

SOLUÇÃO ANALÍTICA DE EQUAÇÕES DIFERENCIAIS POR ALGORITMOS EVOLUCIONÁRIOS DE INTELIGÊNCIA ARTIFICIAL

Waldir J. A. Lobão¹

Marco A. C. Pacheco²

INTRODUÇÃO

É de amplo conhecimento que um número expressivo de projetos de pesquisa realizados por diversas áreas do conhecimento científico utiliza-se de modelos matemáticos que são formulados, parcialmente ou integralmente, por equações diferenciais. No entanto, dada a complexidade dos modelos propostos, os seus formuladores quase sempre se deparam com problemas de difícil solução e com solução analítica desconhecida. Nesta situação, o procedimento usual é a obtenção de uma solução através de métodos numéricos. Contudo, grande parte destes projetos objetivam resultados maiores e usam as equações diferenciais para tentar descrever o comportamento dinâmico de variáveis que são objetos centrais destas pesquisas. Nestes casos, certamente, a solução numérica não é suficientemente completa para dar aos pesquisadores as respostas requeridas para as suas avaliações. Faz-se necessário o conhecimento da solução na sua forma literal, pois esta permite a realização de importantes e diferentes tipos de análises, tais como: estáticas comparativas; conhecimento da magnitude de efeitos parciais e elasticidades; estudos de estabilidade e estacionariedade; etc.

Com a motivação de tentar contribuir com novos métodos que ajudem a dirimir os problemas acima citados, o objetivo principal deste estudo é investigar o potencial de algoritmos computacionais, construídos a partir da combinação das técnicas de diferenciação automática e programação genética (DAPG), na obtenção de soluções analíticas para complexos problemas de equações diferenciais ordinárias (EDOs) e parciais (EDPs). Com essa finalidade, e utilizando o ambiente de programação Matlab, diversos algoritmos foram elaborados e soluções analíticas para diferentes problemas de EDOs e EDPs foram determinadas. Os resultados do trabalho são promissores, com soluções exatas obtidas para a grande maioria dos problemas abordados e que, empiricamente, atestam a consistência e robustez da metodologia desenvolvida.

Para comparar os estudos realizados aos de mesmo gênero existentes na literatura, uma revisão bibliográfica foi realizada e um número significativo de trabalhos foi examinado. Verificou-se que, embora exista uma vasta literatura sobre diferenciação automática (DA) e programação genética (PG), a grande maioria dos trabalhos aborda e aplica estas técnicas de forma separada com outros objetivos. Além disso, dentre os que desenvolvem DA e PG, conjuntamente, são raros aqueles que apresentam estudos sobre a solução de equações diferenciais. Destes, destacam-se os artigos de Burgess (1999), Cao et al. (2000), Iba e Sakamoto (2002), Imae et al. (2004) e Tsoulos e Lagaris (2006). Estes trabalhos serviram de base comparativa na avaliação do desempenho computacional dos algoritmos desenvolvidos e apresentados neste artigo. As comparações foram

¹ Escola Naval – Marinha do Brasil e Fundação IBGE – ENCE. Doutor em Engenharia Elétrica – PUC – Rio.

² Pontifícia Universidade Católica – PUC – Rio. Doutor em Computer Science pela University College of London, UL, Inglaterra.

de fundamental importância, pois possibilitaram perseguir melhores resultados e desenvolver algoritmos mais eficientes e eficazes.

Além desta breve introdução, o artigo apresenta-se organizado em outras três seções. A segunda descreve os aspectos metodológicos do algoritmo de DAPG. A terceira seção apresenta os resultados de duas aplicações de soluções de equações diferenciais por DAPG e na quarta seção são apresentadas as conclusões do trabalho.

ASPECTOS METODOLÓGICOS DO ALGORITMO DE DAPG

Nesta seção são apresentados os procedimentos metodológicos adotados na elaboração do algoritmo DAPG, desenvolvido para solucionar problemas de equações diferenciais. Em virtude do espaço exíguo, os fundamentos teóricos sobre diferenciação automática e programação genética não serão aqui apresentados, a seção limita-se a descrever os aspectos metodológicos relacionados à estrutura e as etapas de funcionamento do algoritmo. No entanto, para os leitores menos familiarizados com estas técnicas, sugere-se a leitura dos seguintes textos: Rall (1981), Koza (1992), Fink (2007) e Silva (2009).

