

Algoritmo híbrido MOEAD/PSO + EDA para solução de problemas com muitos objetivos

Jésus J. S. Santos¹ and Clayton Almeida¹ Breno C. de Oliveira¹

Universidade Federal de Minas Gerais, Av. Antônio Carlos, 6627 - Belo Horizonte -
MG,
jesusjss@ufmg.br

Resumo Existe disponível na literatura uma grande variedade de Algoritmos Evolucionários e Determinísticos para otimização multiobjetivo. Porém a otimização de muitos objetivos ($n > 3$) ainda é um campo em exploração com muitas oportunidades de desenvolvimentos tanto para melhoria das soluções encontradas quanto em custo computacional para obtenção dos resultados. Neste sentido o presente trabalho apresenta uma abordagem de aplicação da evolução das soluções através da metodologia de PSO (*particle swarm optimization*) realizando a Decomposição das soluções e obtenção da Fronteira de Pareto a partir do algoritmo MOEA-D (*Multiobjective Evolutionary Algorithm Based on Decomposition*) e busca local utilizando-se um EDA (*Estimation of distribution algorithm*) a fim de auxiliar a convergência.

Keywords: Otimização, Multiobjetivo, Enxame de Partículas, Estimativa de distribuição, Muitos Objetivos

1 Introdução

Os Algoritmos evolucionários (AEs) têm-se tornado uma importante ferramenta para resolver problemas de otimização multiobjetivo, devido a sua capacidade de lidar com funções genéricas gerando, em uma única execução, um conjunto de estimativas de soluções para o problema em questão ([1], [2]).

Um importante representante da classe dos algoritmos evolucionários é o Algoritmo Genético (AG) [3]. Os algoritmos genéticos são caracterizados pela existência de pelo menos três operadores genéticos: um operador de cruzamento que combina a informação contida em dois ou mais indivíduos para gerar outros indivíduos; um operador de mutação que utiliza a informação contida em um indivíduo para, estocasticamente, gerar outro indivíduo; e um operador de seleção que é utilizado para gerar a nova população através de réplicas de alguns indivíduos e eliminação de outros.

Um AG pode ser construído a partir desses três operadores ou pode conter operadores adicionais tais como elitismo, nicho, busca local, etc. A escolha dos tipos de operadores que serão utilizados para compor um AG, bem como o ajuste dos parâmetros associados a eles, deve ser feita levando-se em conta a classe de problemas que se pretende resolver. Portanto, o desempenho de um

AG é bastante influenciado pelos operadores que o compõem e pelos valores dos parâmetros a eles associados.

1.1 Algoritmos Evolucionários Multiobjetivo

Dentre os algoritmos evolucionários para otimização multiobjetivo, destacam-se os algoritmos genéticos. Estes são caracterizados por operadores de cruzamento e mutação dos indivíduos da população, e um mecanismo probabilístico de seleção que favorece as melhores soluções até o momento.

No contexto da otimização multiobjetivo, a diferença básica entre um algoritmo genético multiobjetivo e sua versão para otimização escalar, reside na forma de atribuir a aptidão dos indivíduos.

Descreveremos, brevemente, os algoritmos evolucionários multiobjetivo estado da arte para otimização até 2 objetivos.

NSGA II: Non-Dominated Sorting Genetic Algorithm II Na década de 90 surge um interessante método de atribuição de aptidão para algoritmos genéticos multiobjetivo: a ordenação por fronteiras não-dominadas que consiste em ordenar os indivíduos da população em termos de fronteiras não dominadas no espaço de objetivos e atribuir um valor de aptidão para cada fronteira não dominada. Nascia então o NSGA. Entretanto, o NSGA clássico apresentava alguns problemas práticos, o principal deles a complexidade computacional da ordenação por fronteiras não dominadas. Além disso, o algoritmo era não elitista. Por essas razões, no começo dessa década os autores do NSGA propuseram uma versão melhorada denominada de NSGA II [4]. As principais mudanças no NSGA II são a criação de um algoritmo de ordenação rápida de menor complexidade computacional do que a ordenação por fronteiras não dominadas original, a utilização de elitismo e a eliminação do parâmetro de nicho, através do uso de outra estratégia de medida de dispersão.

