https://eventos.utfpr.edu.br//sicite/sicite2019

Algoritmos genéticos e sua aplicação na resolução de problemas de controle ótimo de EDPs

Genetic algorithms and its application on the resolution of PDE-Constrained Optimization

Felipe de Carvalho Hosp
felipehosp@alunos.utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Apucarana, Paraná, Brasil

Juliana Castanon Xavier
julianaxavier@utfpr.edu.br
Universidade Tecnológica Federal do Paraná, Apucarana, Paraná, Brasil

RESUMO

O algoritmo genético (AG) é baseado nos mecanismos apresentados na teoria da evolução das espécies e também na biologia. Tais mecanismos, como cruzamento, mutação e seleção, são capazes de a partir de uma população inicial gerada aleatoriamente, chegar em populações melhores adaptadas. Essas populações podem ser interpretadas como soluções ótimas de problemas físicos. Tais problemas são frequentemente modelados matematicamente por um conjunto de equações diferenciais parciais (EDPs), cuja complexidade impossibilita, em geral, a determinação de soluções analíticas; em especial, os chamados problemas de controle ótimo, que buscam determinar o perfil de uma ou mais variáveis que minimizem ou maximizem uma função objetivo. Dessa forma, o objetivo desse trabalho é apresentar o AG na resolução de um problema desse tipo, relacionando cada mecanismo do AG com cada parte do problema analisado. É mostrada a estabilidade do algoritmo quando aplicado nessa situação.

PALAVRAS-CHAVE: Algoritmo genético. Problema de controle ótimo. Equações diferenciais parciais.

Recebido: 19 ago. 2019.
Aprovado: 01 out. 2019.

ABSTRACT

The genetic algorithms (GA) are based on the mechanisms presented by the evolution theory of species and biology. Those mechanisms, such as crossover, mutation and selection, based on a random initial population, are capable to reach a more adapted one. Those population can be interpreted as optimal solutions of physical problems. These problems are frequently modeled by partial differential equations (PDE), that in general are hard to solve analytically; in particular, optimal control problems, which aim to determine one or more variables that minimize or maximize an objective function. In this word, the goal was to present the use of a GA in the resolution of this kind of problem, connecting each on of its mechanism with a part of the problem analyzed. Also, it is showed in this paper the stability of the GA when applied it to this problem.

KEYWORDS: Genetic algorithm. Optimal control problem. Partial differential equations.

INTRODUÇÃO

Entre os métodos heurísticos, os algoritmos genéticos (AGs) em particular, são inspirados na biologia evolutiva e capazes de explorar eficientemente regiões promissoras no espaço de candidatos à solução com informações de gerações anteriores. Desta forma, os AGs proporcionam soluções ótimas ou sub-ótimas de maneira bastante eficaz para problemas complexos ou com espaço de soluções muito grande e de difícil determinação, quando comparados com os métodos de otimização convencionais.

A implementação de um AG baseia-se em uma simulação computacional que gera inicialmente uma população codificada que representa soluções abstratas. Essa é submetida a operadores genéticos biológicos, como por exemplo: a seleção natural, o cruzamento e mutação. Desta forma, uma nova e mais apta população de soluções é gerada e utilizada como entrada para a próxima iteração do algoritmo. Assim, após a repetição deste processo uma determinada quantidade de vezes, é esperada a convergência para a solução aproximada do problema.

Ainda que os AGs sejam probabilísticos, não configuram estritamente uma busca aleatória da solução. Os operadores utilizados sobre a população dirigem a busca para regiões do espaço mais suscetíveis, ou seja, regiões com maior probabilidade de se encontrar uma solução ótima.

Para que se tenha bastante variabilidade genética no AG utiliza-se populações com diversos indivíduos; quanto maior for a área de busca de um espaço de soluções maior a variabilidade gênica. Porém a grande variabilidade genética não é a única responsável por fornecer soluções para o problema, pois nesse caso existiria uma escolha de possíveis soluções, e dificilmente uma delas seria uma solução ótima. Para contornar esse problema, são implementados no algoritmo mecanismos de cruzamento e de mutação, que são responsáveis respectivamente por: combinar os genes de possíveis soluções de uma população para formar filhos que são também soluções e recuperar e/ou criar características genéticas que foram perdidas ou não eram existentes na população randômica inicial.

Os algoritmos genéticos são extremamente adaptáveis, sendo possível desenvolver mecanismos no código para que quanto mais iterações o AG executar, maior seja a chance de convergência para um ótimo global. Bons exemplos dessa adaptabilidade do AG são apontados por Michalawicz (1996), que mostra as diferentes contribuições ao mudar os tipos de mecanismos.

Com a ideia dessa adaptabilidade, é natural imaginar a aplicação dessa ferramenta na resolução dos problemas de controle ótimo relacionados a EDPs, assim como feito em Sahoo *et.al.* (2013). A dificuldade de solução de problemas de controle ótimo, também conhecidos por problemas de otimização dinâmica, é devida a necessidade de determinação do perfil de uma ou mais variáveis que minimizem ou maximizem uma função objetivo.

