A Novel Multi-objective Cultural Algorithm Embedding Five-Element Cycle Optimization

1st Zhengyan Mao
School of Information Science and Engineering
East China University of Science and Technology
Shanghai, China
020130072@mail.ecust.edu.cn

2nd Yue Xiang
School of Information Science and Engineering
East China University of Science and Technology
Shanghai, China
x15800380632@outlook.com

3rd Yijie Zhang
School of Information Science and Engineering
East China University of Science and Technology
Shanghai, China
363477534@qq.com

4th Mandan Liu
School of Information Science and Engineering
East China University of Science and Technology
Shanghai, China
liumandan@ecust.edu.cn

***Abstract*—The cultural algorithm, as a dual-inheritance framework designed for optimization problems, can incorporate any population-adopted evolutionary computation technique in its population space. On the other hand, based on the Five-Elements Cycle Model derived from the ancient Chinese Five Elements (metal, wood, water, fire, earth) theory, the five-elements cycle optimization algorithm was proved to be effective in solving continuous function optimization problems. In this work, we propose a multi-objective cultural algorithm with a five-elements-cycle-optimization-based population space, where the five-element cycle model is adopted as the evolution scheme in the population space of the cultural algorithm framework. Simulation results on 12 classic benchmark problems show that the proposed algorithm can effectively solve continuous optimization functions and obtains satisfactory non-dominated solutions compared with 8 representative multi-objective algorithms.**

***Index Terms*—multi-objective evolutionary optimization, cultural algorithm, five-elements cycle model**

I. Introduction

In real-world applications, such as engineering design, scientific experiments and business decision-making, to balance multiple conflicting objectives is an ineluctable problem that often requires decision makers to find several trade-off solutions in the feasible region [1]. When a problem involves more than one objective to be maximized or minimized, the task of finding such solutions is known as multi-objective optimization. Mathematically, a multi-objective optimization problem (MOOP) with n decision variables, M objectives, and J inequality and K equality constraints can be stated in the following form:

$$\begin{aligned} \text{Minimize} \quad & \mathbf{y} = f(\mathbf{x}) = (f_1(\mathbf{x}), \cdots, f_M(\mathbf{x})) \\ \text{subject to} \quad & g_i(\mathbf{x}) \leq 0, \ i = 1, 2, \cdots, J \\ & h_j(\mathbf{x}) = 0, \ j = 1, 2, \cdots, K \qquad , \\ \text{where} \quad & \mathbf{x} = (x_1, \cdots, x_n) \in \mathbf{X} \\ & \mathbf{y} = (y_1, \cdots, y_M) \in \mathbf{Y} \end{aligned} \tag{1}$$

where the n-dimensional vector $\mathbf{x}$ with n decision variables is called the decision vector, and $\mathbf{X}$ is called the decision space, $\mathbf{y}$ the objective vector, and $\mathbf{Y}$ the objective space.

In light of the versatility of heuristic optimization algorithms, the implementation of evolutionary algorithms for solving MOOPs has received considerable attention in recent years. One of the most popular works in the field of multi-objective evolutionary algorithms (MOEAs) is the NSGA-II [2], which was carried out in 2000. Afterwards, David W. Corne et al. proposed the Pareto envelope-based selection algorithm II (PESA-II, [3]) which assigned the selective fitness to the hyper-boxes and proved its superiority compared with its former version PESA. And in the same year, SPEA2 was presented in [4], which integrated a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method into its predecessor SPEA and performed competitively on both combinatorial and continuous test problems. And one of the classic swarm intelligence algorithms, Particle Swarm Optimization (PSO), was expanded to a multi-objective version (named MOPSO) by [5]. MOEA/D presented in [6] is also a noteworthy algorithm that decomposes an MOOP into a number of scalar optimization subproblems and optimized them simultaneously. In the past five years, quite a few newer MOEAs also have been proposed with competitive performance compared to the representative ones mentioned above. A new multi-objective optimization framework, comprised of non-dominated sorting, local search, and the farthest-candidate approach, named non-dominated sorting and local search (NSLS) algorithm was introduced in [7]. Literature [8] proposed an MOEA based on an enhanced inverted generational distance metric (termed MOEA/IGD-NS) which was able to omit noncontributing solutions during the evolutionary search. In the same year, a framework contained a bi-criterion evolution (BCE) with indicator-based evolutionary algorithm (IBEA) embedded into its non-Pareto criterion (NPC) evolution part was proposed in [9], and the effectiveness of this framework (termed BCE-IBEA) was shown by experiments on seven groups of test problems with various characteristics.

In 1994, Robert G. Reynolds first presented in [10] a conceptual model of Cultural Algorithm (CA) as a dual evolutionary framework metaphorically modeled the cultural evolution of human society. During the last three decades, Reynolds and his students have published a number of studies involving not only the evolutionary scheme adopted in the population space but also the knowledge structure used in the belief space.

Based on the mechanism of generation and restriction in the five elements (metal, wood, water, fire, earth) from the ancient Chinese Yin-Yang theory, in [11] Mandan Liu developed the Five-Elements Cycle Model (FECM) and established the five-elements cycle optimization algorithm (FECO) for solving continuous global optimization problems. Thereafter, the algorithm of FECO was successfully extended to a multi-objective version by Chunling Ye et al., Multi-Objective Five-Elements Cycle Optimization algorithm(MOFECO), and the authors proved its capability of obtaining Pareto solutions [12].

Considering the flexibility of the CA framework and the supreme performance of FECO on solving MOOPs, in this paper, we made a bold attempt of incorporating MOFECO into CA as its evolving scheme in the population space to solve MOOPs, and the principle and effectiveness of the proposed algorithm will be elaborated in the rest sections.

The remainder of this paper is organized as follows. Section II gives some basic ideas of the original algorithms of CA and FECO respectively. Section III focuses on the framework and implementation of the proposed algorithm MOFECO. Experimentation on 12 classic benchmark functions is presented in Section IV. Section V gives the results and statistical analysis. Section VI concludes this work and suggests some future directions.

II. Basic Principles of Cultural Algorithm and Five-Element Cycle Optimization

A. Cultural Algorithms

As can be seen in the framework of CA (Fig. 1), there are three key elements in CA – a population space where evolution occurs, a belief space storing useful experience, and the communication channel between the two spaces. The details of the three will be explained in the following.

1) Population Space: The individuals, each of which denotes a solution to the problem, evolve in the population space. The size of the population is preset as a parameter of the algorithm. According to [13], basically, the evolutionary mechanism adopted in the population space can be selected among any population-adopted evolutionary computation techniques. For instance, genetic algorithm and evolutionary programming were introduced into the population space of a CA-based testbed for solving constrained numerical optimization in [14].

2) Belief Space: The belief space can be viewed as the libraries in our real world that provide information, knowledge, and technology collected by all the appeared elites to help us progress towards a better direction. The knowledge stored in the belief space plays the same role for the population in CA – guide the evolution towards the promising areas. Several typical knowledge sources proposed in the literature are listed below.

