

UTILIZAÇÃO DA PLATAFORMA PLATEMO NA RESOLUÇÃO DE PROBLEMAS DE OTIMIZAÇÃO MULTI OBJETIVO: ESTUDO DE CASO DA META-HEURÍSTICA DA ÁGUA-VIVA

USING THE PLATEMO PLATFORM TO SOLVE MULTI OBJECTIVE OPTIMIZATION PROBLEMS: A CASE STUDY OF THE JELLYFISH METAHEURISTIC

Frederic Menezes Ferreira¹

Alexandre Cesar Muniz de Oliveira²

Resumo: Muitos problemas de otimização do mundo real são caracterizados por vários objetivos conflitantes, e são conhecidos como Problemas de Otimização Multiobjetivo. Nas últimas décadas, algoritmos evolutivos e outras meta-heurísticas têm mostrado desempenho promissor na solução aproximada de vários desses Problemas de Otimização Multiobjetivo, e um grande número de meta-heurísticas têm surgido. Em 2017, foi proposta uma plataforma evolutiva de otimização multiobjetivo chamada de PlatEMO. Este trabalho apresenta um estudo de caso para a plataforma PlatEMO, com a inclusão de uma meta-heurística multiobjetivo baseada em Água-viva (Jellyfish metaheuristic), que é uma das mais recentes meta-heurísticas de Inteligência de Enxame para otimização global. Em seguida, uma suíte de algoritmos é usada para demonstrar o potencial da plataforma como facilitador para a experimentação e análise de resultados.

Palavras-chave: Meta-heurística. Otimização. PlatEMO.

Abstract: Many real-world optimization problems are characterized by multiple conflicting objectives and are known as Multi-Objective Optimization Problems. In the last decades, evolutionary algorithms and other metaheuristics have shown promising performance in the approximate solution of many of these Multi-objective Optimization Problems, and many metaheuristics have emerged. In 2017, an evolutionary multi-objective optimization platform called PlatEMO was proposed. This work presents a case study for the PlatEMO platform, with the Jellyfish-based multi-objective metaheuristics inclusion, one of the most recent Swarm Intelligence metaheuristics for global optimization. Then, a suite of algorithms is used to demonstrate the platform's potential as a facilitator for experimentation and analysis of results.

Keywords: Metaheuristics. Optimization. PlatEMO.

1 - Graduado em Ciências da Computação pela Universidade Federal do Maranhão (UFMA-2022), está cursando o Mestrado em Ciência da Computação também na Universidade Federal do Maranhão (UFMA-2023).
Lattes: <http://lattes.cnpq.br/5595399705074880>. ORCID: <https://orcid.org/0000-0001-9539-9808>.
E-mail: frederic.menezes@ufma.br.

2 - Graduado em Ciência da Computação pela Universidade Federal do Maranhão; mestrado em Engenharia Elétrica pela mesma universidade; e doutorado em Computação Aplicada pelo Instituto Brasileiro de Pesquisas Espaciais (INPE - 2004). Atua na UFMA desde 1994, alcançando o cargo de Professor Titular em 2021, tendo explorado Fundamentos da Computação, Inteligência Artificial e Pesquisa Operacional em sua carreira acadêmica. Assumiu temporariamente a posição de Visiting Senior Lecturer na School of Computing da University of Kent, Reino Unido, em 2017, por um ano, pesquisando Meta-learning e Meta-heurísticas aplicadas a problemas de otimização. Em 2022, iniciou uma posição de pesquisa de pós-doutorado no Insight Center for Data Analytics na University College Cork (UCC), investigando Sistemas de Circuito Fechado de Simulação e Otimização aplicados a Sistemas de Manufatura Flexível.
Lattes: <http://lattes.cnpq.br/5225588855422632>. ORCID: <https://orcid.org/0000-0002-1499-5307>.
E-mail: alexandre.cesar@ufma.br.

