

Implementação de algoritmo NSGA-III com operador de estimativa de distribuição

Francirley Resendes Borges Costa¹, José Robson Mariano Alves¹

¹Instituto Federal de Ciência e Tecnologia do Tocantins (IFTO)

Abstract. *This work presents the optimization algorithm for many objectives NSGA-III (Genetic Algorithm of Unclassified Classification) with the evolutionary estimation distribution algorithm (EDA). In the work development was proposed a way for obtaining and preserving solutions, guided by reference points. As selection operator was chosen by EDA with creation of probabilistic models using normal distribution around the projection distances between the individuals of the search space and reference points. The results and comparisons with the original implementations of the NSGA-III and the MOEAD/PBI are presented in the scope of the paper, a proposal was inferior to the results defined by the idealizers of the other methods. At the end a critical analysis is carried out, as well as proposals for improvements for future work.*

Resumo. *Este trabalho apresenta o algoritmo de otimização de muitos objetivos NSGA-III (Non-dominated Sorting Genetic Algoritm) com o método de seleção evolucionário de estimativa de distribuição (EDA). No desenvolvimento do trabalho foi proposto um mecanismo de obtenção e preservação de soluções, orientado por pontos de referência. Como operador de seleção foi escolhido o EDA com criação de modelos probabilísticos usando distribuição normal em torno das distâncias de projeção entre os indivíduos do espaço de busca e pontos de referência. No decorrer do trabalho são apresentados os resultados e comparações com as implementações originais do NSGA-III e MOEAD/PBI, a proposta mostrou-se inferior frente aos resultados apresentados pelos idealizadores dos outros métodos. Ao final uma análise crítica é realizada, bem como propostas de melhorias para trabalhos futuros.*

1. Introdução

Métodos de otimização evolucionária multi/muitos objetivos (*Evolutionary multi/many-objective optimization* - EMO) surgiram no início dos anos noventa e desde então é crescente a quantidade de pesquisadores nessa área. Impulsionado cada vez mais pela evolução dos computadores, sua popularização também aconteceu na área industrial com o aparecimento de softwares EMO comerciais. No cotidiano os métodos EMO tem demonstrado cada vez mais capacidade para resolver problemas com muitos objetivos. Os métodos de otimização multi/muitos objetivos têm basicamente dois conceitos principais: diminuir a distância entre a frente não dominada e a frente pareto-ótimo e encontrar um conjunto de soluções diversificadas. Já os algoritmos genéticos/evolucionários (AEs) buscam solucionar os problemas de seleção e geração das soluções de forma a garantir uma busca eficaz para o conjunto pareto-ótimo. O foco principal das pesquisas com EMO é buscar maneiras de garantir a manutenção de soluções distintas e mais próximas da

fronteira pareto-ótimo durante o processo evolucionário, o que mantém um bom espalhamento das soluções [Barbosa et al. 2010]. Com base na característica dinâmica dos processos evolucionários, este trabalho propõe a implementação de um EMO fundamentado no NSGA-III [Deb et al. 2005] com operador de geração de soluções EDA. Nesse contexto, o algoritmo evolucionário NSGAEDA foi desenvolvido. Na seção II é apresentada uma revisão da literatura sobre alguns dos principais algoritmos de EMO. Na seção III a estrutura do algoritmo proposto e suas características são apresentadas. Na seção IV são apresentados os problemas usados para testes com o algoritmo proposto. Na seção V os resultados obtidos são confrontados e analisados em comparação com os algoritmos NSGA-III e MOEA/D/PBI. Por fim, na sessão VI são apresentadas as conclusões e uma análise crítica do modelo proposto, bem como melhorias não exploradas para trabalhos futuros.

2. Algoritmos de otimização multi/muitos objetivos

Nesta seção são apresentados dois algoritmos EMO: NSGA-III (*Non-dominated Sorting Genetic Algorithm*) e o MOEA/D [Deb and Jain 2014], ambos são ditos como estado da arte em métodos de otimização multiobjetivo.

