

Algoritmo genético para otimização binária

Genetic algorithm for binary optimization

RESUMO

Walace Rutielo Lopes Santos
walacerutielo@gmail.com
Universidade Tecnológica Federal
do Paraná, Ponta Grossa, Paraná,
Brasil

Hugo Valadares Siqueira
hugosiqueira@utfpr.edu.br
Universidade Tecnológica Federal
do Paraná, Ponta Grossa, Paraná,
Brasil

Este trabalho tem por objetivo aplicar métodos bio-inspirados de otimização, como o algoritmo genético (AG) e a evolução diferencial (ED), e verificar o desempenho destes em suas versões binárias ao *OneMax Problem*. Foram desenvolvidas 10 versões para a evolução diferencial e 6 para o algoritmo genético e verificou-se que a primeira alcança um desempenho superior.

PALAVRAS-CHAVE: Otimização. Algoritmo genético. Evolução Diferencial.

Recebido: 19 ago. 2019.

Aprovado: 01 out. 2019.

Direito autoral: Este trabalho está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.



ABSTRACT

This work aims to apply bio-inspired optimization methods, such as genetic algorithm and differential evolution, and verify the performance of their binary versions to the *OneMax problem*. Ten versions were developed for differential evolution and 6 for the genetic algorithm. We verified that the first reached the best performances.

KEYWORDS: Optimization. Genetic algorithm. Differential evolution.

INTRODUÇÃO

A Computação Natural (inspirada na natureza) surge como uma alternativa as soluções já existentes para determinados problemas (Castro, 2006), isto é, a sua aplicação é mais conveniente quando se possui um problema de alta complexidade, com grande quantidade de variáveis e possíveis soluções. Uma vez que o problema não necessite desse tratamento, provavelmente ele possui alguma solução específica.

Na computação evolucionária são desenvolvidas técnicas de busca e otimização para solucionar problemas complexos baseados na evolução biológica (Linden, 2012). Os algoritmos evolutivos são ferramentas que emulam os processos naturais de evolução das espécies como seleção, reprodução e mutação, para resolverem os citados problemas.

O Algoritmo Genético e a Evolução Diferencial em suas versões binárias são os algoritmos evolutivos abordados nesse trabalho.

ALGORITMOS EVOLUTIVOS

Em 1975, John H. Holland propôs os primeiros algoritmos genéticos, os quais tornaram-se populares. Inicialmente foram desenvolvidos em codificação binária devido a tecnologia computacional disponível na época, porém o objetivo era a otimização real.

O Algoritmo Genético (AG) inspira-se na teoria da evolução de Darwin, na qual os indivíduos mais fortes e adaptados têm maiores chances de transmitir sua carga genética para as próximas gerações (Lewontin, 2013). É uma metaheurística de busca e otimização, que tem como propósito encontrar o melhor indivíduo para uma determinada situação.

A Evolução Diferencial (ED) proposta por Rainer Storn e Kenneth Price em 1995 (Opara et al., 2019), foi inicialmente criada para solucionar problemas de otimização contínua de forma robusta. É um método populacional no qual as possíveis soluções são obtidas através das combinações entre os vetores (agentes) da população.

A ED é um algoritmo evolutivo que se utiliza dos mesmos procedimentos do Algoritmo Genético: seleção, mutação e *crossover* (cruzamento). Nest, cada indivíduo é uma solução candidata. O algoritmo inicia-se com agentes aleatoriamente gerados, de modo que cada um carrega características próprias da mesma forma que um cromossomo. O cromossomo é formado por uma cadeia de genes, com várias posições, sendo cada uma denominada de locus, o qual contém um alelo. Na cadeia binária este valor é 0 ou 1. O agente é avaliado de acordo com a função *fitness* e essa avaliação é realizada a cada geração.

O valor do *fitness* tende a ser maior em relação a geração anterior. Dessa maneira ao final do ciclo, tem como resultado o melhor cromossomo ou agente.

RESULTADOS E DISCUSSÃO

Foram desenvolvidas 6 versões de algoritmos genéticos e 10 versões para a evolução diferencial.

Em relação aos algoritmos de Evolução Diferencial tem-se 10 variações desenvolvidas. Estas possuem uma nomenclatura padrão de classificação para apresentar de maneira simplificada qual estratégia o algoritmo utiliza. Para o tipo de mutação pode-se ter Rand, Best e Rand – to – Best. Na quantidade de diferenças ponderadas tem-se 1 ou 2. Por fim, os tipos de crossover que podem ser usados são o Binomial ou Exponencial.