Introdução

Muitos problemas de otimização do mundo real são Multiobjetivos. Nesse tipo de problema, geralmente esses objetivos são conflitantes, ou seja, a melhoria em um causa prejuízo no desempenho de pelo menos um dos outros objetivos (CHUGH et al., 2019). Problemas assim são conhecidos pela sigla MOPs, abreviação de *Multiobjective Optimization Problems*, ou, em português, Problemas de Otimização Multiobjetivo.

Devido à sua natureza conflitante, encontrar a solução desses problemas (MOPs) não é uma tarefa trivial. Nas últimas décadas, algoritmos evolutivos, e outras meta-heurísticas, têm alcançado grande sucesso na resolução desses MOPs (HUA et al., 2021), atraindo a atenção da comunidade acadêmica. Esses algoritmos são conhecidos como Algoritmos Evolucionários Multiobjetivos (*Multiobjective Evolutionary Algorithms* - MOEAs).

Para determinar o algoritmo (MOEA) mais adequado para um problema (MOP) específico, geralmente são necessários a realização de experimentos para comparar o desempenho de vários MOEAs candidatos neste MOP. Para facilitar a comparação dos MOEAs, uma plataforma para otimização multiobjetivo evolutiva, denominada PlatEMO foi proposta por Tian et al. (2017).

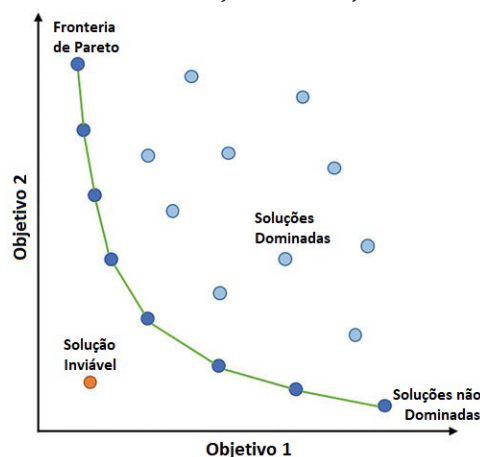
Essa plataforma é totalmente desenvolvida na linguagem MATLAB e contém mais de 200 MOEAs de última geração e mais de 400 MOPs de referências populares (TIAN et al., 2017). Plataformas como PlatEMO foram desenvolvidas para concentrar em um só lugar MOEAs, MOPs e as métricas usadas nas comparações entre os conjuntos *Pareto Front* (PF).

O objetivo deste trabalho é realizar um estudo sobre o PlatEMO, tendo como estudo de caso a adição do Algoritmo de Água-viva Multi-Objetivo (*Multi-Objective Jellyfish Search* - MOJS) (CHOU; TRUONG, 2020), avaliando o desempenho desse algoritmo quando comparado a outros MOEAs já disponíveis no PlatEMO, a partir de métricas como PF, Hiper-volume (HV), Distância Geracional (*Generational Distance* - GD) e Espaçamento (*Spacing* - SP).

Fundamentação teórica

Comparados aos problemas de objetivo único, os problemas multiobjetivos são mais difíceis de resolver, porque não há uma solução única. Em vez disso, há um conjunto de soluções ótimas de compensação aceitáveis. Este conjunto é chamado de Frente de Pareto ou Fronteira Pareto-ótima (Figura 1). Tomando dois objetivos como exemplo, soluções não dominadas formam uma fronteira chamada Fronteira de Pareto que separa as soluções inviáveis na região inferior esquerda das soluções dominadas na região superior direita.

Figura 1. Fronteira de Pareto na otimização multiobjetivo



Fonte: (LIU; IJZERMAN; WESTEN, 2021).

