2023 IEEE Symposium Series on Computational Intelligence (SSCI)
Mexico City, Mexico. December 5-8, 2023

An Adaptive Multiform Evolutionary Algorithm for Global Continuous Optimization

Hongtao Ling, Jinghui Zhong*, Junlan Dong, Shanxia Wang, Shibin Wang, and Qin Zhang

***Abstract*—Evolutionary algorithm(EA) with knowledge transfer is an emerging research topic that has attracted a lot of attention in the EA community. The multiform evolutionary algorithm is a promising algorithm framework concept falling in this direction, but little work has been done so far to design and discuss this new EA framework. In this paper, we proposed an adaptive multiform evolutionary framework, which integrates the idea of transfer optimization and population-based evolutionary algorithms. The main idea is to utilize multiple equivalent or similar formulations to solve the given problem cooperatively. By adaptively adjusting computational resources of different formulations and transferring knowledge among formulations, the search efficiency and the population diversity can be improved. Furthermore, a new multiform algorithm is implemented based on the proposed framework, which utilizes two different coordinate systems, namely the Cartesian and polar coordinate systems, to model the given problem. To test the effectiveness of the proposed algorithm, we conducted numerical experiments on CEC2013 benchmark test functions, and the experimental results have verified the efficacy of the proposed algorithm.**

***Index Terms*—Multiform Optimization, DE, Continuous Optimization**

I. INTRODUCTION

Evolutionary algorithms (EAs) are population-based stochastic search algorithms, which have attracted a lot of attention owing to their flexible representation, ease of implementation, and strong global search ability. Over the past decades, a number of EA variants have been proposed and applied to a wide range of applications, including combinatorial optimization problems [1], continuous optimization problems [2], multi-objective optimization problems [3], and dynamic optimization problems [4].

However, existing EAs focus on solving the given problem through a single formulation modeling. As the saying goes: ”All roads lead to Rome”, in practical applications, a given problem usually can have multiple modeling methods. However, it is usually unclear which modeling strategy is the most effective one for the given problem. Modeling a given problem from different perspectives can result in diverse answers. By analyzing and constructing multiple solutions from different perspectives, we can gain a more comprehensive and deeper understanding of the given problem. To make it clearer, we have used an illustration to explain it, as shown in Fig. 1. There are currently about 500 proofs of the Pythagorean Theorem. In Fig. 1, two of the famous methods are shown, which are Euclidean proof and Shuang Zhang proof. Euclid proved it by the similarity of triangles and the area of triangles. In contrast, Shuang Zhang argues by the sum of the area of a square and four triangles.

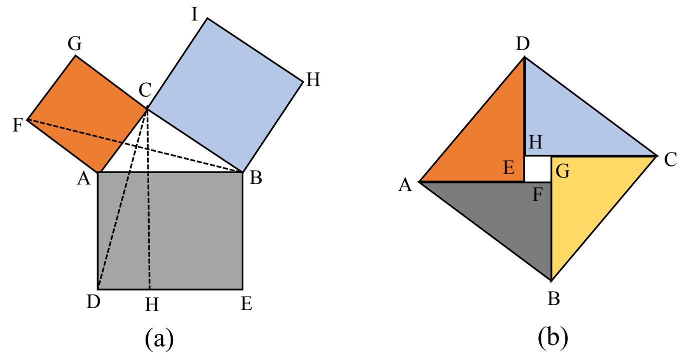

Fig. 1: An example of multiple proof methods to the Pythagorean Theorem. (a) Euclidean proof. (b) Shuang Zhang proof.

Recently, Ong *et al.* [5] proposed the concept of multiform optimization. Multiform optimization applies alternate formulations to a single objective optimization problem. The use of different formulations in a single optimization problem may capture different properties or situations, and the search experience gained from these alternate formulations may help to obtain a better solution to the objective problem [6]. The latest evolutionary transfer optimization (ETO) literature review also indicates that multiform can have great potential to improve the search efficacy of EA [7]. However, multiform EA is still in its infancy and lacks sufficient research attention. Recently, Da *et al.* [8] have proposed a multiform EA for the traveling salesman problem (TSP) and has shown encouraging results, but this algorithm only focuses on the TSP problem and could not be applied to continuous optimization problems.

To solve the above issues, this paper proposes a general and adaptive multiform EA framework for global continuous optimization. In the proposed framework, the given problem is formulated into multiple equivalent modeling forms. Each of these forms can act as a catalyst in the multitasking environment, thereby leveraging the unique advantages offered by each formulation through a continuous process of knowledge transfer. Based on the general multiform framework, we

H. Lin, J. Zhong, and J. Dong are with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510006, China (e-mail:1025813748@qq.com;jinghuizhong@scut.edu.cn;1135580174@qq.com;).

S. Wang and S. Wang are with the Henan Normal University, Xinxiang, China, e-mail: 121085@htu.edu.cn; wangshibin@htu.edu.cn

Q. Zhang is with the Communication University of China, Beijing, China, e-mail: zhangqin@cuc.edu.cn

* corresponding author.

