Handling Constrained Multi-Objective Optimization with Objective Space Mapping to Decision Space Based on Extreme Learning Machine

1st Hao Zhang[1,2,3], 2nd KuTao[1,2,3], 3rd Lianbo Ma[4,*], 4th Yibo Yong[4]
1: Key Laboratory of Networked Control Systems, Chinese Academy of Sciences
2: Shenyang Institute of Automation, Chinese Academy of Sciences
3: Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences
4: Software College, Northeastern University
Shenyang, China
zhanghao@sia.cn, kutao@sia.cn, malb@swc.neu.edu.cn, yyb@swc.neu.edu.cn

***Abstract*—Constrained multi-objective optimization is frequently encountered from the point of view of practical problem solving. The difficulty of constrained multi-objective optimization is how to offer guarantee of finding feasible optimal solutions within a specified number of iterations. To address the issue, this paper proposes an innovative optimization framework with objective space mapping to decision space for constrained multi-objective optimization and a novel multi-objective optimization algorithms are proposed based on this framework. Extreme learning machine implements prediction of decision variables from modified objective values with distance measure and adaptive penalty. This algorithm employs the framework of artificial bee colony to divide this optimization process into two phases: the employed bees and the onlooker bees. In the phase of employed bees, multi-objective strategy employs fast non-dominant sort and crowded distance to push the population toward Pareto front. In the phase of onlooker bees, multi-objective strategy employs Tchebycheff approach to enhance the population diversity. The experimental results on a series of benchmark problems suggest that our proposed algorithm is quite effective, in comparison to other state-of-the-art constrained multi-objective optimizers.**

***Index Terms*—constrained multi-objective optimization, extreme learning machine, artificial bee colony, decomposition, nondomination**

I. Introduction

A majority of real world optimization problems involve simultaneous optimization of several objectives and constraints. Generally, these objective functions are competing and conflicting. Their constraints are various and noncommensurable. The constrained multi-objective optimization problem are difficult to solve, as finding a feasible and optimal solution may require substantial computational resources. In recent years, an increasing number of researchers in this respect pay attention to constrained multi-objective optimization problems [1]- [5].

Generally, constrained multi-objective optimization problem considered is defined as:

$$\begin{aligned} \text{minimize} \quad & \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_m(\boldsymbol{x}))^T \\ \text{subject to} \quad & g_j(\boldsymbol{x}) \le 0, \quad j = 1, \cdots, p \\ & h_j(\boldsymbol{x}) = 0, \quad j = p, \cdots, q \\ & \boldsymbol{x} \in \Omega \end{aligned} \tag{1}$$

where, $\boldsymbol{x} = (x_1, \cdots, x_D)^T$ is a D-dimensional candidate solution. Ω represents the decision space, and $F\colon \Omega \to R^m$ consists of m conflicting objective functions and R^m represents the objective space. $g_j(\boldsymbol{x})$ and $h_j(\boldsymbol{x})$ represent the jth inequality and equality constraint respectively.

For two vectors $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ in the decision space of constrained multi-objective optimization problem, if $f_i(\boldsymbol{x}_a) \le f_i(\boldsymbol{x}_b)$ for $\forall i \in \{1, 2, \cdots, m\}$ and $f_i(\boldsymbol{x}_a) < f_i(\boldsymbol{x}_b)$ for $\exists i \in \{1, 2, \cdots, m\}$, then $\boldsymbol{x}_a$ is said to dominate $\boldsymbol{x}_b$, which is denoted as $\boldsymbol{x}_a \prec \boldsymbol{x}_b$. The set of optimal trade-offs forms the solution set which is called the Pareto optimal set which is denoted by P^*. The set $PF^* = \{\boldsymbol{F}(\boldsymbol{x}) | \boldsymbol{x} \in P^*\}$ is called the Pareto front.

Constrained multi-objective optimization problems are frequently encountered in science, engineering or finance fields. An increasing number of real-world applications are formulated as constrained multi-objective optimization problems. In recent years, some novel multi-objective optimization algorithms are proposed [6] [7]. In [8], an agent-based model is proposed to improve the electricity market efficiency by using different demand response programs and a market power index and the operation cost are used to evaluate the market efficiency by using a multi-objective decision-making approach. In [9], a stochastic framework for day-ahead scheduling of microgrid energy storage systems in the context of multi-objective optimization is presented and the non-dominated sorting genetic algorithm II is employed to effectively cope with the optimization problem. In [10], a multi-objective airships deployment optimization model is proposed considering path loss, user demand, and inner structure and a similarity-based algorithm

This paper is supported by Natural Science Foundation of China (61803367), Natural Science Foundation of Liaoning Province (2019-MS-346), National Key R&D Program of China (2017YFB0306401) and Huawei HIRP project (HO2019085002).

* Corresponding author.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

is proposed to optimize this model under the framework of the multi-objective evolutionary algorithm based on decomposition. In [11], an innovative hybrid multi-objective artificial bee colony algorithm is proposed for the nonlinear constrained multi-objective burdening optimization model, which achieves good results. In [12], a multi-objective evolutionary algorithm is designed to solve the multi-commodity capacitated network design problem that is a very complex network design problem variation. These real-world constrained multi-objective optimization problems usually contain more than one objective functions and several diverse constraints.