SPEA 2: Strength Pareto Evolutionary Algorithm 2 A segunda geração dos algoritmos genéticos multiobjetivo começou quando o elitismo tornou-se um mecanismo padrão dos algoritmos. Apesar de alguns estudos preliminares relacionados com a noção de elitismo nos algoritmos evolucionários multiobjetivo, a introdução formal desse conceito se deve a Eckart Zitzler através da publicação do algoritmo *Strength Pareto Evolutionary Algorithm* (SPEA) em [5]. Após a publicação desse artigo, os novos algoritmos incorporaram o mecanismo de elitismo. Na prática, o uso do elitismo é uma exigência teórica para garantir a convergência dos algoritmos evolucionários multi-objetivo.

Posteriormente, foram identificadas algumas limitações importantes referentes ao SPEA e uma versão mais sofisticada foi proposta em seguida, o SPEA2 [6]. SPEA 2 utiliza um arquivo contendo soluções não-dominadas previamente encontradas. A cada geração, soluções não-dominadas são armazenadas nesse arquivo. Para cada indivíduo nesse arquivo, um valor de *força* (strength) é calculado. Esse valor de *força* é similar à classificação dos indivíduos adotada pelo MOGA,

uma vez que ele é proporcional ao número de soluções que um determinado indivíduo domina. A função de aptidão de cada indivíduo é calculada de acordo com as *forças* de todas as soluções não-dominadas do arquivo que domina esse determinado indivíduo e por todas as soluções que o indivíduo domina. O processo do cálculo da aptidão do SPEA 2 considera ao mesmo tempo proximidade do conjunto Pareto real e distribuição das soluções. Desta forma, no lugar de técnicas de nicho baseadas em distância, o critério de dominância é usado para garantir que as soluções estejam distribuídas ao longo do conjunto de Pareto real (ressalva-se que uma técnica de “clusterização” continua sendo empregada para limitar o tamanho do conjunto de estimativas Pareto-ótimas armazenadas).

1.2 Otimização por Enxame de Partículas

O PSO foi proposto por Kennedy e Eberhart em 1995 [7], sendo inspirado no comportamento social de organismos biológicos, mais precisamente na habilidade de algumas espécies de animais de trabalhar em conjunto para localizar boas regiões com fontes de alimento, assim como ocorre em enxames de abelhas ou bandos de pássaros [8]. Trata-se então de uma meta-heurística baseada em populações que produzem soluções competitivas em muitos domínios de aplicação conforme pode ser visto em ([9] e [10]).

As partículas são caracterizadas pela sua posição e velocidade, onde sua posição refere-se a localização no espaço de soluções e sua velocidade representa a mudança de posição ([11] e [12]). Uma partícula utiliza as posições do líder global selecionado (que representa a inteligência social do bando) e sua própria trajetória de movimento pessoal (que representa a inteligência cognitiva individual) afim de atualizar os valores de velocidade e posição. As equações a baixo (1) e (2) demonstram esta atualização de posição e velocidade:

$$v_i(t+1) = w * v_i(t) + C_1 * r_1 (x_{pbest_i} - x_i(t)) + C_2 * r_2 (x_{lbest_i} - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

Onde:

$pbest_i$: é referente à melhor posição já visitada pelo grupo ao qual a partícula i pertence;

$lbest_i$: é referente à melhor posição já visitada pela partícula i ;

r_1 e r_2 : vetores de valores aleatórios normalmente distribuídos;

c_1 e c_2 : são os fatores de aprendizado.

1.3 Problemas de Otimização com Muitos Objetivos

Os algoritmos evolutivos multiobjetivos otimizam simultaneamente duas ou mais funções objetivo, conseguindo encontrar um conjunto de soluções em um única execução do algoritmo. Recentemente, há um crescente interesse na aplicação de

AEMOs (Algoritmos Evolucionários Multiobjetivos) para resolver problemas de otimização com muitos objetivos, aqueles que tratam de quatro ou mais objetivos.

Desta forma, dentre os AEMOs conhecidos, o MOEA-D (MultiObjective Evolutionary Algorithm based on Decomposition) [13] tem destacado-se por lidar bem com problemas com até dez critérios. Atualmente, MOEA-D é um dos melhores AEMOs para muitos objetivos, apresentando alta capacidade de busca, bem como grande eficiência computacional. Por exemplo, o MOEA-D trabalha adequadamente com quatro ou mais objetivos. Esse algoritmo também pode obter soluções bem distribuídas na fronteira de Pareto usando um número de vetores com peso, com diferentes direções em funções de ponderação escalar.

