several computational strategies aimed at improving the basic GEP when applied to the task of symbolic regression. Four different problems are addressed and the performance is evaluated with different measures. Results demonstrate that the proposed strategies lead GEP to better performance when compared with the basic version.

Keywords— Evolutionary computation, Gene expression programming, Genetic programming, Symbolic regression.

Resumo — A Programação de Expressão Genética é uma técnica de computação evolucionária que une conceitos de Algoritmos Genéticos e de Programação Genética. Um diferencial desta técnica é o uso conceito biológico de *Open Read Frame*, onde a representação genotípica do problema nem sempre é totalmente transcrita em sua representação fenotípica. Este trabalho propõe diversas estratégias visando a melhoria do desempenho da versão original. Dentre estas propostas estão: manipulação de cromossomos de tamanhos diferentes, a utilização do torneio como operador de seleção, a manipulação de constantes aleatórias, novas funções de *fitness* e procedimentos de normalização. Quatro problemas de regressão simbólica são utilizados nos experimentos e avaliados por diferentes métricas. Os resultados demonstram que as estratégias propostas levam a um desempenho significativamente melhor sobre a técnica de programação de expressão genética originalmente proposta.

Palavras-chave— Computação evolucionária, Programação de expressão genética, Programação genética, Regressão simbólica.

1 Introdução

Técnicas de Computação Evolucionária (CE) têm sido empregadas em diversas áreas como: ciências naturais, engenharia, biologia e ciência da computação. A CE é descrita como um conjunto de técnicas e procedimentos genéricos e adaptáveis, a serem aplicados na solução de problemas complexos, para os quais outras técnicas conhecidas são pouco eficazes.

Dentre as diversas técnicas de CE pode-se destacar os Algoritmos Genéticos (AGs) (Goldberg, 1989) e a Programação Genética (PG) (Koza, 1992). Basicamente, todos os modelos de algoritmos evolucionários representam possíveis soluções para um problema sob a forma de indivíduos de uma população. Estes indivíduos evoluem de acordo com regras de seleção e operadores genéticos.

A Programação de Expressão Genética (PEG) (Ferreira, 2001) é um modelo de algoritmo evolucionário, semelhante a AGs e PG. A diferença fundamental entre os três algoritmos, AGs, PG e PEG, reside na natureza dos indivíduos. Nos AGs os indivíduos são cadeias lineares normalmente de tamanho fixo (cromossomos) e na PG os indivíduos são entidades não-lineares de diferentes tamanhos e formas (árvores). Na PEG os indivíduos são codificados em cadeias lineares de tamanho fixo (o genoma ou cromossomo) que são expressas posteriormente como entidades não-lineares com diferentes tamanhos e formas (representações esquemáticas simples ou árvores de expressão) (Ferreira, 2001).

Lopes e Weinert (2004) desenvolveram uma série de melhorias e adaptações no algoritmo da PEG originalmente definida por Ferreira (2001). O produto destas melhorias é uma ferramenta de software conhecida como EGIPSYS[1]. O EGIPSYS é uma ferramenta genérica para solução de problemas de regressão simbólica.

A regressão simbólica objetiva encontrar uma função de identificação que seja capaz de mapear de forma simbólica um conjunto finito de amostras de pontos. Na regressão simbólica conhece-se apenas um conjunto de entradas e saídas (variáveis independentes e dependentes, respectivamente), mas não se conhece nada a respeito do modelo. O objetivo é, então, encontrar uma função de transferência baseada nas variáveis independentes, que seja capaz de minimizar uma medida de erro entre os valores da variável dependente, calculadas pela função encontrada e o valor desejado.

Este trabalho introduz e avalia cinco estratégias introduzidas na PEG original: utilização de cromossomos de tamanho variável, método de seleção por torneio, uma função de escalonamento linear para os valores de *fitness* dos indivíduos, uma função de *fitness* normalizada baseada no erro absoluto e a utilização probabilística de constantes no conjunto de terminais.

