A Hybrid Estimation of Distribution Algorithm with Monarch Butterfly Optimization

Bo Zhu
School of Mechatronics Engineering and Automation
Shanghai University
Shanghai, China
1532116227@qq.com

Li Jia*
School of Mechatronics Engineering and Automation
Shanghai University
Shanghai, China
jiali@staff.shu.edu.cn

Jianfang Li
School of Mechatronics Engineering and Automation
Shanghai University
Shanghai, China
jianfang_li@shu.edu.cn

***Abstract*—The complex optimization problems have been investigated deeply by researchers in the optimization community. The estimation of distribution algorithm (EDA) and the monarch butterfly optimization algorithm (MBO) are meta-heuristic algorithms that attracted wide attention. In this study, an improved algorithm based on Estimation of Distribution of Algorithm combined with Monarch Butterfly Optimization Algorithm named EDMBO is proposed. The weighted average of candidate solutions is embedded to estimate the mean value. A linear search strategy is introduced to enhance the exploitation of the algorithm. The CEC 2017 benchmark test suite is adopted to verify the performance of the algorithm. The experimental results show that the EDMBO is competitive.**

Keywords—hybrid algorithm, estimation of distribution algorithm, monarch butterfly optimization, linear search

I. INTRODUCTION

In the last few years, intelligent algorithms have attracted the wide attention of various researchers as the global optimization problems have become increasingly complex [1]. Intelligent algorithms such as evolutionary algorithms and swarm optimization performed well on these issues while they were rough to solve by mathematical methods[2]. Three categories of meta-heuristic attracted researchers widely investigated and studied. Inspired by the theory of evolution, evolution algorithms including genetic algorithm (GA) [3], particle swarm optimization (PSO [4], ant colony optimization (ACO) [5], artificial bee colony optimization algorithm (ABC) [6] were developed. Water wave optimization algorithm (WWO) [7], simulated annealing (SA) [8], et al. were designed based on physical laws. Related works on other emerging intelligent algorithms such as the estimation of distribution algorithm (EDA) [9] and differential evolution algorithm (DE) [10] are numerous.

Estimation of distribution algorithm (EDA) is an evolutionary algorithm first proposed by H.Mühlenbein and G.Paaβ in 1994 [11]. In contrast with the canonical evolutionary algorithm such as GA, EDA produces the new population through sampling the learned model instead of crossover and mutation operations. EDA is an evolutionary algorithm at the "macro" level compared with the other evolutionary algorithm. Probabilistic models are the key to EDA [12]. By far, various models have been introduced to EDAs, such as the Gaussian model, Cauchy model, and histogram model [13]. The Gaussian model is the most widely adopted model, which has obtained good results in some problems. The Gaussian EDAs generally classified into three categories [14] according to the variable dependencies: 1) Univariate EDA, which assumes that all variables are independent. 2) Bivariate EDA, which only thinks of some pairwise variable interactions. 3) Multivariate EDA considering interactions among multiple variables. Sebag et al. [15] extend population-based increment learning (PBIL) from boolean to continuous search spaces. A Gaussian model was utilized to describe the distribution of the population. The algorithm is validated on several large-sized problems, and the results show that PBILc outperforms standard ES. Zhao et al. [16] developed a new hybrid optimization algorithm that combines EDA and a differential evolution based on chaotic theory. In cDE/EDA, the chaotic strategy was integrated into differential evolution to strengthen the search performance of DE. Experimental results proved that cDE/EDA is an effective method to improve the convergence speed of basic EDA. A new EDA variant named AAVS-EDA [17] was proposed by Ren et al. In AAVS-EDA, a novel anisotropic adaptive variance scaling (AAVS) technique was applied to strengthen the performance of traditional EDA. An auxiliary global monitor was introduced to help the algorithm to converge to a promising area. The results on benchmark functions demonstrate that AAVS-EDA has good efficiency and competitiveness. In [18], an archive mechanism is introduced to EDA, and historical solutions are employed to assist in estimating the covariance matrix of the Gaussian model. The proposed algorithm dubbed EDA^2 is tested on various benchmark functions and compared with the state-of-the-art EAs. The results show the effectiveness of the algorithm.

