Supplementary Document of "A Dynamic Dimensionality Expansion-Based Evolutionary Algorithm for Constrained Many-Objective Optimization"

Jiawei Yuan

I. Specific Steps of the Mechanism using Promising Region and Parallel Distance in PREA [1]

The mechanism using promising region and parallel distance prunes the feasible elite set $Archive$, which exceeds size N, to a specific size of N through the following steps:

1) Translate the positions of all individuals in the objective space to R_+^m using the formula:

$$\bar{f}_i(\mathbf{x}) = f_i(\mathbf{x}) - f_i^{Fmin} + 10^{-6}, \quad i = 1, ..., m, \tag{1}$$

where f_i^{Fmin} is the minimum value found on the i-th objective f_i so far.

2) Calculate the I-indicator value for any individual $\mathbf{x}^1$ in $Archive$ evaluated by other one $\mathbf{x}^2$:

$$I(\mathbf{x}^1|\mathbf{x}^2) = \begin{cases} \max\limits_i \dfrac{f_i'(\mathbf{x}^2)}{f_i'(\mathbf{x}^1)}, & \text{if } \exists f_i'(\mathbf{x}^2) < f_i'(\mathbf{x}^1), \\ \min\limits_i \dfrac{f_i'(\mathbf{x}^1)}{f_i'(\mathbf{x}^2)}, & \text{otherwise.} \end{cases} \tag{2}$$

3) Define the minimum I-indicator value from all other individuals for $\mathbf{x}^1$ as its fitness value within $Archive$:

$$Fitness(\mathbf{x}^1|Archive) = \min_{\mathbf{x}^2 \in Archive \setminus \{\mathbf{x}^1\}} I(\mathbf{x}^1|\mathbf{x}^2). \tag{3}$$

4) Identify the Pareto optimal solution set $L1$ and the inferior solution set $L2$ within $Archive$ based on comparison result between fitness value and 1:

$$\begin{cases} L1 = \{\mathbf{x} \in Archive | Fitness(\mathbf{x}|Archive) > 1\}, \\ L2 = Archive \setminus L1. \end{cases} \tag{4}$$

5) Set $Archive = L1$. If the size of $Archive$ is smaller than N, directly select the $(N - |Archive|)$ individuals with the largest fitness values from $L2$ to fill $Archive$, terminating the process and outputting $Archive$ as the result. Otherwise, proceed to step 6).

6) Select the N individuals with the largest fitness values from $L1$ to construct the promising region PR: $[0, f_1^{Fmax}] \times [0, f_2^{Fmax}] \times ... \times [0, f_m^{Fmax}]$, where $f_1^{Fmax}, ..., f_m^{Fmax}$ are determined as follows

$$f_i^{Fmax} = \max_{\mathbf{x}^j \in L1} \bar{f}_i(\mathbf{x}^j), \quad i = 1, ..., m. \tag{5}$$

7) Remove the individuals of $Archive$ that lie outside of PR.

8) Calculate the crowding distance between any two individuals in $Archive$,

$$Crowd(\mathbf{x}^i|Archive) = \min_{\mathbf{x}^j \in Archive \setminus \{\mathbf{x}^i\}} Paral(\mathbf{x}^i, \mathbf{x}^j), \tag{6}$$

where $Paral(\mathbf{x}^i, \mathbf{x}^j)$ is a function measuring the parallel distance of two individuals, computed as follows:

$$Paral(\mathbf{x}^i, \mathbf{x}^j) = \sqrt{\sum_{p=1}^{m} (\frac{\bar{f}_i(\mathbf{x}^i) - \bar{f}_i(\mathbf{x}^j)}{f_i^{Fmax}})^2 - \frac{(\sum_{p=1}^{m} (\bar{f}_i(\mathbf{x}^i) - \bar{f}_i(\mathbf{x}^j))/f_i^{Fmax})^2}{m}}. \tag{7}$$

9) Iteratively remove the individual with the lowest fitness value among the two individuals with the smallest crowding distance in $Archive$ until the size of $Archive$ is reduced to N. Output $Archive$ as the final result.