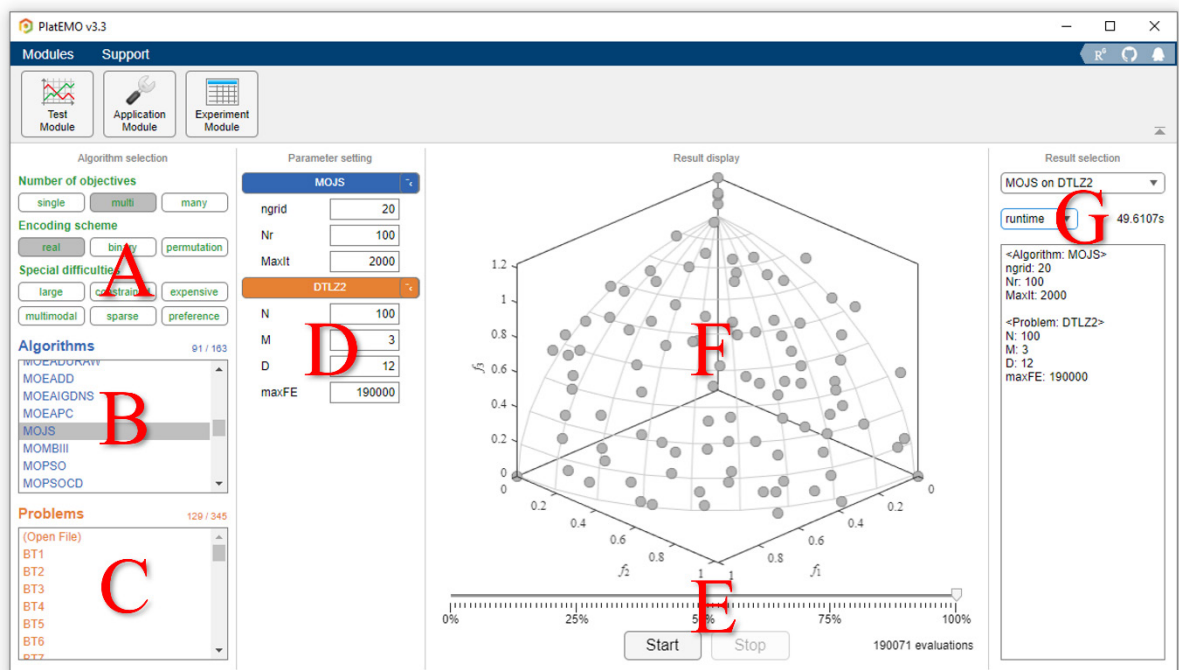
Meta-heurísticas multiobjetivos podem ser usadas para encontrar essas soluções. Elas são heurísticas de alto nível utilizadas na solução de uma ampla gama de problemas de otimização (DOKEROGLU et al., 2019). Recentemente, muitos algoritmos meta-heurísticos estão sendo aplicados com sucesso para resolver problemas intratáveis. Neste trabalho, o MOJS (CHOU; TRUONG, 2020) foi comparado com três MOEAs: NSGA-II; MOPSO; e MOEA/D.

Para comparar essas meta-heurísticas utilizamos o PlatEMO, que é uma plataforma, open source, para otimização multiobjetivo escrita em MATLAB (TIAN et al., 2017) que permite aos usuários comparar facilmente vários algoritmos e coletar resultados estatísticos em arquivos Excel ou LaTeX.

Metodologia

Os usuários podem chamar o PlatEMO de duas formas: podem usar a Interface Gráfica (GUI) do PlateEMO (Figura 2) chamando a função principal sem nenhum parâmetro ou podem chamar essa mesma função passando parâmetros como: o problema de otimização (MOP) e um algoritmo (MOEA), além de poder passar outros parâmetros. Por exemplo: `platemo('algorithm', @MOJS, 'problem', @DTLZ2, 'N', 100, 'M', 3, 'D', 12, 'maxFE', 190000, 'save', 10);`