[1] *http://bioinfo.cpgei.cefetpr.br/br/softwares/Install_EGIPSYS.zip*

Quatro conjuntos de dados são utilizados nos experimentos que objetivam mensurar a contribuição de cada uma das melhorias propostas em relação à PEG original.

A sessão 2 apresenta a fundamentação teórica referente a PEG e às melhorias que são objetivo de estudo deste trabalho. A sessão 3 descreve a metodologia utilizada para realização dos experimentos. A sessão 4 apresenta os resultados obtidos e a última sessão destina-se a discussão e conclusões.

2 Fundamentação Teórica

A Figura 1 apresenta o sistema de codificação adotado na PEG, onde cada indivíduo é composto normalmente de um único cromossomo. Este cromossomo é representado por uma estrutura linear de tamanho fixo.

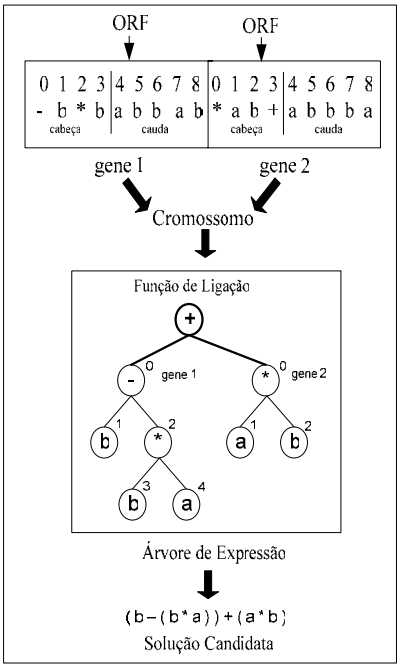

Figura 1: Sistema de codificação da PEG.

O cromossomo é divido em genes e os genes são divididos em cabeça e cauda. Um conjunto de funções e terminais especificado para o problema fornece o material genético para o preenchimento dos genes. A cabeça de um gene pode receber tanto elementos presentes no conjunto de terminais quanto elementos presentes no conjunto de funções, enquanto que a cauda somente pode manipular os elementos especificados dentro do conjunto de terminais.

A PEG utiliza o conceito de ORF (*Open Read Frame*) no processo de codificação. Segundo este conceito biológico nem todo material genético disponível é utilizado para codificar um gene. Assim, a ORF, ou região codificadora de um gene, pode ativar e desativar material genético durante o processo evolutivo.

Todo cromossomo é transcrito para uma árvore de expressão de tamanho variável. O material genético utilizado neste processo de transição é o material ativo presente em cada gene, como mostrado na Figura 1. As regras de transcrição seguem os padrões expressos pela linguagem Karva (Ferreira, 2001). Cada gene é transcrito para uma sub-árvore. Em seguida, une-se as sub-árvores através de uma função de ligação gerando, assim, a árvore de expressão. A árvore de expressão representa a solução candidata para o problema em questão e da avaliação desta árvore obtém-se o *fitness* do indivíduo.

A PEG implementa diversos operadores genéticos: clonagem, replicação, mutação, transposição e recombinação. Todos eles atuam sobre a estrutura linear do cromossomo, sempre gerando estruturas válidas. Maiores detalhes a respeito da codificação destes operadores e a função de *fitness* padrão podem ser encontrados em Ferreira (2001).

O método usual de seleção utilizado na PEG é o da roleta (Goldberg, 1989). As próximas sessões descrevem as melhorias implementas por Lopes e Weinert (2004) no algoritmo da PEG e que serão avaliadas neste trabalho.

2.1 Cromossomos de Tamanho Variável

Segundo o trabalho de Ferreira (2001), um dos problemas da PEG é determinar o tamanho da cabeça de cada gene dentro do cromossomo. Normalmente problemas mais complexos exigem tamanhos de cabeças maiores. Então, o que se faz na maioria das vezes é executar o algoritmo da PEG várias vezes, com tamanhos de cabeças de genes diferenciados, até que se encontre um tamanho que seja capaz de representar o problema, na tentativa de alcançar uma solução aceitável de forma satisfatória.