2.1. NSGA-III

Non-Dominated Sorting Genetic Algorithm (NSGA III) [Deb et al. 2005], proposto para tentar diminuir a complexidade computacional pela técnica de classificação por não-dominância, que faz o elitismo com base em frentes de soluções ditas não-dominadas. O algoritmo NSGA III (Figura 1) é uma evolução do NSGA-II e trabalha com uma população filha, criada a partir de uma população pai inicial, utiliza um operador de seleção/cruzamento/mutação e mantém sua diversidade por meio de um mecanismo de preservação de nichos que busca soluções próximas da projeção criada por pontos de referência.

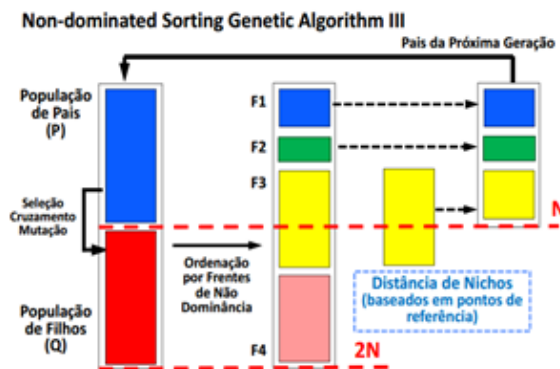


Figura 1. Algoritmo NSGA-III

Os pontos de referência buscam dar a direção para manter a diversidade das soluções de busca e também podem dar suporte para convergência do algoritmo. O próprio usuário pode gerar os pontos de referência, bem como eles podem ser gerados com base em um ponto ideal balanceando os objetivos [Deb et al. 2005].

Para a preservação de soluções é criado um mecanismo de preservação de nichos que através da distância de projeção do espaço normalizado das soluções para o ponto

de referência, associa cada indivíduo da população a um desses pontos. Em seguida os nichos com o menor número de soluções associadas são favorecidos [Deb et al. 2005].

2.2. MOEA/D

O algoritmo MOEA/D (*Multiobjective Evolutionary Algorithm Based in Decomposition*) utiliza a técnica de decompor um problema em subproblemas. O MOEA/D decompõe o problema em N subproblemas escalares de otimização, resolvendo-os simultaneamente. Isto é feito a partir da utilização de um conjunto de N vetores de peso, destinados a obtenção e manutenção de um conjunto diverso de soluções "trade-off" em paralelo. Inicialmente, cada membro da população (N) é associado a um dos vetores de ponderação aleatoriamente [Deb and Jain 2014]. Para tornar o problema escalar, o algoritmo utiliza-se de métodos de agregação de funções que podem ser uma simples soma ponderada. A soma ponderada torna um problema multiobjetivo de minimização em um problema escalar, conforme:

$$\min g^{soma}(x|\lambda) = \sum_{i=1}^{nf} \lambda_i f_i(x) \quad (1)$$

Onde g^{soma} é a nova função a ser otimizada, λ é o vetor de pesos associados aos objetivos, f , do problema multiobjetivo. A única condição imposta aos pesos é a de que $\sum_1^{nf} \lambda_i = 1$. As relações de vizinhança entre os subproblemas são definidas pelas distâncias euclidianas entre os vetores de ponderação. Para cada geração, as soluções associadas aos vetores vizinhos são usadas para geração de novas soluções, conforme a Figura 2.

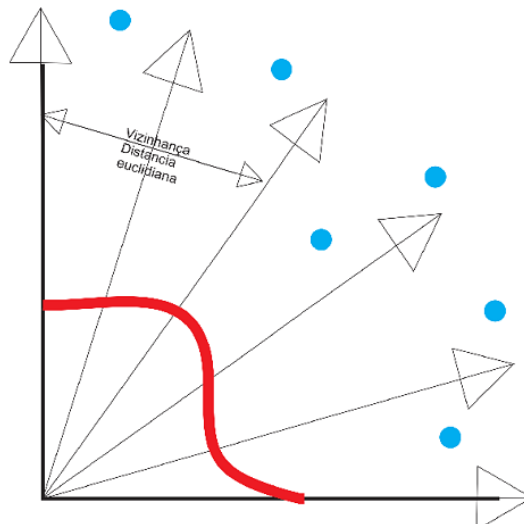


Figura 2. Relação de vizinhança

O Algoritmo 1 mostra o procedimento utilizado pelo MOEA/D para otimizar uma população de soluções.