Although many researchers have a consensus that heuristic algorithms based on biological evolution are a promising way to deal with constrained multi-objective optimization problems [13], there are still some shortcomings that can be improved in these algorithms as follows:

* current biological heuristic algorithms rarely consider the influence of variables and constraints in the objective space on decision variables;
* current constraint handling methods can rarely keep a good balance between convergence and diversity within the feasible region.

Therefore, these algorithms might be easily trapped into some locally optimal or locally feasible regions, or offer no guarantee of finding feasible optimal solutions within a specified number of iterations, especially when the feasible regions are scattered or narrow in the search space.

In this paper, an novel multi-objective algorithm with a innovative optimization framework is proposed that performs objective space to decision space mapping for constrained multi-objective optimization. Extreme learning machine is employed to implement prediction of decision variables from modified objective values with the constraint handling technique of distance measure and adaptive penalty. A two-phase framework in this algorithm is proposed on the basis of artificial bee colony [14], which divides this optimization process into the employed bee phase and the onlooker bee phase. In the phase of employed bees, multi-objective strategy employs fast non-dominant sort and crowded distance to push the population toward Pareto front. In the phase of onlooker bees, multi-objective strategy employs Tchebycheff approach to enhance the population diversity. This algorithm can be called CMOABC-ELM for short.

The rest of this paper is organized as follows. The technical details of CMOABC-ELM are described in Section II. The experimental setup is introduced in Section III. Subsequently, Section IV shows the effectiveness and competitiveness of the proposed algorithm that is compared with other multi-objective algorithms [15] on various benchmark problems. Finally, Section V concludes this paper.

II. Proposed Approach

A. Objective Space to Decision Space Mapping for Constraint Handling

To solve constrained multi-objective optimization problems generically, a search strategy is proposed that enhances local search in the objective space by means of mapping several certain objective vectors back to their corresponding decision vectors in this paper. Fig. 1 shows how to generate candidate solutions in decision space from objective space using this search strategy.

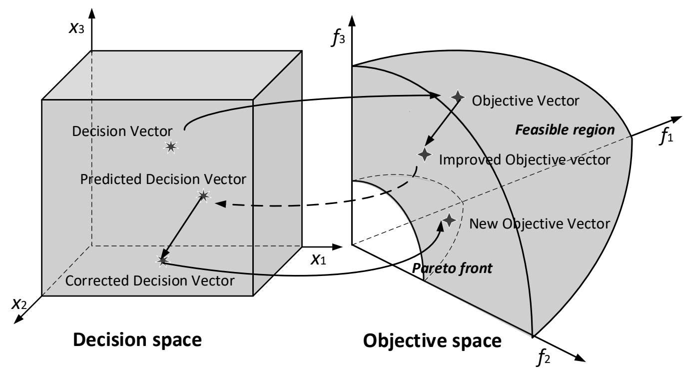

Fig. 1. The actions of this search strategy that map objective vectors in feasible region back to their corresponding decision vectors.

Firstly, the computed objective values by decision variables are updated in the objective space using a method inspired from artificial bee colony [16]. The set of objective vectors is defined as $\boldsymbol{F} = \{\boldsymbol{f}_i | i = 1, 2, \cdots, n\}$, where n is the number of individuals. A objective vector is defined as $\boldsymbol{f} = \{f_j | j = 1, 2, \cdots, m\}$, where m is the number of objective functions. A new objective value $f'_{i,j}$ is calculated by the following expression:

$$f'_{i,j} = f_{i,j} + \gamma_{i,j}(f_{i,j} - f_{r,j}) \tag{2}$$

where $r \in \{1, 2, ..., m\}$, r is a chosen index randomly, but $r \neq i$ and $f_{i,j} > f_{r,j}$ that can keep objective values moving in a good direction; $\gamma_{i,j}$ is random real number in the range of $[-1, 0]$.

In this paper, all original objective values are replaced by modified objective function values with a self-adaptive penalty function [17] to handle the constraints in constrained multi-objective optimization problem. The approach determines the amount of penalty added to infeasible individual by means of the number of feasible individuals of the colony and evolves feasible optimal solutions not only from the feasible space but also from the infeasible space. The search in the infeasible space is designed to encourage those individuals with better objective value and low constraint violation. The number of feasible individuals in the population is employed to guide the search process toward finding more feasible optimal solutions. More details are described in [17].

Then, a prediction model is trained by extreme learning machine (described in Section II-B) to produce decision variables vectors according to the improved objective vectors. The exact data results from earlier evaluations of objective functions at the same generation during the search. If these new produced decision variable vectors are out-of bounds, they are replaced by their nearest boundary values in feasible region of decision space.

Finally, the new decision vectors compete with the exist decision vectors. The better solutions will be retained and the others will be abandoned.

The constraint handling techniques can push the whole population toward the feasible region as much as possible and consider the balance between convergence and diversity within the feasible region simultaneously.