A estrutura e etapas de funcionamento do algoritmo de DAPG

O algoritmo foi desenvolvido em ambiente de programação Matlab e funciona com dois programas básicos que se completam e trabalham simultaneamente. O primeiro é responsável pela execução da programação genética (PG) e o segundo executa a diferenciação automática (DA) e avalia a aptidão (*fitness*) de cada indivíduo gerado pelo primeiro. Os programas funcionam concomitantemente ao longo de muitas repetições, denominadas de gerações, e finalizam as suas tarefas de acordo com um critério de parada previamente definido. A estrutura de representação da PG é a tradicional em forma de árvore, sendo escolhida por permitir rápida leitura e direta interpretação dos resultados. A parametrização da PG é flexível e possibilita intervir na definição do conjunto de parâmetros que se deseja utilizar na execução do programa. Algumas das possibilidades de parametrização são: (i) Tipos e formas de árvores; (ii) Profundidade mínima da árvore inicial; (iii) Profundidade máxima da árvore (*depth*); (iv) Controle do número de nós (*nodes*); (v) Seleção do conjunto de funções e terminais; (vi) Percentual ou número de indivíduos; (vii) Métodos de seleção para reprodução; (viii) Taxas de crossover e mutação; (ix) Tamanho da população; (x) Diferentes critérios de parada.

Um componente essencial para o funcionamento do programa de PG são os conjuntos de funções e terminais, pois estes são necessários para a criação e reprodução da população de indivíduos e, conseqüentemente, implicam diretamente na geração da solução ótima para o problema. Juntos, funções e terminais definem o tamanho da árvore de representação. O conjunto de funções é composto de operadores e funções matemáticas elementares. O conjunto de terminais é constituído de constantes (números reais, complexos, número p etc.) e das variáveis que compõem a equação diferencial.

A primeira etapa que o programa de PG realiza é a criação da população inicial. Funções e terminais são selecionados aleatoriamente e combinados, dando origem aos indivíduos que compõem a população. No caso específico deste trabalho, cada indivíduo criado é uma função matemática e, portanto, uma possível solução da equação diferencial de interesse. Para a avaliação dos indivíduos recém-criados, é necessário definir uma medida de erro (*fitness*) que avalie a qualidade e aptidão de cada indivíduo. Em virtude de o objetivo ser a solução de equações diferenciais, surge, nesse ponto, a necessidade de um programa de diferenciação automática para, via cálculo diferencial, verificar a qualidade da solução proposta. Para ilustrar como esta tarefa é realizada pelo algoritmo, um exemplo simples é apresentado a seguir.

Suponha que se queira obter a solução do seguinte problema de EDO:

$$y'(x) + 2y(x) = \exp(-2x), \quad \text{com } x \in \mathbb{R} \text{ e} \quad \text{inicial } y(0) = 3. \quad (1)$$

Suponha, também, que a função abaixo seja um indivíduo ou uma solução proposta pela PG.

$$f(x) = 2 + \exp(-2x) \quad (2)$$

Então, logo após a função ser gerada pela PG, a sua qualidade é avaliada. Ou seja, o segundo programa executa diferenciação automática e calcula a derivada de $f(x)$ em cada ponto do domínio definido pelo algoritmo (geralmente uma grade com 50 ou mais pontos de x). Em seguida, gera uma medida de *fitness* que compara os resultados obtidos com o verdadeiro modelo de EDO. Então, de acordo com o exemplo, segue que:

$$f'(x) = -2 \exp(-2x) \quad (3)$$

Aplicando este resultado na EDO, tem-se:

$$f'(x) + 2f(x) - \exp(-2x) = -2 \exp(-2x) + 2[2 + \exp(-2x)] - \exp(-2x) = 4 - \exp(-2x) \neq 0. \quad (4)$$

