

Evolução diferencial com mutação ordenada em problemas de otimização monobjetivo com restrições de caixa

Dênis E. C. Vargas¹
DM/CEFET-MG, Belo Horizonte, MG

Rafael de Paula Garcia²
DAU/UFV, Viçosa, MG

Afonso C. C. Lemonge³
MAC/UFJF, Juiz de Fora, MG

Resumo. A Evolução Diferencial (ED) está entre os algoritmos evolucionistas mais eficientes para lidar com problemas de otimização. Sua proposição original utiliza o esquema clássico DE/Rand para selecionar aleatoriamente soluções candidatas da população para o processo de mutação, sem considerar qualquer ordenação entre elas. Recentemente foi proposto o esquema DE/Order para problemas multi-objetivo, uma estratégia de ordenação entre as soluções selecionadas para a mutação. O algoritmo com o esquema DE/Order apresentou melhores resultados em problemas de otimização multi-objetivo quando comparado ao DE/Rand. Esse trabalho avalia a estratégia de mutação DE/Order em problemas de otimização monobjetivo com restrições de caixa. A performance desta estratégia foi comparada com duas outras já consolidadas na literatura, DE/Rand e DE/Best, ao serem aplicadas a problemas monobjetivos *benchmark* da competição do *IEEE Congress on Evolutionary Computation - CEC 2021*. Os resultados mostraram que o esquema DE/Best apresenta o pior desempenho, sugerindo convergência prematura para ótimos locais. Além disso, este trabalho mostra através de testes não paramétricos que as estratégias DE/Order e DE/Rand não demonstraram diferenças estatísticas. Concluiu-se que o DE/Order se mostra competitivo neste conjunto de problemas, apresentando-se como uma estratégia que se beneficia dos conceitos das outras duas abordagens, randomização e elitismo, porém sem ser prejudicado pela estagnação em ótimos locais.

Palavras-chave. Evolução diferencial, mutação ordenada, otimização monobjetivo.

1 Introdução

A Evolução Diferencial (ED) [1] é um algoritmo evolucionista que se destaca devido à sua simplicidade, aos poucos parâmetros de controle utilizados e pelo bom desempenho com rápida convergência. Na sua versão original, a ED seleciona aleatoriamente soluções candidatas da população, sem considerar qualquer ordenação, para aplicar sobre elas mutação e recombinação de suas variáveis. Esta abordagem é conhecida como DE/Rand.

Uma alternativa a este esquema de seleção foi proposta recentemente na quarta versão do algoritmo *Generalized Differential Evolution* (GDE4) [2], que introduziu uma ordenação dessas

¹denis.vargas@cefetmg.br

²rafael.pgarcia@ufv.br

³afonso.lemonge@ufjf.edu.br

soluções antes da aplicação do processo de mutação. Nesta estratégia, três soluções são selecionadas aleatoriamente como na anterior DE/Rand, porém antes da aplicação da mutação, uma ordenação destas soluções é feita como uma tentativa de tender a busca para regiões com melhores valores de função objetivo. Os autores denominaram esse esquema de DE/Order. Os experimentos em [2] foram conduzidos em problemas de otimização multi-objetivo e os resultados da comparação revelam que o algoritmo GDE4 supera o GDE3 [3], versão anterior do algoritmo *Generalized Differential Evolution* e que utiliza o esquema DE/Rand.

Este trabalho se propôs a realizar uma continuidade do estudo [2] para avaliar se a boa performance do DE/Order se estende a problemas de otimização monobjetivo com restrições de caixa. A ED com a estratégia DE/Order foi aplicada aos problemas monobjetivo de otimização com restrições de caixa *benchmark* da Competição do *IEEE Congress on Evolutionary Computation - CEC 2021* [4]. Os resultados foram comparados com os obtidos pela ED usando os esquemas clássicos DE/Rand e DE/Best nos mesmos problemas mantendo os parâmetros da ED inalterados.

2 Problemas Monobjetivo de Otimização com Restrições de Caixa

Problemas monobjetivo de otimização com restrições de caixa podem ser definidos como:

$$\begin{aligned} \min_{\mathbf{x}} \quad & f(\mathbf{x}) \\ \text{onde } \mathbf{x} = & (x_1, \dots, x_D) \\ & x_i \in [l_i, u_i] \end{aligned} \quad (1)$$

onde l_i e u_i são, respectivamente, os limites inferior e superior de cada entrada x_i do vetor $\mathbf{x} \in \mathbb{R}^n$.

