ADA: Evolução do Algoritmo do Duelista Aplicado a Problemas de Otimização Global

Demison Rolins de Souza Alves[1], Mário Tasso Ribeiro Serra Neto[1], Rodrigo Lisbôa Pereira[2,3], Edson Koiti Yasojima[2]

[1]Centro Universitário do Estado do Pará, Belém, Pará, Brasil. Grupo de Estudos Temático em Computação Aplicada -Laboratório de Inteligência do Centro Universitário do Estado do Pará (GET-COM -LINCOMP CESUPA)

[2]Universidade Federal do Pará, UFPA Laboratório de Computação Bio-Inspirada (LCBIO) Belém, Pará, Brasil

[3]Universidade Federal Rural da Amazônia, Paragominas, Pará, Brasil

demisonalves9@gmail.com, mariotnxd@gmail.com, rodrigo.pereira@ufra.edu.br, koitiyasojima@gmail.com

Abstract. *The Duelist Algorithm (DA) is a technique that obtained good performance solving problems of optimization with real variables, however, presents high randomness. Thus, this work presents an adapted model called ADA (Adapted Duelist Algorithm) that applies mathematical formulation to the classical algorithm to obtain a fast convergence. In order to validate its effectiveness, the algorithm to be tested in benchmark functions and its results compared to the canonical versions of the Algorithm Genetic and Duelist, without classification was superior in the test cases.*

Resumo. *O Duelist Algorithm (DA) é uma técnica que obteve bom desempenho solucionando problemas de otimização com variáveis reais, entretanto, apresenta alta aleatoriedade. Sendo assim, este trabalho apresenta um modelo adaptado denominado ADA (Algoritmo do Duelista Adaptado) que aplica formulações matemáticas ao algoritmo clássico para obter uma rápida convergência. Com o propósito de validar a eficiência do mesmo, o algoritmo fora testado em funções de benchmark e seus resultados comparados com as versões canônicas do Algoritmo Genético e Duelist Algorithm, no qual mostrou-se superior nos casos de teste.*

1. Introdução

Há diversos algoritmos utilizados e descritos na literatura solucionando problemas de otimização como o *Genetic Algorithm* (GA) por Holland (1975), *Artificial Bee Colony* (ABC) por Karaboga (2005), entre outros, o *Duelist Algorithm* (AD), desenvolvido por Biyanto et al. (2016). É mais uma técnica plausível a ser utilizada para soluções de tais problemáticas onde o objetivo é atingir o valor máximo ou mínimo de uma função, o foco deste artigo é expor a eficiência da adaptação realizada na técnica de *crossover* concebido por Yasojima et al. (2016), aplicado ao AD.

O trabalho está organizado da seguinte forma: a seção 2 explica o embasamento teórico sobre o AD, bem como são modeladas matematicamente; seção 3 explana a adaptação realizada ao modelo original; a seção 4 descreve as configurações e testes; na seção 5 têm-se os resultados obtidos; por fim, conclui-se o trabalho na seção 6.

2. Algoritmo do Duelista

Proposta por Biyanto et al. (2016), o AD é uma técnica para otimização inspirado em um duelo entre indivíduos. Este pode ser considerado uma competição esportiva como: uma luta de boxe, duelo entre dois lutadores, ou uma partida de futebol, duelo entre duas equipes [Reilly, 2003. Woodward, 2006.].

O modelo original é semelhante ao AG proposto por Holland (1975), que define indivíduos portadores de um cromossomo, e.g. vetor de elementos, e um valor de adaptação baseado em seu cromossomo. Entretanto, o AD trata o arranjo de elementos como um *skillset* (e.g. conjunto de habilidades), uma capacidade de luta como valor de *fitness* e um fator de sorte para cada duelista da população. O algoritmo é composto de seis etapas principais, em que com exceção da primeira etapa, outras estão inseridas em um ciclo finito de iterações, sendo estas:

- **Registro e Pré-qualificação**: Esta etapa corresponde ao processo de inicialização, encontrado em outros algoritmos apresentados por Binitha e Sathya (2012). O registro e pré-qualificação correspondem respectivamente a: 1) gerar aleatoriamente novos conjuntos de habilidade para cada duelista; 2) avaliar cada conjunto de habilidade e determinar sua capacidade de luta.
- **Determinar o quadro de campeões**: O quadro é gerado por uma série de treinamento entre o melhor duelista encontrado e outro duelista aleatório. O melhor duelista tende a passar experiência para seu treinando, este novo duelista substituiria a posição de melhor duelista no jogo e se juntaria ao próximo duelo. [Biyanto et al., 2016. Biyanto et al. 2017]
- **Definir a tabela de duelos e Premiação**: Dois duelistas, A e B, são selecionados aleatoriamente para, posteriormente, entrarem em um duelo baseado em sua capacidade de luta e sorte. Se a sorte do duelista A mais a sua capacidade de luta for maior que a do duelista B, A é dado como vencedor do duelo e B como perdedor. Caso contrário, B é o vencedor e A é o perdedor. A sorte para cada duelista é definida por (1),

$$sorte = D_f * (CS + (\text{rand}(0,1) * CS)) \quad (1)$$

  em que, D_f é a capacidade de luta do duelista e CS é coeficiente de sorte composto por um valor real. O vencedor do duelo continua seu treinamento para tornar-se um melhor duelista, enquanto que, o perdedor tem chances de adquirir determinada habilidade do vencedor através da experiência. [Byanto et al. 2016]
- **Eliminação**: Por fim, a cada competição, conjunto de duelos, o pior duelista é eliminado da competição, que abre uma oportunidade para outro duelista. [Biyanto et al., 2016. Biyanto et al. 2017]

3. Algoritmo do Duelista Adaptado

O trabalho desenvolvido por Biyanto et al. (2016), propôs um novo algoritmo de otimização para problemas de maximização de função e fora comparado com algoritmos presentes na literatura. O algoritmo demonstrou convergência em ambos os casos apresentados, entretanto, devido à alta aleatoriedade encontrada no algoritmo, modificações são necessárias para que o mesmo obtenha melhores resultados em problemas de otimização sem comprometer a analogia dos duelos. No Algoritmo do Duelista Adaptado (ADA), é abordada uma modificação em uma etapa encontrada no algoritmo original, respectivamente: a etapa de premiação foi reescrita, tal que, o novo modelo permite a capacidade de o perdedor explorar sabiamente o que deve melhorar em seu conjunto de habilidades para uma melhor capacidade de luta. Enquanto que o vencedor treina exaustivamente a mesma habilidade para chegar ao topo rapidamente.

3.1. Premiação Adaptada

Para a etapa de premiação, o duelista vencedor recebe um valor aleatório destinado a seu conjunto de habilidade e para o perdedor, o valor do vencedor era designado ao mesmo. Todavia, este modelo de premiação contempla o perdedor de forma mais significativa que para o vencedor, tal que, um valor aleatório inserido ao valor do conjunto de habilidade do vencedor, pode causar a estagnação em um melhor local, pois o vencedor pode não melhorar com o novo treinamento aleatório. Enquanto isso, o perdedor por sua vez, obtém rapidamente um valor melhor que seu anterior.

Após a etapa de duelos e, definir vencedor e perdedor, o ajuste no conjunto de habilidade do perdedor é dado por (2),

$$f(x) = \begin{cases} P_i = P_i + \alpha * (V_i - P_i), & Se,\ V_i \leq P_i \\ P_i = P_i + \alpha * (V_i + P_i), & Se,\ V_i > P_i \end{cases} \quad (2)$$

Em que, P_i corresponde ao conjunto de habilidade aleatório *i* do perdedor, V_i equivale ao conjunto de habilidade aleatório *i* do vencedor e por fim, α equivale a uma constante cujo objetivo é controlar o valor de adaptação do perdedor. Este mecanismo de adaptação ao treinamento é uma simplificação do modelo apresentado por Yasojima et al. (2016), que aborda um fluxo de direções para cada valor do cromossomo dos indivíduos. Para a bonificação do vencedor, (3) é aplicada,

$$V_i = V_i + (V_i * \alpha) \quad (3)$$

em que, V_i e α correspondem ao mesmos valores encontrados em (2). O mecanismo adotado incrementa o treinamento de ambos os duelistas auxiliando a adaptação para que, em outro duelo, estes venham a obter desempenho melhor que outros futuros adversários. A Figura 1 ilustra o mecanismo abordado para bonificação dos treinamentos pôs duelo.