II. Brief Description of the Comparison algorithms

- CMOCSO [2] is a cooperative multi-population, multi-criterion algorithm. It employs a competitive swarm optimizer based on a constraint relaxation mechanism to search for the CPF, while a cooperative cooperative swarm optimizer utilizing constraint-ignoring CHT is used to explore the UPF. Additionally, it incorporates a feasibility-driven CHT to update the discovered set of optimal solutions, known as the elite set.
- CCMO [3] also utilizes two distinct populations to separately search for CPF and UPF. A notable feature of this algorithm is the minimal interaction between the two populations, which only exchange useful information through offspring contributions after the generation of new individuals.
- DSPCMDE [4] dynamically adjusts the quality assessment of individuals throughout the evaluation process. In its early stages, it emphasizes convergence and diversity by ignoring feasibility, thus preventing the population from becoming trapped in local optima. In the later stages, it employs a feasibility-driven CHT to guide the population toward the CPF.
- ICMA [5] adopts a uniform exploration strategy to identify the CPFs. It prioritizes the elimination of the individual with the highest constraint violation among the two closest individuals in promising regions, while maintaining an elite set based on feasibility-driven CHT to record the optimal individuals discovered.
- MTCMO [6] introduces a corresponding auxiliary optimization problem when addressing CMaOP, where the tolerance levels for constraints gradually decrease as the evolutionary process advances. It simultaneously tackles both the original and auxiliary optimization problems through multi-task optimization techniques, effectively escaping local optima regions.

III. Performance Metrics

Two commonly used metrics, inverted generational distance (IGD) [7] and hypervolume (HV) [8], are employed to assess the quality of the final results obtained by the considered algorithms in this paper. Given that infeasible solutions lack practical significance for the optimization problems, only feasible solutions are considered when evaluating the quality of algorithms' outcomes. If a algorithm's final results do not contain any feasible solutions, both its IGD and HV values are marked as "NaN". To prevent excessive numerical values, prior to computing these two metrics, we normalize the variation range of the CPF to the space $[0, 1]^m$. Subsequently, the final outcomes obtained by each algorithm undergo corresponding transformations within the objective space. Let $\mathbf{P}$ be a uniformly distributed set of points across the CPF of considering CMaOP, and $\mathbf{X}^*$ be the final set of feasible solutions obtained by a algorithm on this problem. The formulas for calculating the IGD and HV values of the algorithm on this CMaOP are as follows:

$$\begin{cases} IGD(\mathbf{X}^*|\mathbf{P}) = \frac{\sum_{\mathbf{p}\in\mathbf{P}} \min_{\mathbf{x}^*\in\mathbf{X}^*} \|\mathbf{p}-\mathbf{F}(\mathbf{x}^*)\|_2}{|\mathbf{P}|}, \\ HV(\mathbf{X}^*) = Vol(\biguplus_{\mathbf{x}^*\in\mathbf{X}^*} \prod_{i=1}^{m}[f_i(\mathbf{x}^*), 1.1]), \end{cases} \tag{8}$$

where $Vol(\bullet)$ is the Lebesgue measure. It is evident that when $\mathbf{X}^*$ closely approximate the CPF, it typically exhibits smaller IGD value and larger HV value.