978-0-7381-4408-5/23/$31.00 ©2023 IEEE

further propose an adaptive multiform EA algorithm that can automatically generate and configure Evolutionary Transfer Optimization algorithms for different spatial coordinates of a given problem. In general, the contributions of this paper are as follows:

- We propose a general multiform EA framework, which no longer performs evolutionary search only on a single search space, but rather performs evolutionary search alternately on multiple search spaces by passing useful information in constant interaction.
- A specific multiform EA is designed based on the proposed framework, which is implemented by a classic adaptive differential EA named JADE [30]. In the proposed algorithm, two different coordinate systems, namely the Cartesian and polar coordinate systems, are utilized to model the given problem. The interactions of the two formulations during the evolution are presented in detail to facilitate the practical application of the proposed framework.
- A suite of experiments on benchmarks are set up to prove the effectiveness of the proposed framework. The experimental results show that the multiform EA exhibits good performance on continuous optimization problems.

II. BACKGROUND AND RELATED WORK

A. Evolutionary transfer optimization

Evolutionary transfer optimization has emerged as a hot research topic in evolutionary computing, with the goal of improving optimization efficiency and performance by combining EA with knowledge acquisition and migration in adjacent domains. Recently, many ETO methods have been proposed to solve various optimization problems, including multitask optimization, dynamic optimization, and complex optimization [9]–[11].

Existing ETO methods can be categorized into two types: homogeneous ETO and heterogeneous ETO [7]. The goal of homogeneous ETO is to transfer knowledge in the evolutionary search between issues in the same search space. For example, Feng *et al.* [12] proposed a single-layer denoising autoencoder that can transfer knowledge from past multi-objective optimization. Also, there are studies using kernel-based, manifold transfer learning, and knee point-based imbalance learning to obtain optimal solutions using knowledge learned from past search experiences [13], [14]. Ma *et al.* [15] provided a two-level transfer learning strategy for multiple tasks, including upper-level inter-task knowledge transfer learning and lower-level intra-task transfer learning. Chaabani *et al.* [16] proposed to combine the algorithm of co-evolutionary decomposition with migration learning to improve search efficiency. The goal of heterogeneous ETO is to learn knowledge from problems in different search spaces. For example, Iqbal *et al.* [17], [18]transfer learning-based genetic programming which can learn useful features from simple problems to help solve complex problems. Bali *et al.* [19] proposed a linearized domain adaptive strategy to improve performance by converting the search space for simple tasks to a search space similar to that of complex tasks.

B. Multitask evolutionary Optimization

While both single-objective and multi-objective optimizations try to solve a single problem each time, the concept of multitask optimization (MTO) is proposed to solve several problems simultaneously. Based on the assumption of implicit parallelism, each task(problem) in MTO can help accelerate the optimization of each other by sharing useful information. The mathematical form of MTO is defined as follows,

$$\boldsymbol{x}_i^* = \underset{\boldsymbol{x}\in\Omega_i}{\arg\min} f_i(\boldsymbol{x}), i = 1, 2, \ldots, n \tag{1}$$

where $f_i(x)$ denotes the fitness value of task i and Ω_i are the decision space of task i.

With the advancement of MTO, some studies have used ETO to solve multitask optimization problems. Gupta *et al.* [10] proposed a multifactor evolutionary algorithm (MFEA) in which cross-task knowledge transfer is achieved through the crossover process. Later, many improved MFEA variants were proposed [20], [22], [23]. In addition to using crossover for knowledge transfer, matrix multiplication can be used to transfer optimum answers from one task to another [24], [25]. Da *et al.* [8] proposed to use transferable knowledge to provide useful inductive biases for search progress, thus overcoming local optima and effectively guiding populations towards. Further, there are many high-quality algorithms based on multiple populations have also been proposed [26], [27]. In the latest literature, Wei et al. [21] provide a detailed analysis and elaboration of evolutionary multitask optimization studies.

C. Multiform evolutionary Optimization

Multiform optimization differs from traditional single-task optimization in that it can simplify the structure of the original task or extract useful knowledge from the original task to enhance optimization performance. Useful information is found from the evolutionary search of different formulations, and the search is transferred between different formulations by multitasking.

In the literature, Zhang *et al.* [28] proposed a multiform optimization algorithm based on noise-reducing self-encoder and Gaussian process regression, which is experimentally demonstrated in the commonly used MOP benchmark. In addition, Jiao *et al.* [29] proposed a multiform optimization framework that uses different problem formulations to improve the constraint processing capability of constrained multiobjective optimization problems (CMOP). The results show that the derived CMOP formulation can be used as an auxiliary task to enhance the search processes through knowledge transfer. Although the current research has achieved encouraging results, the research of multiform EA is still in the nascent stage.

III. PROPOSED MULTIFORM EA

In this section, we proposed an adaptable multiform framework at first, providing a seamless integration for EAs. Subsequently, a multiform differential evolutionary algorithm is described in detail based on the proposed multiform framework.