Observe que $f(x)$ apresenta erro diferente de zero e, necessariamente, não é uma solução exata da EDO de interesse, embora satisfaça a condição inicial, pois

$$f(0) = 2 + \exp(-2 \times 0) = 2 + 1 = 3. \quad (5)$$

Contudo, o algoritmo não trabalha com a diferenciação simbólica da solução proposta, como feito acima, e sim com a derivada avaliada em cada ponto do domínio. Diante disso, a medida de *fitness* escolhida para avaliar a qualidade das soluções propostas pela PG é o, comumente utilizado, erro absoluto médio, acrescido de uma penalidade para o erro da condição inicial ou de outras restrições exigidas pelo problema de interesse. Isto é,

$$fitness = \text{erro absoluto médio} + \text{erro de restrições}. \quad (6)$$

$$fitness = \left\{ \frac{1}{n} \sum_{i=1}^n |f'(x_i) + 2f(x_i) - \exp(-2x_i)| \right\} + |f(0) - 3|. \quad (7)$$

Após a avaliação dos indivíduos, feita de acordo com os seus *fitness*, parte daqueles que receberam as melhores avaliações são selecionados para etapa de reprodução e geração da próxima população. Os demais são automaticamente descartados e excluídos do processo. Esta sequência de criação, avaliação, seleção e reprodução se repete ao longo de diversas gerações até que a melhor solução seja encontrada. Portanto, o que o algoritmo realiza ao longo de diversas gerações é tentar criar por PG o indivíduo ou a solução $f(x)$ que minimize o *fitness*. Deve-se ressaltar que, a busca pela solução ótima ocorre de forma evolutiva e a velocidade de convergência depende de algumas importantes etapas, métodos e operadores genéticos que não serão aqui apresentados.

Outro aspecto importante da metodologia são os critérios que definem e elegem a solução analítica de uma equação diferencial pelo algoritmo de DAPG. De acordo com o critério de parada utilizado, o algoritmo funciona até que a última geração seja avaliada e, ao término das gerações, seleciona a melhor solução encontrada, ou seja, aquela que minimiza a medida de erro (*fitness*). A partir deste ponto, a melhor solução encontrada é validada como uma solução final de PGDA se satisfaz os seguintes requisitos: (i) se apresenta baixíssima medida de *fitness*, com valores abaixo de 0,00001; (ii) se satisfaz as condições iniciais e condições de contorno do problema proposto; (iii) se, quando diferenciada pelas regras do cálculo usual, se iguala à equação diferencial proposta em todo o domínio do problema.

Caso uma solução validada apresente *fitness* nulo e satisfaça os demais requisitos com perfeita exatidão, a mesma é considerada como uma *solução analítica exata* do problema proposto, pois esta solução é idêntica à obtida quando se aplica o cálculo diferencial usual, tanto para EDOs quanto para EDPs. Caso contrário, a solução validada é considerada como uma solução analítica aproximada.

EXEMPLOS DE SOLUÇÕES DE EQUAÇÕES DIFERENCIAIS POR DAPG

Nesta seção são apresentados os resultados de duas aplicações, uma para EDO e outra para EDP, que ilustram e ajudam a entender como é obtida a solução de equações diferenciais por DAPG. Contudo, deve-se informar que

mais de 30 problemas foram tratados com o objetivo de testar a eficiência e eficácia do algoritmo, no sentido da obtenção de soluções analíticas exatas.

De forma geral, para as diversas aplicações realizadas, o algoritmo foi parametrizado de acordo com os parâmetros indicados na Tabela 1.