MBO [19] is a classical swarm intelligence meta-heuristic algorithm. The individuals in MBO are updated by the migration operator and butterfly adjusting operator. Feng et al. [20] introduced the chaos theory to enhance the global optimization of the algorithm. The parameters are replaced with chaotic maps. The results that compared with the several algorithms demonstrate the competitiveness of the algorithm. An improved MBO that employed an opposition-based learning strategy and Gaussian perturbation is proposed by [21]. The former guarantees a larger search scale while the latter prevents the algorithm from falling into the local optimal. The experiment study shows that the algorithm can offer high-quality solutions to the knapsack problem. A novel multi-strategy monarch butterfly optimization (MMBO) is designed in [22]. The author introduced two effective strategies into MMBO.

This work was financially supported by the National Natural Science Foundation of China under Grant 92067105, Grant 62373236. It was also supported by the the Natural Science Foundation of Shanghai under Grant 23ZR1422000.

Experimental analysis verified the neighborhood mutation strategy enhances the global search ability, and the Gaussian perturbation strengthens local search ability. Hu et al. [23] developed a modified MBO with a self-adaptive mechanism. In the evolution process, only the individuals who are better than their parents are reserved and pass to the next generation.

This paper hybridizes the estimation of distribution algorithm and monarch butterfly optimization to improve the performance of EDA. The main modifications are summarized as follows:

- A mean estimation method is applied to guide the population to find a more promising area of solution space through calculating the weighted average of candidate solutions.
- A linear search strategy is introduced to strengthen the exploitation ability of the algorithm.
- MBO is combined with EDA to enhance the performance of the algorithm and alleviate the limits of premature convergence and loss of population diversity.

The struct of this article is organized as follows. The basic EDA and MBO are described in Section II. Section III presents the EDMBO. The experimental study and results analysis are presented in Section IV. The conclusion and future research are given in Section V.

II. Background

A. Estimation of Distribution Algorithm

The main steps of EDA include four phases, selection, modeling, sampling, and combination [24]. First of all, EDA generally uses the truncation selection method to obtain a set of good individuals from the population. The solution space is described comprehensively by the selected individuals. The second process constructs the probabilistic model via extracted information from the selected individuals. When the modeling phase is finished, a set of individuals is sampled from the constructed model. In every generation, the modeling individuals and sampled individuals are combined to obtain the offspring population. The general framework of EDA is summed up in Algorithm 1.

Algorithm 1: EDA

Set initial parameters and generate the initial population

While the termination criteria were not satisfied, **do**

1. Evaluate the population and update the best solution
2. Select good solutions via a selecting method
3. Construct a probability model
4. Generate the next generation by sampling the learned model

end

return the best solution

Generally, continuous EDAs adopt the Gaussian model to describe the solution space. The probability density function of the Gaussian model is presented as follows:

$$G_{(\mu,C)}(x) = \frac{1}{(2\pi)^{n/2}(detC)^{1/2}} e^{-\frac{1}{2}(x-\mu)^T(C)^{-1}(x-\mu)} \quad (1)$$

where x is an n-dimensional individual vector. μ is the mean of x and C is the covariance matrix of x. To construct the probability model, we need to know μ and C from the selected individuals. The Maximum-likelihood (ML) method is often applied to estimate μ and C. The ML method is given by the following formulas:

$$\bar{\mu}^t = \frac{1}{|S^t|}\sum_{i=1}^{|S^t|} S_i^t \quad (2)$$

$$\bar{C}^t = \frac{1}{|S^t|}\sum_{i=1}^{|S^t|}(S_i^t - \bar{\mu}^t)(S_i^t - \bar{\mu}^t)^T \quad (3)$$

where $|S^t|$ is the number of selected individuals.

B. Monarch Butterfly Optimization

MBO is a meta-heuristic algorithm based on the behavior of monarch butterflies [25]. In the MBO, the population of the monarch butterfly is divided into two subpopulations. The number of monarch butterflies in *Land* 1 dubbed *Subpopulation* 1 are $ceil(p * NP)$ ($ceil(x)$ is a ceiling function), and the others in *Land* 2 dubbed *Subpopulation* 2 are $(NP - NP1)$. NP is the size of the monarch butterfly population. p is the ratio of *Subpopulation* 1.

The offspring generated by the migration operator is determined by the migration ratio. The positions of the population can also be updated by the butterfly adjusting operator. Therefore, the search direction of the monarch butterfly population is determined by the two operators.