B. Decision Variable Prediction with Extreme Learning Machine

Extreme learning machine was proposed by Huang et al. in 2004 and is a fast and efficient training algorithm for single-hidden layer feed forward neural network [18]. In this paper, extreme learning machine is employed to predict the corresponding decision vectors from the data of several certain objective vectors for the enhancements to the whole population due to its good generalization performance and fast training speed.

Single-hidden layer feed forward neural network with at most n hidden neurons can learn n distinct samples with zero error by adopting any bounded nonlinear activation function. For the set of new objective vectors $\boldsymbol{F}' = (\boldsymbol{f}_1', \boldsymbol{f}_2', \cdots, \boldsymbol{f}_n')$ that are calculated by Expression (2), given an independent and identically distributed training set $\{(\boldsymbol{f}_1', \boldsymbol{t}_1), ..., (\boldsymbol{f}_n', \boldsymbol{t}_n) \subset R_K \times R_s\}$, standard single-hidden layer feed forward neural networks with K hidden nodes are mathematically formulated as follows [19]:

$$\sum_{k=1}^{K} \beta_k g_k(\boldsymbol{f}_i') = \sum_{k=1}^{K} \beta_k g_k(\boldsymbol{w}_k \cdot \boldsymbol{f}_i' + b_k) = \boldsymbol{o}_i, i = 1, ..., n \quad (3)$$

where $g(\boldsymbol{f}')$ is an activation function, $\mathbf{w}_k = [w_k^1, w_k^2, ..., w_k^K]^T$ is input weight vector connecting input nodes and the kth hidden node, $\boldsymbol{\beta}_k = [\beta_k^1, \beta_k^2, ..., \beta_k^s]^T$is the output weight vector connecting output nodes and the kth hidden node, b_k is bias of the kth hidden node.

The hidden layer output matrix $\boldsymbol{H}$ of a standard single-hidden layer feed forward neural network with N hidden nodes is invertible and $\|\boldsymbol{H}\boldsymbol{\beta} - \boldsymbol{T}\| = 0$ with probability one, which is proved for N arbitrary distinct samples and any $(\boldsymbol{w}_k, b_k)$randomly chosen from $R_K \times R_s$ according to any continuous probability distribution if the activation function $g(x)$ is infinitely differentiable in any interval [20]. Then given $(\boldsymbol{w}_k, b_k)$, training a single-hidden layer feed forward neural network equals finding a least-squares solution of the following equation:

$$\boldsymbol{H}\boldsymbol{\beta} = \boldsymbol{T} \quad (4)$$

where

$$\boldsymbol{H} = \begin{bmatrix} g(\boldsymbol{w}_1 \cdot \boldsymbol{f}_1' + b_1) & \cdots & g(\boldsymbol{w}_K \cdot \boldsymbol{f}_1' + b_K) \\ \vdots & \ddots & \vdots \\ g(\boldsymbol{w}_1 \cdot \boldsymbol{f}_n' + b_1) & \cdots & g(\boldsymbol{w}_K \cdot \boldsymbol{f}_n' + b_K) \end{bmatrix}_{n \times K}$$

$$\boldsymbol{\beta} = [\beta_1, ..., \beta_K]^T \qquad \boldsymbol{T} = [\boldsymbol{t}_1, ..., \boldsymbol{t}_K]^T$$

Considering most cases that $K \ll N$, $\boldsymbol{\beta}$ cannot be computed through the direct matrix inversion. Therefore, the smallest norm least-squares solution is calculated as follows:

$$\hat{\boldsymbol{\beta}} = \boldsymbol{H}^{\dagger}\boldsymbol{T} \quad (5)$$

where $\boldsymbol{H}^{\dagger}$ is the Moore-Penrose generalized inverse of matrix H. According to Bartletts theory that the generalization performance of single-hidden layer feed forward neural network will be improved by minimizing training errors as well as the norm of output weights, $\hat{\boldsymbol{\beta}}$ can theoretically pledge the generalization ability of single-hidden layer feed forward neural network.

Based on the above analysis, the framework of extreme learning machine is as follows:

Step 1. Randomly generate input weight and bias $(\boldsymbol{w}_k, b_k)$, $i = k, ..., K$.

Step 2. Compute the hidden layer output matrix $\boldsymbol{H}$.

Step 3. Compute the output weight $\hat{\boldsymbol{\beta}} = \boldsymbol{H}^{\dagger}\boldsymbol{T}$: Therefore, the output of single-hidden layer feed forward neural network can be calculated by $(\boldsymbol{w}_k, b_k)$and $\hat{\boldsymbol{\beta}}$:

$$\boldsymbol{x}(\boldsymbol{f}_i') = \sum_{k=1}^{K} \beta_k g_k(\boldsymbol{w}_k \cdot \boldsymbol{f}_i' + b_k) = \hat{\boldsymbol{\beta}} \cdot h(\boldsymbol{f}_i') \quad (6)$$

where, $\boldsymbol{x}(\boldsymbol{f}_i')$ is a new corresponding decision vectors that is mapped from the objective vector $\boldsymbol{f}_i'$. At this point, the new decision vectors are generated.