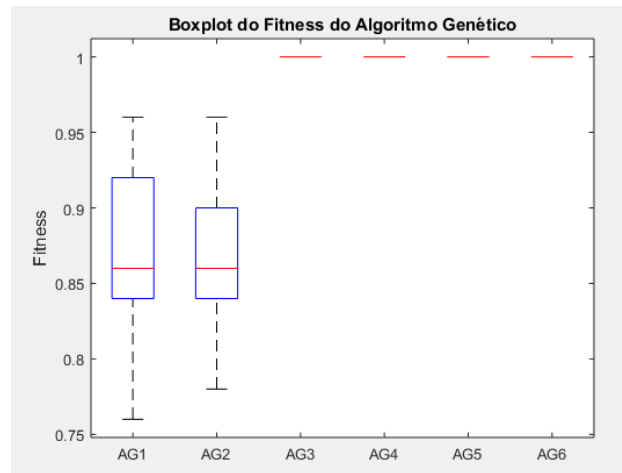
A lista das possibilidades segue abaixo:

- a) AG1: Clássico, seleção por roleta, com 70% de taxa *crossover*, sendo *crossover* de ponto e mutação fixa;
- b) AG2: Seleção por roleta, *crossover* de ponto e mutação fixa;
- c) AG3: Seleção por roleta com subpopulação, *crossover* de ponto e mutação fixa;
- d) AG4: Seleção por torneio, *crossover* de ponto e mutação fixa;
- e) AG5: Seleção por torneio com subpopulação, *crossover* de ponto e mutação fixa;
- f) AG6: Seleção por torneio, *crossover* de ponto e taxa de mutação em 20%;
- g) ED1: Rand/1/Bin;
- h) ED2: Best/1/Bin;
- i) ED3: Rand/2/Bin;
- j) ED4: Best/2/Bin;
- k) ED5: Rand-to-Best/2/Bin;
- l) ED6: Rand/1/Exp;
- m) ED7: Best/1/Exp;
- n) ED8: Rand/2/Exp;
- o) ED9: Best/2/Exp;
- p) ED10: Rand-to-Best/2/Exp.

Consideramos a taxa de mutação de 10% e *crossover* de 100% quando não mencionadas no algoritmo genético.

Para iniciar os testes definiu-se para cada algoritmo uma população com 50 componentes e cada um com 50 bits ($D = 50$). Todos os métodos foram simulados 30 vezes. Na Figura 1 é demonstrado o resultado dos algoritmos genéticos por meio do diagrama boxplot:

Figura 1 – Boxplot do *fitness* do AG



Fonte: Autoria própria (2019).

Em relação ao AG, o tipo de seleção influencia em uma convergência com menor dispersão. No caso do AG1 e AG2, que utilizaram método roleta, apresentaram uma dispersão maior do que os outros e com valor médio de *fitness* de 0,86. De maneira oposta, o AG3 obteve um resultado melhor mesmo utilizando a seleção do tipo roleta. Isso se deve a presença da subpopulação, em que os pais e os filhos competem para a próxima geração, o que evitou a perda dos melhores agentes, consequentemente tendo uma convergência para o ótimo global. A seleção do tipo torneio demonstra um desempenho melhor nos AG4, AG5 e AG6. A média de convergência atingiu o valor máximo de *fitness*.

Para o resultado da evolução diferencial e todos os algoritmos convergiram para o ótimo global, ou seja, todos atingiram o *fitness* igual a 1.

CONCLUSÃO

O presente artigo apresentou a aplicação dos métodos bio-inspirados de otimização, em suas versões binárias, Algoritmo Genético (AG) e Evolução Diferencial (ED) para solução do *OneMax Problem* em 50 dimensões.

A partir desses resultados apresentados é possível verificar que os algoritmos evolutivos são técnicas de otimização que devem ser consideradas para resolver problemas intrinsecamente binários. Algumas versões do AG apresentaram desempenho sub-ótimo, enquanto as propostas de Evolução Diferencial chegaram ao ótimo global.

Em futuros trabalhos podem ser abordados novos problemas como o da mochila, ou outras vertentes do *OneMax Problem*.

AGRADECIMENTOS

Agradecemos Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo suporte financeiro, processo número 405580/2018-5.

REFERÊNCIAS

CASTRO, L. N. **Fundamentals of Natural computing: basic concepts, algorithms, and applications**. Chapman and Hall/CRC, 2006.

HOLLAND, J. H. **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**, 1975.

LEWONTIN, R.; GRIFFITHS, A.; CARROLL,; WESSLER, S. **Introdução à Genética**. São Paulo: Guanabara, 2013.

LINDEN, R. **Algoritmos Genéticos**. Rio de Janeiro, p. 17-23, 2012.

OPARA, R. K.; ARABAS, J. **Differential Evolution: A survey of theoretical analyses, Swarm and Evolucionary Computation**, p. 546-558, 2019.