In *Land* 1, the offspring are generated by the migration operator. The operator is defined as follows:

$$\begin{cases} x_{i,k}^{t+1} = x_{r1,k}^t & (r \le p) \\ x_{i,k}^{t+1} = x_{r2,k}^t & (r > p) \end{cases} \quad (4)$$

$$r = rand * peri \quad (5)$$

where $x_{i,k}^{t+1}$ is the kth element of the ith solution at iteration $t+1$, $x_{r1,k}^t$ is the kth element of the $r1$th solution at iteration t, $x_{r2,k}^t$ is the kth element of the $r2$th solution at iteration t. r is a random number in [0,1] based on Eq. (5), and $peri$ is the migration rate which is set to 1.2 in the MBO. $r1$ is a random number drawn from *Subpopulation* 1, and $r2$ is a random number drawn from *Subpopulation* 2.

In *Land* 2, the offspring are generated by the butterfly adjusting operator. The operator is presented by the following formulas:

$$\begin{cases} x_{j,k}^{t+1} = x_{best,k}^t & (rand \le p) \\ x_{j,k}^{t+1} = x_{r3,k}^t & (rand > p) \end{cases} \quad (6)$$

$$x_{j,k}^{t+1} = x_{best,k}^{t+1} + \alpha \times (d_{x_k} - 0.5) \quad (7)$$

$$\alpha = \frac{S_{max}}{t^2} \quad (8)$$

$$dx = L\acute{e}vy(x_j^t) \quad (9)$$

where $x_{j,k}^{t+1}$ is the kth element of the jth solution at iteration $t+1$, $x_{best,k}^t$ is the kth element of the $best$ solution at iteration t in *Land* 1 and *Land* 2. $x_{r3,k}^t$ is the kth element of the $r3$th solution at iteration t. $rand$ is a random number in [0, 1]. $r3$ is a random number from *Subpopulation* 2. if $rand > BAR$, the kth element of the jth solution at iteration $t+1$ is updated based on Eq. (7). BAR is the butterfly adjustment rate. In Eq. (7), α is the weighting factor based on Eq. (8). α is a crucial parameter for balancing exploration and exploitation. In Eq. (8), S_{max} is the maximum step size. In Eq. (9), dx is the step size from the Lévy flight. The basic MBO is presented in Algorithm 2.

Algorithm 2: MBO

Begin

Initialize the parameters of the MBO

$t = 0$

While (t < MaxGen)

1. Sort the population and divide the population into two subpopulations
2. Generate individuals by **Migration Operator**
3. Generate individuals by Butterfly Adjusting Operator
4. Evaluate the new population

end while

Output the best solution

End

III. EDMBO

In this section, two improvements are proposed to improve the performance of the algorithm. The details of the EDMBO are shown in Algorithm 3.

Algorithm 3: EDMBO

Begin

Initialize the parameters of the EDMBO

$t = 0$

While the termination criteria were not satisfied, **do**

1. Generate the candidate solutions by applying the **T*runcation Selection Method***
2. Select some individuals from candidate solutions
3. **MBO** is executed on the selected solutions
4. Evaluate the offspring attained by **MBO**
5. Select individuals from the offspring attained by MBO and candidate solutions. Generate the offspring of the same size as the candidate solution according to fitness value.
6. **Estimate the μ and C**
7. **Shift μ by the linear search strategy**
8. Generate the new population by Gaussian Model
9. Evaluate the new population

end while

Output the best solution

End

A. The Estimation Method of Mean

The traditional method, ML, which calculates the mean of candidate solutions is shown in Eq. (2). However, this method cannot provide a more promising search center for EDA. Thus, a weighted average of candidate solutions has been adopted to estimate the mean according to the fitness of the candidate solutions. The definition of the mean is presented by the following formulas:

$$\bar{\mu}^t = \frac{\sum_{i=1}^{|s^t|}(FITNESS - fitness(i))S_{(i)}^t}{\sum_{i=1}^{|s^t|}(FITNESS - fitness(i))} \quad (10)$$

$$FITNESS = \sum_{i=1}^{|s^t|} fitness(i) \quad (11)$$

where $|S^t|$ is the number of selected individuals and $S_{(i)}^t$ denotes the ith solution in the candidate solutions.