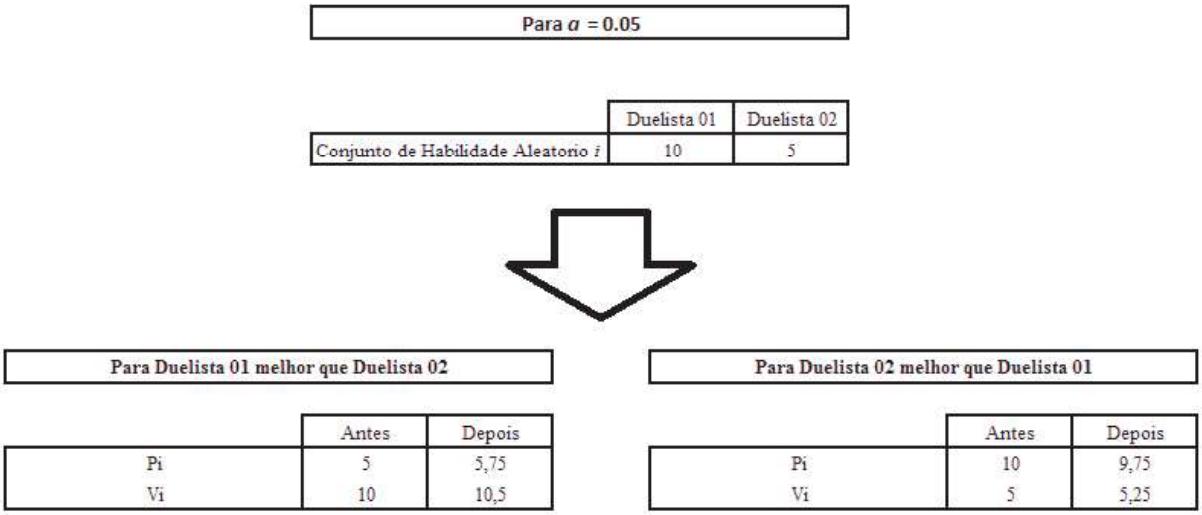


Figura 1. Mecanismo de otimização para vencedores e perdedores.

4. Configurações e testes

No seguinte experimento, a Tabela 1 apresenta as funções de *benchmark* utilizadas para validar a eficiência do algoritmo, ambas com o objetivo de maximização. Estas funções são consideradas representativas pela comunidade cientifica a fim de testar a eficiência de qualquer algoritmo com proposito de otimização numérica.

No que diz a respeito dos parâmetros do algoritmo, as configurações propostas por Byanto et al. (2016), foram utilizadas para o ADA, sendo estas: 1) 100 duelistas; 2) o valor

designado a constante α é igual a 0.05; 3) 200 gerações; 4) coeficiente de sorte 0.1. Por fim, os parâmetros do Algoritmo Genético adotado foram: 1) 100 indivíduos; 2) 200 gerações; 3) probabilidade de mutação 0.5; 4) probabilidade de crossover 0.8.

Tabela 1. Funções de *benchmark* utilizadas.

Função	Limite	Dimensão	Equação	Global
F1	[0-10]	2	$f(x) = -(x.\sin(4.x) + 1{,}1.y.\sin(2,y))$	18.5547
F2	[0-10]	2	$f(x) = -(\sqrt{x^2 + y^2.\cos(x-y)}\,.\,e^{(\cos\left(\frac{x.(y+5)}{7}\right))})$	30.3489

