10.1109/ISCMI.2017.8279601

2017 4th IEEE International Conference on Soft Computing and Machine Intelligence

Five-elements Cycle Optimization Algorithm for Solving Continuous Optimization Problems

Mandan Liu
School of Information Science and Engineering, East China University of Science and Technology
Shanghai 200237, China
E-mail: liumandan@ecust.edu.cn

Abstract—**The Five-elements Cycle Optimization Algorithm (FECO) is proposed in this paper inspired by the theory of Five-elements in Chinese traditional culture. It is built for finding the optimal solution of continuous functions based on the Five-elements Cycle Model which characterizes the mechanism of generation and restriction among five elements. The comparison with 11 optimization algorithms based on various mechanisms for 23 benchmark functions is given, which indicates the suitability and universality of FECO.**

Keywords-continuous optimization; heuristic algorithms; five-elements cycle optimization

I. INTRODUCTION

Optimization problems are common in many disciplines and various domains. To solve optimization problems, we have to find solutions which are optimal or near-optimal with respect to some goals. In the past several decades, different kinds of optimization algorithms have been designed and applied to solve different optimization problems. Most of the popular optimization approaches are nature-inspired, e.g., Simulated Annealing (SA) [1], Evolutionary Programming (EP) [2], Genetic Algorithm (GA) [3], Evolution Strategies (ES) [4], Tabu Search (TS) [5], Ant Colony Optimization (ACO) [6], Particle Swarm Optimization (PSO) [7], Differential Evolution (DE) [8], Artificial Immune Systems (AIS) [9], Cultural Algorithm (CA) [10], Memetic Algorithm (MA) [11], etc. Many improved algorithms based on the above-mentioned optimization algorithms have been designed to obtain better performance. Furthermore, various novel optimization algorithms were proposed and researched to solve continuous optimization problems or combinatorial optimization problems.

The theory of Yin-Yang, Five-elements, and the Eight Trigrams of Chinese traditional cultural indicates the inherent laws of things. It shows us how the states of things develop gradually to homeostasis. It implies a possible approach to solve present-day problems in engineering. Lei Xu [12] proposed Bayesian Ying-Yang Learning, which provides a general framework that accommodates typical learning approaches from a unified perspective. WenRan Zhang [13] proposed Yin-Yang bipolar sets and bipolar dynamic modus ponens for equilibrium-based bipolar knowledge fusion and visualization with different applications. Sik Chung Tam [14] proposed the Algorithm of Changes (AOC) based on the concept of hexagram operators in I Ching, and applied to Bin Packing Problems. Rong Zhao [15] proposed Yin-Yang Optimization (YYO) with only few control parameters, and applied to Traveling Salesman Problems.

We have developed the Five-elements Cycle Model (FECM) based on the mechanism of generation and restriction in five elements, and proposed the Five-elements Cycle Optimization algorithm (FECO) for solving Traveling Salesman problems in [16]. In this paper, we propose the Five-elements Cycle Optimization algorithm for finding the optimal solution of continuous functions.

The rest of this paper is organized as follows. The FECM is explicated in Section II. Based on FECM, Section III gives the framework and implementation of FECO. In Section IV, we compare the performance of FECO with other 11 optimization algorithms. Finally, we conclude this paper and suggest potential future work in Section V.

II. FIVE-ELEMENTS CYCLE MODEL

The theory of the Five-elements in Chinese philosophy conceives the world as dynamic states, or phases, of constant change. It refers to the five elements of wood, fire, earth, metal, and water. There are two main kinds of interactions between five elements, generating interaction and restricting interaction. The generation cycle is the same as a mother and child relationship, where the child is dependent upon the mother for nourishment and thus growth and well-being. The order of generation of five elements is, wood generates fire, fire generates earth, earth generates metal, metal generates water, water in its turn generates wood. The restriction cycle is the same as a grandmother and grandson relationship, where the responsibility of disciplining the children rested with the grandparents. The order of restriction of five elements is, wood inhibits earth, earth inhibits water, water inhibits fire, fire inhibits metal, metal in its turn inhibits wood.