Algoritmo 1: Algoritmo MOEA/D**início**

1. Defina $EP = \emptyset$.
2. Calcula a distância euclidiana entre dois vetores de peso e retorna aqueles T vetores de peso mais próximos de cada vetor. Para cada $B(i) = i_1, \dots, i_T$ onde $\lambda^{i_1}, \dots, \lambda^{i_T}$ são os T vetores de peso associados a λ^i .
3. Gera a população inicial x^1, \dots, x^N randomicamente, e defina $FV^i = F(x^i)$.
4. Inicializa o ponto ideal $z = \{z_1, \dots, z_m\}^T$ (menor valor de objetivo em cada direção).

para $i=1$ até N **faça**

5. Reprodução: Randomicamente elege dois índices k e l de $B(i)$ e gera uma nova solução y de x^k e x^l por uso de operadores genéticos.
6. Atualiza ponto ideal z : Para cada $j = 1, \dots, m$ se $z_j < f_i(y)$, define $z_j = f_i(y)$
7. Atualiza soluções de vizinhança: Para cada índice $j \in B(i)$ se $g^{PBI}(y|\lambda^j, z) \leq g^{PBI}(y|x^j, z)$, então define $x^j = y$ e $FV^j = F(y)$
8. Atualiza EP: Remove todos os vetores dominados por $F(y)$ e adiciona $F(y)$ se nenhum vetor em EP domina $F(y)$

fim**fim**

O algoritmo inicializa o arquivo de soluções não-dominadas EP e o ponto ideal z . Em N gerações, para cada solução em P , dois indivíduos pais x_n^1 e x_n^2 são selecionados da vizinhança e uma nova solução y é gerada através de operadores genéticos/evolucionários. Avaliado o objetivo y , o ponto ideal z é atualizado com os menores valores para cada objetivo encontrado até o momento. Em seguida a vizinhança é atualizada substituindo por y os indivíduos que possuem valor de função de agregação maior que o novo indivíduo, considerando o seu respectivo valor de pesos. O arquivo de soluções não dominadas (EP) é atualizado removendo as soluções não dominadas, adicionando y se esta é não dominada. Por fim o processo é repetido [Deb and Jain 2014].

3. Algoritmo Proposto NSGAEDA

O algoritmo proposto baseia-se na estrutura do NSGA-III e usa um operador de estimativa de distribuição para gerar os filhos para a próxima geração. A proposta cria a nova geração de filhos através de um operador que usa estimativa de distribuição. O operador EDA proposto cria modelos probabilísticos com base em cada ponto de referência, calculando a distância de projeção entre os " $n = 4$ " (valor com melhores resultados) indivíduos mais próximos, por fim, com a média e desvio padrão obtidos pelo modelo é aplicada uma distribuição normal para gerar um novo filho.

3.1. Operador EDA

Para criar os modelos probabilísticos é necessária a informação de distância de projeção de todos os indivíduos, logo, como é proposto no NSGA-III, o algoritmo comporta-se como ilustra a Figura 3.

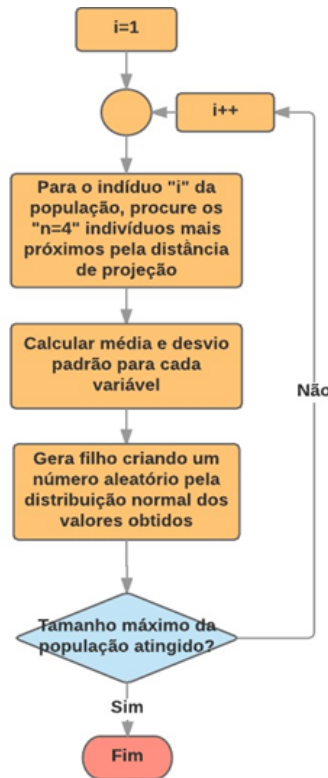


Figura 3. Operador EDA

Depois de gerados os filhos, uma perturbação chamada de mutação gaussiana é aplicada para ajudar na convergência do algoritmo. Aleatoriamente é gerado um valor entre o limite mínimo e máximo do espaço de busca do problema, em seguida, para cada valor de variável j de cada indivíduo X_i da população, verifica-se se o valor gerado é maior que o valor X_{ij} , se sim, o valor da variável em questão é definido como média e define-se um desvio padrão de $\frac{1}{l}$ sendo que l varia de 20 até 1000, de acordo com a evolução das gerações, quanto maior o valor de l , maior é a perturbação na variável, por fim, a mutação é aplicada.