C. A Two-Phase Multi-Objective Strategy

In CMOABC-ELM, a two-phase framework is proposed on the basis of artificial bee colony to handle multiple objectives simultaneously. This framework divides optimization process into two phases: the employed bee phase and the onlooker bee phase. In the phase of employed bees, a fast nondominated sorting procedure, a fast crowded distance estimation procedure, and a simple crowded comparison operator [21] are employed to push the population toward Pareto front and accelerate the convergence of the whole population. In the phase of onlooker bees, multi-objective strategy employs Tchebycheff approach [15] to improve the population diversity.

Artificial bee colony is a newly proposed optimization algorithm by Karaboga [16] and applied widely [22]. In the first step, a randomly distributed population is initialized which includes $n/2$ solutions with D-dimensional vector $\boldsymbol{x} = \{x_{i,j} | i = 1, 2..., n; j = 1, 2, ..., D\}$, where n denotes the size of population. The solution $\boldsymbol{x}_i$ is conducted by

$$x_{i,j} = x_{min,j} + rand(0, 1)(x_{max,j} - x_{min,j}) \quad (7)$$

where $x_{min,j}$ and $x_{max,j}$ are lower and upper bounds of solution $x_{i,j}$, respectively.

In artificial bee colony, employed bees of the food source generate a new candidate solution $\boldsymbol{v}_i$ from $\boldsymbol{x}_i$ using the following equation:

$$v_{i,j} = x_{i,j} + \gamma_{i,j}(x_{i,j} - x_{r,j}) \quad (8)$$

where $r \in \{1, 2, ..., n\}$, $r \neq i$, r is a chosen index randomly; γ_i^j is random real number in the range of [-1,1].

In this phase, these new candidate solutions and old ones are combined into a composite set H. Then, fast non-dominant sort and crowded distance are used to sorting H into different nondomination levels. The crowded distance of each solution is estimated and a crowded comparison operator is performed. Finally, a new population is generated with n individuals.

Unlike the employed bees, the onlookers select a food source to visit by means of calculating the probability values p_i for the solutions using fitness as follows:

$$p_i = \frac{fitness_i}{\sum_{i=1}^{n/2} fitness_i} \tag{9}$$

where $fitness_i$ denotes the fitness value of the solution $\boldsymbol{x}_i$.

After the above process, each onlooker generates a new solution by Expression (8) and Tchebycheff approach is applied in order to decide that the new or old solution will be kept.

In CMOABC-ELM, a predefined parameter called "limit" is used to improve the diversity of the population and avoid trapping in local optimum. That is foraging process of a scout. If a solution cannot be improved after "limit" iterations, the corresponding food source is abandoned and the employed bee becomes the scout. The scout will find a new food source by Expression (7).

D. Proposed Algorithm

The flowchart of CMOABC-ELM algorithm is shown in Fig. 2 and its performance procedure is given as follows:

Step 1: Set maximum number of cycles T and "limit" value;

Step 2: Initialize n individuals that construct a random initial population P using Expression (7);

Step 3: Set $t = 1$;

Step 4: Loop over each individual

1) Generate new solutions as set EP_t for the employed bees by Expression (8) and then form a new combined population $NEP_t = P_t \cup EP_t$;
2) Calculate modified objective function values with distance measure and adaptive penalty for all individuals, and then the bee colony EP_t is sorted according to nondomination;
3) Select exactly n best individuals as new P_t from NEP_t using the crowded-comparison operator;
4) Generate new solutions as set OP_t for the onlooker bees by Expression (8) and form a new combined population $NOP_t = P_t \cup OP_t$;
5) Calculate modified objective function values with distance measure and adaptive penalty for all individuals, and then select exactly n best individuals as new P_t from NEP_t using Tchebycheff approach;
6) Predict new decision vectors as set PP_t with extreme learning machine and then form a new combined population $NEP_t = P_t \cup PP_t$;
7) Select exactly n best individuals as new P_t from PP_t using fast non-dominant sort and crowded distance;
8) If there is no improvement of an individual after the "limit" number of cycles, it will be discarded and a new one is randomly generated using Expression (7);

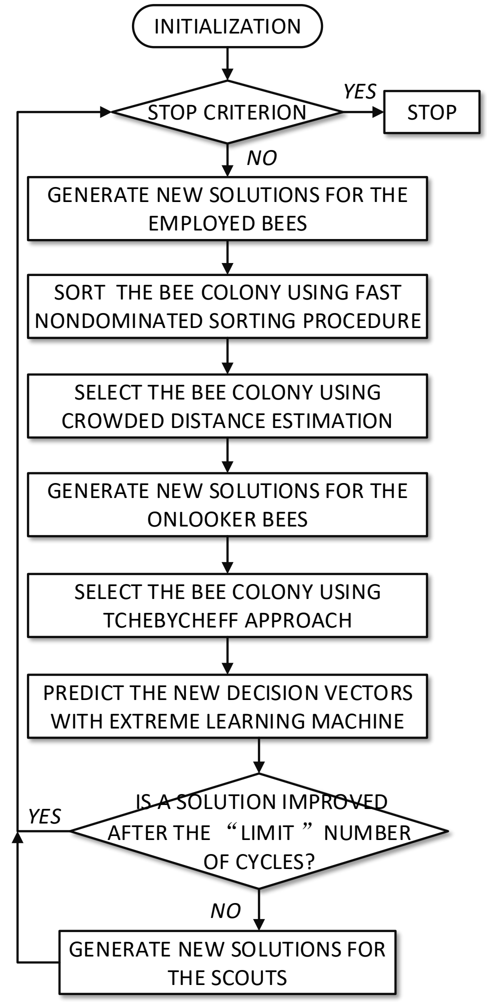

Fig. 2. Flow chart of CMOABC-ELM.