Neste trabalho, 10 problemas *benchmark* de otimização monobjetivo com restrições de caixa da competição do *IEEE Congress on Evolutionary Computation - CEC 2021* foram adotados para os experimentos computacionais, com $D = 10$ e $D = 20$. Estes problemas são frequentemente adotados para comparar a performance de algoritmos evolucionistas. Suas nomenclaturas e principais características podem ser observadas na Tabela 1, enquanto que uma descrição detalhada pode ser acessada em [4].

3 Evolução Diferencial

A Evolução Diferencial (ED) [1] é um algoritmo evolucionista que parte de uma população inicial de NP soluções candidatas, geradas aleatoriamente e a evolui ao longo de gerações. A evolução ocorre pela aplicação de operadores de mutação e recombinação sobre a população de uma geração, produzindo novas soluções candidatas. O esquema de mutação proposto na versão original da ED funciona da seguinte forma: para cada uma das NP soluções candidatas ($\mathbf{x} \in \mathbb{R}^n$ chamado de vetor alvo), sorteia-se três soluções na população \mathbf{x}_{r1} , \mathbf{x}_{r2} e $\mathbf{x}_{r3} \in \mathbb{R}^n$, diferentes entre si e diferentes do vetor alvo. Um vetor $\mathbf{v} \in \mathbb{R}^n$ (chamado de vetor doador) é então calculado pela fórmula a seguir:

$$\mathbf{v} = \mathbf{x}_{r1} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3}) \quad (2)$$

onde $F \in \mathbb{R}$ é um parâmetro de multiplicação maior do que zero.

O vetor experimental \mathbf{u} é encontrado após a aplicação do processo de cruzamento, que mistura informações apresentadas pelas soluções \mathbf{x} e \mathbf{v} . Para isso, um inteiro aleatório j entre 1 e n é sorteado e para cada $i = 1, \dots, n$, um número aleatório $rand_i$ entre 0 e 1 também é sorteado. Se $rand_i \leq CR$ ou $i = j$, então $u_i = v_i$. Caso contrário, $u_i = x_i$. A condição $i = j$ garante a cópia de

Tabela 1: Funções básicas envolvidas nos 10 problemas da Competição do CEC 2021 e principal característica

Problema	Função(ões) envolvida(s)	Característica
1	<i>Bent Cigar</i>	Unimodal
2	<i>Schwefel</i>	Função Básica
3	<i>Lunacek bi-Rastrigin</i>	Função Básica
4	<i>Rosenbrock plus Griewangk</i>	Função Básica
5	<i>Schwefel, Rastrigin e Elliptic</i>	Função Híbrida
6	<i>Schaffer, HGBat, Rosenbrock e Schwefel</i>	Função Híbrida
7	<i>Schaffer, HGBat, Rosenbrock, Schwefel e Elliptic</i>	Função Híbrida
8	<i>Rastrigin, Griewank e Schwefel</i>	Composição de Funções
9	<i>Ackley, Elliptic, Griewank e Rastrigin</i>	Composição de Funções
10	<i>Rastrigin, Happycat, Ackley, Discus e Rosenbrock</i>	Composição de Funções

pelo menos uma variável do vetor doador. Por fim, utiliza-se um critério de seleção para decidir qual das duas soluções \mathbf{u} ou \mathbf{x} continua na população da geração seguinte. Se $f(\mathbf{u}) < f(\mathbf{x})$, então \mathbf{u} substitui \mathbf{x} na população da geração seguinte. Do contrário, \mathbf{u} é descartado e \mathbf{x} permanece na população.

O esquema de mutação apresentado pela Equação 2 é conhecido como DE/Rand, pois os vetores são escolhidos e classificados aleatoriamente. Na literatura existem outros esquemas clássicos como o DE/Best, que substitui o \mathbf{x}_{r-1} da fórmula da Equação 2 pelo \mathbf{x}_b , o melhor de toda a população.