3.2. Mutação Gaussiana

Dado $X_{ij} \in [a, b]$, a mutação gaussiana M_G é calculada por: $M_G(x) = \min(\max(N(x_{ij}, \sigma), a), b)$ onde $\sigma = \frac{1}{l}$ mede o tamanho da perturbação, a e b são os limites superiores e inferiores do espaço de busca e x_{ij} é o valor da variável j do indivíduo X_i .

3.3. Preservação de nichos

Para a preservação de nichos, após calcular a distância de projeção para cada ponto de referência, um vetor com a quantidade de soluções associadas é criado para cada ponto de referência, assim, para cada nicho/ponto de referência que não possui uma solução associada a ele, é escolhida a com menor distância ortogonal.

3.4. Distância de projeção

A distância de projeção d ou projeção ortogonal é calculada por:

$$d(s, w) = \left\| \frac{s - w^t s w}{\|w\|^2} \right\| \quad (2)$$

Onde s é o valor da função objetivo e w o ponto de referência, ambos no espaço normalizado. Assim é possível definir qual o ponto de referência mais próximo para cada solução, como ilustra a Figura 4.

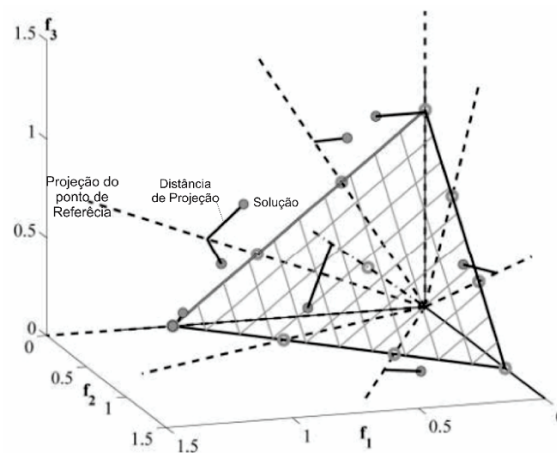


Figura 4. Distância de projeção

4. Resultados

Para medir os resultados do algoritmo proposto foram utilizados os testes de benchmark DTLZ1 e DTLZ2 [Deb et al. 2005] com 3 e 5 objetivos bem conhecidos da literatura. Além disso, o algoritmo foi comparado com o NSGA-III e MOEA/D-PBI, com os mesmos requisitos de testes, apresentados em [Deb et al. 2005]. Primeiramente são apresentados os gráficos com a diversidade de distribuição das soluções sobre a fronteira pareto. Para comparar o desempenho foi usado o IGD (*Inverted Generational Distance*) que demonstra o quão distante os elementos da fronteira pareto real estão da fronteira pareto estimada produzida pelo algoritmo [Zhang and Li 2007]

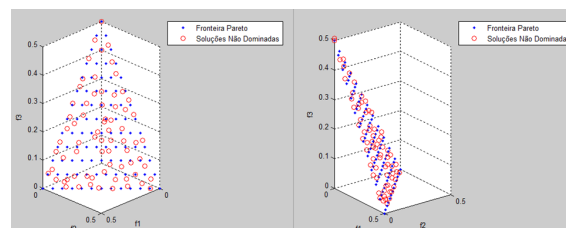


Figura 5. DTLZ1 com 3 objetivos

Como pode ser observado nas figuras 5, 6, 7 e 8, o algoritmo conseguiu uma boa diversidade, tanto nos problemas com 3 quanto nos com 5 variáveis, porém no DTLZ1 não houve uma maior homogeneidade na diversidade como houve no DTLZ2.