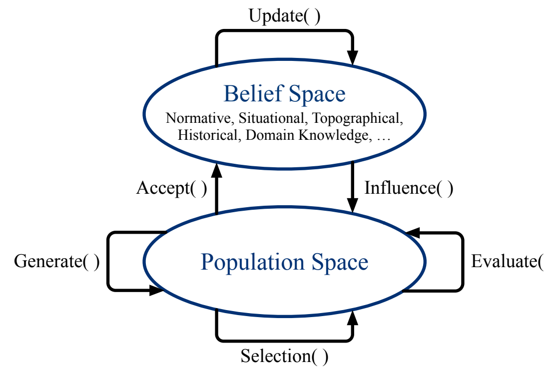

Fig. 1. Framework of Cultural Algorithm.

- Normative Knowledge: it gives standards or guidelines for the population to follow along with the evolution progress. More specifically, the closed intervals where the exemplar individuals locate in are preserved as normative knowledge.
- Situational Knowledge: it was first proposed by Chanjin Chung in [15] as a set of exemplars that provide their individual experience. In other words, the well-performed individuals will be stored in situational knowledge and in its influence functions they act like exemplars for other individuals to follow.
- Topographical Knowledge: Xidong Jin and Robert G. Reynolds used topographical knowledge to handle constraints when solving single-objective optimization problems (SOOPs) in [16] and [17]. The topographical knowledge segments the search region defined by normative knowledge into several hypercubes and then continues to split the promising areas into sub-hypercubes. The information of the hypercubes recorded in the topographical knowledge contains feasibility, weight, the leftmost corner's position, size of each hypercube, and counters for feasible and unfeasible individuals in each hypercube.
- Historical Knowledge and Domain Knowledge: these two were first proposed to deal with dynamic optimization problems. As dynamic objective optimization is not included in this work, refer to the literature [18] for details on these two knowledge sources.

3) Communication Channel:

- Acceptance Function: it determines how many and which of the individuals in the current population space can be used to edit or improve the knowledge in the belief space. One of the common ways to define the acceptance function is to set an appropriate percentage parameter p_a and select the best $p_a \times N$ individuals to impact the update operation of the knowledge.

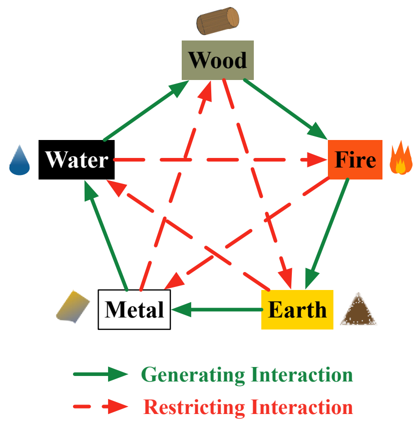

Fig. 2. The generating and restricting interaction among the five elements.

- Influence Function: it is the vehicle by which the knowledge sources to reproduce new individuals in the population. Every knowledge source has its corresponding influence function and the function can be modified according to the specific nature of the optimization problem.

B. Five-Element Cycle Optimization

In FECM, the generating and restricting interactions between the five elements are depicted in Fig. 2. The generating interactions, which denoted by the green arrows, resembles the relationship between mother and child as the mother feeds and raises the child. The red arrows represent the restricting interaction between grandmother and grandson as the grandmother used to take the responsibility of disciplining the grandson rested with the grandparents. Taking wood for instance, in the outer circle wood generates fire and is generated by water, while in the inner circle wood inhibits earth and is inhibited by metal.

In the FECM dynamic system comprised of L elements (in the five-element cycle, $L = 5$), the dynamically varying mass of each element $x_i(k)$ at time k is signified as $m_i(k)$, and the force exerted on the element by other elements, $F_i(k)$, is computed as follows [11]:

$$\begin{aligned} F_i(k) = \omega_{gp} \cdot \ln \left[\frac{m_{i-1}(k)}{m_i(k)}\right] - \omega_{rp} \cdot \ln \left[\frac{m_{i-2}(k)}{m_i(k)}\right] \\ - \omega_{ga} \cdot \ln \left[\frac{m_i(k)}{m_{i+1}(k)}\right] - \omega_{ra} \cdot \ln \left[\frac{m_i(k)}{m_{i+2}(k)}\right], \end{aligned} \tag{2}$$

where $i = 1, 2, \cdots, L$. The subscripts of $m_{i-2}(k)$, $m_{i-1}(k)$, $m_i(k)$, $m_{i+1}(k)$, and $m_{i+2}(k)$ represent that i circulates in the order of $1, 2, \cdots, L$. ω_{gp} is the weight of the force that one element is generated by its parent element; ω_{rp} is the weight of the force that one element is inhibited by its grandparent element; ω_{ga} is the weight of the force that one element generates its child element, ω_{ra} is the weight of the force that one element inhibits its grandchild element. The four parameters should be positive numerical values in the range of $[0, 1]$ and are set as 1 in the original literature [11].

Based on the FECM, the FECO can be established as an iterated optimization algorithm. Suppose there are q cycles in the system and each cycle contains L elements. When iteration number $k = 0$, we randomly generate $L \times q$ initial individuals, each of which is denoted as x_{ij} ($i \in \{1, 2, \cdots, L\}$, $j \in \{1, 2, \cdots, q\}$) corresponding to one solution to the optimization problem. For each individual, calculate its objective function value $f(x_{ij}(k))$ as its mass $m_{ij}(k)$ and then the force exerted on x_{ij}, $F_{ij}(k)$, can be computed by Eq. 2.

According to the value of force $F_{ij}(k)$, each individual x_{ij} updates in the following way. If $F_{ij}(k) > 0$, x_{ij} is regarded as a good solution and stayed unchanged for the next iteration, otherwise x_{ij} will be updated following Eq. 3:

$$\begin{aligned} & x_{ij,d}(k+1) = \\ & \begin{cases} x_{ij,d}(k) + r_s\left(x_{i^*j,d}(k) - x_{ij,d}(k)\right), & \text{if } r_m < P_m \\ x_{ij,d}(k) + r_s\left(x_{\text{best},d} - x_{ij,d}(k)\right), & \text{else} \end{cases}, \\ & \qquad (i = 1, 2, \cdots, L;\ j = 1, 2, \cdots, q;\ d = 1, 2, \cdots, n) \end{aligned} \tag{3}$$

where $x_{ij,d}(k)$ is the d-th component in the n-dimensional vector $x_{ij}(k)$. P_m is a probability predefined for the update operation, and r_m is a randomly generated number in the range of $[0, 1]$. $x_{i^*j}(k)$ is the element with the biggest $F_{ij}(k)$ value in the current j-th cycle. And $\mathbf{x}_{\text{best}}$ is the best solution found so far.

The iteration progresses in the aforementioned way and terminates once the pre-defined termination condition is reached.

III. The Proposed Multi-objective Cultural Algorithm Embedding Five-Element Cycle Optimization (MOCAFECO)

A novel combination of two evolutionary algorithms is put forward in this paper. This section will dwell on how the proposed algorithm MOCAFECO works.

A. Evolution Scheme in Population Space – Five-Element Cycle Optimization

As MOCAFECO is designed for solving MOOPs, similar to the MOFECO presented in [12], some necessary modifications on the computation of the masses and forces and the update and mutation operations in the FECM are made and will be explained in this part.

