



CONEM 2016
CONGRESSO NACIONAL DE
ENGENHARIA MECÂNICA

21 - 25
AGOSTO DE 2016
FORTALEZA - CEARÁ

METAHEURÍSTICAS DE OTIMIZAÇÃO APLICADAS À SINTONIA DE CONTROLE PID DE UM PROCESSO BALANÇO HORIZONTAL

Marco Antonio Reichert Boaretto, marco.boaretto@hotmail.com¹

Edson Gnatkovski Gruska, edson.gruska@gmail.com²

Leandro dos Santos Coelho, lscoelho2009@gmail.com³

¹ Pós-Graduação em Engenharia Elétrica (PPGEE), Universidade Federal do Paraná (UFPR), Curitiba, PR, Brasil

² Pós-Graduação em Engenharia de Produção e Sistemas (PPGEPS), Escola Politécnica, Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, PR, Brasil

³ Pós-Graduação em Engenharia de Produção e Sistemas (PPGEPS), Escola Politécnica, Pontifícia Universidade Católica do Paraná (PUCPR) e Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), Universidade Federal do Paraná (UFPR), Curitiba, PR, Brasil

Resumo: Neste artigo, a solução de três problemas de otimização com uma variante auto adaptativa do algoritmo de evolução diferencial JaDE é proposta. Para provar a efetividade do algoritmo JaDE, o mesmo será comparado com mais dois algoritmos, a Evolução Diferencial clássica e um algoritmo baseado em cosmologia, denominado de Otimizador Multiverso (Multi-verse optimizer). Um dos problemas a ser solucionado é o da sintonia dos parâmetros de um controlador PID (proporcional, integral e derivativo) aplicado a um processo experimental de controle, conhecido como Processo de Balanço Horizontal, utilizando como critério de desempenho o valor do ITAE (integral of time-weighted absolute error) em malha fechada. Os outros dois problemas de otimização adotados são problemas de testes, ambos apresentados na literatura de otimização utilizando metaheurísticas.

Palavras-chave: Evolução diferencial; Otimizador Multiverso, Controle proporcional, integral e derivativo; Processo de Balanço Horizontal.

1. INTRODUÇÃO

O interesse em desenvolver novas técnicas de otimização inspiradas pela natureza, nas mais diversas áreas, tem aumentado a partir das últimas décadas. Existe uma grande variedade de técnicas de otimização e uma classe que vem se destacando como promissora é a dos algoritmos evolutivos. Exemplos recentes da aplicação bem-sucedida de algoritmos evolutivos são os de Ma *et al.* (2013), Bianchi *et al.* (2014) e Reynoso-Meza *et al.* (2016). Um paradigma desta classe de algoritmos é a Evolução Diferencial (ED), (Price e Storn, 1995; Price, Storn e Lampinen, 2005; Plessis e Engelbrecht, 2012; Piotrowski, Napiorkowski e Kiczko, 2012).

A ED é um algoritmo simples de se implementar e possui um baixo custo computacional, Mitchell (1996) mostra que a ED vem sendo utilizada recentemente e superou variantes dos Algoritmos Genéticos (AGs) e dos algoritmos de otimização por enxame de partículas (*Particle Swarm Optimization*, PSO) (Cho *et al.*, 2011; Chou, 2013). Uma das vantagens da ED é a de possuir poucos parâmetros de controle quando comparada com outras técnicas de otimização.

Neste trabalho, com o intuito de apresentar resultados competitivos para a ED, serão realizados testes com três problemas de otimização com o objetivo de minimizar uma função objetivo (*fitness*), sendo um deles um problema real de ajuste de controle para um sistema em malha fechada, denominado Processo de Balanço Horizontal (PBH).

Os outros dois problemas são funções clássicas, geralmente usadas em testes de algoritmos de otimização, a função de Rastrigin (Sathya e Radhika, 2013), e a função de Rosenbrock (Jarre, 2013).

O algoritmo de ED será comparado com uma de suas variantes auto adaptativas denominada JaDE, conforme apresentado em Zhang e Sanderson (2007) e Zhang e Sanderson (2013), e um algoritmo baseado em conceitos cosmológicos denominado otimizador multiverso (*Multi-verse Optimizer*, MVO), conforme proposto por Mirjalili, Mirjalili e Hatamlou (2016).

O restante deste artigo está dividido da seguinte forma. Na seção 2 é apresentado o algoritmo de ED e suas respectivas etapas no ciclo evolutivo. Na seção 3 é apresentado o algoritmo JaDE. Depois disto, na seção 4, o algoritmo MVO e as regras necessárias para o desenvolvimento do algoritmo são detalhados. A seção 5 apresenta uma descrição

do PBH e a técnica de controle adotada. Na seção 6 são mostrados e comentados os resultados obtidos com os testes, seguidos pela conclusão do artigo, dada na seção 7.

