Trabalho apresentado no XXXIX CNMAC, Uberlândia - MG, 2019.

Proceeding Series of the Brazilian Society of Computational and Applied Mathematics

MOPSO com ciclos de buscas

Breno Tiago de Souza Mota[1]
Instituto Politécnico, UERJ, Nova Friburgo, RJ
Adriana da Rocha Silva[2]
Instituto Politécnico, UERJ, Nova Friburgo, RJ
Gilza Santos Simão Ferreira[3]
Instituto Politécnico, UERJ, Nova Friburgo, RJ

Resumo. Este trabalho tem por objetivo apresentar uma nova versão de algoritmo MOPSO para resolver problemas de Otimização multi-objetivo, utilizando a teoria de dominância de Pareto. Nessa nova proposta de algoritmo chamado de MOPSO-CB, a ideia central é utilizar ciclos de iterações com o peso de inércia fixo, dando assim, mais possibilidades de escolha de Líderes para uma mesma partícula e a capacidade de fugir de mínimos locais. Os experimentos computacionais foram feitos utilizando uma função teste e um problema de otimização de um Processo de Alquilação. Os resultados mostram um bom desempenho nos critérios de convergência e distribuição.

Palavras-chave. MOPSO, MOPSO-CB, ciclos de buscas, otimização, multiobjetivo

1 Introdução

Devido ao seu desempenho em problemas mono-objetivos, o PSO [6] foi modificado para atender problemas multi-objetivos. Proposto por Coello e Lechuga [3], o MOPSO (Multi-objective Particle Swarm Optimization) foi a primeira meta-heurística adaptada do PSO a solucionar problemas multi-objetivos utilizando o critério de dominância de Pareto [8].

Definição 1: um determinado vetor $\vec{\boldsymbol{u}} = (u_1, u_2, ..., u_k)$ diz-se dominar o vetor $\vec{\boldsymbol{w}} = (w_1, w_2, ..., w_k)$, e neste caso se escreve $\vec{\boldsymbol{u}} \preccurlyeq \vec{\boldsymbol{w}}$, se e somente se

$$u_i \leq w_i, \forall i \in \{1, 2, ..., k\} \text{ e } \exists i \in \{1, 2, ..., k\} \text{ tal que } u_i < w_i \tag{1}$$

Ao se comparar dois vetores $\vec{\boldsymbol{u}}$ e $\vec{\boldsymbol{w}}$, todo componente u_i atende à condição de ser menor e igual ao seu correspondente w_i. Após satisfeita essa primeira exigência, é necessário que exista pelo menos um elemento u_i que seja menor que seu equivalente w_i .

Henderson et al. [7], ao tratarem problemas de termodinâmica relacionados ao cálculo de pontos críticos, apresentaram uma nova estratégia de otimização ao PSO. Nessa versão,

[1]brenotsm1@gmail.com
[2]arsilva@iprj.uerj.br
[3]gilzasimao@hotmail.com

se utiliza a estratégia de decaimento não linear para ω e são acrescentas iterações internas com esse valor fixo. Apesar da mudança relativamente simples na essência da meta-heurística, os resultados empíricos apresentados mostraram a eficácia do algoritmo em fugir de mínimos locais em problemas altamente não lineares.

Este trabalho tem como objetivo apresentar uma nova estrutura de meta-heurística, adaptando a estratégia de [7] e tendo como estrutura de manipulação do Repositório o MOPSO de Coello et al. [4].

Na seção 2 é feita a descrição da estrutura do método proposto. Os problemas que foram usados nos experimentos são tratados na seção 3. As seções 4 e 5 são destinadas à exposição dos resultados e das conclusões, respectivamente.

2 MOPSO com ciclos de buscas

O método proposto utiliza a estratégia de decaimento não linear para ω [2]:

$$\omega = \left\{ \frac{(iter_{max} - iter)^q}{iter_{max}^q} \right\} (\omega_{inicial} - \omega_{final}) + \omega_{final} \tag{2}$$

em que q o índice de modulação não linear, $iter$ a iteração corrente, $iter_{max}$ é o número máximo de iterações. Esse valor ω afeta a equação que atualiza a velocidade de cada elemento do enxame:

$$\vec{v}_i(t+1) = \omega \cdot \vec{v}_i(t) + c_1 \cdot r_1 \cdot (\vec{p}_i(t) - \vec{x}_i(t)) + c_2 \cdot r_2 \cdot (\vec{g}_j(t) - \vec{x}_i(t)) \tag{3}$$

onde $\vec{v}_i$ é a velocidade atual; $\vec{x}_i$ a posição atual da partícula; r_1 e r_2 são termos aleatórios de uma distribuição uniforme no intervalo de $[0; 1]$; $\vec{p}_i$ é o vetor com a melhor posição (memória) já encontrada pela partícula até a iteração atual; e $\vec{g}$ é o vetor com a melhor posição já encontrada por uma partícula da vizinhança até a iteração atual (é o *Líder* que serve como componente social); c_1 e c_2 são coeficientes de aprendizado, constantes reais positivas.

Os valores de ω diminuem com o decorrer do algoritmo e proporcionam ao mesmo a capacidade de uma busca global no início e uma busca refinada localmente no fim [12]. Porém caso decaia de maneira abrupta essas buscas podem não ser tão eficazes. O pseudo-código do algoritmo é apresentado na Figura 1.

A diferença na estrutura do algoritmo está na etapa 6 (Figura 1). Nesta etapa um número de n "ciclos" é feito com ω fixo. Se o número de ciclos for igual a 1, o algoritmo se comporta exatamente como na versão de [3]. Na equação 3 para o cálculo da velocidade é utilizado um vetor líder ($\vec{g}$). Esse vetor é escolhido dentre os que estão no Repositório. Diferente da versão mono-objetivo do PSO, em que há apenas um líder por iteração para guiar o enxame, nos problemas multiobjetivo, com a utilização do arquivo externo (Repositório), existem NR líderes. Os ciclos de pesquisa possibilitam que o Repositório forneça um líder diferente a cada ciclo, para uma mesma partícula, antes que Repositório seja atualizado no passo 18. Com ω fixo o objetivo é buscar melhores posições dentro de uma região de busca, fugindo de mínimos locais em problemas altamente não lineares, antes que o processo de otimização continue.

DOI: 10.5540/03.2020.007.01.0405

1. Inicializar o Enxame.
2. Avaliar as partículas não dominadas.
3. Inicializar o Repositório com as partículas não dominadas.
4. **para** $iter$ **de** 1 **até** $iter_{max}$ **passo** $iter = iter + 1$, **fazer**
5. | Atualizar o valor de ω.
6. | **para** k **de** 1 **até** n **passo** $k = k + 1$, **fazer** $\{Ciclos\ de\ Buscas\}$
7. | | **para** i **de** 1 **até** NP **passo** $i = i + 1$, **fazer**
8. | | | Selecionar o Líder $\vec{g}$.
9. | | | Calcular a Velocidade $\vec{v}_i$.
10. | | | Atualizar a Posição $\vec{x}_i$.
11. | | | Avaliar a posição da partícula.
12. | | | **se** $(F(\vec{x}_i) \preccurlyeq F(\vec{p}_i))$, **então**
13. | | | | $\vec{p}_i = \vec{x}_i$
14. | | | **fim se**
15. | | **fim para**
16. | **fim para**
17. | Avaliar as partículas não dominadas.
18. | Atualizar o Repositório.
19. **fim para**
20. Retornar as soluções do Repositório.

Figura 1: pseudo-código do algoritmo MOPSO-CB.

3 Problemas testes

Para análise do algoritmo foi utilizada uma função teste e um problema da engenharia química.

Teste 1: Minimizar $T_1(x_1, x_2) = [f_1(x_1), f_2(x_1, x_2)]$ definida como [5]:

$$\begin{aligned} f_1(x_1) &= x_1 \\ g(x_2) &= 2 - \exp\left\{-\left(\frac{x_2 - 0{,}2}{0{,}004}\right)^2\right\} - 0{,}8\exp\left\{-\left(\frac{x_2 - 0{,}6}{0{,}4}\right)^2\right\} \\ f_2(x_1, x_2) &= \frac{g(x_2)}{x_1} \end{aligned} \tag{4}$$

onde $0,1 \leq x_1 \leq 1,0$ e $0,1 \leq x_2 \leq 1,0$. A função $g(x_2)$ é bi-modal, com um mínimo local e um global. Por sua vez, esse mínimo global sendo muito difícil de ser encontrado devido à configuração da própria função, torna a função T_1 multi-modal.