A. The Linear Search Strategy

Additionally, a linear search method is introduced to accelerate the search process as follows:

$$\delta_1 = \bar{\mu}^t - \mu_0 \quad (12)$$

$$\delta_2 = \bar{\mu}^t - \mu_1 \quad (13)$$

$$\hat{\mu}^t = \begin{cases} \bar{\mu}^t + \eta(\delta_1 + \delta_2), if\ f(\Delta) < f(\bar{\mu}^t) \\ \bar{\mu}^t, otherwise \end{cases} \quad (14)$$

$$\Delta = \bar{\mu}^t + \eta(\delta_1 + \delta_2) \quad (15)$$

where $\hat{\mu}^t$ denotes the shifted mean in the tth generation. The $\hat{\mu}^t$ is used to generate the offspring when the estimation process is finished. δ_1 represents the difference between $\bar{\mu}^t$ and μ_0 where $\bar{\mu}^t$ is the mean of the candidate solutions and μ_0 is the mean of the whole population. δ_2 represents the difference between $\bar{\mu}^t$ and μ_1 where μ_1 is the mean of the individuals not to be selected from the population. η is a shifting factor that should be greater than 0. The purpose of this operation is to balance the exploration and exploitation of the algorithm and avoid premature convergence. η_{max} is set to 10 in this paper.

Algorithm 4: Linear Search Strategy

Input: $\bar{\mu}^t$, δ_1, δ_2 and η_{max}

Output: $\hat{\mu}^t$

1. $\hat{\mu}^t = \bar{\mu}^t$, $\eta = 0$
2. **While** $f(\Delta) < f(\bar{\mu}^t)$ && $\eta < \eta_{max}$
3. $\hat{\mu}^t = \Delta = \bar{\mu}^t + \eta(\delta_1 + \delta_2)$
4. $\eta = \eta + 1$
5. **end while**

C. The Combination of EDA and MBO

EDAs that adopt the Gaussian model suffer from premature convergence, and thus, the performance of the algorithm has been limited. In addition, the diversity of the population is lost in the late evolution process. Therefore, this paper integrates MBO into EDA to enhance the search efficiency of EDA and balance the exploration and exploitation of the search process. The main framework of EDMBO is demonstrated as follows:

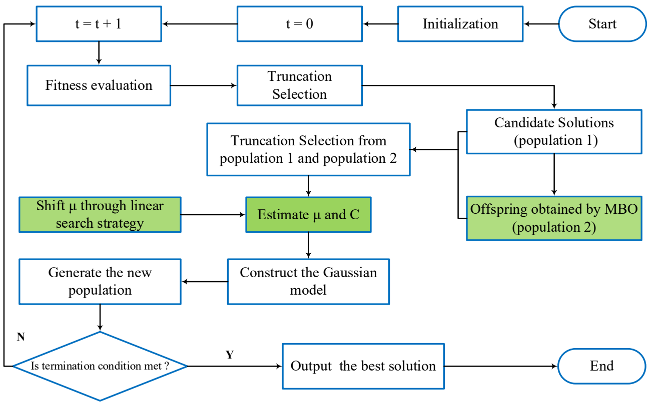

Fig. 1. Flowchart of EDMBO.

In this designed algorithm, MBO is integrated into EDA as a significant component because MBO is an effective meta-heuristic algorithm for the global optimization problem according to the other related works. The exchange of information between the two subpopulations and within the subpolpulation1 has been enhanced by the migration operator. The butterfly adjusting operator aims to improve the population diversity, and the search scope has been expanded by Lévy flight. As is shown in the flowchart of EDMBO, when the selection of candidate solutions is finished, the stage of production which applied MBO on a random part of candidate solutions is executed instead of the mean and covariance matrix estimation phase—then employed the truncation selection to the whole set that included candidate solutions and individuals generated by MBO to obtain the same size as the candidate population. The estimation of mean value and covariance matrix is processed afterward. To enhance the exploitation ability, a linear search mechanism is adopted. As aforementioned in Algorithm 4, this approach tries to find a better value for $\hat{\mu}^t$ by shifting the mean until it is unable to improve. During the process, some fitness evaluations are consumed. When the improved mean value is obtained, the Gaussian model is constructed. The latter phase is the same as basic EDA.