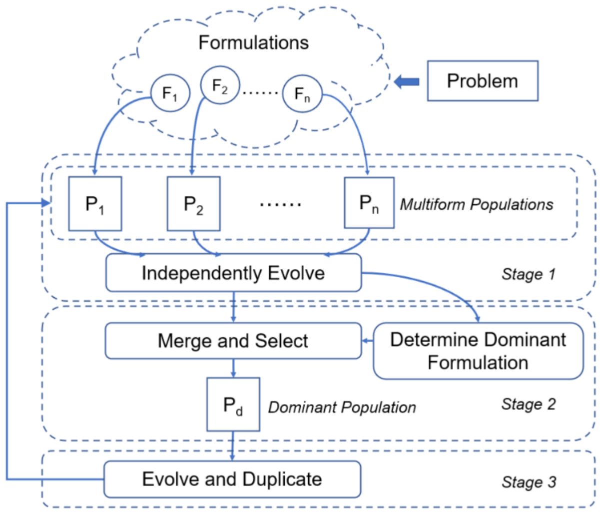

Fig. 2: The multiform Framework

To begin with, the basic idea of the proposed multiform framework is reconciling different formulations to work together for a single problem while keeping computational resources under control to improve search efficiency. To achieve this goal, we proposed to model the given problem into multiple equivalent or similar formulations by transformations and then utilize different solvers to address these formulations cooperatively. In detail, the proposed framework consists of three cyclic stages, as illustrated in Fig. 2, which are 1) *independent evolving*, 2) *merge and selection*, and 3) *dominant evolving*.

Hereby, two formulations, namely the Cartesian coordinate system (denoted as F_C) and polar coordinate system (denoted as F_P), are incorporated into the framework as shown in Fig. 3. The motivation for considering the two coordinate systems as formulations is summarized below. First of all, they can be transformed into each other with a group of formulas, which construct the basis of this evolutionary algorithm. Besides, their function landscapes are quite different, which may make a difference in the convergence pace so that the framework can reconcile. Finally, the two systems are frequently used in the real world. Therefore, they can readily be associated to serve as two formulations.

Commonly, each individual of the two formulations is represented by variable $\mathbf{X}_C$ or variable $\mathbf{X}_P$ respectively: $\mathbf{X}_C = [x_1, x_2, \ldots, x_D]^T$, $\mathbf{X}_P = [\rho, \theta_1, \ldots, \theta_{D-1}]^T$, where D is the dimension of the problem. Notably, the polar coordinate should be confined within the extra angle constraints:

$$\begin{aligned} &-\frac{\pi}{2} \leq \theta_i \leq \frac{\pi}{2}, \text{ for i} = 1, \ldots, D-2 \\ &-\pi \leq \theta_{D-1} \leq \pi \end{aligned} \tag{2}$$

It should be noted that the two coordinate systems also differ in the expression of boundary constraints. For Cartesian coordinates, the solution space is usually limited with the

Algorithm 1 The Procedure of Multiform JADE

Initialization: Generate the Cartesian population P_C with the uniform distribution (3); Generate the polar population P_P with the uniform distribution (2) and $0 \leq \rho \leq 1$; Calculate $\boldsymbol{C}$ and $\boldsymbol{S}$ with the given $\mathbf{L}$ and $\mathbf{U}$. /* by (6) */
while termination conditions not met **do**
 initialize parameters of JADE and perform JADE solvers of P_C and P_P for t_1 generations.
 calculate evolutionary efficiency w of P_C and P_P and determine dominant formulation F_d. /* by (8) */
 merge P_C and P_P to form the mixed population P_m.
 sort P_m on the fitness
 select top *NP* individuals from P_m and transform them into F_d to form the dominant population P_d. /* by (4), (5) and (7) */
 initialize parameters of JADE and perform JADE solver of P_d for t_2 generations.
 duplicate P_d into two copies and transform one copy into another formulation to form the new P_C and P_P.
end while

lower and upper boundary constraint vectors, $\mathbf{L} = [l_1, \ldots, l_n]^T$ and $\mathbf{U} = [u_1, \ldots, u_n]^T$, respectively.

$$\begin{aligned} &\mathbf{L} \leq \mathbf{X} \leq \mathbf{U} \\ &l_i \leq x_i \leq u_i, \text{ for } i = 1, \ldots, n \end{aligned} \tag{3}$$

The transformation between Cartesian and polar coordinate systems can be summarized into simplified formats [31]:

$$Cartesian \leftarrow Polar:$$

$$\begin{cases} x_1 &= \rho \prod_{j=1}^{n-1} sin\theta_j \\ x_i &= \rho cos\theta_{n-i+1} \prod_{j=1}^{n-i} sin\theta_j, \text{for i} = 2, \ldots, n-1 \\ x_n &= \rho cos\theta_1 \end{cases} \tag{4}$$

$$Polar \leftarrow Cartesian:$$

$$\begin{cases} \rho = \sqrt{\sum_{j=1}^{n} x_j^2} \\ \theta_i = arctan\frac{\sqrt{\sum_{j=1}^{n-i} x_j^2}}{x_{n+i-1}}, \text{for i} = 1, \ldots, n-1 \end{cases} \tag{5}$$

Under the constraint of the two vectors, the solution space is constrained as a (hyper)cube. However, the polar coordinate system is more suitable for representing the (hyper)spherical space. To alleviate the complexity of boundary constraints expressed by polar coordinates, especially those that represent the space where the centroids deviate from the origin, two vectors $\mathbf{C} = [c_1, \ldots, c_n]^T$ and $\mathbf{S} = [s_1, \ldots, s_n]^T$ are utilized. $\mathbf{C}$ is used to store the centroid coordinate of solution space in the Cartesian coordinate system. $\mathbf{S}$ can scale up or down