9) $t = t + 1$;
10) When $t >= T$, stop the procedure; otherwise, go to step 4;

Step 5: Output: the population P.

III. Experimental Setup

A. Benchmark Functions

The experiments on CMOABC-ELM contain four benchmark functions which are constrained multi-objective problems (C1-DTLZ1, C1-DTLZ3, C2-DTLZ2 and C3-DTLZ4) [23]. All these test problems are scalable to any number of objectives, where we set $m = 12$ here. The experiments are performed to demonstrate the proposed algorithm is a powerful search and optimization technique for constrained multi-objective problems.

B. Performance Metrics

Three widely used indicators are chosen to assess the performance of different algorithms in our experiments.

1) *Inverted Generational Distance (IGD)* [24]: Given P^* as a set of points uniformly sampled along the PF and

P as the set of solutions obtained from algorithms. The IGD value of P is calculated as:

$$IGD(P) = \frac{1}{|P^*|} \sum_{z \in P*} distance(z, P) \qquad (10)$$

where, $distance(z, P)$ is the minimum Euclidean distance between z and its nearest neighbor in P. The smaller the IGD value, the better the performance of a constrained multi-objective optimization algorithm.

2) *Hypervolume (HV)* [25]: Let $z^r = (z_1^r, \cdots, z_m^r)^T$ be a worst point dominated by all the Pareto optimal objective vectors. The HV of P is defined as the volume of the objective space dominated by solutions in P and bounded by z^r:

$$HV(P) = \text{VOL}(\bigcup_{z \in P} [z_1, z_1^r] \times \cdots \times [z_m, z_m^r]) \qquad (11)$$

where VOL indicates the lebesgue measure. The larger the HV value, the better the performance of a constrained multi-objective optimization algorithm.

3) *Spacing (SP)* [26]: SP numerically describes the spread of the obtained nondominated set of solutions in the Pareto Front. This Pareto noncompliant metric measures the distance variance of neighboring vectors in PF_{known}, which is defined as follows:

$$SP = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (\bar{d} - d_i)^2} \qquad (12)$$

and

$$d_i = \min_j (\sum_{k=1}^{M} |f_k^i - f_k^j|) \qquad (13)$$

where $i, j = 1, ..., n$, is the mean of all d_i, $\bar{d} = \frac{1}{n} \sum_n^1 d_i$, n is the number of the obtained nondominated solutions. The smaller the SP value, the better the performance of a constrained multi-objective optimization algorithm.

C. Parameter Settings

In order to prove the numerical correctness, efficiency and superiority of CMOABC-ELM for constrained multi-objective problems, The proposed algorithm is compared with CMOABC without ELM, NSGAII [21] and MOEA/D [15].

The parameters of CMOABC-ELM were set as follows: the number of hidden neurons $H = 40$, a colony of bee size N = 600, the maximum number of cycles MNC = 200 and LIMIT = $MNC \times D/2$ (D is the number of decision variables). Twenty independent runs were performed with the means and standard deviation being presented for benchmark functions. For NSGAII, the population size, maximum number of generations, crossover and mutation probabilities are selected as 600, 200, 0.85 and 0.25. For MOEA/D, the population size and maximum number of generations are selected as 200, 100. The other parameters are the same with [15].

IV. Results and Discussion

Fig. 3 shows comparative results of four algorithms on C1-DTLZ1. It is seen from Fig. 3 that the four algorithms can obtain Pareto Fronts with good distribution and they can find three intersections between Pareto Fronts and three coordinate axes. CMOABC-ELM can find more points in the intersection lines between Pareto Fronts and three coordinate planes.

Fig. 4 shows comparative results of four algorithms on C1-DTLZ3. It is seen from Fig. 4 that CMOABC-ELM and MOEA/D can obtain Pareto Fronts with good distribution and the points obtained by NSGA-II and CMOABC in Pareto Fronts are sparser than the others. All of them can find three intersections between Pareto Fronts and three coordinate axes.

Fig. 5 shows comparative results of four algorithms on C2-DTLZ2.It is seen from Fig. 5 that CMOABC can obtain Pareto Fronts with poor distribution and there is no significant difference among the others. All of them can find three intersections between Pareto Fronts and three coordinate axes.

Fig. 6 shows comparative results of four algorithms on C3-DTLZ4. It is not easy to find Pareto Fronts of C3-DTLZ4 in the optimization process. It is seen from Fig. 4 that only MOABC-ELM can obtain Pareto Fronts with the best distribution. The other algorithms can only find the intersection lines between Pareto Fronts and two coordinate planes.