IV. Experiment And Performance Analysis

The simulation experiments on CEC 2017 benchmark problems are utilized to test the performance of the proposed algorithm and three comparison algorithms. Four different test problems are included in the CEC 2017 test suite. $f1 \sim f3$ are unimodal functions, $f4 \sim f10$ are multimodal functions, $f11 \sim f20$ are hybrid functions, and $f21 \sim f30$ are composition functions. During the experiment, each algorithm is tested 51 times the error value was smaller than 10^{-8} will be taken zero. The parameters of each algorithm are set as follows:

TABLE I. Parameter Setting

	Population size	***FES***	***Truncation ratio***
PBILc	300	$D \cdot 10^4$	0.35
cDE/EDA	100	$D \cdot 10^4$	0.35
AAVS_EDA	1000	$D \cdot 10^4$	0.35
EDMBO	1000	$D \cdot 10^4$	0.35

The EDMBO is compared with the PBILc, cDE/EDA, and the AAVS_EDA. The simulated results of four algorithms including mean error and standard deviation with the dimension $D = 10$ were listed in TABLE I. The convergence curves of function $f1, f4, f14, f29$ are shown in Fig. 2, and the convergence speed of the EDMBO is faster than the three algorithms. Boxplots show that EDMBO is more stable than comparison algorithms.

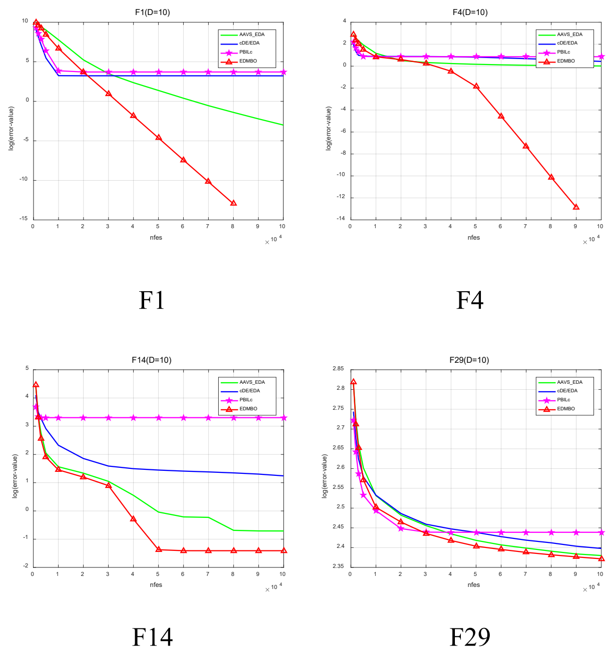

Fig. 2. The convergence curves of four benchmark functions.

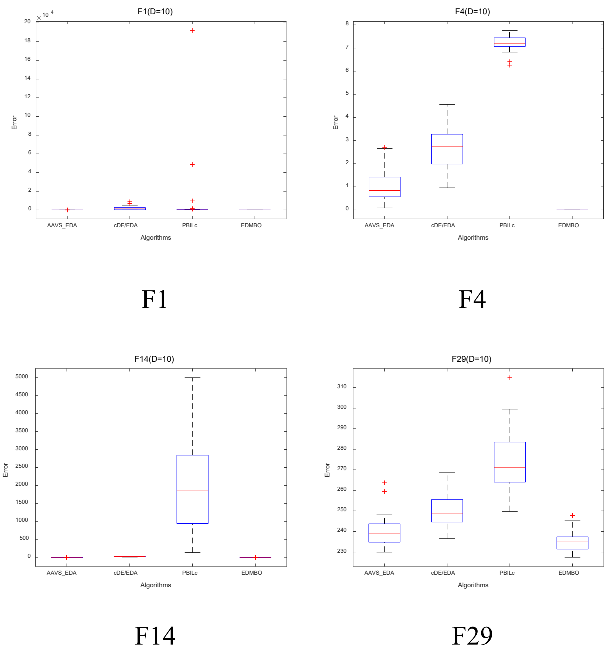

Fig. 3. The boxplots of four benchmark functions.

To verify the performance difference between EDMBO and comparison algorithms, the Wilcoxon-test is adopted, and the results are shown in TABLE III. From TABLE III, the proposed EDMBO outperforms PBILc and cDE/EDA. The two former p-value is less than the α, which means there are conspicuous differences between algorithms. Although there are no significant differences between EDMBO and AAVS_EDA, EDMBO performed well on 17 functions.