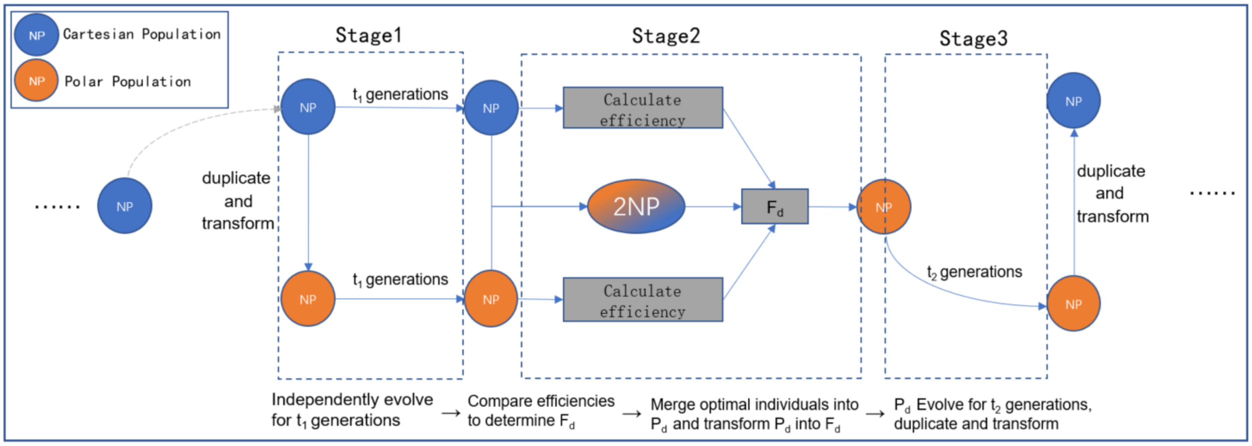

Fig. 3: The Multiform JADE Algorithm

the unit (hyper)sphere to fit Cartesian solution space which is usually a (hyper)cube as close as possible. They can be calculated by

$$c_i = \frac{u_i + l_i}{2}$$
$$s_i = \frac{u_i - l_i}{2} \tag{6}$$

Supposing $\mathbf{X}_P$ is transformed to Cartesian coordinate $\mathbf{X}_C^*$ by (4) and (5). Hereafter, the overall transformation equation is

$$\mathbf{X}_C = \mathbf{C} + \mathbf{S} \circ \mathbf{X}_C^* \tag{7}$$

where $\circ$ is the sign of the element-wise product. Hence, We can use a unit (hyper)sphere with $\mathbf{C}$ and $\mathbf{S}$ to approximately represent the given Cartesian search space. By this expression, there is no additional boundary constraint for polar coordinates during coordinates conversion except (2) and $0 \leq \rho \leq 1$.

In the following part, the implementation of the multiform JADE is presented in a step-by-step manner.

Step 1-*Initialization*: First of all, the vectors $\mathbf{C}$ and $\mathbf{S}$ are calculated according to the given boundary constraints $\mathbf{L}$ and $\mathbf{U}$ by (6). Then, the Cartesian population is randomly generated with the uniform distribution $\mathbf{L} \leq \mathbf{X}_C \leq \mathbf{U}$. Similarly, the polar population is generated with the uniform distribution by (2) and $0 \leq \rho \leq 1$. Both populations have the same size *NP*. Besides, JADE solvers will initialize their parameters as the default settings of the original JADE.

Step 2-*Indepedent Evolving*: With JADE as the EA solver, the Cartesian and polar populations independently evolve for t_1 generations without any interactions. At this step, the two populations update the parameters of JADE and record the best fitness of each population in order to evaluate the dominant formulation(F_d).

Step 3-*Merge and Selection*: To be simple, all individuals of both populations will be sorted on the objective function value. Then, the top *NP* optimal individuals will be selected to form the new dominant population P_d. Meanwhile, those individuals that are not F_d will be transformed by (4) or (5) into F_d.

Step 4-*Dominant Evolving*: Because the dominant formulation was determined, the dominant population P_d will evolve for t_2 generations. Before the next independent evolving step, it replicates itself into Cartesian and polar populations. For instance, if the current F_d is F_C, P_d will be duplicated and one copy of it will be transformed into another formulation, namely F_P.

Except for the initialization, the above steps are repeated until the termination conditions are met. The whole procedure of the multiform JADE is described in Algorithm 1.

IV. EXPERIMENT STUDIES

In this section, a series of numerical optimization experiments and corresponding result analyses are presented to demonstrate the effectiveness and efficiency of the proposed Multiform JADE. All experiments are conducted using Python Language on a PC with Intel(R) Core i7-11800H 2.30 GHz and 16GB RAM.

A. Benchmark Functions and Parameter Settings

For the experiment, we adopt the commonly used CEC2013 benchmark functions [32] for testing. Among the benchmark functions, $F1$-$F5$ are unimodal functions, while $F6$-$F20$ are multimodal functions.

For each test problem, the dimension is set as 10 and each algorithm runs for 30 independent times. Besides, the population size *NP* is set as 100. The termination conditions are set to *TES*, the function value of termination, equals 1E-8 and *MaxGen*, the maximum generation, equals 1000. Once either of the two conditions is satisfied, this independent run terminates.