5. Resultados

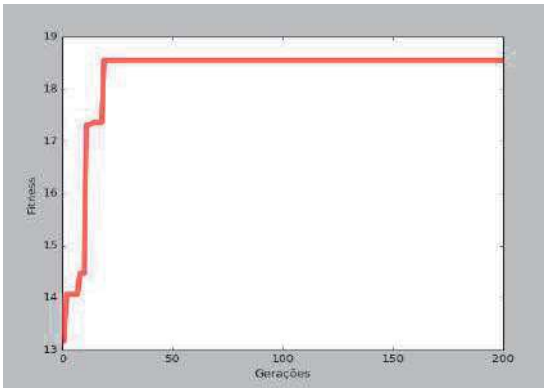
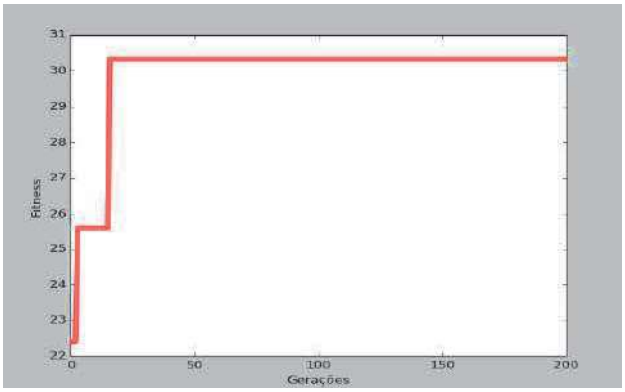
A Tabela 2 apresenta os resultados estatísticos associados às funções. Estes resultados foram obtidos por intermédio dos valores encontrados em 30 execuções distintas e *seeds* aleatórias diferentes, sendo assim, garantindo que nenhuma técnica fora influenciada pela máquina.

Tabela 2. Resultados estatísticos para as funções de *benchmark*.

Função	Resultados	ADA	DA Biyanto et al. (2016)	GA Biyanto et al. (2016)
F1	Fitness	**18.5495**	18.5285	-
	Média	18,1099		
	DP	0,32955		
F2	Fitness	**30.3248**	30.3060	30.3017
	Média	30,2277		
	DP	0,4960		

A Tabela 3 apresenta a evolução das melhores soluções obtidas, onde para F1 encontrou-se o resultado após a vigésima iteração e para F2 após a decima sétima.

Tabela 3. Evolução das soluções máximas encontradas utilizando o ADA.

F1	F2
	

Após a análise de resultados, constatou-se que o algoritmo aqui proposto obteve melhor desempenho quando comparado à versão canônica, encontrando valores mais aproximados do máximo global. Portanto, as modificações inseridas permitiram uma ampla busca no espaço, encontrando duelistas mais aptos para estes problemas.

6. Conclusão

O objetivo deste estudo foi analisar o desempenho do ADA em comparativo com a versão canônica do mesmo para funções de *benchmark,* onde por meio destes testes efeituados obteve resultados superiores ao GA e o AD desenvolvido por Biyanto et al. (2016). A modificação realizada no AD provou-se efetiva, por meio da adaptação da técnica de *crossover* concebida por Yasojima et al. (2016), onde pela qual conseguiu aproveitar de forma mais ampla o espaço de busca deparando-se com soluções melhores.

Como trabalhos futuros, pretende-se ampliar os testes do algoritmo proposto com o intuito de encontrar suas limitações, para talvez aplicar uma próxima modificação no mesmo assim buscando conseguir resultados melhores.

Referências

Binitha, S., and S. Siva Sathya. "A survey of bio inspired optimization algorithms." International Journal of Soft Computing and Engineering 2.2 (2012): pp 137-151.

Biyanto, Totok R., Henokh Yernias Fibrianto, and Hari H. Santoso. "Duelist Algorithm: An Algorithm in Stochastic Optimization Method." Advances in Swarm Intelligence, The Seventh International Conference on Swarm Intelligence, Junho. 2016.

Biyanto, Totok R., et al. "Techno Economic Optimization of Petlyuk Distillation Column Design Using Duelist Algorithm." Procedia Engineering 170 (2017): pp 520-527.

J. Holland. Adaptation in Natural and Artificial Systems. University of Michigan Press, 1975.

K. Woodward, Boxing, Masculinity and Identity. The" I" of the Tiger: Routledge, 2006.

Karaboga D. “An idea based on honey bee swarm for numerical optimization”. Vol. 200. Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.

T. Reilly, Science and soccer: Routledge, 2003.

Yasojima, Edson Koiti Kudo, et al. "Analyzing Genetic Algorithm with Game Theory and Adjusted Crossover Approach on Engineering Problems." International Conference on Hybrid Intelligent Systems. Springer, Cham, 2016.