TABLE II. OPTIMIZATION RESULTS OF PBILC, CDE/EDA, AAVS_EDA AND EDMBO

Fun	PBILc		cDE/EDA		AAVS_EDA		EDMBO	
	Mean	*Std*	*Mean*	*Std*	*Mean*	*Std*	*Mean*	*Std*
F1	5.10E+03	2.73E+04	1.66E+03	1.83E+03	9.93E-04	5.16E-04	**0.00E+00**	0.00E+00
F2	1.10E+07	1.17E+07	1.53E-05	4.99E-05	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
F3	2.90E+03	1.67E+03	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00	**0.00E+00**	0.00E+00
F4	7.22E+00	2.89E-01	2.69E+00	8.07E-01	1.04E+00	6.66E-01	**0.00E+00**	0.00E+00
F5	**4.68E-01**	6.34E-01	5.50E+00	2.05E+00	1.93E+00	8.90E-01	3.36E+00	1.92E+00
F6	**0.00E+00**	0.00E+00	4.10E-07	1.61E-06	3.89E-04	1.91E-04	**0.00E+00**	0.00E+00
F7	**1.10E+01**	4.07E-01	1.58E+01	3.29E+00	1.22E+01	3.08E+00	1.31E+01	1.51E+00
F8	5.85E-01	7.92E-01	4.69E+00	2.28E+00	**7.80E-02**	2.67E-01	3.05E+00	1.82E+00
F9	5.87E-04	1.25E-03	**0.00E+00**	0.00E+00	9.44E-08	6.59E-08	**0.00E+00**	0.00E+00
F10	**2.20E+01**	6.46E+01	7.07E+02	3.93E+02	2.33E+01	5.78E+01	4.76E+01	6.54E+01
F11	7.96E+02	4.76E+02	1.00E+00	1.16E+00	7.87E-01	6.78E-01	**1.87E-01**	4.80E-01
F12	7.59E+05	6.55E+05	4.32E+03	4.47E+03	4.87E+01	5.75E+01	**1.05E+01**	3.25E+01
F13	6.99E+03	2.12E+03	3.67E+01	1.10E+02	**1.65E+00**	2.63E+00	4.52E+00	2.43E+00
F14	2.01E+03	1.20E+03	1.74E+01	6.21E+00	1.95E-01	3.95E-01	**3.90E-02**	1.93E-01
F15	2.96E+03	1.60E+03	3.86E+00	2.98E+00	8.87E-01	8.87E-01	**3.62E-01**	1.76E-01
F16	2.35E+00	4.98E-01	2.83E+00	3.34E+00	1.15E+00	4.23E-01	**8.78E-01**	2.14E-01
F17	1.20E+01	1.04E+01	3.72E+01	6.52E+00	1.92E+01	1.02E+01	**7.36E+00**	8.10E+00
F18	4.25E+03	2.42E+03	2.07E+01	3.59E+00	6.30E-01	3.44E-01	**4.57E-01**	8.11E-02
F19	4.97E+03	1.80E+03	2.58E+00	6.36E-01	3.88E-01	3.08E-01	**9.54E-02**	6.88E-02
F20	8.17E+00	9.43E+00	1.21E+01	9.77E+00	6.64E+00	8.97E+00	**6.63E+00**	9.00E+00
F21	2.02E+02	3.98E+00	1.87E+02	4.05E+01	1.99E+02	9.23E+00	**1.83E+02**	2.42E+01
F22	**1.00E+02**	1.10E+00	**1.00E+02**	2.49E-01	**1.00E+02**	1.10E-05	**1.00E+02**	0.00E+00
F23	3.04E+02	2.18E+00	3.08E+02	2.23E+00	**3.01E+02**	1.84E+00	3.02E+02	1.14E+01
F24	3.30E+02	1.01E+00	3.35E+02	2.95E+00	**3.14E+02**	1.29E+01	3.21E+02	2.12E+01
F25	4.48E+02	1.40E+00	4.41E+02	1.54E+01	**4.04E+02**	1.50E+01	4.35E+02	1.72E+01
F26	5.05E+02	2.09E+02	3.43E+02	6.72E+01	3.04E+02	6.34E+00	**3.00E+02**	0.00E+00
F27	3.98E+02	1.81E+00	**3.73E+02**	1.83E+00	3.94E+02	1.37E+00	3.86E+02	6.68E+00
F28	6.42E+02	1.68E+01	4.17E+02	4.45E+01	**3.00E+02**	1.48E-04	4.88E+02	4.69E+01
F29	2.75E+02	1.31E+01	2.50E+02	7.76E+00	2.40E+02	6.42E+00	2.35E+02	4.50E+00
F30	3.94E+05	3.07E+05	**2.03E+02**	7.13E-01	3.99E+02	5.51E+00	1.42E+04	5.55E+04