B. Results and Discussion

Table I shows the optimization results of the multiform JADE compared to the standard JADE and the polar JADE which utilize the Cartesian and the polar coordinate systems as formulations, respectively. To further explore how the performance of the multiform JADE is influenced by the two coefficients t_1 and t_2, we fixed t_1 as 5, the smaller the better to be fair, and varied t_2 from 20, 30 to 50. Besides, a

TABLE I: the Median Results on the CEC-2013 Test Benchmark Functions with 10 dimensions.

Function	Standard JADE	Polar JADE	Multiform JADE		
			$t_1 = 5$		
			$t_2 = 20$	$t_2 = 30$	$t_2 = 50$
F1	1.61E−7	1.28E−6	**1.04E−8**(+,+)	2.30E−8(+,+)	2.63E−8(+,+)
F2	**1.03E−8**	1.78E+6	1.37E−4(−,+)	4.03e−6(−,+)	8.49E−1(−,+)
F3	**1.63E−8**	8.53E+1	5.31E−4(−,+)	3.98E+1(−,≈)	2.57E+1(−,+)
F4	8.63E+3	1.18E+4	**1.03E+4**(≈,+)	1.08e+4(≈,+)	8.68E+3(≈,+)
F5	3.28E−8	1.07E−4	**1.03E−8**(+,+)	2.45E−8(≈,+)	9.28E−8(−,+)
F6	9.81E+0	1.03E−4	1.09E−7(+,+)	**1.16E−8**(+,+)	1.03E−7(+,+)
F7	1.27E−8	1.26E−2	**1.01E−8**(≈,+)	4.07E−8(≈,+)	1.34E−7(≈,+)
F8	2.02E+1	2.02E+1	**2.01E+1**(+,+)	**2.01E+1**(+,+)	**2.01E+1**(+,+)
F9	3.09E+0	1.74E+0	9.99E−1(+,+)	**7.89E−1**(+,+)	1.50E+0(+,+)
F10	8.84E−3	1.35E−2	**1.02E−8**(+,+)	1.31E−4(≈,≈)	2.66E−4(≈,≈)
F11	2.43E−8	9.12E+0	**1.23E−8**(≈,+)	1.83E−7(≈,+)	6.67E−6(−,+)
F12	**5.57E+0**	7.38E+0	6.47E+0(−,≈)	6.47E+0(−,≈)	6.03E+0(−,≈)
F13	8.17E+0	1.27E+1	1.01E+1(−,≈)	8.97E+0(≈,≈)	**7.79E+0**(≈,≈)
F14	**1.27E−4**	2.49E+2	6.26E−2(−,+)	6.26E−2(−,+)	6.26E−2(−,+)
F15	4.92E+2	**2.09E+2**	3.11E+2(+,−)	3.74E+2(+,−)	2.77E+2(+,≈)
F16	4.16E−1	3.82E−1	2.94E−1(+,≈)	3.01E−1(+,+)	**2.32E−1**(+,+)
F17	**1.01E+1**	2.34E+1	1.08E+1(−,+)	1.04E+1(−,+)	1.03E+1(−,+)
F18	1.96E+1	2.77E+1	1.98E+1(≈,+)	2.05E+1(≈,+)	**1.93E+1**(≈,+)
F19	6.30E−1	6.48E−1	6.36E−1(≈,+)	5.62E−1(≈,≈)	**5.00E−1**(+,+)
F20	2.54E+0	2.90E+0	**2.36E+0**(+,+)	2.38E+0(≈,+)	2.52E+0(≈,+)
+ (v.s. Standard JADE, Polar JADE)			(9,15)	(6,15)	(7,16)
− (v.s. Standard JADE, Polar JADE)			(6,1)	(5,1)	(7,0)
≈ (v.s. Standard JADE, Polar JADE)			(5,4)	(9,4)	(6,4)

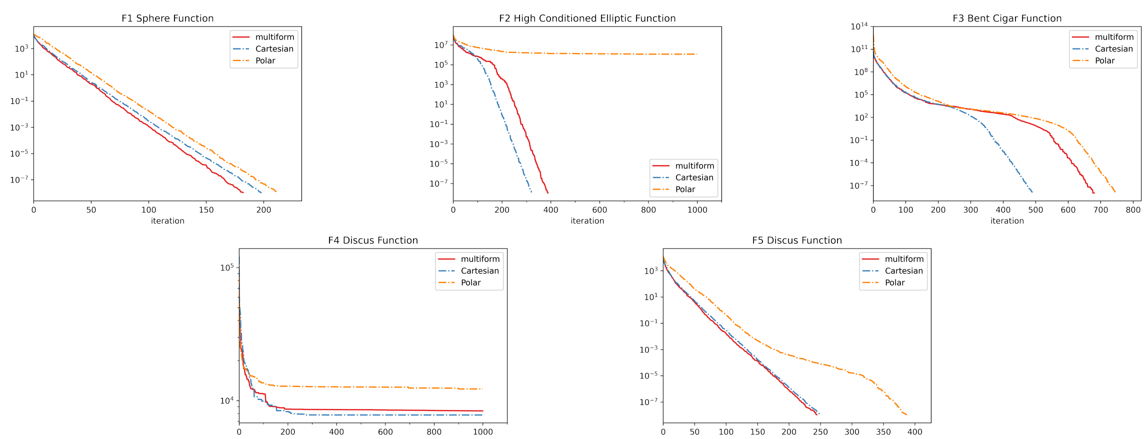

Fig. 4: Convergence Curves of the Three Algorithms on Unimodal Functions $F1$-$F5$