1) Expression of Element Space: In the population space of MOCAFECO, each individual, namely a solution to the problem, can be seen as an element in the FECM. For example, suppose there are q L-element cycles in the system, $\mathbf{x}_{ij}(k)$ represents the i-th element in the j-th cycle at the k-th iteration. Hence the population of size $L \times q$ can be expressed as:

$$\mathbf{p} = \overbrace{\begin{bmatrix} \mathbf{x}_{11} & \mathbf{x}_{21} & \cdots & \mathbf{x}_{L1} \\ \mathbf{x}_{12} & \mathbf{x}_{22} & \cdots & \mathbf{x}_{L2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{x}_{1q} & \mathbf{x}_{2q} & \cdots & \mathbf{x}_{Lq} \end{bmatrix}}^{L \text{ elements in each cycle}} \left.\vphantom{\begin{matrix} a \\ a \\ a \\ a \end{matrix}}\right\} q \text{ cycles}, \tag{4}$$

where each element $\mathbf{x}_{ij}$ has n decision variables: $\mathbf{x}_{ij} = [x_{1,ij}, \cdots, x_{n,ij}]$.

In the j-th cycle, each individual (element) $\mathbf{x}_{ij}(k)$ has M objective values, so its M masses are represented as $m_{r,ij}(k)$ where $r \in \{1, 2, \cdots, M\}$. Accordingly, there are M forces exerted on $\mathbf{x}_{ij}(k)$, denoted by $F_{r,ij}(k)$ with $r \in \{1, 2, \cdots, M\}$. And $F_{r,ij}(k)$ is computed by

$$\begin{aligned} F_{r,ij}(k) = \ln\left[\frac{m_{r,(i-1)j}(k)}{m_{r,ij}(k)}\right] - \ln\left[\frac{m_{r,(i-2)j}(k)}{m_i(k)}\right] \\ - \ln\left[\frac{m_{r,ij}(k)}{m_{r,(i+1)j}(k)}\right] - \ln\left[\frac{m_{r,ij}(k)}{m_{r,(i+2)j}(k)}\right]. \end{aligned} \quad (5)$$

Thereafter, the update and mutation operation can be done according to the two randomly selected force values $F_{u1,ij}(k)$ and $F_{u2,ij}(k)$ $(u1,\ u2 \in \{1, 2, \cdots, M\})$. Individuals with $F_{u1,ij}(k)$ and $F_{u2,ij}(k)$ that satisfy

$$\begin{cases} F_{u1,ij}(k) > 0 \\ F_{u2,ij}(k) > 0 \end{cases} \quad (6)$$

are regarded as good solutions and will be kept unchanged, otherwise the individuals would be updated and mutated.

2) Update of Elements: If an individual has at least one negative force in the two forces selected, which means it is probably not a good solution, first perform update operation on it:

$$x_{d,ij}^{\text{updated}}(k) = x_{d,ij}(k) + V_{d,ij}(k), \quad (7)$$

where the notion of velocity vector $V_{d,ij}(k)$ borrowed from particle swarm optimization algorithm [19] is defined as follows:

$$V_{d,ij}(k) = \begin{cases} \omega \times V_{d,ij}(k-1) + r_1 \times r_s \times (x_{j,d}^{\text{local}}(k) - x_{d,ij}(k)), \\ \qquad \text{if } r_m < p_u(k) \\ \omega \times V_{d,ij}(k-1) + r_2 \times r_s \times (x_d^{\text{best}}(k) - x_{d,ij}(k)), \\ \qquad \text{if } r_m \geq p_u(k) \end{cases}, \quad (8)$$

where ω is the inertia weight; d is randomly selected within $\{1, 2, \cdots, n\}$; r_1 and r_2 are pre-defined constants; r_s is a random coefficient in the range of $[0, 1]$. In MOCAFECO, a local-global update probability p_u is introduced to decide the basis of the update – the local optimal individual in the j-th cycle $x_{j,d}^{\text{local}}(k)$, or the global best individual $x_d^{\text{best}}(k)$. And r_m is a random scalar drawn from the standard normal distribution $\mathbb{N}(0, 1)$. The calculation of p_u follows Eq. 9:

$$p_u(k) = 1 - \left[p_{u_{\min}} + (p_{u_{\max}} - p_{u_{\min}}) \times e^{-20\cdot\left(\frac{k}{T}\right)^6}\right], \quad (9)$$

where $p_{u_{\min}}$ and $p_{u_{\max}}$ are predefined minimum and maximum update probability; k is the current iteration and T is the maximum iteration. Therefore, the value of p_u is low at the early period, increases gradually in the medium, and then stays in a high level at the late stage of evolution. When $p_{u_{\min}}$ and $p_{u_{\max}}$ are set as 0.2 and 0.8, the variation curve of p_u is shown in Fig. 3.

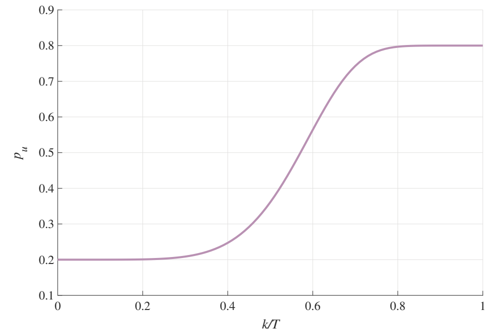

Fig. 3. Nonlinear variation of local-global update probability p_u.

Algorithm 1 Update($\mathbf{x}_{ij}(k)$)

Input: $\mathbf{x}_{ij}(k)$, $p_u(k)$
Output: $\mathbf{x}_{ij}^{\text{updated}}(k)$
Randomly generate r_m from $\mathbb{N}(0, 1)$
if $r_m < p_u(k)$ **then**
$\mathbf{x}_{ij}^{\text{updated}}(k) \Leftarrow \text{LocalUpdate}\left(\mathbf{x}_{ij}(k)\right)$
else
$\mathbf{x}_{ij}^{\text{updated}}(k) \Leftarrow \text{GlobalUpdate}\left(\mathbf{x}_{ij}(k)\right)$
end if

3) Mutation of Elements: Afterwards, perform mutation operation on the updated individuals:

$$x_{ij}^{\text{evolved}}(k) = \begin{cases} x_{ij}^{\text{updated}}(k) + \mathbb{U}(-\sigma_1, \sigma_1), & \text{if } k \leq \dfrac{T}{4} \\ x_{ij}^{\text{updated}}(k) + \mathbb{C}(0, \sigma_2), & \text{if } \dfrac{T}{4} < k \leq \dfrac{3T}{4} \\ x_{ij}^{\text{updated}}(k) + \mathbb{N}(0, \sigma_3), & \text{else} \end{cases}, \quad (10)$$

where uniform, Cauchy and Gaussian distribution mutation operators are used in different phases of iteration.

Algorithm 2 Mutate($\mathbf{x}_{ij}^{\text{updated}}(k)$)

Input: $\mathbf{x}_{ij}^{\text{updated}}(k)$
Output: $\mathbf{x}_{ij}^{\text{evolved}}(k)$
if $k \leq \frac{T}{4}$ **then**
$\mathbf{x}_{ij}^{\text{evolved}}(k) \Leftarrow \text{UniformMutate}(\mathbf{x}_{ij}^{\text{updated}}(k))$
else if $\frac{T}{4} < k \leq \frac{3T}{4}$ **then**
$\mathbf{x}_{ij}^{\text{evolved}}(k) \Leftarrow \text{CauchyMutate}(\mathbf{x}_{ij}^{\text{updated}}(k))$
else
$\mathbf{x}_{ij}^{\text{evolved}}(k) \Leftarrow \text{GaussianMutate}(\mathbf{x}_{ij}^{\text{updated}}(k))$
end if

The pseudocode of the evolution process within FECM adopted in the population of MOCAFECO is given in Algorithm 3.