Tabela 1. Parametrização do Algoritmo de DAPG

Parâmetros Básicos do DAPG		Parâmetros Básicos do DAPG	
Gerações	20 a 100	Inicialização	Growinit
População (indivíduos)	30 a 600	Seleção e Reprodução	Lexictour
Funções Elementares	8 a 15	Elitismo	totalelitism
Terminais	5 a 15	Critério de Parada	Nº de Gerações
Variáveis	2 a 6	Taxa de Crossover	Variável
Modo de DA (fixo)	<i>forward</i>	Taxa de Mutação	Variável

O número de gerações variou de 20 a 100 e o tamanho da população de 30 a 600, dependendo do número de variáveis, da ordem de diferenciação, e do tamanho e complexidade da equação diferencial. O número de funções e operações matemáticas elementares variou de 8 a 15, e o número de terminais, variáveis mais constantes, variou de 5 a 15. A inicialização da população, a seleção para reprodução e a técnica de elitismo, para a maioria das aplicações, foram utilizados, respectivamente, os métodos *growinit*, *lexictour*, de acordo com a metodologia proposta por Luke e Panait (2022), e *totalelitism*. As taxas dos operadores genéticos de *crossover* e mutação são variáveis, iniciando com 50% para cada operador. O critério de parada utilizado foi o número de gerações. Foram utilizados conjuntos ou grades de 50 ou mais pontos do domínio da função para a realização das equações.

Exemplo 1: a primeira aplicação aborda a solução da seguinte EDO de 2ª ordem:

EDO:	$y''(x) + 0,3 y'(x) + 25 y(x) = 25,12 + 10x - 1,5 \cos(5x) \exp(-0,3x)$
-------------	---

com $x \in \mathbb{R}$; e condições iniciais $y(0) = 1$ e $y'(0) = 5,4$.

Solução exata obtida:	$y(x) = 1 + 0,4x + \text{sen}(5x) \exp(-0,3x)$
------------------------------	--

Os resultados são apresentados graficamente na Figura 1 – Representação em Árvore da Solução da EDO e na Figura 2 – Gráfico da Função Solução da EDO. Na árvore da Figura 1, a variável independente (x) é representada por ($x = x_1$).

A solução analítica da EDO foi obtida no indivíduo 6031 (*bestsofar*), para uma população de 400 indivíduos e 40 gerações. A solução é exata, conforme mostra *fitness* nulo na Figura 1. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 6 (*depth*) e 16 nós (*nodes*).

Exemplo 2: a segunda aplicação aborda a solução da seguinte EDP de 2ª ordem:

EDP:	$f_t - f_{xx} = [x - t(4x^3 - 6x)] e^{-x^2}$
-------------	--

com $0 \leq x \leq 1$; $0 \leq t \leq 1$; onde $f_{xx} = \frac{\partial^2 f(x, t)}{\partial x^2}$; $f_t = \frac{\partial f(x, t)}{\partial t}$; e

condições $f(0, t) = 0$; $f(x, 0) = 0$; e $f_x(1, t) + f(1, t) = 0$.

Solução exata obtida:	$f(x, t) = t x e^{-x^2}$
------------------------------	--------------------------

Os resultados são apresentados na Figura 3 – Representação em Árvore da Solução da EDP e na Figura 4 – Gráfico da Função Solução da EDP. Na árvore da Figura 3, as variáveis independentes x e t são representadas, respectivamente, pelas variáveis ($x = x_1$) e ($t = x_2$).

A solução da EDP foi obtida no indivíduo 1336 (*bestsofar*), para uma população de 50 indivíduos e 50 gerações. A solução analítica encontrada é exata, conforme mostra *fitness* nulo na Figura 1. A árvore de representação obtida é a mais parcimoniosa possível, com profundidade 5 (*depth*) e 9 nós (*nodes*).