Wilcoxon's Rank Sum Test [33] at α = 0.05 was conducted to show the statistical significance between the multiform JADE and the other two formulations of JADE under the above three settings of t_2. In Table I, We use $+$,$-$, and $\approx$ to denote "significantly better", "significantly worse", and "not significantly different", respectively. Through the comparison of the experimental results, the multiform JADE has exhibited a promising performance compared with the two basic single-formulation JADEs.

One potential explanation for the results is that the constant switching between the two formulations during the evolutionary process keeps the solver evolving in the optimal formulation overall. For some functions, the currently used formulations may not be conducive to jumping out of the domain of local optima. Nonetheless, the multiform JADE alters it to another formulation after comparing the evolutionary efficiencies, which enables the fitness to converge faster.

Not surprisingly, the standard JADE and the polar JADE obtain optimal performance on several functions as well. This is probably because the determination of the dominant formulation is temporally wrong during periods of evolution in these problems. If the historical information is not sufficient to correctly evaluate the suitable formulation, the dominant computational resource will be falsely allocated to a less suitable formulation, which will slow down the search efficiency. As the latency of historical information is inevitable, the ratio of t_1 to t_2 plays an essential role because the smaller ratio means less frequency of updating the dominant formulation. More latency brings about less accuracy. As shown in Table I, the multiform JADE with the smallest ratio of t_1 to t_2 outperforms the other two multiform JADE.

The convergence curves of the three algorithms on all test functions are illustrated in Fig. 4 and Fig. 5 where the parameter t_2 of multiform JADE is 30. Among the figures, multiform JADE is almost the best or the second best in every generation. Nevertheless, early convergence can also be found on the curves of $F14$ and $F15$. It may result from the loss of accuracy in converting coordinates, which does damage to the effect of differential vectors that can push an individual close to the current optimal solution. On $F10$, multiform JADE exhibits an extraordinary search performance over the two single formulation JADEs, which demonstrates that the framework successfully evaluates the best formulation for each stage to accelerate the evolution by alternating the two formulations. Much to our delight, the curve of $F6$ illustrates that polar formulation is much better than Cartesian formulation, which verifies the idea that different formulations may be suitable for different functions for specific periods.

In short, although this multiform framework combined with JADE does not achieve the best results on all test functions, it shows an overall best performance compared with JADE and its polar formulation. Furthermore, the curves in Fig. 4 and Fig. 5 suggest that the polar coordinate system, as a formulation, actually performs better on specific functions in contrast with the Cartesian one.

V. Conclusion

In this paper, a multiform evolutionary framework based on the concept of ETO is proposed, which can reconcile different formulations of a single problem. The core idea is to make full use of different formulations to accelerate the evolutionary process. Moreover, a specific algorithm named multiform JADE is designed based on the proposed framework and JADE. The proposed method is tested on the CEC2013 benchmark test functions and the acquired results proved their superiority and the effectiveness of the proposed method.

In the future, we plan to consider utilizing more formulations and better formulation evaluation mechanisms to further improve the search performance. Besides, extending the proposed framework to multi-objective optimization applications is another promising research topic.

VI. ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62076098),in part by the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2023A1515012291), and in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X183)

References

[1] Z. M. Huang, W. N. Chen, Q. Li, X. N Luo, H. Q. Yuan, J. Zhang. Ant colony evacuation planner: An ant colony system with incremental flow assignment for multipath crowd evacuation. IEEE Transactions on Cybernetics, vol. 51, no. 11, pp. 5559-5572, 2020.

[2] Z. H. Zhan, L. Shi, K. C. Tan, J. Zhang. A survey on evolutionary computation for complex continuous optimization. Artificial Intelligence Review, pp. 1-52, 2022.

[3] Y. Tian, R. Liu, X. Zhang, H. P. Ma, K. C. Tan, Y. C. Jin. A multi-population evolutionary algorithm for solving large-scale multimodal multi-objective optimization problems. IEEE Transactions on Evolutionary Computation, vol. 25, no. 3, pp. 405-418, 2020.

[4] Yazdani D, R. Cheng, Yazdani D, Jürgen B, Y. C. Jin, X. Yao. A survey of evolutionary continuous dynamic optimization over two decades—Part A. IEEE Transactions on Evolutionary Computation, vol. 25, no. 4, pp. 609-629, 2021.

[5] A. Gupta, Y. S. Ong. Back to the roots: Multi-x evolutionary computation. Cognitive Computation, vol. 11, no. 1, pp. 1-17, 2019.

[6] A. Gupta, Y. S. Ong, L. Feng. Insights on transfer optimization: Because experience is the best teacher. IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 2, no. 1, pp. 51-64, 2017.

[7] K. C. Tan, L. Feng, Jiang M. Evolutionary transfer optimization-a new frontier in evolutionary computation research. IEEE Computational Intelligence Magazine, vol. 16, no. 1, pp. 22-33, 2021.