Figura 2. Interface Gráfica do PlatEMO



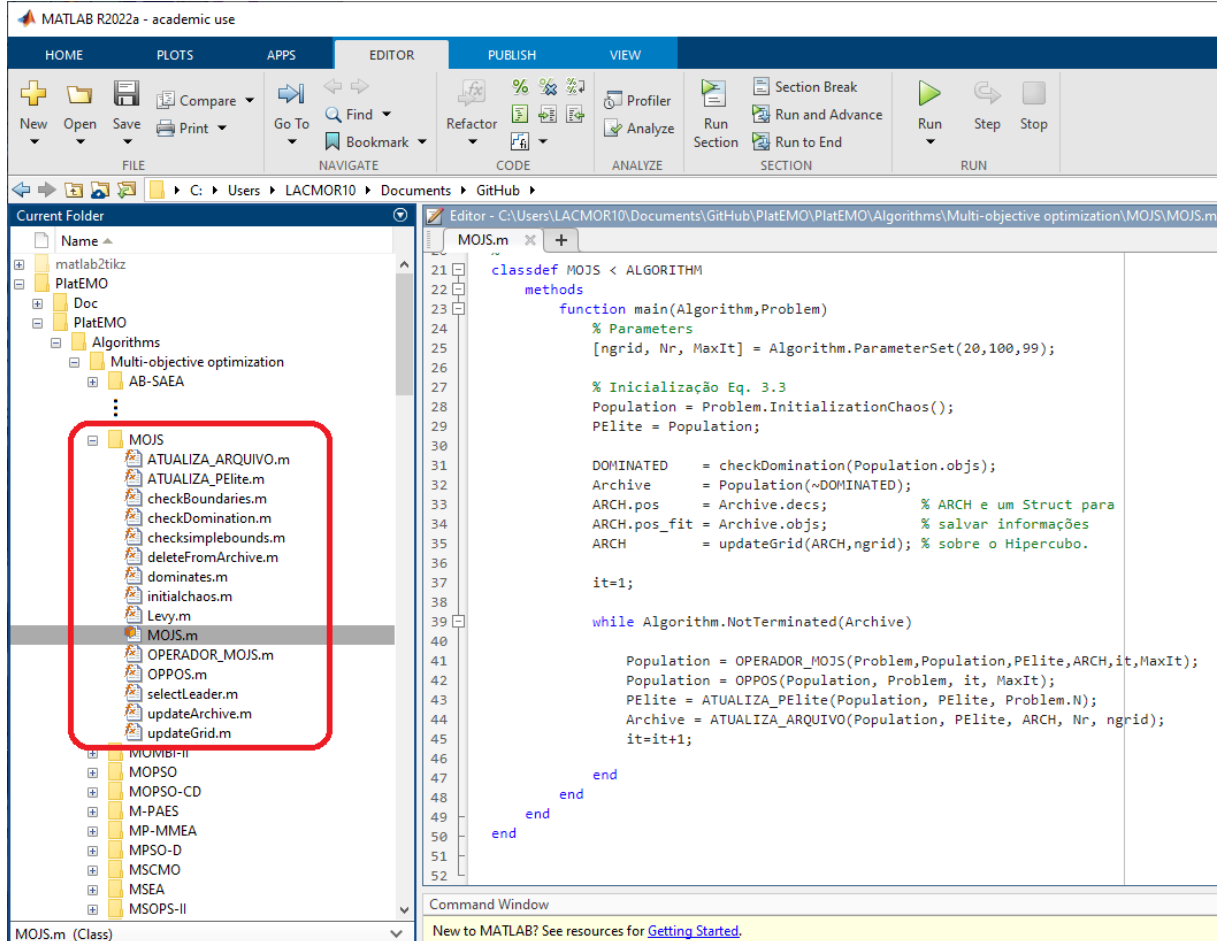
Fonte: Frederic Menezes Ferreira (2022).

O código acima executa o MOJS no problema DTLZ2 com uma população de 100 águas-vivas, 3 objetivos, 12 variáveis, para 190 mil avaliações de funções, onde as populações são salvas em um arquivo. Na Figura 2, os usuários podem determinar o tipo de problema na Região A, selecionar um algoritmo na Região B, selecionar um problema de referência na Região C e definir os valores dos parâmetros na Região D. Em seguida, o processo de otimização pode ser iniciado e controlado na Região E, onde o resultado em tempo real é exibido na Região F e os resultados históricos podem ser revisados na Região G.