IV. Experimental Setting Details

- Platform: The numerical experiments were conducted using the platform PlatEMO, which can be downloaded from the website https://github.com/BIMK/PlatEMO.
- Common Parameters: Each test problem was evaluated across three objective dimensions: 5, 8, and 10. The corresponding population sizes were set to 126, 156, and 275, respectively. Each algorithm was independently run 20 times with a maximum of 1500 evolutionary generations per test problem.
- Specific Parameters: The private parameters of the five comparison algorithms were set to their recommended values in their publications, matching the default settings on the platform PlatEMO. Specifically, in CMOCSO, the parameters used to adjust the value of ϵ are set as follows: $cp = 2$, $\alpha = 0.95$, and $\tau = 0.05$. In CCMO, the parameter $type$ controlling the genetic operation operator is set to 1, indicating the use of SBX. In ICMA, the probability p_c of an individual selecting a mate from its neighbors is set to 0.7. Parameters for DSPCMDE, MTCMO, and the proposed DDEEA were predefined within their algorithmic frameworks and did not include any additional private parameters. Besides, for the three genetic operators, i.e., SBX, DE, and the polynomial mutation, their parameters are set to $\{p_c = 1, \eta_c = 20\}$, $\{CR = 1, F = 0.5\}$, and $\{p_m = 1/D, \eta_m = 20\}$, respectively.
- Results Presentation: The Wilcoxon rank-sum test was employed to assess whether there exists a statistically significant difference in performance between the proposed DDEEA and the other comparison algorithms. For clarity, we utilized symbols "+", "–", and "=" to respectively indicate whether a comparison algorithm's performance was superior to, inferior to, or approximately equal to that of DDEEA. Additionally, symbol "*" were used to highlight the algorithm that achieved the best performance at each test problem.

TABLE A-1: The comparison results among DDEEA-D, DDEEA-I, DDEEA-M, and DDEEA in T1 test series.

Metric	Type	Problem	M	DDEEA-D	DDEEA-I	DDEEA-M	DDEEA
IGD	T1	C1–DTLZ1	5	1.286e-1 =	1.225e-1 +*	1.346e-1 =	0.1291
			8	2.095e-1 =	2.014e-1 +*	2.119e-1 –	0.2085
			10	2.193e-1 =	2.083e-1 +*	2.255e-1 –	0.2185
		C3–DTLZ4	5	1.600e-1 =	1.579e-1 +*	1.609e-1 =	0.1594
			8	2.623e-1 =	2.722e-1 –	2.626e-1 =	0.2622*
			10	3.006e-1 =	3.142e-1 –	2.986e-1 +*	0.3002
		DC2–DTLZ1	5	1.343e-1 =	1.363e-1 –	1.367e-1 =	0.1335*
			8	2.730e-1 =	3.088e-1 –	2.666e-1 =*	0.2702
			10	3.896e-1 –	3.888e-1 –	5.788e-1 –	0.3548*
		LYO1	5	2.494e-1 –	2.565e-1 –	2.643e-1 –	0.2357*
			8	6.202e-1 =	5.789e-1 +*	6.272e-1 =	0.6164
			10	6.820e-1 =	6.447e-1 +*	6.875e-1 =	0.6880
		LYO2	5	1.390e-1 =	1.374e-1 =*	3.301e-1 –	0.1382
			8	6.478e-1 –	2.647e-1 =*	7.500e-1 –	0.2798
			10	8.668e-1 –	3.108e-1 –	8.950e-1 –	0.3094*
		LYO3	5	3.401e-1 –	3.185e-1 =	1.171e+0 –	0.1794*
			8	9.003e-1 –	1.455e-1 =*	4.037e-1 –	0.1556
			10	5.545e-1 –	1.350e-1 =	5.605e-1 –	0.1319*
		LYO4	5	1.543e-1 –	1.520e-1 –	1.704e-1 =	0.1502*
			8	2.804e-1 –	2.639e-1 –	2.680e-1 =	0.2609*
			10	3.180e-1 –	3.056e-1 –	3.144e-1 =	0.2959*
	Summary (+/–/=)			0/10/11	6/10/5	1/10/10	
HV	T1	C1–DTLZ1	5	9.722e-1 =	9.683e-1 –	9.581e-1 =	0.9723*
			8	9.955e-1 =	9.925e-1 –	9.955e-1 =	0.9957*
			10	9.984e-1 =	9.963e-1 –	9.987e-1 +*	0.9983
		C3–DTLZ4	5	9.572e-1 =	9.567e-1 –	9.579e-1 +*	0.9572
			8	9.961e-1 =	9.954e-1 –	9.962e-1 =*	0.9961
			10	9.994e-1 =	9.991e-1 –	9.994e-1 +*	0.9993
		DC2–DTLZ1	5	9.707e-1 =	9.662e-1 –	9.711e-1 =*	0.9709
			8	9.883e-1 =	9.568e-1 –	9.918e-1 +*	0.9894
			10	9.500e-1 –	9.230e-1 –	7.104e-1 –	0.9776*
		LYO1	5	7.119e-1 –	6.885e-1 –	7.132e-1 –	0.7248*
			8	5.834e-1 =	6.449e-1 +*	5.897e-1 =	0.5900
			10	6.747e-1 =	7.445e-1 +*	6.678e-1 =	0.6597
		LYO2	5	9.704e-1 =	9.701e-1 –	7.563e-1 –	0.9706*
			8	4.556e-1 –	9.909e-1 =*	3.617e-1 –	0.9741
			10	2.153e-1 –	9.909e-1 +*	2.196e-1 –	0.9644
		LYO3	5	6.605e-1 –	7.608e-1 =	1.491e-1 –	0.9121*
			8	2.660e-1 –	9.245e-1 =	6.584e-1 –	0.9486*
			10	4.523e-1 –	9.115e-1 =	6.109e-1 –	0.9280*
		LYO4	5	9.593e-1 –	9.599e-1 –	9.437e-1 =	0.9615*
			8	9.881e-1 –	9.918e-1 –	9.914e-1 =	0.9930*
			10	9.961e-1 –	9.971e-1 –	9.952e-1 =	0.9980*
	Summary (+/–/=)			0/10/11	3/14/4	4/8/9	