[8] B. Da, A. Gupta, Y.S. Ong, L. Feng. Evolutionary multitasking across single and multi-objective formulations for improved problem solving. IEEE Congress on Evolutionary Computation (CEC), pp. 1695-1701, 2016.

[9] M. Jiang, Z. Q. Huang, L. M. Qiu, W. Z. Huang, G. G. Yen. Transfer learning-based dynamic multiobjective optimization algorithms. IEEE Transactions on Evolutionary Computation, vol. 22, no. 4, pp. 501-514, 2017.

[10] A. Gupta, Y. S. Ong, L. Feng. Multi-factorial evolution: toward evolutionary multitasking. IEEE Transactions on Evolutionary Computation, vol. 20, no. 3, pp. 343-357, 2015.

[11] L. Feng, Y. S. Ong, Lim M H, I. W. Tsang. Memetic search with interdomain learning: A realization between CVRP and CARP. IEEE Transactions on Evolutionary Computation, vol. 19, no. 5, pp. 644-658, 2014.

[12] L. Feng, Y. S. Ong, S. Jiang, A. Gupta. Autoencoding evolutionary search with learning across heterogeneous problems. IEEE Transactions on Evolutionary Computation, vol. 21, no. 5, pp. 760-772, 2017.

[13] G. Ruan, L. L. Minku, S. Menzel, M. Z. Stefan, S. Bernhard, X. Yao. When and how to transfer knowledge in dynamic multi-objective optimization. IEEE Symposium Series on Computational Intelligence (SSCI), pp. 2034-2041, 2019.

[14] M. Jiang, Z. Wang, H. Hong, G. G. Yen. Knee point-based imbalanced transfer learning for dynamic multiobjective optimization. IEEE Transactions on Evolutionary Computation, vol. 25, no. 1, pp. 117-129, 2020.

[15] X. L. Ma, Q. J. Chen, Y. N Yu, Y. W. Sun, L. J. Ma, Z. X. Zhu. A two-level transfer learning algorithm for evolutionary multitasking. Frontiers in Neuroscience, vol. 13, pp. 1408, 2020.

[16] A. Chaabani, L.B. Said. Transfer of learning with the co-evolutionary decomposition-based algorithm-II: a realization on the bi-level production-distribution planning system. Applied Intelligence, vol. 49, no. 3, pp. 963-982, 2019.

[17] M. Iqbal, B. Xue, M. J. Zhang. Reusing extracted knowledge in genetic programming to solve complex texture image classification problems, Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, pp. 117-129, 2016.

[18] M. Iqbal, B. Xue, H. Al-Sahaf, M. J. Zhang. Cross-domain reuse of extracted knowledge in genetic programming for image classification. IEEE Transactions on Evolutionary Computation, vol. 21, no. 4, pp. 569-587, 2017.

[19] K. K. Bali, A. Gupta, L. Feng, Y. S. Ong, T. P. Siew. Linearized domain adaptation in evolutionary multitasking. IEEE Congress on Evolutionary Computation (CEC), pp. 1295-1302, 2017.

[20] J. L. Ding, C. Yang, Y. C. Jin, T. Y. Chai. Generalized multitasking for evolutionary optimization of expensive problems. IEEE Transactions on Evolutionary Computation, vol. 23, no. 1, pp. 44-58, 2017.

[21] T. Y. Wei, S. B. Wang, J. H. Zhong, D. Liu, J. Zhang. A Review on Evolutionary Multi-Task Optimization: Trends and Challenges. IEEE