No entanto, no trabalho de Ishibuchi [14], foi verificado que o seu desempenho em problemas multiobjetivos com objetivos altamente correlacionados é prejudicado, enquanto que os algoritmos NSGA-II e SPEA2 apresentam poucos efeitos negativos quando tratam de objetivos com alta similaridade. Também na literatura não há resultados de sucesso do MOEA-D para problemas com mais de dez objetivos. Por fim, o MOEA-D requer o pré-conhecimento das soluções ótimas de cada problema mono-objetivo que compõe o problema de muitos objetivos para calcular distâncias de cada solução aos extremos da fronteira.

Neste contexto, o presente trabalho desenvolve uma abordagem híbrida com a utilização do MOEA-D evoluindo as partículas de acordo com a metodologia baseada em PSO e realiza uma busca local aplicando-se a estimativa da distribuição de probabilidades das variáveis de forma semelhante ao algoritmo CMAES [15] que utiliza um modelo Gaussiano para evoluir a matriz de covariâncias a fim de encontrar as melhores soluções.

2 Descrição do algoritmo proposto

Como pode ser visto na figura 1 o algoritmo proposto aplica uma abordagem híbrida com a utilização do MOEA-D evoluindo as partículas de acordo com a metodologia baseada nas equações (1) e (2) e realiza uma busca local aplicando-se a estimativa da distribuição de probabilidades das variáveis correlacionadas através de uma matriz de covariâncias.

Para a obtenção das soluções a partir da aplicação da busca local por EDA, executam-se os seguintes passos:

- Recebe N pontos não dominados;
- Calcula as médias e a matriz de Covariâncias;
- Realiza a decomposição de cholesky;
- Gera os novos indivíduos considerando pdf normal;
- Retorna o indivíduo com melhor valor de função.

A busca local utilizando-se o EDA foi executada durante 25% das gerações, isto é entre a geração 550 até a geração 800 para um problema de 1000 gerações, como por exemplo o DTLZ-3.

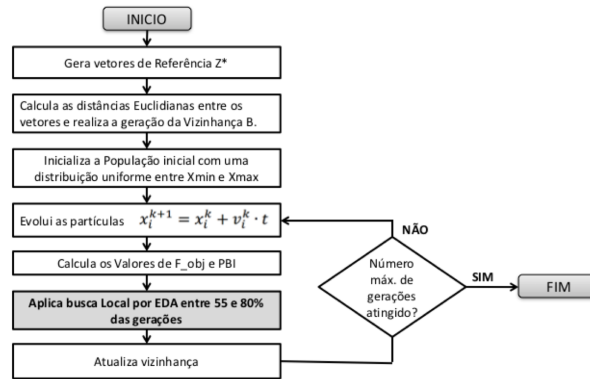


Figura 1: Fluxograma Simplificado do Algoritmo

3 Apresentação dos problemas solucionados

Neste trabalho serão utilizados os problemas com muitos objetivos apresentados por Deb, Thiele, Laumanns e Zitzler que podem ser vistos em [16] para a avaliação de desempenho do algoritmo proposto aqui.

Os autores destes problemas sugerem o uso de diferente números de variáveis e objetivos de acordo com cada problema, sendo $n = M + k + 1$, onde M é o número de objetivos e k depende de cada problema. Para o problema DTLZ-1 é indicado $k = 5$, para o problema DTLZ-7 é recomendado $k = 20$ e para os problemas restantes é sugerido $k = 10$ ([17]).

Foram utilizados três problemas de teste neste trabalho: DTLZ-1, DTLZ-2 e DTLZ-3. Que serão descritos a seguir:

- **DTLZ-1** - Apresenta uma Fronteira Pareto linear e multimodal, sendo que a principal característica deste problema explorada para testes de benchmarking é a dificuldade de convergência para a Fronteira Pareto, podendo estagnar em ótimos locais;
- **DTLZ-2** - Apresenta uma Fronteira Pareto esférica, neste caso a concavidade é explorada nesta função;
- **DTLZ-3** - Apresenta uma Fronteira Pareto esférica e multimodal, de forma que o problema DTLZ3 é uma modificação do problema DTLZ-2 somando-se a característica de multimodalidade imposta pela função DTLZ-1. Essa modificação torna o problema DTLZ-3 um desafio para os algoritmos em termos de convergência e mapeamento da Fronteira Pareto.