2. O ALGORITMO DA ED

O algoritmo de ED proposto por Price e Storn (1995) foi adotado originalmente para resolver o problema de Chebyshev. A ED mostrou ser uma simples, porém eficaz abordagem de otimização para resolver uma variedade de problemas de referência como também em aplicações na indústria. O trabalho de Chiha, Ghabi e Liouane (2012) mostra que a ED envolve uma população de potenciais soluções (indivíduos), buscando aumentar a convergência para soluções ótimas. Por ser um algoritmo baseado em busca populacional, consegue resolver problemas de mínimo local e convergência prematura de uma maneira fácil e rápida em contraste com algoritmos baseados em solução simples.

O algoritmo de ED é derivado dos AG, portanto certa semelhança é observada ao comparar a estrutura de ambos algoritmos. Baseado em Price e Storn (1995), o algoritmo de ED pode ser definido pelas etapas descritas nas próximas subseções.

2.1. Inicialização da população

Após serem definidos os parâmetros de controle, o fator de perturbação F e a probabilidade de cruzamento Cr , são determinados os limites máximo $b_{j,U}$ e mínimo $b_{j,L}$ para gerar o domínio do espaço de busca para o cálculo do vetor inicial $x_{i,j,0}$ Eq. (1). Em contraste com outros algoritmos evolutivos a ED não usa funções de distribuição de probabilidade com intuito de adicionar variações na população, ao invés, utiliza a diferença de vetores aleatoriamente selecionados como fonte de variações aleatórias.

$$x_{i,j,0} = rand_j(0,1) \times (b_{j,U} - b_{j,L}) + b_{j,L} \quad (1)$$

onde $rand_j$ é a função geradora de números aleatórios uniformemente distribuídos no intervalo entre 0 e 1. O subscrito j indica que a cada parâmetro é gerado um valor aleatório novo.

2.2. Operação de mutação

O processo de mutação, Eq. (2), é a diferença entre dois vetores escolhidos aleatoriamente, que passam a ser multiplicados à um fator de perturbação, que controla a taxa de evolução populacional. Um terceiro vetor, também selecionado aleatoriamente, é adicionado à equação, originando o vetor mutante $v_{i,g}$. Uma possível razão para ED funcionar tão bem, segundo Mayer, Kinghorn e Archer (2005), é que a mutação é acionada por diferenças entre o valor dos parâmetros de membros da população contemporânea.

$$v_{i,g} = x_{r0,g} + F \cdot (x_{r1,g} - x_{r2,g}) \quad (2)$$

onde, enquanto $x_{r0,g}$, $x_{r1,g}$ e $x_{r2,g}$ são os vetores selecionados aleatoriamente, e g é o índice da geração atual.

2.3. Operação de cruzamento

Conhecido também por reprodução, Eq. (3), é onde ocorre o cruzamento entre os vetores mutante e alvo para gerar um novo vetor teste, formado pela combinação de informações entre os dois vetores.

$$u_{i,g} = u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{se } rand_j(0,1) \leq Cr \text{ ou } j = j_{rand} \\ x_{j,i,g} & \text{caso contrário} \end{cases} \quad (3)$$

onde Cr é a probabilidade de cruzamento que possui valores no intervalo [0,1], e corresponde a fração média de componentes que são geradas no vetor mutante.

2.4. Seleção

É na seleção, Eq. (4), que o vetor alvo e o vetor teste competem de acordo com seu *fitness*, o vetor que tiver o melhor *fitness* passa para a próxima geração.

$$x_{i,g+1} = \begin{cases} u_{i,g} & \text{se } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{caso contrário} \end{cases} \quad (4)$$

Para um problema de maximização é trocado o operador $\leq$ por $\geq$. Os passos se repetem até que um critério de parada seja encontrado. Na Fig. 1 o pseudocódigo da ED.

Pseudocódigo ED*Definição de $b_{j,U}$ e $b_{j,L}$, e dos parâmetros de controle F e Cr .**Geração da população inicial Eq. (1).***enquanto** critério de parada não atingido.**para** 1 até população máxima (N_p).*Selecionar três indivíduos diferentes r_1 , r_2 e r_3 .**Mutação Eq. (2)**Cruzamento Eq. (3).**Seleção dos melhores indivíduos resultantes dos processos anteriores Eq. (4).***fim para****fim enquanto**

...

Figura 1. Pseudocódigo ED**3. JADE**

O algoritmo JaDE proposto por Zhang e Sanderson em 2007, é uma variação auto adaptativa da ED, significa que ao decorrer das iterações a JaDE automaticamente ajusta os parâmetros de controle da ED de acordo com a função a ser otimizada, para obter o melhor desempenho possível. A JaDE também conta com uma estratégia de mutação considerada uma das mais bem-sucedidas da ED, “*DE/Current-to-pbest*”, Eq. (5), que seleciona vetores não só da população atual, mas também da união entre população atual com um grupo de soluções anteriores arquivadas.

$$v_{i,g} = x_{i,g} + F_i \cdot (x_{best,g}^p - x_{i,g}) + F_i \cdot (x_{r1,g} - x_{r2,g}) \quad (5)$$

onde $x_{best,g}^p$ é aleatoriamente escolhido como um dos top 100% p indivíduos da população atual $p \in [0,1]$, e F_i é o fator de mutação associado com x_i que é regenerado a cada geração pelo processo de adaptação.