B. Knowledge Sources in Belief Space

After the update and mutation operations, update the knowledge in the belief space and then influence all the current

Algorithm 3 Evolution in FECM

Input: $\mathbf{x}_{ij}(k)$, k
Output: $\mathbf{x}_{ij}^{\text{evolved}}(k)$
  Calculate $p_u(k)$
  for $j = 1 : q$ **do**
    for $i = 1 : L$ **do**
      for $r = 1 : M$ **do**
        Calculate $F_{r,ij}(k)$
      end for
      Select $u1,\ u2$ from $\{1, 2, \cdots, M\}$
      if $F_{u1,ij}(k) < 0$ or $F_{u2,ij}(k) < 0$ **then**
        $\mathbf{x}_{ij}^{\text{updated}}(k) \Leftarrow \text{Update}\left(\mathbf{x}_{ij}(k)\right)$
        $\mathbf{x}_{ij}^{\text{evolved}}(k) \Leftarrow \text{Mutate}\left(\mathbf{x}_{ij}^{\text{updated}}(k)\right)$
      else
        $\mathbf{x}_{ij}^{\text{evolved}}(k) = \mathbf{x}_{ij}(k)$
      end if
    end for
  end for

population by a randomly selected influence function. In MOCAFECO, normative, situational, topographical knowledge are utilized in the belief space and the data structures of them will be elaborated in this part. It should be noted that, when it comes to sorting the solutions in the population, the techniques of non-dominated sorting and calculation of crowding distance from [2] are adopted.

- Normative Knowledge: in MOCAFECO, normative knowledge is used to preserve the intervals where the accepted individuals located. The data structure is similar to the original one designed for SOOPs mentioned in Section II-A2:

$$NK(k) = \{\mathbf{l},\ \mathbf{u}\}, \tag{11}$$

  where $\mathbf{l} = \{l_1, \cdots, l_n\}$ and $\mathbf{u} = \{u_1, \cdots, u_n\}$ are the minimum and maximum values of each decision variable that are found in all the individuals sorted out by the acceptance function. While being updated, the accepted individuals sorted out from the current population by the acceptance function are used to modify the normative knowledge.
- Situational Knowledge: naturally, for MOOPs, the non-dominated individuals are saved in situational knowledge as the elite solutions.

$$SK(k) = \left\{\mathbf{E}^1,\ \mathbf{E}^2,\ \cdots,\ \mathbf{E}^s\right\}, \tag{12}$$

  where s is the pre-defined capacity of situational knowledge and here we make it equal to the size of the population N, and $\mathbf{E}^i$ is the i-th non-dominated individual with $i \in \{1, 2, \cdots, s\}$. When situational knowledge is updated, perform the non-dominated sorting method on the current population and stored the non-dominated individuals in the repository SK.
- Topographical Knowledge: it was originally used for preserving promising regions of constrained problems, whereas in MOCAFECO, we use it to expand the spread the non-dominated solutions. Similarly, we define a parameter called $nGrid$ and split the search area into $nGrid$ segments along each dimension of the objective space, so that we have $(nGrid)^M$ hypercubes in the objective space defined by the normative knowledge. And then find out the edge area and perform fine tuning on the individuals located inside the edge area. It is necessary to mention here that the update of topographical knowledge is not done at every iteration of the algorithm. Instead it only updates before it is chosen to influence the population, the purpose of which is to save computation resource.

After the update and mutation operations in the FECM, the population will be influenced by a randomly selected influence function in the belief space. The influence functions will be illustrated in the next part.

C. Communication Channel

1) Acceptance Function: The acceptance function is to accept a proportion of high-performing individuals to update the belief space. In MOCAFECO, we use an acceptance rate $p_a = 35\%$, which means 35% of the individuals in the current population with the best non-dominated ranks are accepted. And the accepted individuals are allowed to update the normative knowledge.

2) Influence Functions:

- Influence function of normative knowledge: it applies increment or decrement upon the individuals according to the intervals saved in normative knowledge. The adjustment can be expressed as:

$$x_{d,ij}^{\text{influenced}}(k) = \begin{cases} x_{d,ij}^{\text{evolved}}(k) + |\Delta x_d|, & \text{if } x_{d,ij}^{\text{evolved}}(k) < l_d \\ x_{d,ij}^{\text{evolved}}(k) - |\Delta x_d|, & \text{if } x_{d,ij}^{\text{evolved}}(k) > u_d \\ x_{d,ij}^{\text{evolved}}(k) + \beta \times \Delta x_d, & \text{otherwise} \end{cases}, \tag{13}$$

  where d is randomly selected from set $\{1, 2, \cdots, n\}$, which means the adjustment is exerted on a random dimension of the decision vector $\mathbf{x}_{ij}$. And the increment/decrement Δx_d is computed by

$$\Delta x_d = \alpha \times (u_d - l_d) \times r_c, \tag{14}$$

  where α is an influence coefficient and r_c is a random scalar drawn from the standard normal distribution $\mathbb{N}(0, 1)$. If the individual to be influenced already lies in the normative region, use another influence coefficient β.
- Influence function of situational knowledge: it aims to urge the individuals towards the exemplars stored in the situational knowledge:

$$x_{d,ij}^{\text{influenced}}(k) = \begin{cases} x_{d,ij}^{\text{evolved}}(k) + |\Delta x_d|, & \text{if } x_{d,ij}^{\text{evolved}}(k) < E_d^{\text{random}} \\ x_{d,ij}^{\text{evolved}}(k) - |\Delta x_d|, & \text{if } x_{d,ij}^{\text{evolved}}(k) > E_d^{\text{random}} \\ x_{d,ij}^{\text{evolved}}(k) + \Delta x_d, & \text{if } x_{d,ij}^{\text{evolved}}(k) = E_d^{\text{random}} \end{cases}, \tag{15}$$

where E_d^{random} is the d-th component of a randomly chosen elite $\mathbf{E}^{\text{random}}$ from the repository set SK and d is randomly selected from $\{1, 2, \cdots, n\}$. And the adjustment Δx_d is calculated by

$$\Delta x_d = \gamma \times (x_d^{\max} - x_d^{\min}) \times r_c \ , \tag{16}$$

where γ is an influence coefficient; $x_d^{\max}$ and $x_d^{\min}$ are the d-th element of $\mathbf{x}^{\max}$ and $\mathbf{x}^{\min}$, which are the upper and lower bounds of the decision variables given by the optimization problem; r_c is a random scalar drawn from the standard normal distribution $\mathbb{N}(0, 1)$.