Recentemente foi proposto em [2] um novo esquema de mutação de ED chamado DE/Order, que introduziu uma ordenação das soluções sorteadas antes da aplicação do processo de mutação. No esquema DE/Order, as 3 soluções candidatas sorteadas \mathbf{x}_{r-1} , \mathbf{x}_{r-2} e $\mathbf{x}_{r-3} \in \mathbb{R}^n$ são classificadas em \mathbf{x}_b (*best*), \mathbf{x}_{2b} (*second best*) e \mathbf{x}_w (*worst*) de acordo com os valores de suas funções objetivos. Os experimentos em [2] foram conduzidos em problemas de otimização multi-objetivo e a classificação considerava a posição das soluções no ranqueamento por dominância de Pareto e uma medida de diversidade chamada *Crowding Distance*. Os autores compararam o DE/Order apenas com o DE/Rand. Nesse trabalho, além de adaptar o DE/Order para problemas de otimização monobjetivo com restrições de caixa, também utilizou-se o esquema DE/Best para comparação. Todos os esquemas de mutação utilizados nesse trabalho estão detalhados na Tabela 2.

4 Experimentos Computacionais

Nesse trabalho avaliou-se a performance do esquema de mutação DE/Order para problemas de otimização monobjetivo com restrições de caixa. Eles foram comparados aos esquemas clássicos DE/Rand e DE/Best nos 10 problemas monobjetivo de otimização da Competição do *IEEE Congress on Evolutionary Computation - CEC 2021*, com dimensões $D = 10$ e $D = 20$, resultando, portanto, na experimentação em 20 diferentes problemas. Os parâmetros adotados para todos os algoritmos foram os sugeridos pela própria competição, além de serem popularmente utilizados na

Tabela 2: Esquemas de Mutaç o de ED utilizados nesse trabalho.

Esquema	F�rmula	Definiç�o
DE/Rand	$\mathbf{v} = \mathbf{x}_{r1} + F(\mathbf{x}_{r2} - \mathbf{x}_{r3})$	\mathbf{x}_{r1} , \mathbf{x}_{r2} e \mathbf{x}_{r3} aleat�rios
DE/Best	$\mathbf{v} = \mathbf{x}_b + F(\mathbf{x}_{r1} - \mathbf{x}_{r2})$	\mathbf{x}_{r1} e \mathbf{x}_{r2} aleat�rios \mathbf{x}_b : soluç�o com melhor valor de funç�o objetivo em toda a populaç�o
DE/Order	$\mathbf{v} = \mathbf{x}_b + F(\mathbf{x}_{2b} - \mathbf{x}_w)$	\mathbf{x}_{r1} , \mathbf{x}_{r2} e \mathbf{x}_{r3} aleat�rios \mathbf{x}_b , \mathbf{x}_{2b} e \mathbf{x}_w escolhidos entre \mathbf{x}_{r1} , \mathbf{x}_{r2} e \mathbf{x}_{r3} de acordo com a melhor, segunda melhor e pior valor de funç�o objetivo, respectivamente

literatura: $NP = 100$, $F = 0.5$, $CR = 0.9$, 200.000 e 1.000.000 de avaliaç es da funç o objetivo como crit rio de parada para $D = 10$ e $D = 20$, respectivamente, e toler ncia de 1×10^{-8} . Foram conduzidas 30 execuç es independentes. Apesar dos problemas da competiç o do CEC 2021 considerarem possibilidades de parametrizaç es como polarizaç o, deslocamento e rotaç o, nenhuma dessas configuraç es foram adotadas nesse trabalho.

A competiç o prop e que os algoritmos sejam avaliados segundo a soma de dois *scores*. Para o primeiro, $Score_1$, inicialmente, cada algoritmo tem uma Soma de Erros Normalizados (*SNE*) calculada atrav s da f rmula

$$SNE = \frac{1}{2} \sum_{i=1}^{20} ne_i \quad \text{onde} \quad ne = \frac{f(\mathbf{x}_{best}) - f(\mathbf{x}^*)}{f(\mathbf{x}_{best})_{max} - f(\mathbf{x}^*)}$$

sendo $f(\mathbf{x}_{best})$ o melhor resultado do algoritmo nas 30 execuç es independentes, $f(\mathbf{x}^*)$ o valor  timo conhecido da funç o e $f(\mathbf{x}_{best})_{max}$ o maior $f(\mathbf{x}_{best})$ entre todos os algoritmos para a funç o considerada. Uma vez que *SNE* foi determinado para todos os algoritmos, o $Score_1$   calculado como

$$Score_1 = \left(1 - \frac{SNE - SNE_{min}}{SNE} \right) \times 50 \quad (3)$$

onde SNE_{min}   o menor *SNE* entre todos os algoritmos.