Para contornar esta limitação, adicionou-se um recurso que permite a inclusão de cromossomos de diferentes tamanhos numa mesma população. Para tanto, o usuário determina um intervalo apropriado para o tamanho da cabeça dos genes que constituem o cromossomo, lembrando que todos os genes de um cromossomo compartilham a mesma configuração, ou seja, tamanhos de cabeça e cauda iguais.

Definiu-se, então, que metade da população é gerada de maneira uniforme, respeitando uma proporcionalidade parametrizada por um intervalo de tamanho de cabeças definido pelo usuário. A outra metade é gerada aleatoriamente dentro deste mesmo intervalo. Esta abordagem foi inspirada na técnica *ramped-half-and-half* (Koza, 1992) que guia a criação das árvores na PG e é baseada num parâmetro de profundidade máxima. Todos os operadores genéticos foram adaptados de modo a gerarem sempre indivíduos válidos.

2.2 Método de Seleção por Torneio

A estratégia de seleção por torneio estocástico (Banzhaf, Nordin, Keller *et al*, 1998) para a escolha

dos indivíduos é guiada por um parâmetro que indica a quantidade de indivíduos que serão escolhidos ao acaso dentro da população. Em seguida retorna o melhor deles, neste caso, o de maior *fitness*. Esta abordagem é menos elitista do que a estratégia de seleção por roleta (Goldberg, 1989), uma vez que dá mais oportunidade de seleção aos indivíduos da população sem perder a característica evolucionária.

2.3 Escalonamento Linear

Muitas vezes ocorrem grandes disparidades entre os valores de *fitness* dos indivíduos dentro de uma população. Isto pode prejudicar o processo evolutivo de busca da solução para um determinado problema, pois a pressão seletiva causada pela aplicação dos operadores de seleção baseados no valor de *fitness* dos indivíduos acaba normalmente selecionando os mesmos indivíduos. Como conseqüência, o algoritmo converge prematuramente, ou seja, toda a população fica praticamente igual, sem atingir uma solução satisfatória e estagnando a evolução.

Lopes e Weinert (2004) apresentam a implementação de um procedimento de escalonamento linear para os valores de *fitness*, conforme proposto por Goldberg (1989) para AGs. Este escalonamento preserva o valor do *fitness* médio da população após sua aplicação. A intenção deste escalonamento é aproximar os indivíduos fazendo com que estes tenham mais chances de se manterem dentro da população no decorrer do processo evolutivo, evitando uma possível convergência prematura da população.

2.4 Função de Fitness

A função de *fitness* quantifica o grau de aptidão de um indivíduo. Lopes e Weinert (2004) propuseram utilizar o valor de *fitness* normalizado entre 0 e 1, onde 0 representa o pior indivíduo e 1 o melhor.

Para problemas de regressão simbólica trabalha-se com o conceito de erro absoluto onde o *fitness* é dado pela equação 1.

$$fitness(i,t) = \left(\frac{valor_referência}{(valor_referência + multiplicador * \sum_{j=1}^{Ne} |S(i,j) - C(j)|} \right) \quad (1)$$

onde:

fitness(i,t): *fitness* do indivíduo *i* na geração *t*.
valor_referência: determina a inclinação das curvas de distribuição.
multiplicador: fator escolhido pelo usuário.
Ne: número de casos de *fitness*.
S(i,j): valor retornado pela função de transferência *i* para o caso de *fitness j*.
C(j): valor correto para o caso de *fitness j*.

O *multiplicador* e o *valor_referência* têm um papel importante dentro das equações. Eles são responsáveis pelo maior ou menor achatamento das curvas que os valores de *fitness* representam, ou seja, servem para ampliar ou comprimir as diferenças de *fitness* entre os indivíduos. Não existe um valor ideal estas variáveis, pois dependem do valor do somatório dos erros e da distinção que se deseja obter entre o *fitness* do melhor e do pior indivíduo dentro da população.