4 Apresentação dos resultados

A seguir serão apresentados os resultados das simulações do algoritmo proposto para a solução dos problemas DTLZ de 1 a 3. As tabelas mostram as médias, melhores e piores casos para cinco execuções do método para cada problema.

A tabela 1 a seguir apresenta os tempos médios e desvio padrão para 100 execuções realizadas, observa-se que os maiores tempos computacionais são advindos dos problemas com cinco variáveis, sendo o problema DTLZ-3 com cinco variáveis o de maior custo computacional conforme esperado pela sua complexidade.

Tabela 1: Tempos Médios para as Soluções dos Problemas

PROBLEMA	Nº OBJETIVOS	TEMPO MÉDIO (s)	DESVIO PADRÃO (%)
DTLZ-1	3	55,672	1,0
DTLZ-1	5	373,024	1,7
DTLZ-2	3	41,4729	0,8
DTLZ-2	5	222,6628	1,2
DTLZ-3	3	127,395	2,0
DTLZ-3	5	973,224	3,7

Outro aspecto importante a ser observado nos resultados é a cardinalidade, que apresenta o total de pontos que estão presentes na Fronteira Pareto. Como pode ser visto na tabela 2 o pior caso obteve 92% de pontos presentes na frente de soluções não dominadas.

Tabela 2: Cardinalidade para as Soluções dos Problemas

PROBLEMA	Nº OBJETIVOS	CARDINALIDADE
DTLZ-1	3	91
DTLZ-1	5	210
DTLZ-2	3	85
DTLZ-2	5	208
DTLZ-3	3	84
DTLZ-3	5	206

A tabela 3 e o gráfico 3 apresentam os resultados para o problema DTLZ-1 com 3 objetivos. Observa-se que o algoritmo encontrou a solução obtendo uma Fronteira Pareto bem distribuída e com ótima cardinalidade em razoável tempo computacional. A ótima distribuição pode ser atribuída a existência de pontos de Referência pelo MOEA/D que auxilia a obtenção de pontos ao longo de toda superfície da Fronteira Pareto.

A seguir estão a tabela 4 e a figura 3 que apresentam os resultados para o problema DTLZ-1 de 5 objetivos. Pode-se observar que a solução deste problema apresentou bons resultados com estabilidade na ordem de grandeza de $10 - 2$ para pior caso, caso médio e melhor caso. Além de apresentar ótima cardinalidade (100% dos pontos) e boa distribuição da Fronteira de Pareto.

Tabela 3: Resultado DTLZ 1 para 3 Objetivos

ANÁLISE	MOEA-DD	NSGAIII	PROPOSTO
Pior Caso	6,57E-04	4,88E-03	1.31e-02
Caso Médio	5,84E-04	1,31E-03	1.07e-02
Melhor Caso	3,19E-04	4,88E-04	2.61e-03

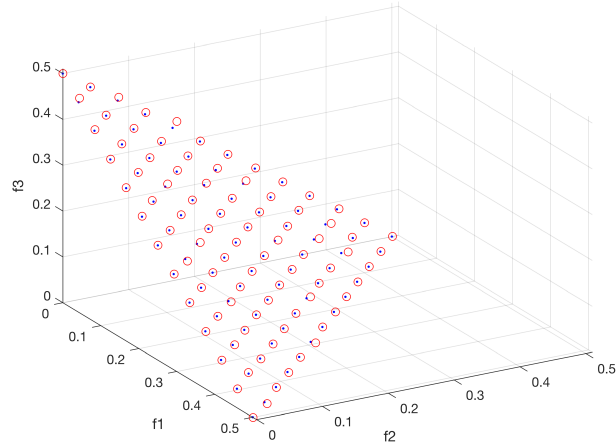


Figura 2: Resultado DTLZ 1 para 3 Objetivos

Tabela 4: Resultado DTLZ 1 para 5 Objetivos

ANÁLISE	MOEA-DD	NSGAIII	PROPOSTO
Pior Caso	3,11E-04	1,98E-03	3,53E-02
Caso Médio	2,92E-04	9,79E-04	3,25E-02
Melhor Caso	2,63E-04	5,12E-04	3,06E-02