Para o $Score_2$, considera-se para cada algoritmo uma Soma de Ranks (*SR*) calculada atrav s da f rmula

$$SR = \frac{1}{2} \sum_{i=1}^{20} Rank_i$$

onde $Rank_i$   a classificaç o do algoritmo entre todos os algoritmos para uma determinada funç o baseada em seu valor de erro m dio n o normalizado $f(\mathbf{x}_{best}) - f(\mathbf{x}^*)$. Uma vez que *SR* foi determinado para todos os algoritmos, o $Score_2$   calculado como

$$Score_2 = \left(1 - \frac{SR - SR_{min}}{SR} \right) \times 50 \quad (4)$$

onde SR_{min}   o menor *SR* entre todos os algoritmos. A pontuaç o final   a soma $Score_1 + Score_2$.

A Tabela 3 mostra a pontuaç o de cada esquema de mutaç o nos problemas adotados, enquanto que a Figura 1 mostra os boxplots dos erros, $f(\mathbf{x}_{best}) - f(\mathbf{x}^*)$, obtidos nas 30 execuç es por cada esquema de mutaç o em cada um dos problemas.

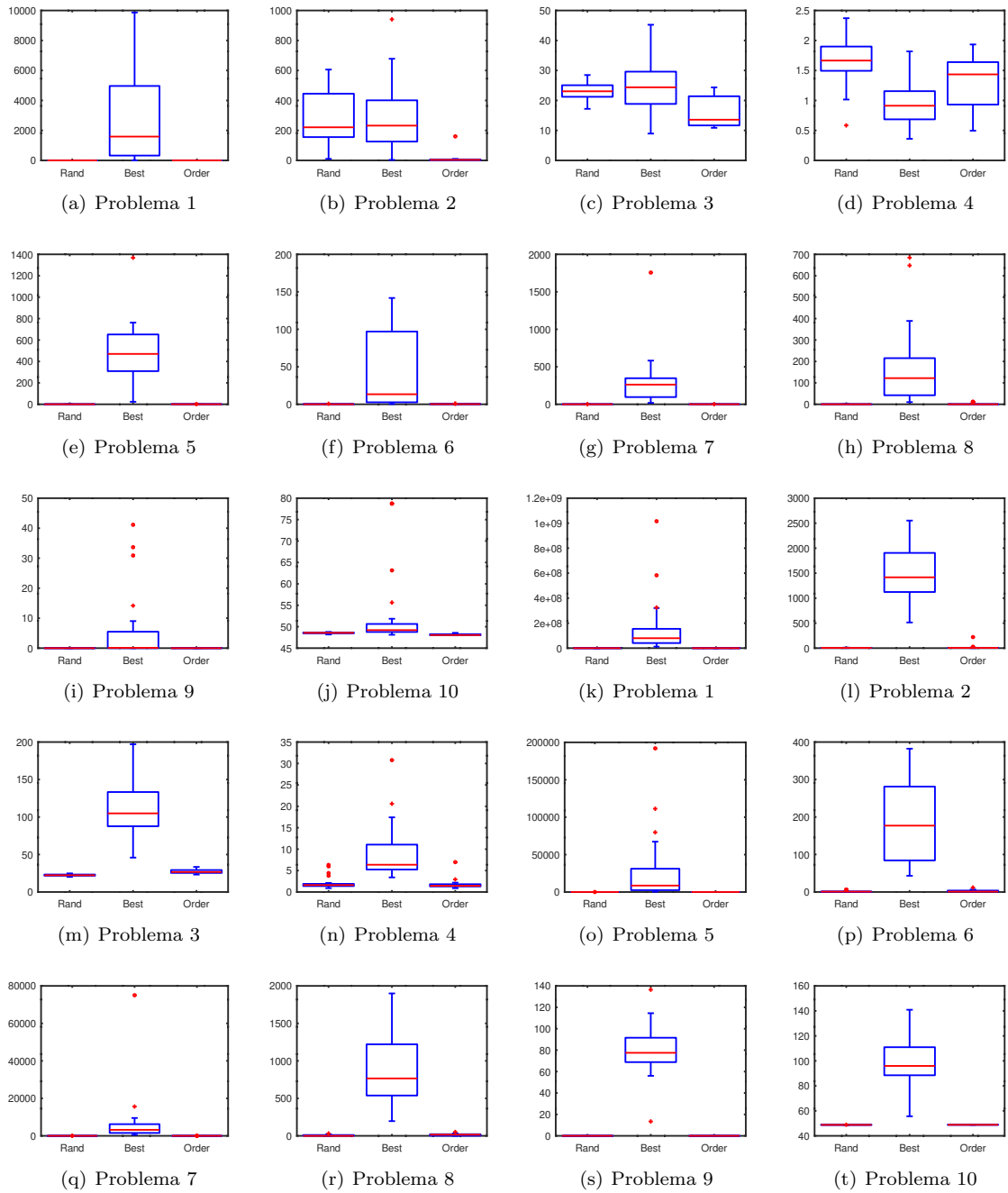


Figura 1: Boxplots com os erros encontrados, $f(\mathbf{x}_{best}) - f(\mathbf{x}^*)$, nas 30 execuções de cada algoritmo em cada problema. As letras (a)-(j) referem-se à $D = 10$ e as letras (k)-(t) referem-se à $D = 20$.

Tabela 3: Pontuação

	DE/Rand	DE/Best	DE/Order
$Score_1$	50	8	48.18
$Score_2$	50	25.47	42.19
Pontuação Final	100 (1 ^o Lugar)	33.47 (3 ^o Lugar)	90.37 (2 ^o Lugar)

5 Discussão dos Resultados