2.5 Manipulação de Constantes

Adota-se uma política de utilização de constantes onde o usuário determina uma taxa percentual de probabilidade para a ocorrência de constantes dentro das soluções candidatas de um determinado problema, sem que seja necessário aumentar o tamanho do cromossomo e criar um vetor de constantes para cada gene, como originalmente proposto por Ferreira (2001), o que torna o processo mais natural e de fácil manipulação.

Vale lembrar que uma constante só pode ocorrer em posições específicas dentro de cada gene, mais especificamente, onde poderiam ser inseridos os terminais. O usuário também determina um intervalo de valores que as constantes devem respeitar, pelo menos no momento de sua criação, uma vez que o operador de mutação pode realizar alterações nestes valores fazendo com que o intervalo original de valores possa ser extrapolado. Este intervalo de valores sempre varia de um número negativo até este mesmo número positivo. Por padrão adota-se o intervalo de –10 a 10. Também é permitida a inserção de certas constantes previamente definidas, como π (3,141592), *e* (número de Euler – 2,718281) ,a constante 1,0 ou qualquer outra desejada pela usuário. O símbolo utilizado para indicar a presença de uma constante na solução candidata é "?" seguido da referida constante, conforme a notação proposta por Ferreira (2001). Um operador de busca local realiza ajustes finos nos valores das constantes (Lopes e Weinert, 2004).

3 Metodologia

Para avaliar as estratégias de melhoria propostas para o algoritmo da PEG utilizou-se quatro bases de dados para problemas de regressão simbólica. Foram analisadas quatro medidas de desempenho disponibilizadas pela ferramenta EGIPSYS.

3.1Problemas Abordados

3.1.1 Função Quadrática com Ruído

Este trata-se de um problema fictício com 201 casos de *fitness* e apenas uma variável independente *x*, e uma variável dependente *y*, conforme a Equação 2. Os valores de *x* são extraídos do seno de valores ordenados pertencentes ao intervalo de [–10..10] com um passo de 0,1. Para cada saída da função foi somado um ruído aleatório (variável *a*) no intervalo real [0..1]. A escolha desta função matemática em particular foi arbitrária, sendo usual na literatura tal

abordagem. Entretanto, a adição de ruído visa tornar o problema menos trivial para o algoritmo.

$$y = 2x^2 - 3x + 4 + \left(\left(\frac{a}{5}\right) - 0{,}1\right) \qquad (2)$$

3.1.2 Lago Eire

Este problema trata de dados de simulação retirados da base de dados de identificação de sistemas Daisy[2].

Estes dados foram apresentados pela primeira vez em Guidorzi, Losito e Muratori (1980). Esta base tem 57 amostras de um conjunto de 5 parâmetros de entrada e 2 de saída. Neste caso, a função de transferência entre os dados de entrada e os de saída é desconhecida. Os parâmetros de entrada são: x_1: temperatura da água; x_2: condutividade da água; x_3: alcalinidade da água; x_4: concentração de NO3; x_5: dureza (*total hardness*). Os parâmetros de saída são: y_1: dissolução de oxigênio; y_2: concentração de algas.

Estes dados estão separados em quatro conjuntos: sem ruído e com ruído de 10%, 20% e 30%. Neste trabalho utilizou-se apenas os dados sem ruído e os com ruído de 30% para a saída y_1.

3.1.3 Manchas Solares

Este problema busca encontrar uma função de transferência que utiliza como variáveis dependentes *n* entradas anteriores da série, gerando a partir delas, a saída atual. Este problema de séries temporais foi reportado em Weigened, Huberman e Rumelhart (1992) e Ferreira (2002). Ele apresenta uma série de 100 dados a respeito do número observado de manchas solares. Estes dados são então reformulados de maneira que a combinação das 10 primeiras amostras gera a 11ª e assim sucessivamente, ou seja, a base foi formatada de 90 casos de *fitness* com 10 entradas e uma saída, onde os valores de saída se encontram num intervalo de 0 a 154.