A cada nova iteração os parâmetros fator de perturbação e fator de cruzamento são atualizados por suas respectivas médias μ_F Eq. (6) e μ_{CR} Eq. (7) que por sua vez dependem de seus respectivos antecessores armazenadas em um conjunto de médias (S_F e S_{CR}) que obtiveram um bom desempenho.

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot mean_L(S_F) \quad (6)$$

onde c é uma constante positiva entre 0 e 1 e $mean_L$ é a média de Lehmer.

$$\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot mean(S_{CR}) \quad (7)$$

onde $mean$ é uma operação comum de média aritmética.

Os passos são repetidos até um critério de parada ser encontrado. Como pode ser visto na Fig. 2, o pseudocódigo da JaDE.

Pseudocódigo JaDE

Iniciar população $x_{0,g}$ e definir $\mu_{Cr} = 0,5$ e $\mu_F = 0,5$.

para $g=1$ até G

Gerar $CR_i = \text{rand}_i(\mu_{Cr}; 0,1)$, para todo $i=1,2,\dots, NP$

Gerar um terço de todos $F_i = \text{rand}_i(0; 1,2)$

e outros $F_i = \text{rand}_i(\mu_F; 0,1)$

S_F e $S_{Cr} = \emptyset$

para $i=1$ até NP

Escolha aleatória de $x_{best,g}^p$

Escolha aleatória de $r_1 \neq r_2 \neq i$

Mutação “DE/Current-to-pbest” Eq. (5)

Gerar $j_{rand} = \text{randint}_i(1,D)$

para $j=1$ até D

se $j=j_{rand}$ ou $\text{rand}_i(0,1) \leq CR_i$

$u_{j,i,g} = v_{j,i,g}$

senão

$u_{j,i,g} = x_{j,i,g}$

fim se

fim para

se $f(x_{i,g}) \leq f(u_{i,g})$

$x_{i,g+1} = x_{i,g}$

senão

$x_{i,g+1} = u_{i,g}$, $S_{Cr} = S_{Cr} \cup CR_i$, $S_F = S_F \cup F_i$

fim se

fim para

Equação (6)

Equação (7)

fim para

$g=g+1$

Figura 2. Pseudocódigo JADE

4. MVO

O algoritmo da MVO proposto em 2016 por Mirjalili, Mirjalili e Hatamlou, se baseia na teoria do *Big Bang* partindo do princípio em que não existe apenas o universo em que vivemos, mas também uma infinidade de outros universos paralelos. São escolhidos três conceitos da teoria do multiverso: buraco negro, buraco branco e buraco de minhoca.

Guth (2007) mostra que cada universo possui uma taxa de dilatação na qual o universo se expande infinitamente, em busca de um equilíbrio entre os universos paralelos, os mesmos interagem entre si por meio de buracos brancos, buracos negros e buracos de minhoca, de acordo com Steinhart e Turok (2005).

Durante o processo de otimização certas regras devem ser aplicadas para os universos da MVO, segundo Mirjalili, Mirjalili e Hatamlou (2016):

- Quanto maior a taxa de dilatação, maior a probabilidade de haver um buraco branco.
- Quanto maior a taxa de dilatação, menor a probabilidade de haver um buraco negro.
- Universos que possuem uma alta taxa de dilatação tendem a enviar objetos pelos buracos brancos.
- Universos com uma baixa taxa de dilatação tendem a receber objetos pelos buracos negros.
- Os objetos, em todos os universos, podem sofrer movimentos aleatórios em direção ao melhor universo através de buracos de minhoca independente de sua taxa de dilatação.

Baseado nessas regras, os universos podem livremente trocar objetos um com o outro através de buracos branco e negro, havendo sempre uma possibilidade maior de um universo com alta taxa de dilatação enviar objetos para um universo com baixa taxa de dilatação, como mostra Mirjalili, Mirjalili e Hatamlou (2016). Para manter a diversidade entre os universos são considerados os buracos de minhoca para enviar objetos aleatoriamente entre os universos.

Mirjalili, Mirjalili e Hatamlou (2016) mostra que o algoritmo MVO, provou ser uma ferramenta muito eficaz na otimização de problemas uni modais, multimodais e problemas reais de engenharia, garantindo convergência por meio de busca local proporcional ao número de iterações.

Na Fig. 3, segue o pseudocódigo do algoritmo MVO.

Pseudocódigo MVO

Gerar universos aleatórios (U).