Essa função está associada a um conjunto de restrições de igualdades e desigualdades nas chamadas variáveis de estado e de controle, que caracterizam um sistema de equações algébrico-diferenciais. Com exceção de poucos problemas em que métodos analíticos são aplicáveis, a maioria destes problemas necessita da abordagem de métodos numéricos para a obtenção do perfil das variáveis de controle e de estado.

A determinação de soluções de problemas de controle ótimo é de grande interesse científico e industrial, pois a maioria de problemas físicos modelados matematicamente apresenta difícil solução analítica, como observado por Gen e Cheng (2000). Alguns métodos matemáticos, como por exemplo função lagrangeana e método de Newton, fornecem soluções ou aproximações dessas soluções. Todavia, estes métodos apresentam dificuldades em sair de soluções ótimas locais para soluções ótimas globais.

MATERIAIS E MÉTODOS

Para as implementações foi utilizado um computador com o sistema operacional Windows 10 64 bits, 16GB de memória RAM, processador Intel CORE i5. O código foi desenvolvido no *software* Octave®.

O trabalho foi dividido em quatro etapas: a primeira se deu com o estudo do AG; a segunda como a escolha do problema de controle ótimo; a terceira foi a implementação do AG para a resolução desse problema; e na quarta foram realizados testes numéricos com variações dos parâmetros do AG quando utilizado nessa situação. A seguir, cada uma dessas etapas encontra-se detalhada.

Para a abordagem inicial do estudo AG foi utilizado o problema clássico do caixeiro-viajante, que tem como objetivo minimizar a distância percorrida pelo caixeiro ao visitar um certo número de cidades. A implementação do código para esse problema serviu para identificar como diferentes operadores presentes na literatura afetam a resolução desse problema.

O problema de controle ótimo a ser resolvido neste trabalho é o representado pela Eq. (1).

$$min_{y,u}\frac{1}{2}\|y-\bar{y}\|^2{}_{L_2(\Omega)}+\frac{\beta}{2}\|u\|^2{}_{L_2(\Omega)},\ \beta>0 \tag{1}$$

onde consideramos o domínio $\Omega=[0,1]\times[0,1]$ e β uma constante. A Eq. (1) representa a função objetivo que está submetida a algumas restrições, como mostrado na Eqs (2) e (3), sendo a Eq. (3) válida na fronteira de Ω. O problema descrito com mais detalhes pode ser encontrado Rees *et al.* (2010).

$$-\nabla^2 y=u, \tag{2}$$

$$y=\bar{y},\ \bar{y}=-x_1 exp\left(-\left(\left(x_1-\frac{1}{2}\right)^2+\left(x_2-\frac{1}{2}\right)^2\right)\right). \tag{3}$$

Na teoria do controle ótimo o objetivo é determinar u de modo a obter uma função y boa (ou a melhor possível), de forma que neste sentido, u é chamado de variável de controle e y é chamado de variável de estado. Neste trabalho, o controle u deve satisfazer os limites dados pela Eqs. (4) e (5).

$$\underline{u}_a=\begin{cases}-0.35\text{ se }x_1<0.5\\ -0.4\text{ se }x_1\geq 0.5\end{cases} \tag{4}$$

$$\bar{u}_b=-0.1exp\left(-({x_1}^2+{x_2}^2)\right)\ se\ (x_1,x_2)\epsilon[0,1]^2. \tag{5}$$

Inicialmente foi utilizado o método das diferenças finitas (MDF) centralizado para a discretização da EDP, fazendo com que as derivadas parciais das funções sejam dependentes somente da função avaliadas em pontos específicos. O domínio $\Omega = [0,1] \times [0,1]$ da função foi discretizado considerando $n = m = 5, 10$ e 20 divisões nessa região, gerando uma malha de pontos uniformemente espaçados.

Para iniciar a implementação do AG, é definida uma população de indivíduos de forma aleatória, que recebe informações em seus genes formando seu DNA. Nesse caso, foram utilizados os limites superior e inferior do controle, dados pela Eqs. (4) e (5), para a formação de possíveis soluções iniciais randômicas, que são compostas de números reais. Na sequência, cada um dos indivíduos é classificado com base em quão boa solução ele representa. Nesse trabalho, os membros da população serão organizados com o método de ranqueamento não linear, sendo a probabilidade de seleção (função *fitness*) dada a cada indivíduo com base na sua função de minimização representada pela Eq. (1).

Após a seleção de dois indivíduos com a função *fitness*, é feito o cruzamento aritmético, ou seja, ocorre uma combinação linear dos materiais genéticos desses indivíduos. Esses indivíduos são selecionados a partir da geração anterior (população de pais), e são cruzados para darem origem a novos indivíduos que formarão a população de filhos, que tem tamanho reduzido em 30% em relação ao tamanho da geração anterior. Desta forma, para que haja uma regeneração do tamanho inicial da população, 30% dos melhores indivíduos são selecionados da população de pais sem qualquer modificação.