As pontuações apresentadas na Tabela 3 mostram que o esquema de mutação DE/Best foi o pior dentre os avaliados, com pontuação muito baixa quando comparado aos demais. Sua baixa eficiência também pode ser observada na Figura 1, cujos boxplots ficam majoritariamente acima dos outros. A principal razão deste desempenho ruim pode ser atribuída ao comportamento altamente elitista do DE/Best, contribuindo para convergências prematuras para ótimos locais. Uma vez preso nessas regiões, áreas promissoras do espaço de busca deixam de ser visitadas e a falta de diversidade compromete o desempenho da abordagem para se desvencilhar e evoluir em busca da solução ótima.

Com relação aos esquemas DE/Order e DE/Rand, apesar dos resultados da Tabela 3 mostrarem o DE/Rand como o de melhor desempenho, o DE/Order ficou em 2^o Lugar e com pontuação próxima. O p -valor ($= 0.000000853$) obtido pela aplicação do teste não paramétrico de Kruskal Wallis [5] aos resultados, indica que há diferença estatística entre as abordagens. Para a comparação em pares, o também não paramétrico *Wilcoxon Signed Test* [6] foi aplicado. Os testes indicam diferença significativa entre as estratégias DE/Order e DE/Rand com a DE/Best, com p -valores iguais a 0.000003815 e 0.000009537, respectivamente. Por outro lado, com p -valor $= 0.277530251$, o teste não assegura a rejeição da hipótese nula na comparação entre DE/Order e DE/Rand, sugerindo que as distribuições entre estas duas abordagens são semelhantes. A semelhança dos desempenhos do DE/Order e DE/Rand pode ser visualmente percebida pelos boxplots da Figura 1, onde em praticamente todos os problemas eles possuem distribuições semelhantes.