Iniciar WEP (Prob. de existência de buraco de minhoca), TDR (Taxa de distância percorrida), Best_Un;

SU=>universos sortidos.

NI=>normaliza a taxa de inflação (fitness) dos universos.

enquanto critério não atingindo

Avaliar fitness de todos os universos

para cada universo i

Buraco Negro Índ=i;

para cada objeto j

r1=rand([0,1]);

se r1<NI(U_i)

Buraco Branco Índ=Seleção (-NI);

U(Buraco Negro Índ,j)=SU(Buraco Branco Índ,j);

fimse

r2=rand([0,1]);

se r2<WEP

r3=rand([0,1]);

r4=rand([0,1]);

se r3<0,5

U(i,j)=Best_Un(j) + TDR((max(j)- min(j))*r4+min(j));*

senão

U(i,j)=Best_Un(j) - TDR((max(j)- min(j)) *r4+min(j));*

fimse

fimse

fimpara

fimpara

fimequanto

Figura 3. Pseudocódigo MVO

5. PBH

O PBH, como mostra a Fig. 4, consiste em um processo de controle com o objetivo de controlar o ângulo da barra horizontal, que trabalha em um ângulo limitado de 30 graus em ambos os sentidos.

Os atuadores do processo são dois motores DC cada um com uma hélice acoplada ao seu eixo, que se encontram nas extremidades da barra horizontal. No centro de intersecção das duas barras transversais existe um potenciômetro que faz a identificação do ângulo da barra, Fig. 5.



Figura 4. Vista frontal do PBH

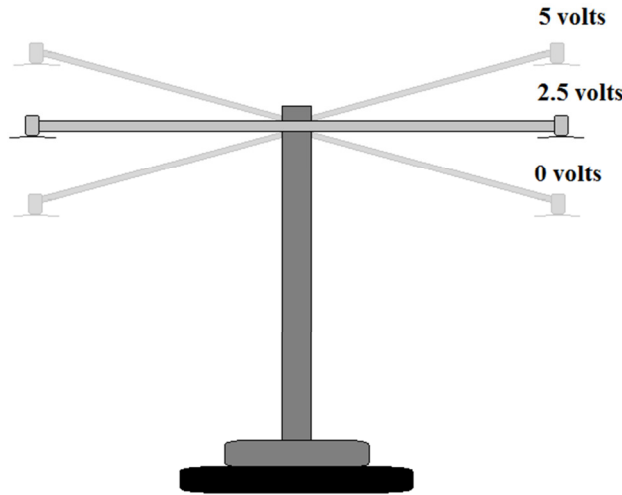


Figura 5. Diagrama do PBH

O controle de ambos os motores é feito por um controlador Proporcional-Integral-Derivativo (Shigemasa, Negishi e Baba, 2013). Como pode ser visto na Fig. 6, o diagrama de controle em malha fechada com o controlador, otimizador e o processo. Controladores PID possuem três parâmetros (proporcional, integral e derivativo), que devem ser sintonizados para obter o melhor desempenho para o processo, como pode ser vista na Eq. (8).

$$C_{PID} = k_p \cdot \left(1 + \frac{1}{T_i s} + \frac{T_d s}{T_d s + 1} \right) \quad (8)$$

onde k_p , T_i e T_d correspondem respectivamente ao ganho proporcional, tempo integral e tempo derivativo.

Segundo Chiha Ghabi e Liouane (2012) por ser tratar de um processo não linear, técnicas convencionais de sintonia de controladores PID, como Ziegler-Nichols, não são recomendadas. Portanto, é ideal utilizar métodos mais complexos para obter um controlador mais robusto e adequado para o PBH.

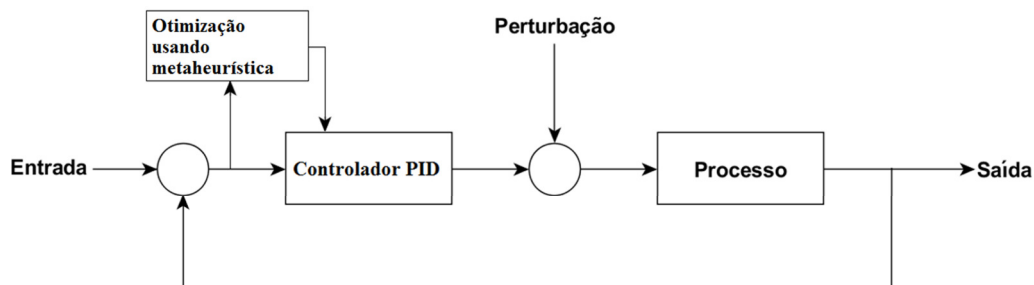


Figura 6. Diagrama de controle e otimização

O tempo integral de erro absoluto (*ITAE*) gerado em malha fechada do esquema mostrado na Fig. 6 será usado como a função objetivo a ser minimizada para encontrar os melhores ganhos do controlador PID para o PBH.