REFERENCES

[1] J. Yuan, H.-L. Liu, F. Gu, Q. Zhang, and Z. He, “Investigating the properties of indicators and an evolutionary many-objective algorithm using promising regions,” *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 75–86, 2021.

[2] F. Ming, W. Gong, D. Li, L. Wang, and L. Gao, “A competitive and cooperative swarm optimizer for constrained multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 5, pp. 1313–1326, 2023.

[3] Y. Tian, T. Zhang, J. Xiao, X. Zhang, and Y. Jin, “A coevolutionary framework for constrained multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 1, pp. 102–116, 2021.

[4] K. Yu, J. Liang, B. Qu, Y. Luo, and C. Yue, “Dynamic selection preference-assisted constrained multiobjective differential evolution,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 5, pp. 2954–2965, 2022.

[5] J. Yuan, H.-L. Liu, Y.-S. Ong, and Z. He, “Indicator-based evolutionary algorithm for solving constrained multiobjective optimization problems,” *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 2, pp. 379–391, 2022.

[6] K. Qiao, K. Yu, B. Qu, J. Liang, H. Song, C. Yue, H. Lin, and K. C. Tan, “Dynamic auxiliary task-based evolutionary multitasking for constrained multiobjective optimization,” *IEEE Transactions on Evolutionary Computation*, vol. 27, no. 3, pp. 642–656, 2023.

[7] J. Yuan, H.-L. Liu, and S. Yang, “An adaptive parental guidance strategy and its derived indicator-based evolutionary algorithm for multi-and many-objective optimization,” *Swarm and Evolutionary Computation*, vol. 84, p. 101449, 2024.

[8] J. Yuan, S. Yang, and W.-L. Yan, “An adaptive uniform search framework for constrained multi-objective optimization,” *Applied Soft Computing*, vol. 162, p. 111800, 2024.

TABLE A-2: The comparison results among DDEEA-D, DDEEA-I, DDEEA-M, and DDEEA in T2 and T3 test series.