It is seen from these four figures that CMOABC-ELM obtains better convergence, diversity and feasibility, especially for C3-DTLZ4. They prove feasibility and effectiveness of CMOABC-ELM for solving constrained multi-objective problems.

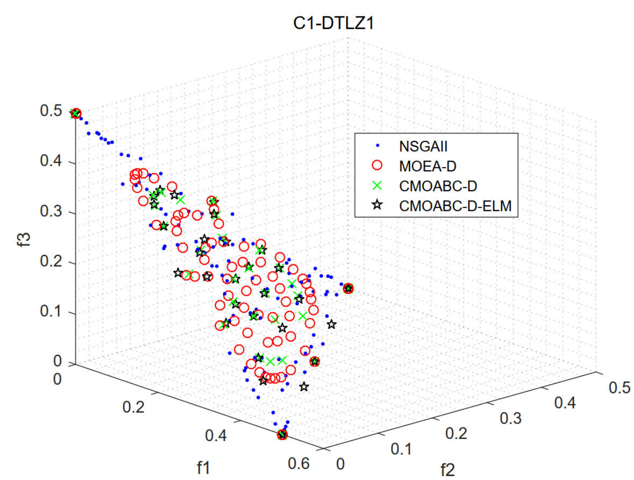

Fig. 3. Comparative results on the 3-objective C1-DTLZ1.

Table I shows mean and deviation of the IGD, HV, and SP metrics using four algorithms NSGA-II, MOEA/D, CMOABC, and CMOABC-ELM for four benchmark functions with constraints. The best values have been marked in bold. It is seen from Table I that CMOABC-ELM can obtain the best results for most benchmark functions in terms of three indicators. The detailed discussions are given below.

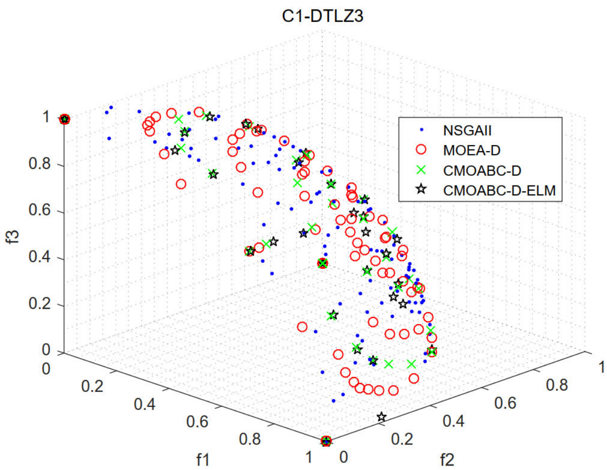

Fig. 4. Comparative results on the 3-objective C1-DTLZ3.

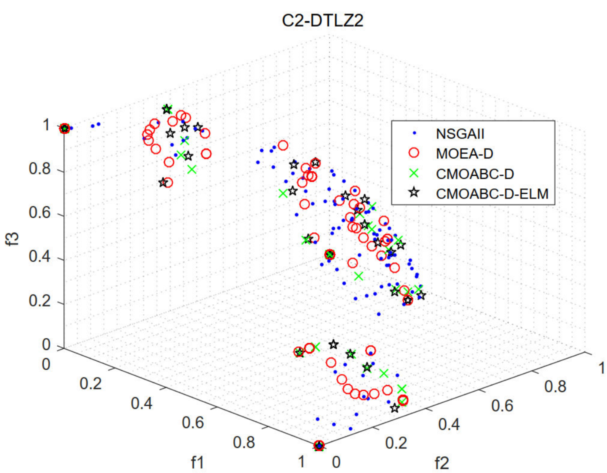

Fig. 5. Comparative results on the 3-objective C2-DTLZ2.

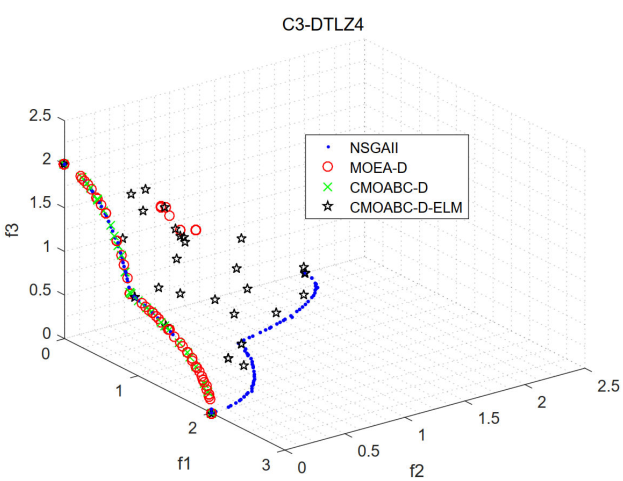

Fig. 6. Comparative results on the 3-objective C3-DTLZ4.