- Influence function driven by normative and situational knowledge:

$$x_{d,ij}^{\text{influenced}}(k) = \begin{cases} x_{d,ij}^{\text{evolved}}(k) + |\Delta x_d|, & \text{if } x_{d,ij}^{\text{evolved}}(k) < E_d^{\text{random}} \\ x_{d,ij}^{\text{evolved}}(k) - |\Delta x_d|, & \text{if } x_{d,ij}^{\text{evolved}}(k) > E_d^{\text{random}} \\ x_{d,ij}^{\text{evolved}}(k) + \Delta x_d, & \text{if } x_{d,ij}^{\text{evolved}}(k) = E_d^{\text{random}} \end{cases} , \tag{17}$$

where E_d^{random} is the d-th component of $\mathbf{E}^{\text{random}}$, a randomly chosen elite from the external repository SK. Different from the influence function of situational knowledge, the adjustment Δx_d here is calculated according to the intervals in normative knowledge:

$$\Delta x_d = \alpha \times (u_d - l_d) \times r_c \ , \tag{18}$$

where r_c is a random scalar drawn from the standard normal distribution $\mathbb{N}(0, 1)$.

- Influence function of topographical knowledge: it simply exerts fine-tuning on the individuals located in the edge areas:

$$x_{d,ij}^{\text{influenced}}(k) = x_{d,ij}^{\text{evolved}}(k) + \Delta x_d, \tag{19}$$

where

$$\Delta x_d = 0.1 \times (x_d^{\max} - x_d^{\min}) \times r_c \ , \tag{20}$$

d is randomly selected from $\{1, 2, \cdots, n\}$, and again, r_c is a random scalar drawn from the standard normal distribution $\mathbb{N}(0, 1)$.

D. Implementation Framework

The implementation flowchart of MOCAFECO is shown in Fig. 4.

IV. Simulation on Benchmarks and Performance Comparison Between MOCAFECO and Other 8 Multi-objective Algorithms

A. Multi-objective Continuous Test Benchmarks

For simulation, 3 bi-objective test benchmarks (ZDT1/2/3 selected from the ZDT set [20]) and 9 tri-objective benchmarks (DTLZ2/4/5 selected from the DTLZ set [21] and MaF1/5/6/8/11/12 from the MaF set [22]) are used in the comparison between MOCAFECO and 5 classic multi-objective algorithms and 3 newer MOEAs.

B. Algorithms for Comparison and Parameter Settings

In this paper, we choose 8 multi-objective algorithms, including five classic multi-objective algorithms NSGA-II, PESA-II, SPEA2, MOPSO and MOEA/D, and three well-performed algorithms NSLS, MOEA/IGD-NS and BCE-IBEA selected from more recent researches, to investigate the performance of MOCAFECO.

For all the 9 algorithms, the population size is set equally to $25 \times M$, and for MOCAFECO, PESA-II, SPEA2 and MOEA/D, which are algorithms with external repositories, the size of the repositories is equal to the population size. The other parameter settings of the 8 algorithms are the same as in the original literature.

For MOCAFECO, the parameters are set as: $p_a = 0.35$, $s = N = 25 \times M$, $nGrid = 10$, $\alpha = 0.3$, $\beta = 0.5$, $\gamma = 0.1$, $p_{u_{\min}} = 0.2$, $p_{u_{\max}} = 0.8$, $\omega = 0.05$, $r_1 = r_2 = 1$. For L and q in the FECM, as we have $N = L \times q$ and $N = 25 \times M$, we set L fixed as 5 and therefore $q = 5 \times M$. Most of the above are the same as they were configured in the original works.

C. Experimental Configurations

The algorithm of MOCAFECO and the other 8 algorithms are written in Matlab scripts and the testing experiments are implemented in Matlab R2019a on a 3.60GHz Intel(R) Core(TM) i7-4790 processor under Windows 10. The source code of the NSGA-II, PESA2, SPEA-II, MOPSO and MOEA/D is available from the EMOO repository in [23], and the implementations of NSLS, MOEA/IGD-NS, BCE-IBEA are from the PlatEMO [24]. All the algorithms run 20 times independently for each test benchmark problem. Additionally, to conduct a fair comparison between the nine methods, instead of adopting a maximum number of iterations as the termination condition, we terminate each algorithm once the calculation times of the objective functions reaches a certain value (50,000 times for the ZDT and the DTLZ suites; $\max\{100000, 10000 \times n\}$ times for the MaF suites).

D. Performance Metrics

Inverted generational distance (IGD) [25], unary ϵ-indicator ($I_{\epsilon 1}$) [26], and hypervolume (HV) [27] are adopted in this paper to evaluate the performance of the 9 algorithms.

V. Results and Discussion

In this simulation, MOCAFECO is proved to have the ability to find non-dominated solutions with satisfactory convergence to the true PFs of the 12 multi-objective problems. Due to space limitations, two typical results on test problems ZDT3 and MaF12 obtained by all the 9 algorithms are presented in Fig. 5 and Fig. 6. In both figures, the non-dominated solutions obtained in the 20 runs are marked by circles in 20 different colors (solutions of each run are denoted by each color). Fig. 5 shows that MOCAFEO and BCE-IBEA are the only two algorithms that possibly converge to the true PF of ZDT3 in all the 20 runs. For example, in the second subfigure of NSGA-II, it can be noted that at least one set of solutions obtained by NSGA-II fails to find the rightmost piece of the ZDT3' PF.