3.2 Medidas de Desempenho Analisadas

- Tempo de Processamento (p_{tempo}): Esta medida ajuda a mensurar a complexidade computacional do algoritmo, uma vez que o aumento de complexidade está diretamente ligado ao aumento de instruções a serem processadas que, por sua vez, exigem maior tempo computacional. O tempo é mensurado em segundos.
- Coeficiente de Correlação (r^2): quantifica a semelhança existente entre o conjunto de pontos dados e o encontrado. Este coeficiente sempre varia de –1 a +1. Uma relação diretamente proporcional exata ocorre quando o coeficiente é igual a +1, ao passo que uma relação inversa exata ocorre quando o coeficiente é igual a –1. Quando o coeficiente é igual a 0, significa que não existe relação linear entre os pontos.
- Geração: A geração na qual se obteve o melhor resultado serve como indicativo de esforço computacional necessário à obtenção da resposta.
- Número de Nós: O número de nós da árvore indica o grau de complexidade da função obtida. Quanto maior o número de nós, mais complexa é a solução obtida, sendo difícil a sua compreensão ou mesmo simplificação.

3.3 Experimentos

Sete estratégias de melhoria foram avaliadas utilizando-se os quatro problemas citados. A sigla FQR refere-se ao problema simulado pela função quadrática com ruído, a sigla LESR refere-se ao problema do Lago Eire para saída y_1 sem ruído e a sigla LESR para saída y_1 com ruído de 30% e a sigla MS representa o problema de manchas solares. O objetivo destes experimentos é avaliar o impacto de cada uma das estratégias segundo as medidas anteriormente citadas. As estratégias denominadas A, B, C, D, E, F e T representam uma determinada configuração utilizada para executar a PEG no EGIPSYS. A Tabela 1 apresenta a descrição destas estratégias, sendo que a primeiro (A) apresenta a configuração padrão proposta por (Ferreira, 2001), os intermediários agem utilizando os parâmetros padrões com apenas uma modificação proposta na metodologia e o último aplica todas as modificações em conjunto. Um total de 100 rodadas foram realizadas sobre cada experimento e reportadas as soluções médias do tempo de processamento, do coeficiente de correlação, da geração em que a melhor solução foi obtida e do número de nós da árvore.

Tabela 1: Estratégias de melhoria propostas para a PEG padrão.

Estratégia	Configuração
A	PEG original
B	cromossomos de tamanho variável (cabeças: de 2 a 10)
C	seleção por torneio
D	escalonamento linear
E	*fitness* com erro absoluto, *mult*=0,1 e *ref_val*=10
F	constantes aleatórias com *p*=0,2
T	todas as opções anteriores

4 Resultados

Todos os resultados estão reportados em tabelas, sendo que os valores grafados em vermelho significam uma melhora em relação ao valor de comparação e os valores grafados em azul representam uma diminuição do desempenho.

[2] *http://www.esat.kuleuven.ac.be/~tokka/daisydata.html*

A Tabela 2 apresenta os resultados para os testes A e B: a PEG original e a PEG utilizando cromossomos de tamanho variável.

Tabela 2: Comparação das estratégias A e B.

Problema	Estratégia	p_{tempo}	r^2	Geração	N°
FQR	A	133,57	0,93	33,92	26,12
	B	**129,35**	**0,89**	**34,64**	**25,50**
LESR	A	36,26	0,33	40,90	23,89
	B	**32,41**	**0,35**	**39,65**	**20,40**
LECR	A	34,90	0,19	41,93	21,71
	B	**32,40**	**0,22**	**39,99**	**19,64**
SP	A	56,19	0,82	44,23	22,24
	B	**48,21**	**0,84**	**42,38**	**19,06**

A Tabela 3 apresenta os resultados para os testes A e C: PEG original e a PEG utilizando o torneio como método de seleção.

Tabela 3: Comparação das estratégias A e C.