Para adicionar o novo MOJS ao PlatEMO, basta modificar ligeiramente a entrada e saída do algoritmo conforme exigido pelo PlatEMO, e colocar o arquivo do código em MATLAB (MOJS.m) na pasta (\PlatEMO\Algorithms\Multi-objective optimization\MOJS) do diretório raiz (Figura 3).

Na avaliação do desempenho do MOJS foram utilizadas Funções de Benchmark (Figura 4) tiradas da literatura (CHOU; TRUONG, 2020). Ao todo, foram utilizadas nessa pesquisa 12 funções matemáticas (MOPs) e quatro meta-heurísticas (MOEAs) (contando com o MOJS).

Figura 3. Estrutura básica do diretório de arquivos do PlatEMO



Fonte: Frederic Menezes Ferreira (2022).

As métricas utilizadas são o Hiper volume (HV), Distância Geracional (GD) e Espaçamento (SP) pelos quais o desempenho do MOJS é comparado a cada algoritmo (MOEA/D, MOPSO e NSGA-II), um de cada vez, a um nível de significância $\alpha = 5\%$. Considerando o desempenho médio do algoritmo MOJS, μ_1 , e dos demais, μ_2 , então as hipóteses são:

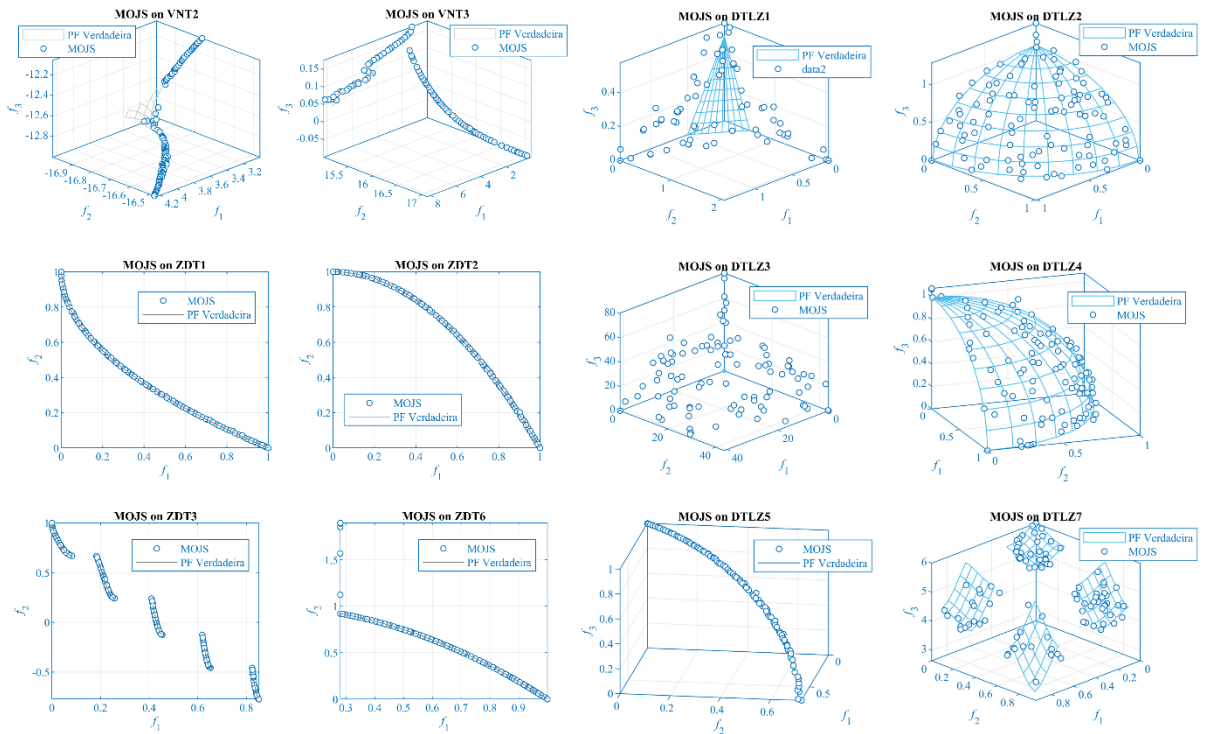
$$\begin{cases} H_0: \mu_1 - \mu_2 \\ H_1: \mu_1 \neq \mu_2 \end{cases}$$