Metric	Type	Problem	M	DDEEA-D	DDEEA-I	DDEEA-M	DDEEA
IGD	T2	C2–DTLZ2	5	1.712e-1 =	1.661e-1 +*	1.739e-1 –	0.1714
			8	2.677e-1 =	2.620e-1 +*	2.738e-1 –	0.2682
			10	2.792e-1 =	2.768e-1 +*	2.848e-1 –	0.2807
		DC1–DTLZ1	5	9.883e-2 =	9.904e-2 –	9.782e-2 =	0.0976*
			8	3.880e-1 =*	4.288e-1 –	4.365e-1 =	0.3936
			10	3.441e-1 –	3.261e-1 –	3.181e-1 =	0.3086*
		DC3–DTLZ1	5	9.909e-2 =	7.026e-2 +*	2.036e+0 –	0.0777
			8	3.216e+0 –	2.812e-1 =*	4.347e+0 –	1.4000
			10	4.516e+0 =	5.417e+0 =	4.498e+0 =	3.8800*
	T3	LYO5	5	1.384e-1 =	1.393e-1 –	1.386e-1 =	0.1381*
			8	2.310e-1 =	2.370e-1 –	2.289e-1 +*	0.2304
			10	2.496e-1 =	2.561e-1 –	2.474e-1 =*	0.2489
		LYO6	5	6.671e-2 =	6.416e-2 +*	6.870e-2 =	0.0666
			8	9.111e-2 =	9.217e-2 =	9.020e-2 =*	0.0915
			10	9.018e-2 =	9.120e-2 =	8.855e-2 =	0.0883*
		LYO7	5	2.029e-1 +*	2.145e-1 –	2.055e-1 =	0.2056
			8	3.659e-1 –	3.729e-1 –	3.663e-1 =	0.3633*
			10	4.246e-1 –	4.318e-1 –	4.168e-1 +*	0.4193
		LYO8	5	1.492e-1 –	1.564e-1 –	1.466e-1 =*	0.1470
			8	2.613e-1 =	2.704e-1 –	2.597e-1 =*	0.2601
			10	2.886e-1 =	3.039e-1 –	2.887e-1 =	0.2880*
	Summary (+/–/=)			1/5/15	5/12/4	2/5/14	
HV	T2	C2–DTLZ2	5	7.067e-1 =	7.201e-1 +*	6.996e-1 –	0.7062
			8	8.227e-1 =	8.337e-1 +*	8.111e-1 –	0.8209
			10	8.863e-1 =	8.969e-1 +*	8.762e-1 –	0.8842
		DC1–DTLZ1	5	8.067e-1 =	8.056e-1 –	8.069e-1 =*	0.8068
			8	5.936e-1 =	5.514e-1 –	5.654e-1 =	0.5940*
			10	8.167e-1 –	8.158e-1 –	8.277e-1 =	0.8283*
		DC3–DTLZ1	5	7.235e-1 –	7.409e-1 –	5.268e-2 –	0.7467*
			8	8.862e-2 –	5.339e-1 =*	0.000e+0 –	0.4507
			10	4.756e-2 =*	3.165e-2 =	2.441e-2 =	0.0009
	T3	LYO5	5	9.712e-1 =*	9.700e-1 –	9.711e-1 =	0.9711
			8	9.968e-1 =	9.960e-1 –	9.969e-1 =*	0.9969
			10	9.996e-1 =	9.994e-1 –	9.996e-1 =*	0.9996
		LYO6	5	9.997e-1 =*	9.997e-1 =	9.997e-1 =	0.9997
			8	1.000e+0 =	1.000e+0 =	1.000e+0 =*	1.0000
			10	1.000e+0 =*	1.000e+0 =	1.000e+0 =	1.0000
		LYO7	5	7.793e-1 =	7.574e-1 –	7.763e-1 –	0.7803*
			8	8.965e-1 –	8.827e-1 –	8.984e-1 –	0.9024*
			10	9.447e-1 –	9.336e-1 –	9.498e-1 =	0.9511*
		LYO8	5	9.626e-1 –	9.590e-1 –	9.643e-1 =*	0.9642
			8	9.935e-1 –	9.919e-1 –	9.939e-1 =	0.9939*
			10	9.983e-1 –	9.974e-1 –	9.984e-1 =	0.9984*
	Summary (+/–/=)			0/8/13	3/13/5	0/7/14	