TABLE III. p-VALUE OF WILCOXON'S RANK-SUM TEST FOR $D = 10$

EDMBO *VS*	R+	R-	+	≈	−	Z	p-value	α = 0.05	α = 0.1
PBILc	374.00	32.00	24	4	2	-3.591	3.30E-04	**Yes**	**Yes**
cDE/EDA	313.00	65.00	24	3	3	-3.849	1.19E-04	**Yes**	**Yes**
AAVS_EDA	187.00	191.00	17	3	10	-1.155	2.48E-01	No	No

V. Conclusion And Future Research

In this paper, EDMBO is proposed to solve complex continuous optimization problems. In the algorithm, the MBO is combined with EDA to balance the exploration and exploitation. A mean estimation method that weighted the average of candidate solutions is designed to obtain a more promising search center, and the solution space is rationally described by the improved Gaussian model. The developed method speeds up the convergence speed of the algorithm. In addition, a linear search strategy is introduced into the EDMBO to enhance the exploitation capability of the algorithm. The integration of MBO alleviates the algorithm trap into the local optima. Experiments on the CEC 2017 benchmark test suite testified that the EDMBO is effective in solving these problems. Wilcoxon test demonstrated that the proposed algorithm is competitive.

The performance of the EDA is highly dependent on the probability model adopted. Therefore, capturing the characteristics of solution space and precisely constructing a probability model is vital for future research. The theory study of EDA should be focused on in the future. Additionally, the hybrid algorithm research based on the EDA and practical application of EDA needs to be further investigated.

Acknowledgment

This work was financially supported by the National Natural Science Foundation of China under Grant 92067105, Grant 62373236. It was also supported by the the Natural Science Foundation of Shanghai under Grant 23ZR1422000.

References

[1] M. Zhang, F. Tao, Y. Zuo, F. Xiang, L. Wang, and A. Y. C. Nee, “Top ten intelligent algorithms towards smart manufacturing,” *J. Manuf. Syst.*, vol. 71, pp. 158–171, Dec. 2023, doi: 10.1016/j.jmsy.2023.09.006.

[2] A. P. Engelbrecht and C. W. Cleghorn, “Recent advances in particle swarm optimization analysis and understanding 2021,” in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, New York, NY, USA, Jul. 2021. doi: 10.1145/3449726.3461431.

[3] B. B. Oliveira, M. A. Carravilla, and J. F. Oliveira, “A diversity-based genetic algorithm for scenario generation,” *Eur. J. Oper. Res.*, vol. 299, no. 3, pp. 1128–1141, Jun. 2022, doi: 10.1016/j.ejor.2021.09.047.

[4] A. Madani, A. Engelbrecht, and B. Ombuki-Berman, “Cooperative coevolutionary multi-guide particle swarm optimization algorithm for large-scale multi-objective optimization problems,” *Swarm. Evol. Comput.*, vol. 78, p. 101262, Apr. 2023, doi: 10.1016/j.swevo.2023.101262.

[5] J. Yu, X. You, and S. Liu, “Dynamically induced clustering ant colony algorithm based on a coevolutionary chain,” *Knowledge-based Syst.*, vol. 251, p. 109231, Sep. 2022, doi: 10.1016/j.knosys.2022.109231.

[6] F. Zhao, Z. Wang, and L. Wang, “A Reinforcement Learning Driven Artificial Bee Colony Algorithm for Distributed Heterogeneous No-Wait Flowshop Scheduling Problem With Sequence-Dependent Setup Times,” *IEEE Trans. Automat. Sci. Eng.*, pp. 1–16, 2022, doi: 10.1109/tase.2022.3212786.

[7] F. Zhao, D. Shao, L. Wang, T. Xu, N. Zhu, and Jonrinaldi, “An effective water wave optimization algorithm with problem-specific knowledge for the distributed assembly blocking flow-shop scheduling problem,” *Knowledge-based Syst.*, vol. 243, p. 108471, May 2022, doi: 10.1016/j.knosys.2022.108471.

[8] V. F. Yu, N. Y. Salsabila, S.-W. Lin, and A. Gunawan, “Simulated annealing with reinforcement learning for the set team orienteering problem with time windows,” *Expert Systems with Applications*, vol. 238, p. 121996, Mar. 2024, doi: 10.1016/j.eswa.2023.121996.

[9] F. Zhao, B. Zhu, L. Wang, T. Xu, N. Zhu, and J. Jonrinaldi, “An offline learning co-evolutionary algorithm with problem-specific knowledge,” *Swarm. Evol. Comput.*, vol. 75, p. 101148, Dec. 2022, doi: 10.1016/j.swevo.2022.101148.