O teste estatístico não paramétrico “Wilcoxon rank sum test” (CHOU; TRUONG, 2020) foi usado na comparação. Dado um nível de significância α , este teste produz a significância dos resultados de qualquer algoritmo, confirmando a hipótese nula H_0 (que não existe diferença ou melhoria significativa entre os resultados obtidos usando um par de algoritmos, onde $p_{valor} \geq \alpha$), ou confirmando a hipótese alternativa H_1 (que existe uma diferença significativa entre os resultados obtidos de um par de algoritmos, onde $p_{valor} < \alpha$).

Resultados e discussão

Depois de inserir o MOJS no PlatEMO (Figura 3), comparou-se seu desempenho com outros MOEAs. A Figura 4 mostra as Fronteiras Pareto-ótimas que foram obtidas com o MOJS. O MOJS se saiu muito bem, principalmente nas funções com 2 objetivos e nas funções com 3 objetivos, só não se saiu muito bem em duas: DTZ1 e DTZ3. Sendo que seu pior resultado foi com a DTZ3.

Figura 4. Aproximações da Fronteira Pareto-ótima geradas pelo MOJS, para as funções VNT2, VNT3, ZDT1, ZDT2, ZDT3, ZDT6 (as duas colunas da direita), DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5 e DTLZ7 (as duas colunas da esquerda)



Fonte: Frederic Menezes Ferreira (2022).

Pode-se ver na Figura 4, que o MOJS não conseguiu chegar na Fronteira Pareto-ótima verdadeira para a função DTZ3. Por outro lado, obteve seu melhor resultado na função ZDT3, na qual o MOJS superou alguns resultados dos outros algoritmos, como mostra as Tabelas 1, 2 e 3.

A Tabela 1 mostra a média dos resultados de Hiper volume (HV) dos 4 Algoritmos (MOEAD, MOPSO, NSGAI e MOJS) após 30 execuções para cada um dos Problemas (ZDT, Viennet e DTLZ).

Tabela 1. p-valor de Wilcoxon para a métrica HV

Problema	M	D	MOEAD	MOPSO	NSGAI	MOJS
ZDT1	2	30	7.1950e-1 ≈	7.8450e-2 –	7.1923e-1 –	7.1953e-1
ZDT2	2	30	4.4496e-1 +	6.6877e-3 –	4.4396e-1 –	4.4432e-1
ZDT3	2	30	5.9645e-1 –	1.7346e-1 –	5.9933e-1 –	5.9956e-1
ZDT6	2	10	3.8833e-1 +	3.6873e-1 –	3.8773e-1 –	3.8811e-1
VNT2	3	2	3.3552e-1 +	3.3256e-1 ≈	3.3134e-1 +	3.3085e-1
VNT3	3	2	1.7516e-1 –	1.7545e-1 –	1.7567e-1 –	1.7586e-1

DTLZ1	3	7	8.4170e-1 +	1.1816e-2 -	8.2202e-1 +	3.0112e-1
DTLZ2	3	12	5.5962e-1 +	1.5795e-1 -	5.2858e-1 +	4.8201e-1
DTLZ3	3	12	5.5849e-1 +	4.6580e-3 ≈	5.3148e-1 +	0.0000e+0
DTLZ4	3	12	5.2300e-1 -	3.9160e-1 -	5.0312e-1 -	5.2758e-1
DTLZ5	3	12	1.8185e-1 -	1.8972e-1 -	1.9874e-1 +	1.9560e-1
DTLZ7	3	22	2.5559e-1 -	2.5294e-2 -	2.6522e-1 ≈	2.6351e-1
+ / - / ≈			6/5/1	0/10/2	5/6/1	

Fonte: Frederic Menezes Ferreira (2022).