6. RESULTADOS E DISCUSSÃO

Os testes foram executados no seguinte sistema, sistema operacional Windows 8.1 64-bit, CPU core i7-4700HQ 2.4 GHz, RAM (Random Access Memory) de 8 GB e no ambiente computacional MATLAB® R2014a.

A função de Rastrigin, Fig. 7, é uma função multimodal e separável, uma das características que a tornam um desafio para encontrar uma solução ótima, é que o seu contorno é constituído de diversos mínimos locais que se distanciam do mínimo global, Sathya e Radhika (2013). O que pode trazer dificuldades para o algoritmo discernir entre mínimo local e mínimo global.

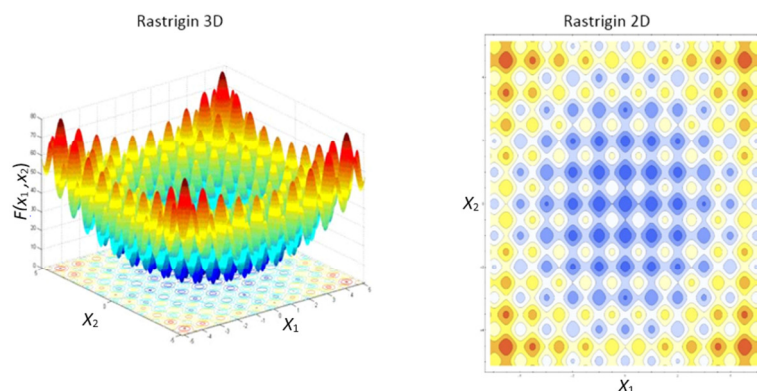


Figura 7. Função de Rastrigin, à esquerda em sua representação em 3D, e à direita o contorno da função

A função de Rosenbrock, Fig. 8, é uma função de duas dimensões que apresenta um vale em forma de parábola, segundo Sathya e Radhika (2013), em razão da não linearidade apresentada pela função de Rosenbrock muitos algoritmos de otimização levam mais tempo para convergir a uma solução ótima, tornando-a uma ótima opção para testar a robustez de algoritmos de otimização.

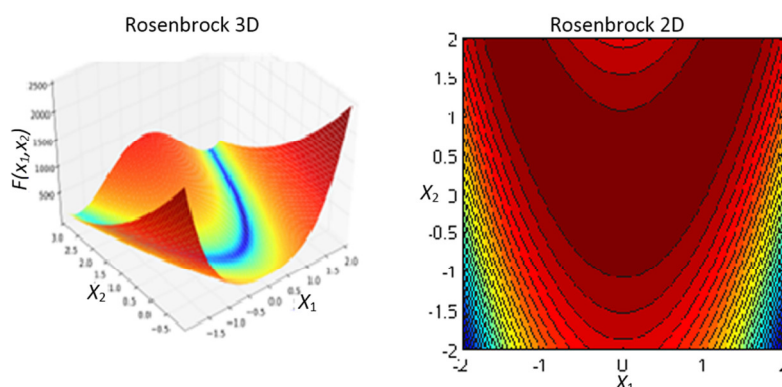


Figura 8. Função de Rosenbrock, à esquerda em sua representação em 3D, e à direita o contorno da função

Ao realizar os testes com os três algoritmos, foram feitos cinquenta experimentos (simulações) com cada. Para ambas as funções Rastrigin e de Rosenbrock foram determinados respectivamente dez dimensões, e para o PBH foram determinados apenas 3.

Para a ED o fator de perturbação e probabilidade de cruzamento foram determinadas respectivamente em 0,9 e 0,5. O tamanho da população foi de 30 indivíduos, foram utilizadas 50 iterações (gerações) por experimento. A estratégia utilizada neste processo foi “DE/local-to-best/1”.

Para o algoritmo da JaDE, os parâmetros iniciais de μ_F e μ_{CR} foram respectivamente 0,5 e 0,5. O tamanho da população utilizada foi de 30 indivíduos, foram rodadas 50 iterações por experimento.

Para o algoritmo da MVO, os parâmetros iniciais foram determinados em 100 universos com um WEP entre 0,2 e 1, e 50 iterações por experimento.

As Tabelas 1, 2 e 3 mostram os valores resultantes dos testes.

Tabela 1. Tabela com os valores da função Rastrigin

Algoritmo	Rastrigin		
	Melhor	Média	Desvio Padrão
ED	30,1560	57,8704	10,5564
JaDE	0,0000	0,0020	0,0141
MVO	5,4464	23,4833	9,7866

Para a função de Rastrigin, o algoritmo da JaDE obteve o melhor desempenho dentre os três algoritmos, chegando mais próximo do mínimo global em 0. Provando ser uma técnica eficiente para solução de funções que possuem muitos mínimos locais. Caso o método não possua robustez suficiente, pode acabar ficar preso nos mínimos locais da função ou convergir prematuramente.