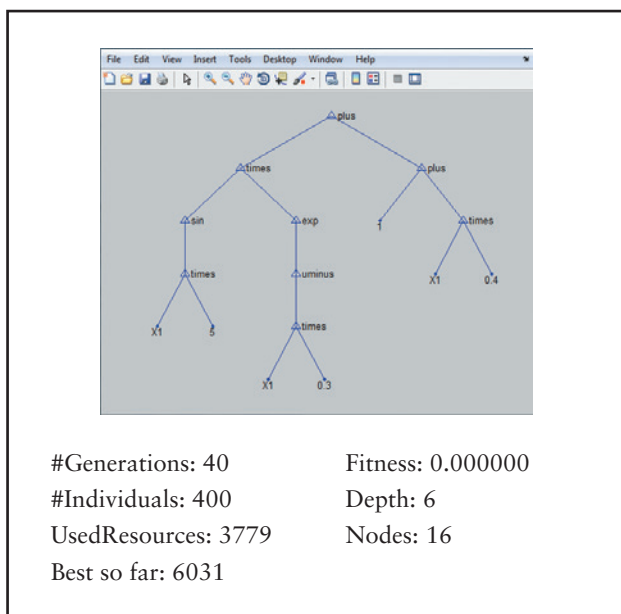


Figura 1. Representação em Árvore da Solução da EDO

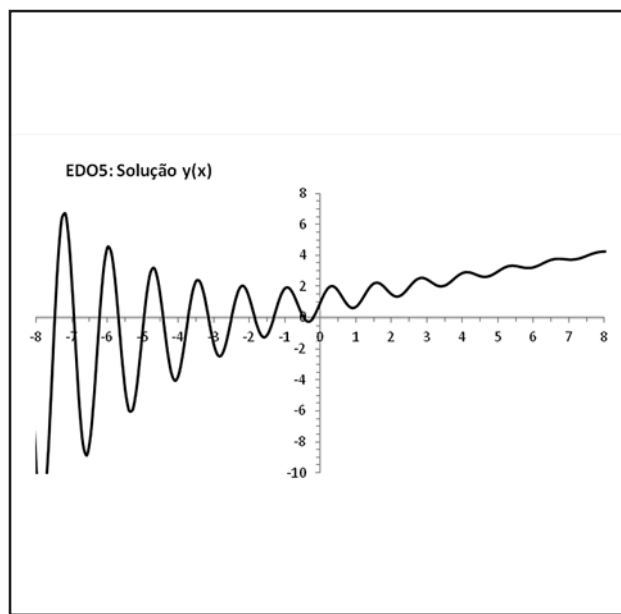


Figura 2. Gráfico da Função Solução da EDO

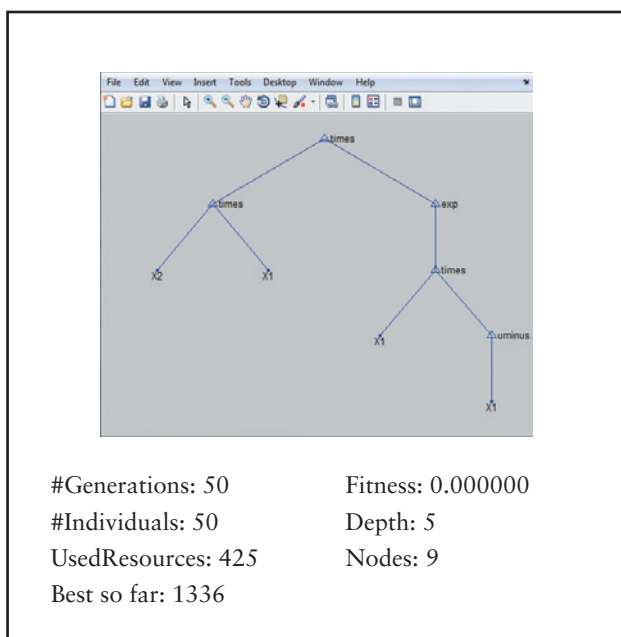


Figura 3. Representação em Árvore da Solução da EDP

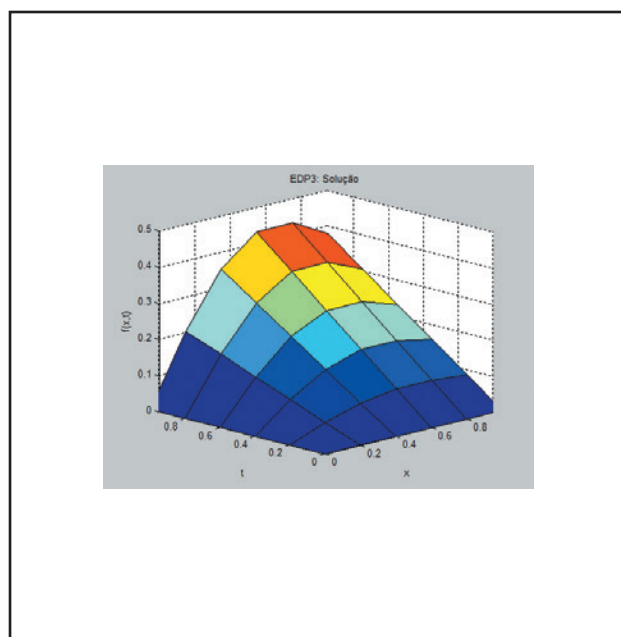


Figura 4. Gráfico da Função Solução da EDP

CONSIDERAÇÕES FINAIS

Como dito inicialmente, o estudo foi realizado com o objetivo de investigar o potencial de algoritmos evolucionários de inteligência artificial, desenvolvidos com a combinação de algoritmos de programação genética e diferenciação automática, na obtenção de soluções analíticas para problemas de equações diferenciais ordinárias e parciais. Acreditamos que o objetivo proposto foi alcançado, pois os resultados das aplicações são muito promissores, com soluções exatas obtidas para a grande maioria dos problemas abordados e que, empiricamente, comprovam a consistência e robustez da metodologia desenvolvida.

É importante informar que todos os problemas de equações diferenciais desenvolvidos neste estudo também foram testados na sub-rotina *DSolver* do programa *Mathematica* e nas *ODE-PDE-Solver Functions* do programa *Matlab*. Os resultados dos testes mostraram que mais de 50% das equações, aproximadamente 16 equações, não foram solucionadas por estes programas, embora tenham sido solucionadas de forma exata e analítica pelo algoritmo de DAPG. Estes resultados são relevantes, pois confirmam a potencialidade do algoritmo. Para finalizar, gostaríamos que o estudo realizado fosse entendido como uma metodologia alternativa e elaborada com a finalidade de auxiliar pesquisadores na difícil tarefa de obter soluções analíticas para complexos problemas de equações diferenciais. Além disso, gostaríamos de informar que estudos semelhantes estão sendo desenvolvidos pelos autores para solucionar problemas equações diferenciais estocásticas (EDE), também com resultados muito promissores. Esperamos, futuramente, apresentar os resultados destes estudos em um novo artigo.