1) In terms of IGD, CMOABC-ELM beats the other algorithms on four benchmark functions and consistently obtain smaller IGD values except deviation values on C1-DTLZ3 and C2-DTLZ2.
2) In terms of HV, CMOABC-ELM provides higher HV values on each benchmark function than the other algorithms except deviation values on C1-DTLZ1, C1-DTLZ3 and C2-DTLZ2.
3) In terms of SP, CMOABC-ELM doesn't perform well. MOEA/D and NSGA-II obtain the best results on mean values on C1-DTLZ3, C2-DTLZ2 and C3-DTLZ4, respectively. MOEA/D and NSGA-II obtain the best results on deviation values on C1-DTLZ1, C2-DTLZ2 and C3-DTLZ4, respectively.

In conclusion, CMOABC-ELM performs best of the four algorithms, which proves its effectiveness for constrained multi-objective problems. But it still need to be improved in order to obtain the better SP metric.

V. Conclusion

In this paper, a novel constraint handling technique with objective space mapping to decision space is proposed for constrained multi-objective optimization to offer guarantee of finding feasible optimal solutions within a specified number of iterations. Due to its good generalization performance and fast training speed, extreme learning machine is employed to predict the corresponding decision vectors from the data of several certain objective vectors. To obtain better feasible optimal solutions with the novel constraint handling technique, a two-phase framework is proposed on the basis of artificial bee colony to handle multiple objectives simultaneously. This framework divides optimization process into two phases: the employed bee phase and the onlooker bee phase. In the phase of employed bees, a fast nondominated sorting procedure, a fast crowded distance estimation procedure, and a simple crowded comparison operator are employed to push the population toward Pareto front and accelerate the convergence of the whole population. In the phase of onlooker bees, multi-objective strategy employs Tchebycheff approach to improve the population diversity.

The four benchmark functions have been used to test CMOABC-ELM in comparison with CMOABC without ELM, NSGAII and MOEA/D. It is seen from the comparison that CMOABC-ELM can obtain better results than CMOABC without ELM, NSGAII and MOABC for constrained multi-objective optimization problem with respect to the HV, IGD and SP performance measures. With the properties of the two-phase framework and efficient constraint handling technique, CMOABC-ELM is very suitable for solving constrained multi-objective optimization problem and should be applied in real world problems in the near future.

References

[1] K. Li, R. Chen, G. Fu and X. Yao, "Two-Archive Evolutionary Algorithm for Constrained Multiobjective Optimization," IEEE Transactions on Evolutionary Computation, vol. 23, no. 2, pp. 303–315, April 2019.

TABLE I
COMPARISON OF NSGA-II, MOEA/D, CMOABC AND CMOABC-ELM ON IGD, HV AND SP METRIC

Functions	Algorithm	IGD		HV		SP	
		Mean	Deviation	Mean	Deviation	Mean	Deviation
C1-DTLZ1	NSGA-II	0.0468	1.60e-06	0.1358	1.17e-06	0.0214	8.55E-06
	MOEA/D	0.0692	6.21e-07	0.1341	**2.87e-07**	0.0288	**3.85e-07**
	CMOABC	0.0536	4.40e-07	0.1292	7.64e-07	0.0198	3.33e-06
	CMOABC-ELM	**0.0445**	**3.67e-07**	**0.1396**	1.45e-06	**0.0128**	1.15e-05
C1-DTLZ3	NSGA-II	0.1272	1.5249e-05	0.6123	3.9949e-05	0.1545	5.3582e-05
	MOEA/D	0.1214	**8.0620e-07**	0.6077	**6.3848e-06**	0.1866	5.3251e-06
	CMOABC	0.1177	6.6234e-06	0.6669	1.2581e-05	**0.1221**	5.4484e-05
	CMOABC-ELM	**0.1101**	1.3577e-06	**0.6684**	3.3519e-05	0.1303	**5.2397e-05**
C2-DTLZ2	NSGA-II	0.1051	7.2271e-06	0.6471	3.4690e-05	**0.0436**	2.2266e-05
	MOEA/D	0.1287	**9.8021e-07**	0.6524	**2.1994e-05**	0.0830	**1.7476e-06**
	CMOABC	0.0984	7.5398e-06	0.6300	3.8331e-05	0.1317	3.5429e-05
	CMOABC-ELM	**0.0982**	1.4081e-05	**0.6674**	5.2598e-05	0.1303	6.7888e-05
C4-DTLZ4	NSGA-II	0.4500	0.0475	7.0760	0.64303	**0.0527**	**0.0008**
	MOEA/D	0.7031	0.5116	5.8514	10.8190	0.0859	0.0044
	CMOABC	0.4268	0.2315	7.1035	4.7342	0.1823	0.0069
	CMOABC-ELM	**0.2905**	**0.0343**	**7.7415**	**0.5204**	0.0682	0.0029

[2] Z. Liu and Y. Wang, "Handling Constrained Multiobjective Optimization Problems With Constraints in Both the Decision and Objective Spaces," IEEE Transactions on Evolutionary Computation, vol. 23, no. 5, pp. 870–884, October 2019.