Tabela 2. Tabela com os valores da função Rosenbrock

Algoritmo	Rosenbrock		
	<i>Melhor</i>	<i>Média</i>	<i>Desvio Padrão</i>
ED	4,1035	2302,9	3996,6
JaDE	0,0000	0,0000	0,0000
MVO	8,8457	279,6731	582,8186

Apesar de ambas as técnicas ED e MVO terem alcançado valores muito baixos para a função Rastrigin, não chegaram próximas ao mínimo global em 0 da função Rosenbrock. A técnica JaDE novamente alcançou o valor de mínimo global, mostrando ser eficaz ao lidar com a minimização de funções não convexas.

Tabela 3. Tabela com os valores do PBH

Algoritmo	PBH		
	<i>Melhor</i>	<i>Média</i>	<i>Desvio Padrão</i>
ED	78,6842	101,2485	63,5154
JaDE	78,1433	80,8247	8,8580
MVO	78,2052	79,3454	1,2771

Na Tab. 3, referente ao PBH, as três técnicas chegaram a valores muito próximos do valor de *ITAE* e só apresentam diferença ao aumentar o número de casas decimais, os três algoritmos trabalham muito bem ao encontrar soluções ótimas para problemas não lineares e complexos como o PBH. Entretanto o algoritmo da JaDE se saiu melhor em relação a ED e ao MVO.

A Fig. 9 mostra a resposta em malha fechada do sistema já mostrado anteriormente na Fig. 6, para os valores das constantes proporcional, integral e derivativa encontradas pelos algoritmos da ED, JaDE e MVO, Tabela 4. Como pode ser visto, em razão das respostas obtidas nos testes com o PBH terem sido muito próximas, torna-se um pouco complicado notar uma diferença acentuada, tanto no gráfico de resposta em malha fechada (Fig. 9) como nos valores dos parâmetros do controlador, Tabela 4. Entretanto pode se perceber que os três algoritmos se saíram muito bem no controle do PBH, mostrando respostas rápidas que apresentam pouca instabilidade, e se adequam rapidamente as variações da referência de entrada.

Tabela 4. Tabela com os ganhos do controlador PID

Algoritmo	PBH		
	<i>K_p</i>	<i>T_i</i>	<i>T_d</i>
ED	4,8186	0,3304	0,2487
JaDE	4,6581	0,3186	0,2564
MVO	4,5675	0,3191	0,2585

Nas Figuras 9 e 10 podem ser observadas as curvas de convergência para cada função dos algoritmos MVO, ED e JaDE.