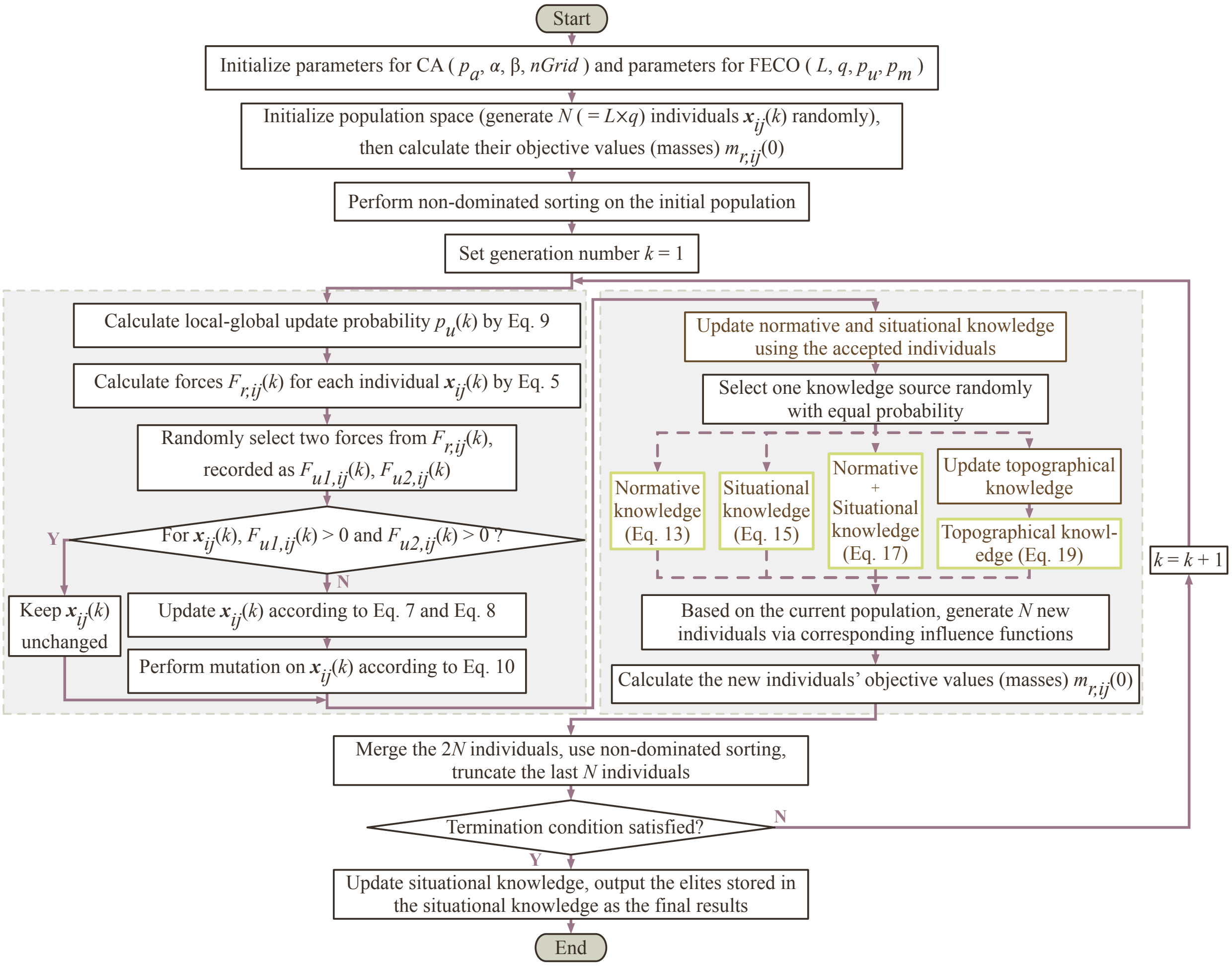

Fig. 4. Framework of MOCAFECO.

Also, as can be seen from Fig. 6, the solutions obtained by MOCAFECO cover most areas of the Pareto surface, second only to BCE-IBEA. Here, MOCAFECO's ability to better cover the Pareto fronts is generally attributed to the use of topographical knowledge in the belief space of CA and the combined mutation strategy in the FECM.

To quantitatively measure the quality of the solutions obtained by MOCAFECO, median values of the three indicators (inverted generational distance (IGD), hypervolume (HV), and unary ϵ-indicator ($I_{\epsilon 1}$)) over the 20 runs are computed. Also, upon the median results, the Friedman test is performed as it was recommended in [28], and the statistics of the three indicators are presented in Table I, II and III respectively.

The first indicator IGD is a comprehensive metric that measures the convergence of solutions to the true PF and their distribution. It can be seen from the data in Table I that, compared with Table II and III, the Friedman mean ranks achieved by the algorithms on indicator IGD are quite close (in the interval of $[3.2500, 7.3333]$). And according to the final rank in Table I, the proposed MOCAFECO performs better than half of the rest 8 algorithms, which can also be noted from the median values of MOCAFECO.

As one of the typical Pareto-compliant metrics that can be used to measure the comprehensive performance of a Pareto solution set, HV provides a qualitative measure of convergence as well as diversity [1]. In Table II, it is apparent that MOCAFECO has outstanding performance on the indicator of hypervolume as it achieves the first rank in the Friedman test among all the 9 algorithms with a quite small p-value. It should be noted that, as the Friedman test is performed in ascending order, and the fact that larger hypervolume value indicates better performance, in Table II larger Friedman mean rank is preferred.

Another Pareto-compliant metric, the unary ϵ-indicator, is also adopted in this paper to comprehensively evaluate the obtained solution sets of the selected multi-objective optimizers. Intuitively, the unary ϵ-indicator measures how much do we need to translate/scale the solution set so that it covers the reference set or the true PF [29]. Table III compares the unary ϵ-indicators of the 9 algorithms, and the data shows that, similar to the results in Table II, the proposed algorithm achieves the first rank in the Friedman test over the unary ϵ-indicator with a small p-value.

In general, as a novel attempt of introducing the FECM into

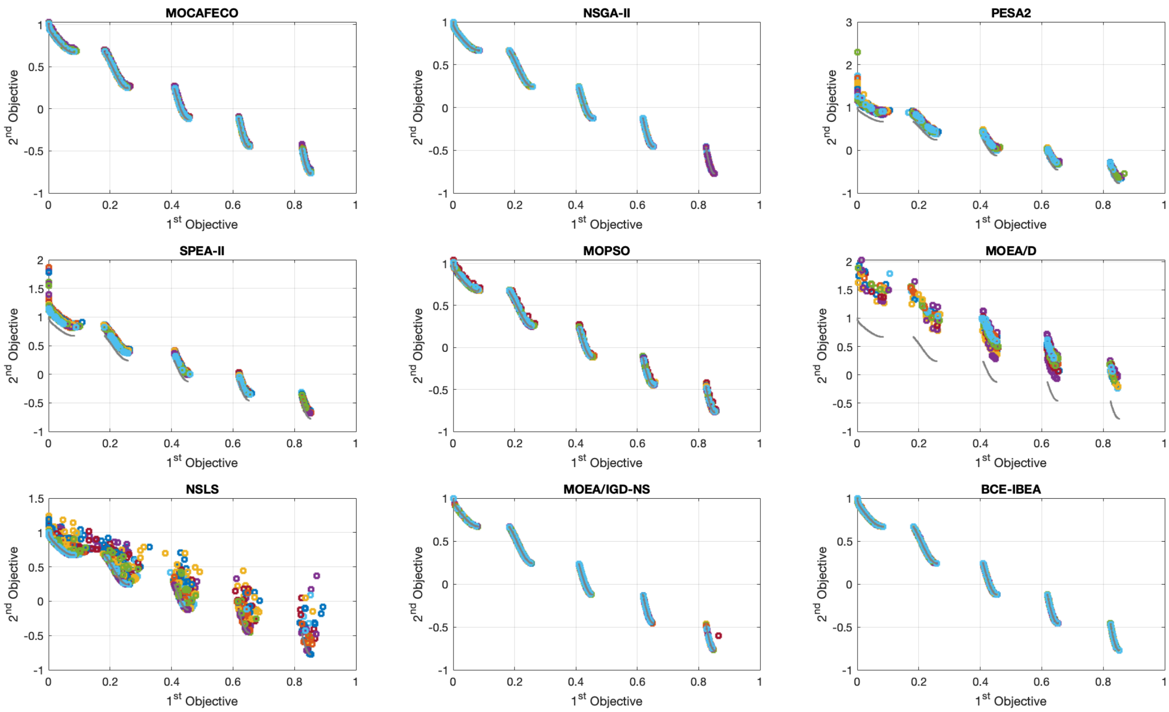

Fig. 5. Plots of the non-dominated solutions in the objective space of ZDT3 obtained by 9 algorithms in 20 independent runs (solutions obtained in each run are denoted by each color).