Nessas tabelas, “-” indica desempenho inferior ao do MOJS; “+” indica melhor desempenho do que o MOJS, e “≈” indica que não houve diferença significativa entre os desempenhos do algoritmo comparado e do MOJS. O destaque em cinza é para os melhores resultados.

Tabela 2. p-valor de Wilcoxon para a métrica GD

Problema	M	D	MOEAD	MOPSO	NSGAI	MOJS
ZDT1	2	30	1.0592e-5 +	6.9880e-2 -	1.4692e-4 -	1.2589e-4
ZDT2	2	30	3.6239e-5 +	1.4577e-1 -	1.3354e-4 -	1.1039e-4
ZDT3	2	30	1.0746e-4 -	7.0112e-2 -	7.7049e-5 ≈	6.8522e-5
ZDT6	2	10	7.8094e-5 +	6.1003e-2 -	3.5920e-6 +	5.4802e-2
VNT2	3	2	1.0047e-3 +	1.7449e-3 +	2.1113e-3 +	2.1665e-3
VNT3	3	2	1.4832e-3 +	3.3464e-3 +	4.0743e-3 -	3.4736e-3
DTLZ1	3	7	1.9019e-4 +	3.3467e+0 -	2.9967e-4 +	9.6401e-2
DTLZ2	3	12	5.0550e-4 +	1.3374e-1 -	1.2634e-3 +	7.0193e-3
DTLZ3	3	12	5.3288e-4 +	2.4573e+1 -	2.2644e-1 +	6.6985e+0
DTLZ4	3	12	4.9489e-4 +	4.8261e-2 -	1.0851e-3 +	2.4258e-3
DTLZ5	3	12	6.9191e-6 +	5.0654e-4 +	2.7584e-4 +	7.3500e-4
DTLZ7	3	22	3.0402e-3 +	2.1234e-1 -	3.1617e-3 +	4.3447e-3
+ / - / ≈			11/1/0	3/9/0	8/3/1	

Fonte: Frederic Menezes Ferreira (2022).

Na Tabela 2, com valores de GD do teste de Wilcoxon, destaca-se que o MOJS teve desempenho melhor que todos os outros algoritmos apenas para o problema ZDT3. O MOEA/D ganhou dos outros algoritmos em praticamente todos os problemas. O MOEA/D perdeu só duas vezes: para o MOJS no ZDT3 e para o NSGA-II no problema ZDT6. Isso mostra que o MOEA/D produz melhores soluções aproximadas do que todos os algoritmos comparados.

Tabela 3. p-valor de Wilcoxon para a métrica SP

Problema	M	D	MOEAD	MOPSO	NSGAI	MOJS
ZDT1	2	30	4.9301e-3 +	1.3129e-2 -	7.0474e-3 ≈	6.6545e-3
ZDT2	2	30	4.2735e-3 +	1.2664e-3 +	7.3930e-3 -	6.3258e-3
ZDT3	2	30	1.6713e-2 -	1.5520e-2 -	8.6609e-3 -	7.0815e-3
ZDT6	2	10	2.9171e-3 +	2.7306e-2 ≈	7.4400e-3 +	8.3328e-2
VNT2	3	2	1.5239e-2 -	1.9971e-2 -	2.1513e-2 -	1.4870e-2
VNT3	3	2	3.7402e-2 +	7.2608e-2 -	7.2166e-2 -	6.6680e-2
DTLZ1	3	7	1.5316e-5 +	1.8751e+0 -	2.4033e-2 +	9.9291e-2
DTLZ2	3	12	5.7192e-2 +	1.4693e-1 -	6.0840e-2 ≈	6.2247e-2
DTLZ3	3	12	5.7240e-2 +	1.9374e+1 -	1.7253e+0 +	8.4823e+0
DTLZ4	3	12	5.1483e-2 +	4.8214e-2 +	5.9665e-2 -	5.2011e-2
DTLZ5	3	12	1.3616e-2 -	1.4408e-2 -	1.0535e-2 -	8.7862e-3