Por fim, destaca-se aqui o DE/Order, pois além de se mostrar competitivo, principalmente quando comparado com o DE/Rand, ainda apresenta resultados substancialmente melhores em alguns problemas, como, por exemplo, os Problemas 2, 3 e 4 com $D = 10$. O DE/Order, portanto, apresenta-se como uma alternativa: é competitivo ao DE/Rand, melhor em alguns problemas e também se beneficia de uma abordagem elitista semelhante ao DE/Best (Problema 4, $D = 10$), quando se propõe a ordenação, porém sem se estagnar em ótimos locais. Como exemplo, a Figura 2 ilustra a evolução das médias dos erros obtidos pelas abordagens no Problema 2, com $D = 10$. Mesmo obtendo soluções melhores já no início da evolução, o DE/Best permanece nela por todo o restante da evolução. O DE/Order, por outro lado, evolui suas soluções na direção do ótimo, competindo bem com o DE/Rand.

6 Conclusões

Nesse trabalho avaliou-se o recentemente proposto esquema de mutação DE/Order em problemas de otimização monobjetivo com 10 e 20 dimensões. A performance desta estratégia foi comparada com outras duas, DE/Rand e DE/Best. Os resultados mostraram que o esquema DE/Best apresenta o pior desempenho, e a principal razão para isso recai sobre seu comportamento altamente elitista, apresentando convergência prematuramente para ótimos locais. Apesar do DE/Rand ter melhor desempenho que o DE/Order segundo uma métrica de *scores*, testes não paramétricos mostram que eles não possuem diferenças estatísticas. Destaca-se que o DE/Order se mostrou competitivo ao DE/Rand por não ter resultados muito piores que ele e ainda ter resulta-

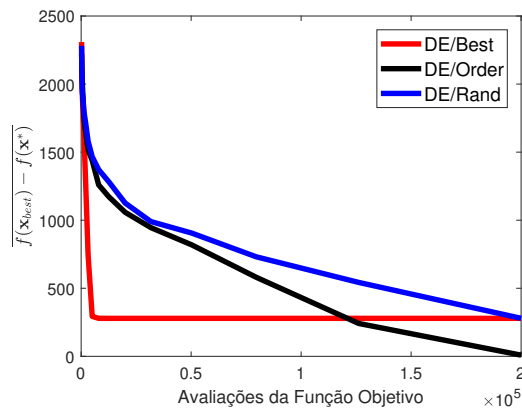


Figura 2: Evolução das médias dos erros, $f(\mathbf{x}_{best}) - f(\mathbf{x}^*)$, nas 30 execuções no Problema 2 com $D = 10$.

dos significativamente melhores em alguns problemas, além de se beneficiar de estratégia elitista, como o DE/Best, porém sem ficar preso em ótimos locais. Trabalhos futuros poderão avaliar o DE/Order em problemas com restrições, além de formas híbridas e *ensemble* de variantes.

Agradecimentos

O terceiro autor agradece ao CNPq (306186/2017-9) e FAPEMIG (TEC PPM-174-18).

Referências

- [1] Storn, R. e Price, K. Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11(4):341–359, 1997. DOI:10.1023/A:1008202821328.
- [2] Bidgoli, A. A., Mahdavi, S., Rahnamayan, S. e Ebrahimpour-Komleh, H. GDE4: The Generalized Differential Evolution with Ordered Mutation. In *Evolutionary Multi-Criterion Optimization. EMO 2019* - Deb K. et al. (eds). Lecture Notes in Computer Science, Springer, Cham. volume 11411, pages 101-113, 2019. DOI: 10.1007/978-3-030-12598-1_9.
- [3] Kukkonen, S. e Lampinen, J. GDE3: the third evolution step of generalized differential evolution, *Proceedings of the IEEE Congress on Evolutionary Computation. CEC 2005*, Edinburgh, Scotland, volume 1, pages 443-450, 2005. DOI: 10.1109/CEC.2005.1554717.
- [4] Wagdy, A., Hadi, A. A., Mohamed, A. K., Agrawal, P., Kumar, A e Suganthan, P. N.. Problem Definitions and Evaluation Criteria for the CEC 2021 Special Session and Competition on Single Objective Bound Constrained Numerical Optimization, Technical Report, Nanyang Technological University, Singapore.
- [5] Kruskal, W. H., & Wallis, W. A. Use of ranks in one-criterion variance analysis. *Journal of the American statistical Association*, 47(260), 583-621, 1952.
- [6] Pratt, J. W. Remarks on zeros and ties in the Wilcoxon signed rank procedures. *Journal of the American Statistical Association*, 54(287), 655-667, 1959.