REFERÊNCIAS

- BURGESS, G. *Finding approximate analytic solutions to differential equations using genetic programming*. Surveillance Systems Division, Electronics and Surveillance Research Laboratory, Department of Defense, Australia, 1999.
- CAO, H. et al. *Evolutionary modeling of systems of ordinary differential equations with genetic programming*, Genetic Programming and Evolvable Machines, vol. 1, pp. 309-337, 2000.
- FINK, M. *Automatic Differentiation for MATLAB*, 2007. MATLAB Central, www.mathworks.com/matlabcentral. (Martin Fink, Director of Hewlett-Packard – HP Labs research.)
- IBA, H.; SAKAMOTO, E. *Inference of differential equation models by genetic programming*. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2002), 2002, pp. 788-795.
- IMAE, J. et al. *Design of nonlinear control systems by means of differential genetic programming*. 43rd IEEE, Conference on Decision and Control. Atlantis, Paradise Island, Bahamas. December 14-17, 2004.
- KOZA, R. J. *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. 1992, MIT – Institute Technology. ISBN 0-262-11170-5.
- LUKE, S.; PANAIT, L. *Lexicographic parsimony pressure*. In: Langdon, W.B. et al., editors, Proceedings of GECCO-2002. Morgan Kaufmann (2002) 829-836.
- RALL, L. B. *Automatic Differentiation: Techniques and Applications*. Lecture Notes in Computer Science 120. Springer. ISBN 3-540-10861-0. (1981).
- SILVA, S. A. *Genetic Programming Toolbox for MATLAB*, 2009. Universidade Nova de Lisboa, PO Box 127, 2780-156 Oeiras, Portugal.
- TSOULOS, I. G.; LAGARIS, I. E. *Solving differential equations with genetic programming*. Genetic Program Evolvable Mach (2006) 7: 33-54.