Assim que a população de novos indivíduos é formada, existe a chance de ocorrer a mutação. Esse mecanismo serve para criar ou recuperar características perdidas dentro da população; em uma visão mais matemática, a mutação serve para mover a população de um ótimo local para um ótimo global. A mutação não uniforme, utilizada neste caso, consiste na substituição de um gene por um número extraído de uma distribuição não uniforme. Essa modificação ocorre sempre com uma chance de 10% em cada indivíduo da nova população gerada no cruzamento; entretanto, os genes são modificados de maneira mais expressiva nas primeiras populações e de maneira mais sutil em populações posteriores. Deste modo, as chances de se chegar a um ótimo global são maiores, como demonstrado em Sahoo *et.al.* (2013).

Para a quarta etapa foram feitos testes com: diferentes tamanhos de população (tamanho da população); diferentes refinamentos da malha (número de divisões); diferentes números de iterações (número de iterações). Esses testes tinham como objetivo verificar a estabilidade do AG para esse problema e o respectivo tempo de processamento (tempo do AG).

Como parâmetro de comparação na quarta etapa, também foi utilizada o valor da função de minimização na Eq. (5) e a média das funções de minimização de todos os indivíduos da população (MDI). Foi analisado também $|\mathrm{MDI} - \mathrm{MI}|$ diferença entre o valor da função de minimização do melhor indivíduo (MI) e da MDI.

RESULTADOS E DISCUSSÕES

Na Tabela 1 estão ilustrados os resultados da implementação do AG considerando $\beta = 10^{-2}$. Foram efetuadas 10 repetições do algoritmo para cada valor de número de divisões, tamanho da população e número de iterações.

Tabela 1 – Tempo de processamento para resolução com diferentes parâmetros

Número de divisões	Tamanho da população	Número de iterações	Melhor indivíduo (MI)	Média dos indivíduos (MDI)	MDI – MI	Tempo do AG
5	10	100	0.8363	0.8363	5.551E-16	0.2812s
5	10	250	0.8364	0.8364	3.331E-16	0.7031s
5	10	500	0.8360	0.8360	1.110E-16	1.3481s
5	20	100	0.8362	0.8362	2.220E-16	0.6434s
5	20	250	0.8359	0.8359	4.441E-16	1.5711s
5	20	500	0.8359	0.8359	5.551E-16	3.2661s
10	10	100	4.4087	4.4087	1.954E-14	0.3058s
10	10	250	4.4085	4.4085	1.066E-14	0.7293s
10	10	500	4.4090	4.4090	1.599E-14	1.5361s
10	20	100	4.4079	4.4079	2.132E-14	0.6894s
10	20	250	4.4077	4.4077	1.688E-14	1.4780s
10	20	500	4.4075	4.4075	3.040E-14	3.0401s
20	10	100	25.001	25.001	8.100E-13	0.3389s
20	10	250	25.001	25.001	5.329E-13	0.7952s
20	10	500	25.001	25.001	6.430E-13	1.6104s
20	20	100	24.997	24.997	8.313E-13	0.7223s
20	20	250	24.999	24.999	7.923E-13	1.7747s
20	20	500	24.997	24.997	6.573E-13	3.6109s

Fonte: Autoria própria (2019)

É possível observar que para um mesmo número de divisões no domínio, os valores da função objetivo para o melhor indivíduo, na Tabela 1 representados por MI, sugerem que o AG se comporta de maneira estável para esse problema. O que também é indicado pelo fato de que a distância entre MI e MDI é pequena e da mesma ordem para o mesmo número de divisões.

CONCLUSÃO

Neste trabalho estudou-se o AG, desde sua definição inicial até sua utilização na resolução de um problema de controle ótimo com restrições. Durante esse processo, utilizou-se o MDF como técnica de discretização do problema, para posterior determinação dos mecanismos de seleção, cruzamento e mutação do AG. Os resultados da utilização do método nesse caso sugerem que o AG se comporta de maneira estável. Pode-se indicar como um trabalho futuro, um estudo mais aprofundado do modelo utilizado, bem como a convergência numérica da solução obtida pelo AG nesse caso, e a utilização desse método para outros problemas da literatura.

REFERÊNCIAS

GEN, M.; CHENG, R. ***Genetic Algorithms and Engineering Optimization.*** Jhon Wiley & Sons Inc., 2000.

MICHALAWICZ, Z. ***Genetic Algorithm + Data structure = Evolution Programs.*** 3ºed, Springer Verlag, Berlin, Heidelberg, New York, 1996.

SAHOO, L.; BHUMIA, A. K.; PAL, D.; MANDAL, B. N. ***An Alternative Approach for PDE-Constrained Optimization via Genetic Algorithm.*** *Journal of Information and Computing Science*, v. 8, nº 1, p. 41-54, 2013.

REES, T.; STOLL, M.; WATHEN A. ***All-At-Once Preconditioning in PDE-Constrained Optimization.*** Kybernetika, 2010.