[10] S. Das, S. S. Mullick, and P. N. Suganthan, “Recent advances in differential evolution – An updated survey,” *Swarm. Evol. Comput.*, vol. 27, pp. 1–30, Apr. 2016, doi: 10.1016/j.swevo.2016.01.004.

[11] S. Baluja, “Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning,” *Tech. Rep. C. Sch. Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA*, pp. 1–41, 1994.

[12] Z. Lin, Q. Su, and G. Xie, “NMIEDA: Estimation of distribution algorithm based on normalized mutual information,” *Concurr. Comput. Pract. Exp.*, vol. 33, no. 6, pp. 1–16, 2021, doi: 10.1002/cpe.6074.

[13] C. Witt, “Theory of estimation-of-distribution algorithms,” in *Proceedings of the 2020 Genetic and Evolutionary Computation Conference Companion*, New York, NY, USA, Jul. 2020. doi: 10.1145/3377929.3389888.

[14] P. Gholamnezhad, A. Broumandnia, and V. Seydi, “An inverse model-based multiobjective estimation of distribution algorithm using Random-Forest variable importance methods,” *Computational Intell*, Apr. 2020, doi: 10.1111/coin.12315.

[15] M. Sebag and A. Ducoulombier, “Extending population-based incremental learning to continuous search spaces,” Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 418–427. doi: 10.1007/bfb0056884.

[16] F. Zhao, Z. Shao, R. Wang, C. Zhang, and J. Wang, “A hybrid EDA with Chaotic DE algorithm and its performance analysis,” *J. Comput. Inf. Syst.*, vol. 11, no. 4, pp. 1505–1512, 2015, doi: 10.12733/jcis13564.

[17] Z. Ren, Y. Liang, L. Wang, A. Zhang, B. Pang, and B. Li, “Anisotropic adaptive variance scaling for Gaussian estimation of distribution algorithm,” *Knowledge-based Syst.*, vol. 146, pp. 142–151, Apr. 2018, doi: 10.1016/j.knosys.2018.02.001.

[18] Y. Liang, Z. Ren, X. Yao, Z. Feng, A. Chen, and W. Guo, “Enhancing Gaussian Estimation of Distribution Algorithm by Exploiting Evolution Direction With Archive,” *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 140–152, Jan. 2020, doi: 10.1109/tcyb.2018.2869567.

[19] G.-G. Wang, S. Deb, and Z. Cui, “Monarch butterfly optimization,” *Neural Comput & Applic*, vol. 31, no. 7, pp. 1995–2014, May 2015, doi: 10.1007/s00521-015-1923-y.

[20] Y. Feng, J. Yang, C. Wu, M. Lu, and X.-J. Zhao, “Solving 0–1 knapsack problems by chaotic monarch butterfly optimization algorithm with Gaussian mutation,” *Memetic Comp.*, vol. 10, no. 2, pp. 135–150, Sep. 2016, doi: 10.1007/s12293-016-0211-4.

[21] L. Sun, S. Chen, J. Xu, and Y. Tian, “Improved Monarch Butterfly Optimization Algorithm Based on Opposition-Based Learning and Random Local Perturbation,” *Complexity*, vol. 2019, pp. 1–20, Feb. 2019, doi: 10.1155/2019/4182148.

[22] Y. Feng, G.-G. Wang, W. Li, and N. Li, “Multi-strategy monarch butterfly optimization algorithm for discounted 0-1 knapsack problem,” *Neural Comput & Applic*, vol. 30, no. 10, pp. 3019–3036, Feb. 2017, doi: 10.1007/s00521-017-2903-1.

[23] H. Hu, Z. Cai, S. Hu, Y. Cai, J. Chen, and S. Huang, “Improving Monarch Butterfly Optimization Algorithm with Self-Adaptive Population,” *Algorithms*, vol. 11, no. 5, p. 71, May 2018, doi: 10.3390/a11050071.

[24] Q. Dang, W. Gao, and M. Gong, “An efficient mixture sampling model for gaussian estimation of distribution algorithm,” *Inf. Sci.*, vol. 608, pp. 1157–1182, Aug. 2022, doi: 10.1016/j.ins.2022.07.016.

[25] Y. Feng, S. Deb, G.-G. Wang, and A. H. Alavi, “Monarch butterfly optimization: A comprehensive review,” *Expert Systems with Applications*, vol. 168, p. 114418, Apr. 2021, doi: 10.1016/j.eswa.2020.114418.