Os resultados para o problema DTLZ-2 com 3 objetivos estão representados na tabela 5 e na figura 4. Pode-se observar boa estabilidade na ordem de grandeza de 10^{-2} para pior caso, caso médio e melhor caso. Além de apresentar boa cardinalidade (85 de 91 dos pontos) e boa distribuição da Fronteira de Pareto.

Tabela 5: Resultado DTLZ 2 para 3 Objetivos

ANÁLISE	MOEA-DD	NSGAIII	PROPOSTO
Pior Caso	1,24E-03	2,11E-03	4,54E-02
Caso Médio	8,07E-04	1,36E-03	4,23E-02
Melhor Caso	6,66E-04	1,26E-03	3,59E-02

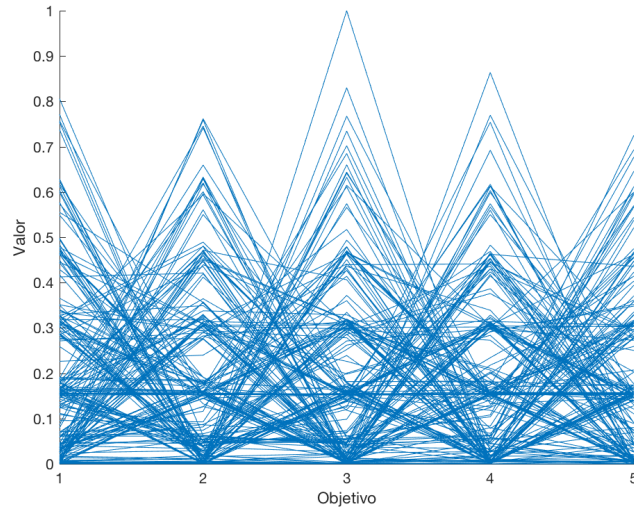


Figura 3: Resultado DTLZ 1 para 5 Objetivos

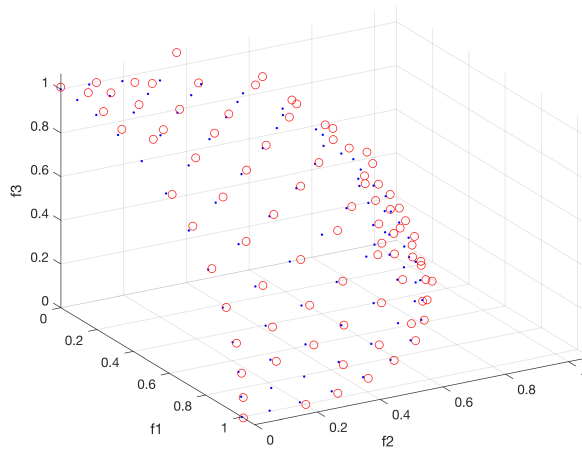


Figura 4: Resultado DTLZ 2 para 3 Objetivos

A seguir estão a tabela 6 e a figura 5 que apresentam os resultados para o problema DTLZ-2 de 5 objetivos. Pode-se observar que a solução deste problema apresentou bons resultados com estabilidade na ordem de grandeza de 10^{-2} para pior caso, caso médio e melhor caso. Além de apresentar ótima cardinalidade (208 de 210 dos pontos) e boa distribuição da Fronteira de Pareto.

Tabela 6: Resultado DTLZ 2 para 5 Objetivos

ANÁLISE	MOEA-DD	NSGAIII	PROPOSTO
Pior Caso	1,42E-03	5,86E-03	1,03E-01
Caso Médio	1,29E-03	4,98E-03	9,84E-02
Melhor Caso	1,13E-03	4,25E-03	9,45E-02