DTLZ7	3	22	1.9607e-1 -	1.4637e-2 +	7.1626e-2 ≈	8.2941e-2
+ / - / ≈			8/4/0	3/8/1	3/6/3	

Fonte: Frederic Menezes Ferreira (2022).

Na Tabela 3, com valores de SP do teste de Wilcoxon, mais uma vez o MOJS teve melhor desempenho para o problema ZDT3, também se destacando no VNT2 e DTLZ5. Comparada aos outros algoritmos, com relação a todos os problemas, o MOJS ganhou do NSGA-II e do MOPSO, respectivamente, ganhando 6 vezes e 8 vezes. Contra o MOEA/D o MOJS perdeu 8 vezes das 12. O MOEA/D teve o melhor desempenho em 6 problemas, mostrando que consegue uma melhor distribuição das soluções do que os outros algoritmos.

Considerações Finais

Esse trabalho apresentou um estudo de caso sobre a plataforma PlatEMO para algoritmos meta-heurísticos multiobjetivos. Para tanto, foi incluído um recente algoritmo de otimização meta-heurístico multiobjetivo, MOJS, que é baseado no Algoritmo da Água-viva (JS) de objetivo único. O experimento foi conduzido sobre 12 funções *benchmark* e comparado com 3 algoritmos de otimização multiobjetivos conhecidos (MOEA/D, MOPSO, NSGA-II). As métricas HV, GD e SP e o teste de soma de postos de Wilcoxon foi usada para avaliação de desempenho. O PlatEMO apresenta ótimos recursos, com um número relativamente alto de algoritmos implementados, sendo eficaz para como ferramenta para comparação de abordagens meta-heurísticas.

Referências

CHOU, J.-S.; TRUONG, D.-N. **Multiobjective optimization inspired by behavior of jellyfish for solving structural design problems**. *Chaos, Solitons & Fractals*, Elsevier, v. 135, p. 109738, 2020. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0960077920301405>

CHUGH, T. et al. **A survey on handling computationally expensive multiobjective optimization problems with evolutionary algorithms**. *Soft Computing*, Springer, v. 23, n. 9, p. 3137-3166, 2019. Disponível em: <https://link.springer.com/article/10.1007/s00500-017-2965-0>

DOKEROGLU, T. et al. **A survey on new generation metaheuristic algorithms**. *Computers & Industrial Engineering*, Elsevier, v. 137, p. 106040, 2019. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0360835219304991>

HUA, Y. et al. **A survey of evolutionary algorithms for multi-objective optimization problems with irregular pareto fronts**. *IEEE/CAA Journal of Automatica Sinica*, IEEE, v. 8, n. 2, p. 303-318, 2021. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9321268>

LIU, X.; IJZERMAN, A. P.; WESTEN, G. J. van. **Computational approaches for de novo drug design: past, present, and future**. *Artificial neural networks*, Springer, p. 139-165, 2021. Disponível em: https://link.springer.com/protocol/10.1007/978-1-0716-0826-5_6

TIAN, Y. et al. **Platemo: A matlab platform for evolutionary multi-objective optimization [educational forum]**. *IEEE Computational Intelligence Magazine*, IEEE, v. 12, n. 4, p. 73-87, 2017. Disponível em: <https://ieeexplore.ieee.org/abstract/document/8065138>.

Recebido em: 30 de novembro de 2022.

Aceito em: 20 de janeiro de 2023.