Teste 2:

Um das fases mais importante no processo de refinação do petróleo é o processo de alquilação. Essa etapa se caracteriza quando uma das olefinas leves, tais como propeno, buteno ou penteno reage com isobutano na presença de um catalisador de ácido sulfúrico forte para produzir o produto alquilado [9]. Sauer et al. [10], desenvolveram um modelo para este processo com base em uma combinação judiciosa de princípios fundamentais,

DOI: 10.5540/03.2020.007.01.0405

equações empíricas e uma série de hipóteses simplificadoras. Como resultado, o modelo resultante tem 10 variáveis e sete restrições de igualdade.

Bracken e McCormick [1] reformularam e apresentaram este modelo e o problema de otimização de uma maneira diferente. Ao notarem que quatro restrições de igualdade, derivadas por análise de regressão, não precisam ser satisfeitas exatamente, eles as converteram em oito restrições de desigualdade.

Problema: O problema bi-objetivo, do processo de alquilação, pode ser definido em maximizar o Lucro ($\$/dia$) e maximizar o Número (Índice) de Octano (x_7). A importância de um produto alquilato com um maior número de octano se da pelo fato de que este produto se torna melhor para misturar com produtos de refinaria. A função objetivo para este problema é definida [9]:

$$\begin{array}{lll} f_1: & Lucro & = 0,063x_4x_7 - 5,04x_1 - 0,035x_2 - 10,0x_3 - 3,36x_5 \\ f_2: & \text{N. } Octano & = x_7 \end{array} \tag{5}$$

com relação a x_1, x_7 e x_8 sujeito às seguintes restrições e equivalências, $0,0 \leq x_1 \leq 2000,0$; $90,0 \leq x_7 \leq 95,0$; $3,0 \leq x_8 \leq 12,0$; $0,0 \leq x_4 \equiv x_1(1,12 + 0,13167x_8 - 0,006667x_8^2) \leq 5000,0$; $0,0 \leq [x_5 \equiv 1,22x_4 - x_1] \leq 2000,0$; $0,0 \leq [x_2 \equiv x_1x_8 - x_5] \leq 16000,0$; $85,0 \leq x_6 \equiv 89.0 + \frac{(x_7-(86,35+1,098x_8-0,038x_8^2))}{0,325} \leq 93,0$; $145,0 \leq [x_{10} \equiv -133,0 + 3x_7] \leq 162,0$; $1,2 \leq [x_9 \equiv 35,82 - 0,222x_{10}] \leq 4,0$ e $0,0 \leq x_3 \equiv 0,001\frac{(x_4x_6x_9)}{(98,0-x_6)} \leq 120,0$.

As unidades das variáveis do problema são [9]: x_1 é a alimentação de Olefina ($barris/dia$), x_2 o isobutano reciclado ($barris/dia$), x_3 a taxa de adição de ácido ($mil\ libras/dia$), x_4 a taxa de produção de alquilado, x_5 a alimentação de isobutano ($barris/dia$), x_6 a força do ácido gasto (*porcentagem em peso*), x_7 o número de octano, x_8 o isobutano para olefina, x_9 o fator de diluição de ácido, x_{10} o número de desempenho f-4.

4 Resultados e análise

As métricas de desempenho, espaçamento (SP) [11] (valores iguais a zero significam que todas as soluções na Frente de Pareto são equidistantes, logo, estão bem distribuídas entre si) e distância geracional (GD) [13] (permite observar se o algoritmo converge, logo, todos os pontos pertencem à frente de Pareto global), evidenciam a convergência, diversidade e uniformidade de distribuição das frentes de Pareto.

Em ambos os testes se utilizou uma população (NP) de 100 elementos, um repositório (NR) com no máximo 100 elementos, 30 divisões em cada dimensão, c_1 e c_2 iguais a $2,05$, $\omega_{inicial}$ de $0,9$, ω_{final} de $0,4$, com q sendo $1,2$ e o número de ciclos internos n igual a 6. Na primeira função teste se usou 100 iterações (a mesma quantidade de [4]), no segundo problema o número de iterações foi de 250.