TABLE I
MEDIANS OF INVERTED GENERATIONAL DISTANCE (IGD) AND RANKS ACHIEVED BY THE FRIEDMAN TEST (A: ALGORITHMS, P: PROBLEMS)

IGD A / P	**MOCAFECO**	NSGA-II	PESA2	SPEA-II	MOPSO	MOEA/D	NSLS	MOEA/IGD-NS	BCE-IBEA
ZDT1	0.014183	0.009645	0.129334	0.09108	0.024368	0.109832	0.010571	0.007697	0.007998
ZDT2	0.011489	0.009845	0.148760	0.113129	0.019088	1.294850	0.022095	0.007650	0.007808
ZDT3	0.152613	0.165316	0.159438	0.128541	0.155690	0.273838	0.166409	0.155044	0.165305
DTLZ2	0.071753	0.075651	0.149795	0.131218	0.174605	0.086025	0.063563	0.059212	0.063741
DTLZ4	0.364195	0.300299	0.144345	0.117881	0.176252	0.146677	0.079404	0.168412	0.054655
DTLZ5	0.153638	0.153369	0.168825	0.160782	0.160016	0.156620	0.153442	0.153304	0.153231
MaF1	0.065853	0.060712	0.114662	0.090476	0.169750	0.078548	0.048772	0.045060	0.048744
MaF5	4.380506	4.110485	3.118054	3.446501	3.667022	4.030302	4.010777	4.055071	3.994221
MaF6	0.178618	0.178943	0.500827	0.188395	0.190435	0.181181	0.178557	0.178422	0.178199
MaF8	0.685032	0.711462	28.170782	0.782971	8.592302	198.930815	0.674738	0.711051	0.684380
MaF11	2.222541	2.247190	2.053610	2.107801	2.098570	2.442144	2.239839	2.293998	2.274423
MaF12	3.161307	3.159197	2.897493	2.945645	2.948057	3.177798	3.137569	3.164169	3.169458
Friedman mean rank	5.0000	5.1667	5.9167	5.0000	5.8333	7.3333	3.9167	3.5833	3.2500
final rank	4	6	8	4	7	9	3	2	1
p-value	0.0066								

the CA framework, MOCAFECO has acceptable performance on the indicator of inverted generational distance and shows significant superiority over the other 8 algorithms on two comprehensive performance indicators HV and unary ϵ-indicator.

VI. CONCLUSIONS AND FUTURE WORK

The feasibility of adopting the FECM as the evolutionary scheme in the CA framework is verified in this paper. Compared with 5 classic and 3 latest multi-objective algorithms, the proposed algorithm MOCAFECO performs competitively and shows its great potential for solving continuous optimization problems.

There is still abundant room to refine the proposed algorithm. First, a mechanism for dealing with constraints in MOOPs can be included in the MOCAFECO. In addition,

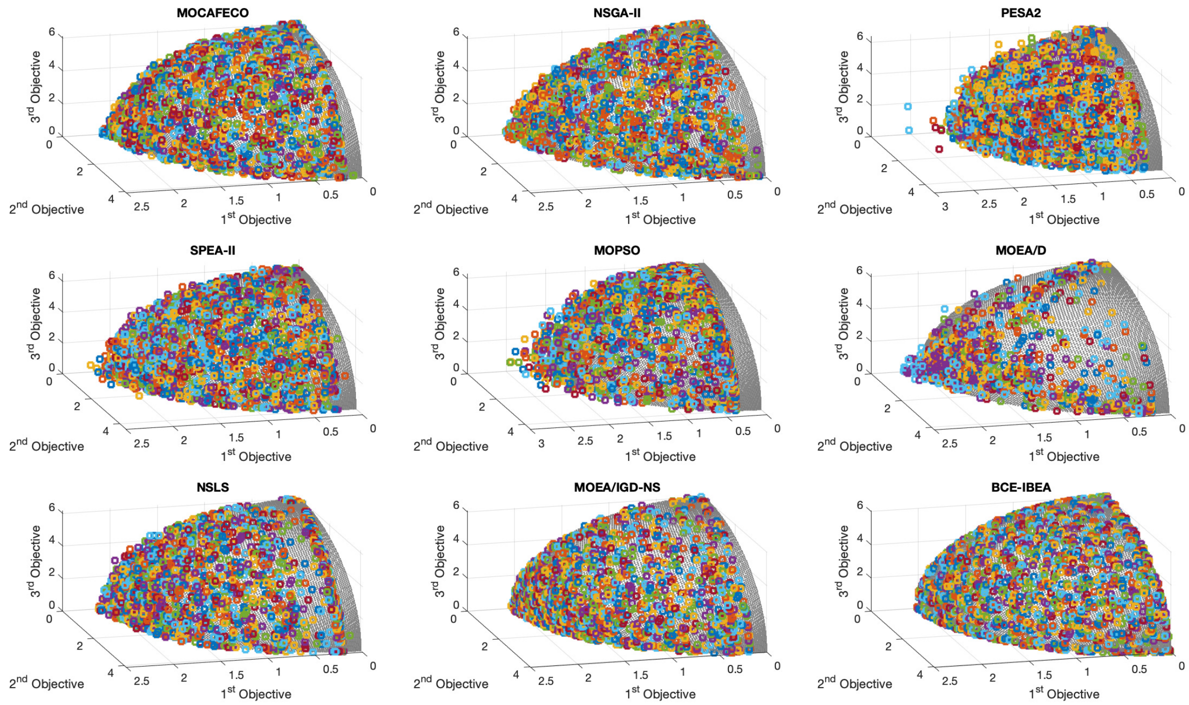

Fig. 6. Plots of the non-dominated solutions in the objective space of MaF12 obtained by 9 algorithms in 20 independent runs (solutions obtained in each run are denoted by each color).

TABLE II
MEDIANS OF HYPERVOLUME (HV) AND RANKS ACHIEVED BY THE FRIEDMAN TEST (A: ALGORITHMS, P: PROBLEMS)

HV \ A / P	**MOCAFECO**	NSGA-II	PESA2	SPEA-II	MOPSO	MOEA/D	NSLS	MOEA/IGD-NS	BCE-IBEA
ZDT1	1.095070	1.099928	0.830576	0.922692	1.041203	0.806723	1.015229	0.892170	0.927179
ZDT2	1.098024	1.100000	0.720335	0.696774	0.734238	0.100000	0.889709	0.728187	0.719629
ZDT3	1.665168	0.937016	0.801689	0.795810	0.681363	0.516177	0.758055	0.927594	0.884965
DTLZ2	1.100000	1.100000	0.885209	0.959991	0.916623	0.943219	1.100000	0.914728	1.074353
DTLZ4	1.100000	1.100000	1.099999	0.900187	1.100000	1.100000	1.100000	1.100000	0.890715
DTLZ5	0.777817	0.777817	0.394298	0.458433	0.538302	0.409302	0.479101	0.352878	0.388038
MaF1	1.099999	1.099999	0.731315	0.744027	0.639410	0.657070	0.819296	0.709609	0.739269
MaF5	8.800000	8.800000	8.800000	8.791239	8.800000	8.628677	8.800000	8.727918	8.335463
MaF6	0.777817	0.777817	0.632291	0.351541	0.501087	0.431472	0.517511	0.546969	0.485437
MaF8	1.888281	1.834780	0.000000	1.175753	0.000000	0.000000	1.891454	1.335510	1.303236
MaF11	2.198133	2.190809	1.917869	2.052736	1.874455	1.831368	1.938912	1.993403	2.050974
MaF12	2.170949	2.160338	1.166010	1.506147	1.726413	1.188865	2.144023	1.864810	1.705992
Friedman mean rank	8.2500	8.0833	3.3333	3.9167	4.4583	2.1667	6.3750	4.5000	3.9167
final rank	1	2	8	6	5	9	3	4	6
p-value	8.2972e-10								

in this work, limited parameters comparison experiment has been made and most parameters are used as their default values. Hence a thorough analysis of the parameters used in the MOCAFECO can be made for achieving better performance. Furthermore, the effect of each knowledge source on searching for promising solutions can be further analyzed, and also other effective knowledge sources can be exploited in the CA part of MOCAFECO.