Problema	Estratégia	p_{tempo}	r^2	Geração	N°
FQR	A	133,57	0,93	33,92	26,12
	C	**159,37**	**0,97**	**30,78**	**29,22**
LESR	A	36,26	0,33	40,90	23,89
	C	**38,90**	**0,51**	**42,55**	**24,26**
LECR	A	34,90	0,19	41,93	21,71
	C	**38,88**	**0,29**	**41,80**	**24,96**
SP	A	56,19	0,82	44,23	22,24
	C	**61,23**	**0,82**	**44,78**	**24,18**

A Tabela 4 apresenta os resultados para os testes A e D: a PEG original e a PEG utilizando escalonamento linear para diminuição da pressão seletiva.

Tabela 4: Comparação das estratégias A.e D.

Problema	Estratégia	p_{tempo}	r^2	Geração	N°
FQR	A	133,57	0,93	33,92	26,12
	D	**139,83**	**0,96**	**36,56**	**27,98**
LESR	A	36,26	0,33	40,90	23,89
	D	**36,13**	**0,37**	**41,39**	**22,48**
LECR	A	34,90	0,19	41,93	21,71
	D	**35,79**	**0,21**	**42,79**	**21,73**
SP	A	56,19	0,82	44,23	22,24
	D	**56,43**	**0,81**	**43,48**	**22,78**

A Tabela 5 apresenta os resultados para os testes A e E: a PEG original e a PEG onde a função de *fitness* se baseia no erro absoluto normalizado.

Tabela 5: Comparação das estratégias A e E.

Problema	Estratégia	p_{tempo}	r^2	Geração	N°
FQR	A	133,57	0,93	33,92	26,12
	E	**145,99**	**0,96**	**38,36**	**28,62**
LESR	A	36,26	0,33	40,90	23,89
	E	**35,83**	**0,49**	**42,87**	**23,50**
LECR	A	34,90	0,19	41,93	21,71
	E	**35,75**	**0,33**	**42,76**	**22,72**
SP	A	56,19	0,82	44,23	22,24
	E	**57,99**	**0,83**	**44,42**	**22,66**

A Tabela 6 apresenta os resultados para os testes A e E: a PEG original e a PEG com manipulação de constantes aleatórias.

Tabela 6: Comparação das estratégias A e F.

Problema	Estratégia	p_{tempo}	r^2	Geração	N°
FQR	A	133,57	0,93	33,92	26,12
	F	**134,40**	**0,97**	**35,09**	**24,44**
LESR	A	36,26	0,33	40,90	23,89
	F	**36,15**	**0,58**	**42,76**	**22,44**
LECR	A	34,90	0,19	41,93	21,71
	F	**35,79**	**0,52**	**42,26**	**21,70**
SP	A	56,19	0,82	44,23	22,24
	F	**55,28**	**0,83**	**44,90**	**20,02**

A Tabela 7 apresenta os resultados para os testes A e T: a PEG original e a PEG com todas as estratégias de melhoria implementadas.

Tabela 7: Comparação das estratégias A e T.

Problema	Estratégia	p_{tempo}	r^2	Geração	N°
FQR	A	133,57	0,93	33,92	26,12
	T	**145,10**	**0,97**	**36,27**	**25,46**
LESR	A	36,26	0,33	40,90	23,89
	T	**32,22**	**0,67**	**42,87**	**19,40**
LECR	A	34,90	0,19	41,93	21,71
	T	**37,19**	**0,54**	**44,53**	**22,60**
SP	A	56,19	0,82	44,23	**22,24**
	T	**50,50**	**0,83**	**46,43**	**18,44**

5 Discussão e Conclusões

O uso de cromossomos de tamanho variável na população inicial (estratégia B) diminui o tempo total de processamento (p_{tempo}), pois o algoritmo tem de processar cromossomos menores (em média) do que no caso da população criada com cromossomos de tamanho fixo. Isto também reflete no tamanho médio das soluções obtidas, que tende a ser menor. O uso deste artifício também causa uma melhora de desempenho em relação ao método básico.