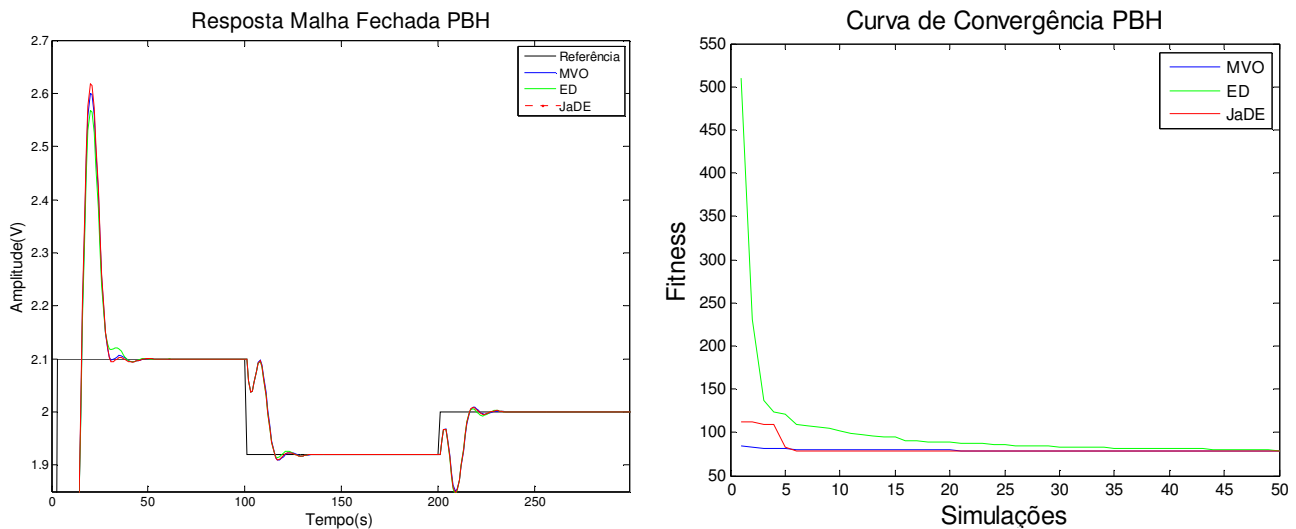


Figura 9. Resposta em malha fechada e curvas de convergência para o PBH

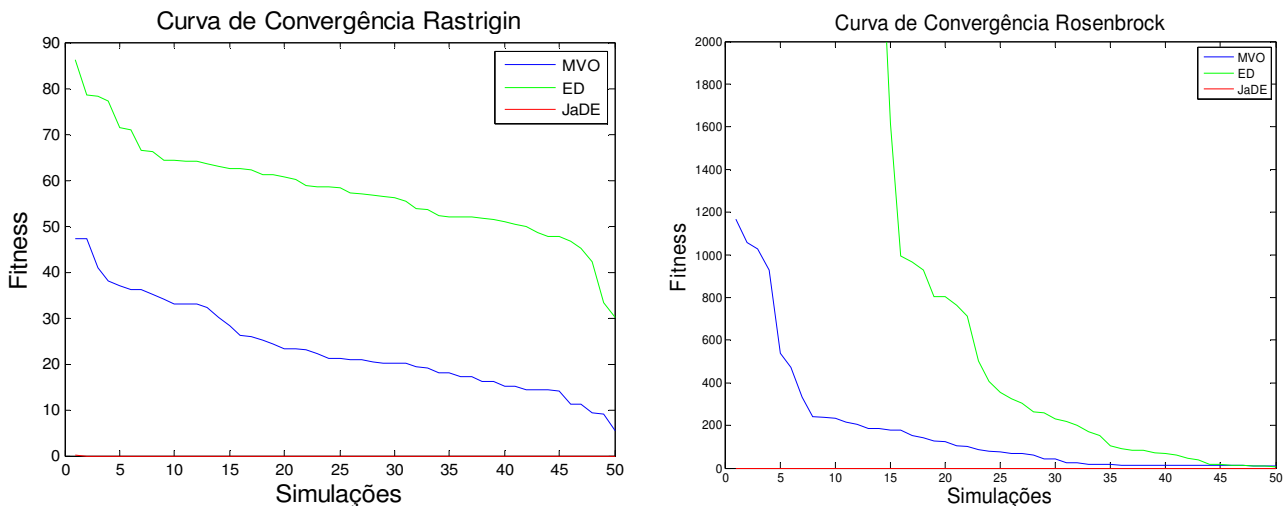


Figura 10. Curvas de convergência para Rastrigin e Rosenbrock

7. CONCLUSÃO

Neste trabalho compara-se o desempenho de três diferentes técnicas de otimização para três diferentes tipos de problemas. Foram obtidos bons resultados das técnicas ED e MVO, principalmente ao analisar o problema não linear do PBH na qual todas as técnicas ficaram muito próximas em relação ao valor da função objetivo.

Entretanto a técnica que apresentou um melhor desempenho geral foi a JaDE, atingindo o mínimo global nas funções Rastrigin (Tab. 1) e Rosenbrock (Tab. 2) e o menor valor do erro *ITAE* do sistema de controle para o PBH (Tab. 3).

Pela robustez e eficácia do algoritmo auto adaptativo JaDE, observadas nos testes realizados, para futuras referências é possível estender o trabalho, colocando o algoritmo da JaDE para competir com outras técnicas bem-sucedidas baseadas na ED, como por exemplo os algoritmos CoDE (Wang, Cai e Zhang, 2011) e EPSDE (Mallipeddi *et al.*, 2011).

8. REFERÊNCIAS

- Bianchi, E., Doppelbauer, G., Filion, L., Dijkstra, M., Kahl, G., 2014, "Predicting patchy particle crystals : variable box shape simulations and evolutionary algorithms". The Journal of Chemical Physics, vol. 136, no. 21, pp.
- Chiha, I., Ghabi, J., Liouane, N., 2012, "Tuning PID controller with multi-objective differential evolution". 5th International Symposium on Communications, Control and Signal Processing, Rome, Italy.
- Cho, H., Kim, D., Olivera, F., Guikema, S.D., 2011, "Enhanced speciation in particle swarm optimization for multi-modal problems". European Journal of Operational Research, vol. 213, no. 1, pp. 15–23.
- Chou, F., 2013, "Particle swarm optimization with cocktail decoding method for hybrid flow shop scheduling problems with multiprocessor tasks". International Journal of Production Economics, vol. 141, no. 1, pp. 137–145.
- Du Plessis, M.C., Engelbrecht, A.P., 2012, "Using competitive population evaluation in a differential evolution algorithm for dynamic environments". European Journal of Operational Research, vol. 218, no. 1, pp. 7–20.