A Tabela 1 compara os resultados do MOPSO com busca cíclica (MOPSO-CB) proposto nesse trabalho, aos resultados obtidos pelo MOPSO com a estrutura normal, ambos aplicados a função teste 1. O dados para a comparação são provenientes de [4]. Os resultados mostram um melhor desempenho por parte do MOPSO-CB nos dados estatísticos.

Tabela 1: Comparação entre os métodos.

	SP		GD	
	MOPSO-CB	MOPSO	MOPSO-CB	MOPSO
Melhor	0,01777	0,04007	0,00570	0,00043
Pior	0,09023	0,58185	0,06151	0,18531
Média	0,03611	0,08358	0,01891	0,03227
Mediana	0,03360	0,05494	0,00668	0,00051
Desv, Pad.	0,01221	0,11821	0,01921	0,06062

O problema Teste 2 aborda a maximização do Lucro (L), e do número de Octano, x_7. A função do Lucro resolvida como um problema mono-objetivo tem valor ótimo igual 1162.02 ($/dia$) [9], com a variável x_7 igual a $94,19$, dessa forma, qualquer acréscimo em x_7 provocará um decréscimo no lucro. Uma frente de Pareto ideal seria aquela em que as soluções possibilitem aumentar o número de Octano com o mínimo de perda ao lucro.

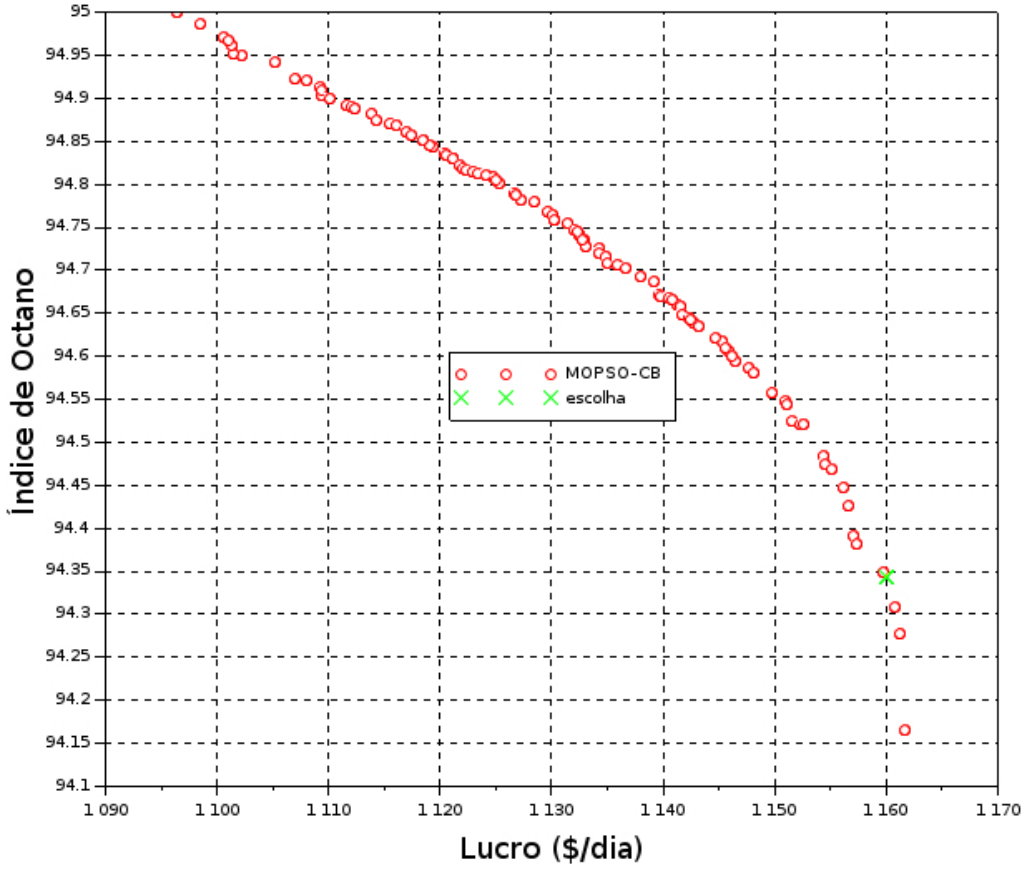

Figura 2: Frente de Pareto do processo de alquilação.