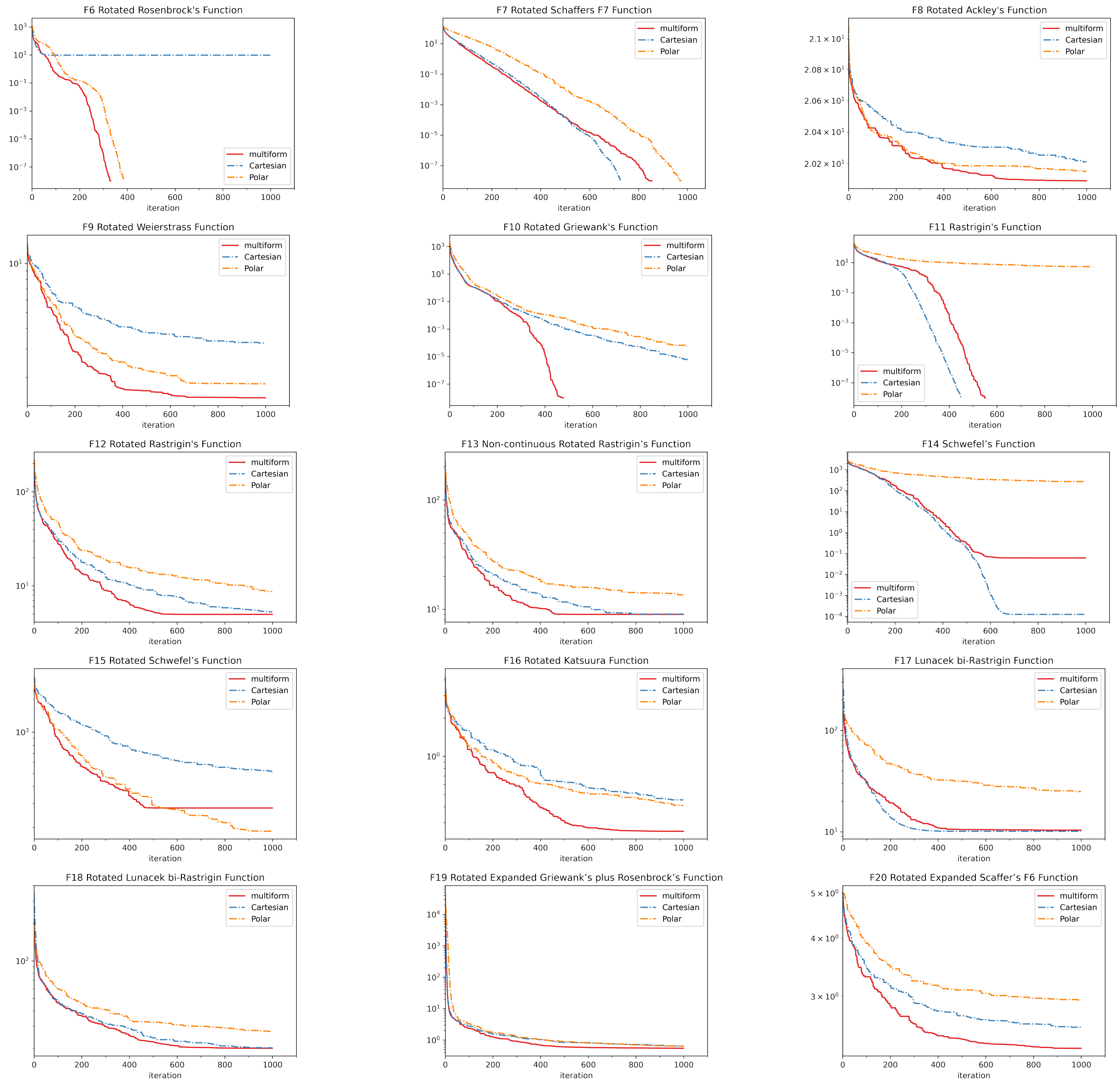

Fig. 5: Convergence Curves of the Three Algorithms on Multimodal Functions $F6$-$F20$

Transactions on Evolutionary Computation, vol. 26, no. 5, pp. 941-960, 2022,

[22] J. H. Zhong, L. Feng, W. T. Cai , Y. S. Ong. Multifactorial genetic programming for symbolic regression problems. IEEE transactions on systems, man, and cybernetics: systems, vol. 50, no. 11, pp. 4492-4505, 2018.

[23] K. K. Bali, Y. S. Ong, A. Gupta, P. S. Tan. Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II. IEEE Transactions on Evolutionary Computation, vol. 24, no. 1, pp. 69-83, 2019.

[24] L. Feng, L. Zhou, J. H. Zhong, A. Gupta; Y. S. Ong, K. C. Tan, A. K. Qin. Evolutionary multitasking via explicit autoencoding. IEEE transactions on cybernetics, vol.49, no. 9, pp. 3457-3470, 2018.

[25] L. Feng, Y. X. Huang, L. Zhou, J. H. Zhong, A. Gupta, K. Tang, K.C. Tan. Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem. IEEE transactions on cybernetics, vol. 51, no. 6, pp. 3143-3156, 2020.

[26] Y. L Chen, J. H Zhong, L. Feng, J. Zhang. An adaptive archive-based evolutionary framework for many-task optimization. IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 4, no. 3, pp. 369–384, 2020.

[27] S. j Huang, J. H. Zhong, W. J. Yu. Surrogate-assisted evolutionary framework with adaptive knowledge transfer for multi-task optimization. IEEE Transactions on Emerging Topics in Computational Intelligence, vol. 9, no. 4, pp. 1930–1944, 2021.

[28] L. J. Zhang, Y. L. Xie, J.J. Chen, L. Feng, C. Chen, K. Liu. A study on multiform multi-objective evolutionary optimization. Memetic Computing, vol. 13, no. 3, pp. 307-318, 2021.

[29] R. W. Jiao, B. Xue; M. J. Zhang. A Multiform Optimization Framework for Constrained Multiobjective Optimization. IEEE Transactions on Cybernetics, PP. 1-13, 2022.

[30] J. Q. Zhang, A. C. Sanderson. JADE: Adaptive Differential Evolution With Optional External Archive. IEEE Transactions on Evolutionary Computation, vol. 13, no. 5, pp. 945-958, 2009.

[31] L. E. Blumenson. A Derivation of n-Dimensional Spherical Coordinates. The American Mathematical Monthly, vol. 67, no. 1, pp. 63–66, 1960.

[32] J. J. Liang, B. Y. Qu, P.N. Suganthan, Alfredo G. Hernández-Díaz. Problem Definitions and Evaluation Criteria for the CEC 2013 Special Session on Real-Parameter Optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou, China and Nanyang Technological University, Singapore, Technical Report, pp. 281-295, 2013.

[33] J. Derrac, S. García, D. Molina, and F. Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput., vol. 1, no. 1, pp. 3–18, 2011.