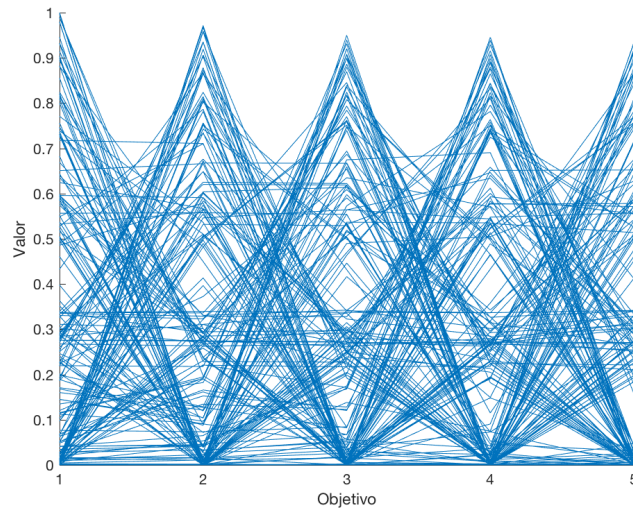


Figura 5: Resultado DTLZ 2 para 5 Objetivos

A tabela 7 e o gráfico 7 apresentam os resultados para o problema DTLZ-3 com 3 objetivos. Apesar deste problema ser um desafio maior que os outros dois problemas anteriores, observa-se que o algoritmo encontrou a solução obtendo uma Fronteira Pareto bem distribuída e com boa cardinalidade (84 de 91 pontos) em tempo computacional compatível com a complexidade do problema. Além de encontrar uma boa distribuição da superfície da Fronteira Pareto.

Tabela 7: Resultado DTLZ 3 para 3 Objetivos

ANÁLISE	MOEA-DD	NSGAIII	PROPOSTO
Pior Caso	6,23E-03	6,67E-03	3,83E-02
Caso Médio	1,89E-03	4,01E-03	3,09E-02
Melhor Caso	5,69E-04	9,75E-04	2,59E-02

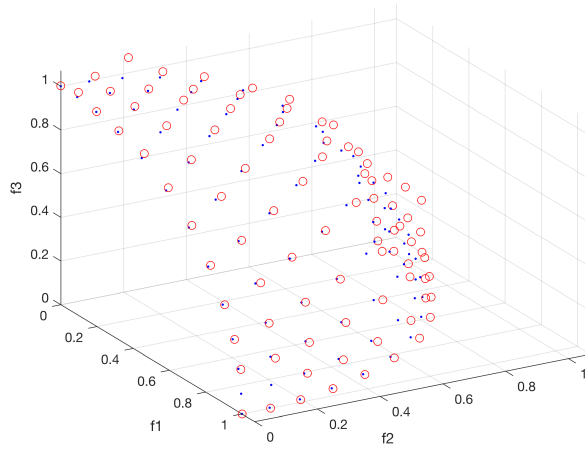


Figura 6: Resultado DTLZ 3 para 3 Objetivos

A seguir estão a tabela 8 e a figura 7 que apresentam os resultados para o problema DTLZ-3 de 5 objetivos. Pode-se observar que a solução deste problema apresentou bons resultados com ligeira instabilidade com média na ordem de grandeza de 10^{-1} e melhor caso em 10^{-2} . Além de apresentar ótima cardinalidade (206 de 210 pontos) e boa distribuição da Fronteira de Pareto com uniformidade do gráfico de linhas. O tempo computacional foi elevado, porém trata-se do problema mais oneroso para solução com o maior número de objetivos.

Tabela 8: Resultado DTLZ 3 para 5 Objetivos

ANÁLISE	MOEA-DD	NSGAIII	PROPOSTO
Pior Caso	4,74E-03	1,20E-02	8,21E+00
Caso Médio	1,18E-03	5,96E-03	9,10E-01
Melhor Caso	6,18E-04	3,09E-03	8,59E-02