[3] H. Jain and K. Deb, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point Based Nondominated Sorting Approach, Part II: Handling Constraints and Extending to an Adaptive Approach," IEEE Transactions on Evolutionary Computation, vol. 18, no. 4, pp. 602–622, August 2014.

[4] K. Deb and H. Jain, "An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints," IEEE Transactions on Evolutionary Computation, vol. 18, no. 4, pp. 577–601, August 2014.

[5] L.Ma, R.Wang, M.Chen, W.Wang, S.Cheng and Y.Shi, "A Novel Many-objective Evolutionary Algorithm Based on Transfer Learning with Kriging model," Information Sciences, vol. 509, pp. 437–456, 2020.

[6] X. Cai, P. Wang, L. Du, Z. Cui, W. Zhang and J. Chen, "Multi-Objective Three-Dimensional DV-Hop Localization Algorithm With NSGA-II," IEEE Sensors Journal, vol. 19, no. 21, pp. 10003–10015, 2019.

[7] M. Zhang, H. Wang, Z. Cui and J. Chen, "Hybrid multi-objective cuckoo search with dynamical local search," Memetic Computing, vol. 10, no.1, pp. 199–208, 2018.

[8] M. Shafie-khah, P. Siano and J. S. Catalão, "Optimal Demand Response Strategies to Mitigate Oligopolistic Behavior of Generation Companies Using a Multi-Objective Decision Analysis," IEEE Transactions on Power Systems, vol. 33, no. 4, pp. 4264–4274, July 2018.

[9] H. Farzin, M. Fotuhi-Firuzabad and M. Moeini-Aghtaie, "A Stochastic Multi-Objective Framework for Optimal Scheduling of Energy Storage Systems in Microgrids," IEEE Transactions on Smart Grid, vol. 8, no. 1, pp. 117–127, Jan. 2017.

[10] M. Gong, Z. Wang, Z. Zhu and L. Jiao, "A Similarity-Based Multiobjective Evolutionary Algorithm for Deployment Optimization of Near Space Communication System," IEEE Transactions on Evolutionary Computation, vol. 21, no. 6, pp. 878–897, Dec. 2017.

[11] H. Zhang, Y. Zhu, W. Zou, Xiaohui Yan, "A hybrid multi-objective artificial bee colony algorithm for burdening optimization of copper strip production," Applied Mathematical Modelling, vol.36, no. 6, pp. 2578–2591, 2012.

[12] M. P. Kleeman, B. A. Seibert, G. B. Lamont, K. M. Hopkinson and S. R. Graham, "Solving Multicommodity Capacitated Network Design Problems Using Multiobjective Evolutionary Algorithms," IEEE Transactions on Evolutionary Computation, vol. 16, no. 4, pp. 449–471, August 2012.

[13] A. Zhou, B. Qu, H. Li, S. Zhao, "Ponnuthurai Nagaratnam Suganthan, Qingfu Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art," Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 32–49, 2011.

[14] D. Karaboga, B. Akay, "A comparative study of Artificial Bee Colony algorithm," Applied Mathematics and Computation, vol. 214, no. 1, pp. 108–132, 2009.

[15] Q. Zhang, W. Liu and H. Li, "The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances," 2009 IEEE Congress on Evolutionary Computation, Trondheim, 2009, pp. 203–208.

[16] D. Karaboga, B. Akay, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," Journal of Global Optimization, vol. 39, no. 3, pp. 459–471, 2007.

[17] Y. G. Woldesenbet, G. G. Yen and B. G. Tessema, "Constraint Handling in Multiobjective Evolutionary Optimization," IEEE Transactions on Evolutionary Computation, vol. 13, no. 3, pp. 514–525, June 2009.

[18] G. B. Huang, Q.Y. Zhu, C. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," Proceedings of International Joint Conference on Neural Networks, Budapest, Hungary, vol. 2, 2004, pp. 25–29.

[19] Q. Y. Zhu, A. Qin,P. Suganthan, G. B. Huang, "Evolutionary extreme learning machine," Pattern Recognition, vol. 38, no. 10, pp 1759–1763, 2005.

[20] G. B. Huang, Q.Y. Zhu, C. Siew, "Extreme learning machine: Theory and applications," Neurocomputing, vol. 70, no.1–3, pp. 489–501, 2006.

[21] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182–197, April 2002.

[22] L. Ma, X. Wang, M. Huang, Z. Lin, L. Tian and H. Chen, "Two-Level Master-Slave RFID Networks Planning via Hybrid Multiobjective Artificial Bee Colony Optimizer," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 49, no. 5, pp. 861–880, May 2019.

[23] K. Li, K. Deb, Q. Zhang and S. Kwong, "An Evolutionary Many-Objective Optimization Algorithm Based on Dominance and Decomposition," IEEE Transactions on Evolutionary Computation, vol. 19, no. 5, pp. 694–716, Oct. 2015.

[24] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," IEEE Transactions on Evolutionary Computation, vol. 7, no. 2, pp. 174-188, April 2003.

[25] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, pp. 257-271, Nov. 1999.

[26] C. Carlos, V. David and L. Gary, "Evolutionary Algorithms for Solving Multi-Objective Problems Second Edition," Springer, New York, 2007.