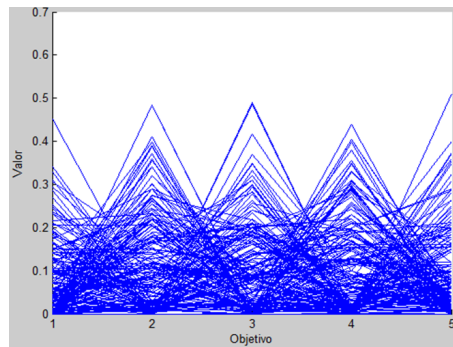


Figura 6. DTLZ1 com 5 objetivos

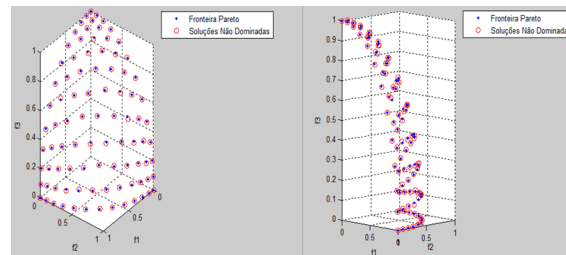


Figura 7. DTLZ2 com 3 objetivos

Os valores de IGD (tabela 1) obtidos demonstram o desempenho do algoritmo em alcançar a fronteira pareto-ótimo, percebe-se que esses valores não conseguiram superar os algoritmos do estado da arte, porém demonstraram certa estabilidade quanto a variação dos resultados obtidos entre melhor, média e pior caso.

5. Conclusão

A proposta de utilização de EDA como operador de geração de soluções obteve resultados levemente inferiores quanto ao estado da arte, porém comportou-se com certa estabilidade quanto as soluções obtidas. Percebeu-se também que para o problema DTLZ2 o resultado obtido foi próximo do estado da arte. Durante o processo de desenvolvimento do problema foram realizados testes com o uso de outros operadores de geração de soluções como o *Diferencial Evolution* (DE) e o Cruzamento Binário Simulado (*Simulated Binary Crossover* - SBX), com os quais melhores resultados foram obtidos, provavelmente pela abordagem do operador, assim percebe-se que este ainda pode ser refinado. Para geração de modelos probabilísticos, investigar outras soluções também poderia mostrar resultados melhores, por exemplo, identificar melhor as soluções vizinhas usando proximidade vetorial, ou verificar dominância entre as soluções e identificar a direção em relação ao ponto de referência, podendo assim definir pontos que circundam os pontos de referência. Com base nos resultados, recomenda-se para trabalhos futuros uma abordagem de geração de prole com um algoritmo híbrido, visto que EDA, DE e SBX comportam-se de forma diferente para problemas de mesma natureza.

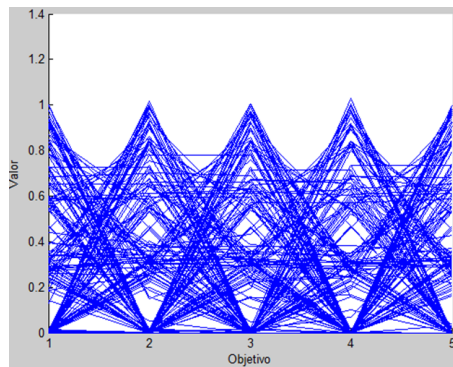


Figura 8. DTLZ2 com 5 objetivos

Tabela 1. Valores de Avaliação IGD

DTLZ	IGD	MOEAD/DD	NSGA-III	EADNS
DTLZ1 m = 3	Melhor	3,19E-04	4,88E-04	2.64E-02
	Média	5,84E-04	1,31E-03	2.77E-02
	Pior	6,57E-04	4,88E-03	2.94E-02
DTLZ1 m = 5	Melhor	2,63E-04	5,12E-04	5,82E-02
	Média	2,92E-04	9,79E-04	6,15E-02
	Pior	3,11E-04	1,98E-03	6,34E-02
DTLZ2 m = 3	Melhor	6,66E-04	1,26E-03	5,42E-03
	Média	8,07E-04	1,36E-03	6.01E-03
	Pior	1,24E-03	2,11E-03	6.46E-03
DTLZ2 m = 5	Melhor	1,13E-03	4,25E-03	4,91E-02
	Média	1,29E-03	4,98E-03	5,42E-02
	Pior	1,42E-03	5,86E-03	6,44E-02

Referências

- Barbosa, A. M., de Carvalho Ribeiro, L., and de Oliveira Arantes, J. M. (2010). Algoritmo genético multiobjetivo: sistema adaptativo com elitismo. *Citado*, 2:16.
- Deb, K. and Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints. *IEEE Trans. Evolutionary Computation*, 18(4):577–601.
- Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005). *Scalable test problems for evolutionary multiobjective optimization*. Springer.
- Zhang, Q. and Li, H. (2007). Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731.