Para todos os casos, o tempo de processamento quando se utiliza o torneio estocástico (estratégia C) como método de seleção é maior do que utilizando o método tradicional da roleta. Isto já era esperado devido ao fato de que o primeiro método de seleção é computacionalmente mais custoso do que o segundo. Porém, utilizando este método de seleção obtém-se melhores resultados (mostrados por r^2) e com esforço computacional e complexidade de soluções equivalentes ao método original. A grande utilidade deste método é o controle da pressão seletiva ao longo das gerações, evitando a convergência prematura para máximos locais. Isto é particularmente importante em problemas difíceis (p.ex. LECR) onde o uso deste artifício, por si só, já leva a bons resultados de desempenho.

O uso do escalonamento linear de *fitness* (estratégia D) não influencia significativamente os resultados obtidos, principalmente para problemas

mais fáceis (LESR). Porém, para problemas mais difíceis (LECR) onde, ao longo das gerações ocorre pouca discrepância entre os indivíduos, este artifício parece ser proveitoso. O efeito final deste artifício é semelhante àquele da seleção por torneio estocástico, especialmente para problemas mais difíceis. O r^2 teve uma redução insignificante para o problema de manchas solares.

A função de *fitness* baseada no erro absoluto (estratégia E) é definitivamente superior àquela originalmente proposta, pois consistentemente obtém resultados melhores em todos os problemas.

O uso de constantes é imperativo em problemas de regressão simbólica, visto que parece ser natural o seu uso em uma equação que descreva o comportamento de um sistema físico. É muito improvável que a função obtida não se beneficie do uso de constantes. Isto é demonstrado em todos os experimentos onde, dentre todos os artifícios testados separadamente, o uso de constantes junto com o operador especial de mutação (estratégia F) foi o que melhor desempenho mostrou. Além disto, tal desempenho foi obtido sem sobrecarga computacional, já que p_{tempo} e *geração* foram aproximadamente equivalentes à PEG original.

Finalmente, o uso conjunto de todos os artifícios (estratégia T) demonstrou claramente a sua superioridade, obtendo sempre os melhores resultados (considerando r^2) para todos os casos de teste (exceto para MS, mas por uma diferença mínima).

Os experimentos deixam claro que as estratégias de melhoria propostas para PEG realmente proporcionam soluções melhores. Entretanto, também foi observado o aumento do tempo de processamento e um pequeno aumento no número de gerações necessárias para se obter a melhor solução. Outro ponto favorável é que o número de nós manteve-se praticamente estável indicando que não houve aumento da complexidade das soluções.

Referências Bibliográficas

Banzhaf, W., Nordin, P., Keller, R.E. e Francone, F.D. (1998). *Genetic Programming: An Introduction***.** San Francisco: Morgan Kaufmann.

Ferreira, C. (2001) Gene expression programming: a new adaptive algorithm for solving problems. *Complex Systems* **13**(2):87-129.

Ferreira, C. (2002) Function finding and a creation of numerical constants in gene expression programming. In: *7th Online World Conference on Soft Computing in Industrial Applications.*

Goldberg D.E. (1989). Genetic Algorithms in Search, Optimization and Machine Learning. Reading, USA: Addison-Wesley.

Guidorzi, R.P., Losito, M.P. e Muratori, T. (1980). On the least eigenvalue test in the structural identification of linear multivariable systems. In: *Proc. of the V European Meeting on Cybernetics and Systems Research*, Vienna, [s.p.].

Koza, J.R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press.

Lopes, H.S., Weinert, W.R. (2004). EGIPSYS: a enhanced gene expression programming approach for symbolic regression problems. *International Journal of Applied Mathematics and Computer Science* **14**(3):375-384.

Weigened, A.S., Huberman, B.S. e Rumelhart D.E. (1992). Predicting sunspots and exchange rates with connectionist networks. In: Eubank, S. e Casdagli, M. (eds.) *Nonlinear Modeling and Forecasting*, Redwood City (CA): Addison-Wesley, 395-432.