- Guth, A., 2007, "Eternal inflation and its implications". *Journal of Physics A: Mathematical and Theoretical*, vol. 40:6811.
- Jarre, F., 2013, "On Nesterov's smooth Chebyshev – Rosenbrock function", *Optimization Methods & Software*, vol. 28, no. 3, pp. 478–484.
- Ma, H., Simon D., Fei, M., Chen, Z., 2013, "On the equivalences and differences of evolutionary algorithms. Engineering Applications of Artificial Intelligence", pp. 2397–2407.
- Mallipeddi, R., et al., 2011, "Differential evolution algorithm with ensemble of parameters and mutation strategies". *Applied Soft Computing*, vol. 11, no. 2, pp. 1679–1696.
- Mayer, D., Kinghorn, B., Archer, A., 2005, "Differential evolution: an easy and efficient evolutionary algorithm for model optimization". *Agricultural Systems*, vol. 83, no. 3, pp. 315–328.
- Mirjalili, S., Mirjalili, S.M., Hatamlou, A., 2016, "Multi-verse optimizer: a nature-inspired algorithm for global optimization". *Neural Computing and Applications*, vol. 27, no. 2, pp. 495–513.
- Mitchell, M., 1996, "An introduction to genetic algorithms". Massachusetts: MIT Press, pp 162.
- Piotrowski, A.P., Napiorkowski, J.J., Kiczko, A., 2012, "Differential evolution algorithm with separated groups for multi-dimensional optimization problems". *European Journal of Operational Research*, vol. 216, no. 1, pp. 33–46.
- Price, K., Storn, R., 1995, "Differential evolution – a simple and efficient adaptive scheme for global optimization over continuous spaces," ICSI, USA, Tech. Rep. TR-95-012, Berkeley, USA.
- Price, K., Storn, R., Lampinen, J., 2005, "Differential evolution: a practical approach to global optimization". Berlin: Germany, Springer-Verlag.
- Reynoso-Meza, G., Sanchis, J., Blasco, X., Freire, R. Z., 2016, "Evolutionary multi-objective optimisation with preferences for multivariable PI controller tuning". *Expert Systems with Applications*, vol. 51, pp. 120–133.
- Sathya, S.S., Radhika, M.V., 2013, "Convergence of nomadic genetic algorithm on benchmark mathematical functions". *Applied Soft Computing Journal*, vol. 13, no. 5, pp. 2759–2766.
- Shigemasa, T., Negishi, Y., Baba, Y., 2013, "A TDOF PID control system design by referring to the MD-PID control system and its sensitivities". *European Control Conference*, Switzerland, Zurich.
- Steinhardt, P., Turok, N., 2005 "The cyclic model simplified". *New Astronomy Reviews*, vol. 49, pp. 43–57.
- Wang, Y., Cai, Z., Zhang, Q., 2011, "Differential evolution with composite trial vector generation strategies and control parameters". *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 55–66.
- Zhang, J., Sanderson, A., 2007, "JADE self-adaptive differential evolution with fast and reliable convergence performance". *IEEE Congress on Evolutionary Computation (CEC)*, Singapore, p. 2251–2258.
- Zhang, J., Sanderson, A., 2009, "JADE: adaptive differential evolution with optional external archive". *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 5, pp. 945–958.
- Zhang, J., Sanderson, A., 2013, "Adaptive differential evolution: a robust approach to multimodal problem optimization". Berlin: Germany, Springer-Verlag, vol. 1, pp. 164.

9. RESPONSABILIDADE AUTORAL

Os autores são os únicos responsáveis pelo conteúdo deste trabalho.

OPTIMIZATION METAHEURISTICS APPLIED TO PID CONTROL TUNING FOR A HORIZONTAL SWING PROCESS

Marco Antonio Reichert Boaretto, marco.boaretto@hotmail.com¹

Edson Gnatkovski Gruska, edson.gruska@gmail.com²

Leandro dos Santos Coelho, lscoelho2009@gmail.com³

¹Pós-Graduação em Engenharia Elétrica (PPGEE), Universidade Federal do Paraná (UFPR), Curitiba, PR, Brasil

²Pós-Graduação em Engenharia de Produção e Sistemas (PPGEPS), Escola Politécnica, Pontifícia Universidade Católica do Paraná (PUCPR), Curitiba, PR, Brasil

³Pós-Graduação em Engenharia de Produção e Sistemas (PPGEPS), Escola Politécnica, Pontifícia Universidade Católica do Paraná (PUCPR) e Programa de Pós-Graduação em Engenharia Elétrica (PPGEE), Universidade Federal do Paraná (UFPR), Curitiba, PR, Brasil

Abstract. *In this paper, we propose the solution of three optimization problems with the self-adaptive approach of the Differential Evolution algorithm called JaDE. To prove its effectiveness, JaDE will compete with two other algorithms, the classical Differential Evolution and a cosmology-based algorithm called Multi-Verse Optimizer. One of the problems that will be solved is a real life control problem known as Horizontal Swing Process, as this process uses a PID (proportional, integral and derivative) controller, the optimization algorithms must find the good combination of parameters that outcomes the lowest ITAE (integral of time-weighted absolute error) for the control loop. The other two optimization problems are benchmark functions, both presented in the optimization literature using metaheuristics.*

Keywords: *Differential evolution; Multi-Verse Optimizer; JaDE; Tuning PID controller; Horizontal Swing.*