O resultado gráfico (Figura 2) mostra a frente de Pareto encontrada pelo método

MOPSO-CB para o problema. O ponto escolhido foi de $1160,02$ ($\$/dia$) para o Lucro e $94,3433$ para o número de Octano. Ao analisar os resultados, nota-se que foi possível aumentar em aproximadamente $19,0\%$ o índice de octano (em comparação a solução de $94,19$ e o limite superior de 95) com uma diminuição no lucro de apenas $1,98$ ($\$/dia$). A solução para esse ponto é: $x_1 = 1727,8$, $x_2 = 15999,4$, $x_3 = 101,203$, $x_4 = 3055$, $x_5 = 1999,3$, $x_6 = 91,089$, $x_7 = 94,3433$, $x_8 = 10,4171$, $x_9 = 2,51336$ e $x_{10} = 150,03$. Fisicamente essa solução propicia que o produto tenha maior capacidade de se misturar com outras substâncias.

No caso específico do segundo problema teste, a redução no número de varáveis livres é definida pelos autores [10] na formulação do problema. Caso fossem considerados as restrições de igualdade, ambos os métodos (MOPSO e MOPSO-CB) teriam que ser adaptados para tratar desse tipo de restrição, oque foge ao objetivo deste trabalho.

5 Conclusões

Neste trabalho foi apresentado uma nova versão para o algoritmo PSO Multi-objetivo, o MOPSO-CB. A diferença central na estrutura do algoritmo proposto está em manter o peso de inércia (ω) fixo por n ciclos internos, permitindo mais possibilidades de escolha para a seleção do líder, antes de atualizar o Repositório.

Para avaliar o desempenho, realizou-se uma comparação com outra versão desse mesmo algoritmo. Foi utilizada uma função teste multi-modal com grande dificuldade de otimização, uma vez que o método é destinado a esse tipo de problema, e também foi aplicado em um problema de engenharia química. Os resultados da função teste mostram que o MOPSO-CB teve um ótimo desempenho nas métricas de convergência e distribuição da frente de Pareto. Além disso, obteve uma ótima solução para o problema de Processo de Alquilação bi-objetivo.

Como trabalhos futuros, deseja-se aplicar o método em outros problemas reais altamentes não lineares e com maiores dimensões.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) com código de financiamento 001.

Referências

[1] J. Bracken e G. P. Mccormick. *Selected applications of nonlinear programming*, Research Analysis Corporation, Mclean, 1968.

[2] A. Chatterjee and P. Siarry. Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Computers & operations research*, 33:859-871, 2006. DOI:10.1016/j.cor.2004.08.012.

[3] C. C. Coello and M. S. Lechuga. MOPSO: A proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation*, 2:1051-1056, 2002. DOI: 10.1109/CEC.2002.1004388.

[4] C. A. C. Coello, G. T. Pulido and M. S. Lechuga. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on evolutionary computation*, 8:256-279, 2004. DOI:10.1109/TEVC.2004.826067.

[5] K. Deb. Multi-objective genetic algorithms: Problem difficulties and construction of test problems. *Evolutionary Computation*, 7:205–230, 1999. ISSN: 1063-6560

[6] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 39-43, 1995. DOI: 10.1109/MHS.1995.494215.

[7] N. Henderson, J. Sartori and W. F. Sacco. Phase stability analysis using a polarization technique and the randomness of a stochastic method in an unconstrained optimization framework. *Industrial & Engineering Chemistry Research*, 53:3342–3352, 2014. DOI: 10.1021/ie402819r

[8] V. Pareto. *Cours d'économie politique, vol. I-II*, F. Rouge, Lausanne, 1896.

[9] G. P. Rangaiah. Multi-objective optimization: techniques and applications in chemical engineering. *World Scientific*, 2009. DOI: 10.1142/7088.

[10] R. N. Sauer, A. R. Colville and C. W. Burwick. Computer points way to more profits. *Hydrocarbon Processing*, v. 84, 1964. (to appear)

[11] J. R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Dissertação de mestrado, MIT, 1995.

[12] Y. Shi e R. C. Eberhart. Parameter selection in particle swarm optimization. In: *International conference on evolutionary programming*. Springer, 1998. DOI: 10.1007/BFb0040810.

[13] D. A. Van Veldhuizen and G. B. Lamont. Multiobjective evolutionary algorithm research: A history and analysis. Technical Report TR-98-03, Air Force Institute of Technology, 1998.