REFERENCES

[1] K. Deb, *Multi-objective optimization using evolutionary algorithms*. John Wiley & Sons, Jul. 2001.

[2] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II," in *International conference on parallel problem solving from nature*. Paris, France: Springer, 2000, pp. 849–858.

TABLE III
MEDIANS OF UNARY ϵ-INDICATOR ($I_{\epsilon 1}$) AND RANKS ACHIEVED BY THE FRIEDMAN TEST (A: ALGORITHMS, P: PROBLEMS)

$I_{\epsilon 1}$ A / P	**MOCAFECO**	NSGA-II	PESA2	SPEA-II	MOPSO	MOEA/D	NSLS	MOEA/IGD-NS	BCE-IBEA
ZDT1	0.004930	0.000072	0.269424	0.177308	0.058797	0.293277	0.084771	0.207830	0.172821
ZDT2	0.001976	0.000000	0.379665	0.403226	0.365762	1.000000	0.210291	0.371813	0.380371
ZDT3	0.045217	0.773369	0.908696	0.914575	1.029022	1.194208	0.952330	0.782792	0.825420
DTLZ2	0.000000	0.000000	0.214791	0.140009	0.183377	0.156781	0.000000	0.185272	0.025647
DTLZ4	0.000000	0.000000	0.000001	0.199813	0.000000	0.000000	0.000000	0.000000	0.209285
DTLZ5	0.000000	0.000000	0.383519	0.319384	0.239515	0.368516	0.298716	0.424940	0.389779
MaF1	0.000000	0.000000	0.368684	0.355972	0.460589	0.442929	0.280703	0.390390	0.360730
MaF5	0.000002	0.000002	0.000002	0.008759	0.000002	0.171321	0.000002	0.072080	0.464535
MaF6	0.000001	0.000001	0.145526	0.426276	0.276730	0.346344	0.260306	0.230848	0.292379
MaF8	0.000192	0.053309	28.713661	0.712335	25.443304	197.41299	0.003365	0.552578	0.584852
MaF11	0.001867	0.009191	0.282131	0.147264	0.325545	0.368632	0.261088	0.206597	0.149026
MaF12	0.029049	0.039660	1.033988	0.693851	0.473585	1.011133	0.055975	0.335188	0.494006
Friedman mean rank	1.6250	1.8750	6.6667	6.0833	5.6250	7.9167	3.5417	5.5833	6.0833
final rank	1	2	8	6	5	9	3	4	6
p-value	2.8581e-10								

[3] D. W. Corne, N. R. Jerram, J. D. Knowles, and M. J. Oates, "PESA-II: Region-based selection in evolutionary multiobjective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference*, 2001, pp. 283–290.

[4] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the strength Pareto evolutionary algorithm," *TIK-report*, vol. 103, 2001.

[5] C. A. C. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256–279, May 2004.

[6] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, Nov. 2007.

[7] B. Chen, W. Zeng, Y. Lin, and D. Zhang, "A new local search-based multiobjective optimization algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 1, pp. 50–73, Jan. 2015.

[8] Y. Tian, X. Zhang, R. Cheng, and Y. Jin, "A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric," in *2016 IEEE Congress on Evolutionary Computation (CEC)*, Jul. 2016, pp. 5222–5229.

[9] M. Li, S. Yang, and X. Liu, "Pareto or non-Pareto: Bi-criterion evolution in multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 645–665, Sep. 2016.

[10] R. G. Reynolds, "An introduction to cultural algorithms," in *Proceedings of the third annual conference on evolutionary programming*. Singapore: World Scientific, 1994, pp. 131–139.

[11] M. Liu, "Five-elements cycle optimization algorithm for solving continuous optimization problems," in *2017 IEEE 4th International Conference on Soft Computing & Machine Intelligence (ISCMI)*, 2017.

[12] C. Ye, Z. Mao, and M. Liu, "A novel multi-objective five-elements cycle optimization algorithm," *Algorithms*, vol. 12, no. 11, p. 244, Nov. 2019.

[13] C. A. C. Coello and R. L. Becerra, "Evolutionary multiobjective optimization using a cultural algorithm," in *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No.03EX706)*. IEEE, 2003, pp. 6–13.

[14] C.-J. Chung and R. G. Reynolds, "A testbed for solving optimization problems using cultural algorithms." in *Evolutionary programming*, 1996, pp. 225–236.

[15] C. Chung, "Knowledge-based approaches to self-adaptation in cultural algorithms," Ph.D. dissertation, Wayne State University, Detroit, Michigan, 1997.

[16] X. Jin and R. G. Reynolds, "Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach," in *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, vol. 3. IEEE, 1999, pp. 1672–1678.

[17] ——, "Using knowledge-based system with hierarchical architecture to guide the search of evolutionary computation," in *Proceedings 11th International Conference on Tools with Artificial Intelligence*. Chicago, IL, USA: IEEE, 1999, pp. 29–36. [Online]. Available: http://gen.lib.rus.ec/scimag/index.php?s=10.1109/tai.1999.809762

[18] S. M. Saleem, "Knowledge-based solution to dynamic optimization problems using cultural algorithms," Ph.D. dissertation, Wayne State University, Detroit, Michigan, 2001.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, Nov 1995, pp. 1942–1948 vol.4.

[20] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary computation*, vol. 8, no. 2, pp. 173–195, 2000.

[21] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multiobjective optimization," in *Evolutionary multiobjective optimization*. Springer, 2005, pp. 105–145.

[22] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "A benchmark test suite for evolutionary many-objective optimization," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 67–81, 2017.

[23] C. A. C. Coello, "EMOO Repository," http://delta.cs.cinvestav.mx/~ccoello/EMOO/, 2015.

[24] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "PlatEMO: A MATLAB platform for evolutionary multi-objective optimization [educational forum]," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[25] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *Trans. Evol. Comp*, vol. 7, no. 2, p. 117132, Apr. 2003. [Online]. Available: https://doi.org/10.1109/TEVC.2003.810758

[26] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: An analysis and review," *IEEE Transactions on evolutionary computation*, vol. 7, no. 2, pp. 117–132, 2003.

[27] E. Zitzler, *Evolutionary algorithms for multiobjective optimization: Methods and applications*. Citeseer, 1999, vol. 63.

[28] 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 and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, Mar. 2011.

[29] J. Knowles, L. Thiele, and E. Zitzler, "A tutorial on the performance assessment of stochastic multiobjective optimizers," Computer Engineering and Networks Laboratory (TIK), ETH Zurich, Switzerland, Tech. Rep., Feb. 2006, revised version.