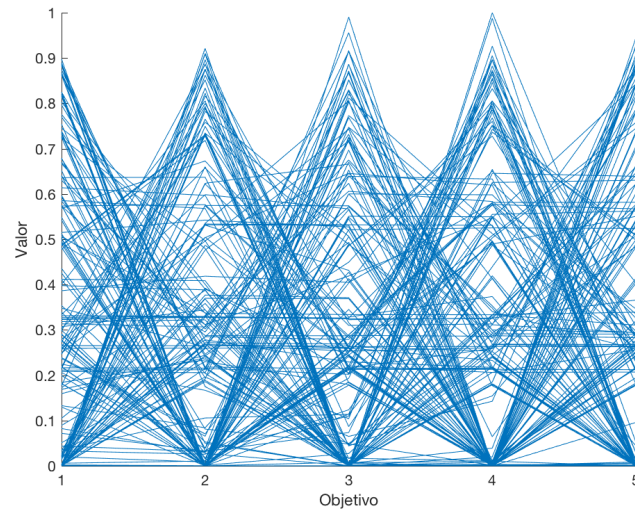


Figura 7: Resultado DTLZ 3 para 5 Objetivos

5 Conclusão

O método proposto alcançou bons resultados, próximos aos *benchmarks*. O mecanismo sugerido, a partir da aplicação de EDA como busca local, gerou ótimos resultados em comparação as execuções sem este método.

Observa-se que o algoritmo tem potencial de melhoria, principalmente em ajustes de parâmetros e sintonia da estratégia de Busca Global Vs. Busca Local e Convergência.

Nota-se também o aumento significativo do tempo computacional com o número de objetivos, o que já era esperado. Porém existem oportunidades de otimização no método que podem ser implementadas para redução do tempo computacional.

Como trabalho futuro, pretende-se aplicar este método a um problema real e realizar a sintonia dos parâmetros a fim de reduzir o tempo computacional para cada tipo de problema tratado.

Referências

1. C. A. Coello Coello A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques, Knowledge and Information Systems, vol. 1, no. 3, pp. 269-308, 1999.
2. C. M. Fonseca and P. J. Fleming An Overview of Evolutionary Algorithms in Multi-objective Optimization, Evolutionary Computation, vol. 7, no. 3, pp. 205-230, 1995.
3. T. Back Evolutionary Algorithms in Theory and Practice, Oxford University Press, 1996.

4. K. Deb and A. Pratap and S. Agarwal and T. Meyarivan A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II, anGAL Report No. 200001, 2000.
5. E. Zitzler and L. Thiele Multiobjective evolutionary algorithms: A comparative case study and the strenght Pareto approach, *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 4, pp. 257-271, 1999.
6. E. Zitzler and M. Laumanns and L. Thiele SPEA 2: Improving the Strenght Pareto Evolutionary Algorithms, UROGEN 2001. *Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems*, pp. 95-100, 2002.
7. Kennedy, J., Eberhart, R. Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks IV*. pp. 1942-1948, 1995.
8. Bratton, D., and Kennedy, J. Defining a standard for particule swarm optimization, In *IEEE Symposium on Swarm Intelligence, SIS' 2007*, pages 120–127, 2007.
9. Z. Wang, G. L. Durst, R. C. Eberhart, D. B. Boyd and Z. B. Miled, Particle swarm optimization and neural network application for QSAR, *18th International Parallel and Distributed Processing Symposium, 2004. Proceedings.*, Santa Fe, NM, USA, 2004, pp. 194-.
10. Jaishia, B., and Ren, W, Finite element model updating based on eigenvalue and strain, *Mechanical Systems and Signal Processing*, Volume 21, Issue 5, Pages 2295-2317, ISSN 0888-3270 21(5):2295–2317, 2007.
11. Reyes, S. M., and Coello, C. C., Multi-objective particle swarm optimizers: A survey of the state-of-the-art, *International Journal of Computational Intelligence Research*, 2(3):287–308, 2006.
12. Kennedy, J., Eberhart, R., and Shi, Y., *Swarm intelligence*, San Mateo, CA: Morgan Kaufmann, 2001.
13. Zhang, Q. and Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. Evol Comput.*, vol. 11, no. 6, pp. 712–731, Dez. 2007.
14. H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: A short review, *IEEE Congress on Evolutionary Computation*, páginas 2419–2426, Jun. 2008.
15. Nikolaus Hansen The CMA Evolution Strategy: A Tutorial, *CoRR*, vol. abs/1604.00772, 2016.
16. Figueiredo, E. M. N., *Algoritmo Baseado em Enxame de Partículas para Otimização de Problemas com Muitos Objetivos*, Dissertação de mestrado, Recife, PE, 2013.
17. Strickler, A., Deb, K., Thiele, L., Laumanns, M., e Zitzler, E., Scalable multi-objective optimization test problems. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, volume 1, páginas 825–830.