Acting on the generating and restricting interactions, each element is under the influence of the other four elements, in this way, all five elements establish an intricate homeostasis in nature. We proposed FECM to model the transformation process of five-elements. FECM extends five elements to L elements.

Suppose there is a dynamic system which is comprised of L elements. The mass of L elements at k time is defined as $m_i(k)$, $i = 1,2,\cdots,L$. The force exerted on every element by other elements at k time is defined as $F_i(k)$, which is corresponding to $m_i(k)$. FECM can be described as (1) which is explicated in [16].

$$
\begin{cases}
F_i(k) = w_{gp} \cdot ln\left[\frac{m_{i-1}(k)}{m_i(k)}\right] - w_{rp} \cdot ln\left[\frac{m_{i-2}(k)}{m_i(k)}\right] \\
\qquad -w_{ga} \cdot ln\left[\frac{m_i(k)}{m_{i+1}(k)}\right] - w_{ra} \cdot ln\left[\frac{m_i(k)}{m_{i+2}(k)}\right] \\
m_i(k+1) = m_i(k) \cdot \frac{2}{1+\exp(-F_i(k))}
\end{cases} \quad (1)
$$

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 $m_{i+2}(k)$ mean i circulates in order of $1, 2, \cdots, L$. w_{gp} is the weight of force that one element is generated by its parent element, w_{rp} is the weight of force that one element is inhibited by its grandparent elementw_{ga}, is the weight of force that one element generates its child element, w_{ra} is the weight of force that one element inhibits its grandchild element. w_{gp}, w_{rp}, w_{ga}, and w_{ra} should be positive numerical value in the range of [0, 1].

FECM can be converged in pace with the mass of five elements changing by generating and restricting interaction.

III. Five-Elements Cycle Optimization Algorithm Implementation

A continuous box-constrained global optimization problem can be stated as (2):

$$
\min f(x) \quad \text{subject to } x \in \left[x_d^{(low)}, x_d^{(high)}\right]^D \quad (2)
$$

where $x = (x_1, x_2, \cdots, x_D)^T \in R^D$ is the decision vector, $\left[x_d^{(low)}, x_d^{(high)}\right]^D (d = 1,2,\cdots,D)$ is the feasible region of the search space, $f\colon R^D \to R$ is the objective function.

Based on FECM, we propose the Five-elements Cycle Optimization (FECO) to find the optimal solution of $f(x)$.

A. Expression of Element Space

Every solution x of (2) corresponds with an element in FECM. To establish the relationship of competition and collaboration among elements, we construct the Element Space (see Fig. 1).

In the Element Space, there are q cycles, and L elements in each cycle. We use $x_{ij}(k)$ to denote the ith element of jth cycle at kth iteration, $x_{ij}(k)$ represents one of solution of $f(x)$. $m_{ij}(k)$, which is the mass of element $x_{ij}(k)$, represents the objective function $f(x_{ij}(k))$. The force exerted on $x_{ij}(k)$ by other elements in jth cycle is denoted by $F_{ij}(k)$, which is calculated by (3).

$$
\begin{aligned}
F_{ij}(k) = w_{gp} \cdot ln\left[\frac{m_{(i-1)j}(k)}{m_{ij}(k)}\right] - w_{rp} \cdot ln\left[\frac{m_{(i-2)j}(k)}{m_{ij}(k)}\right] \\
-w_{ga} \cdot ln\left[\frac{m_{ij}(k)}{m_{(i+1)j}(k)}\right] - w_{ra} \cdot ln\left[\frac{m_{ij}(k)}{m_{(i+2)j}(k)}\right]
\end{aligned} \quad (3)
$$

$F_{ij}(k)$ represents the force among elements in jth cycle, it is independent with elements in other cycles.

B. Establishment of FECO

After constructing the Element Space, FECO can be established to solve the optimization problem described by (2). To illustrate clearly, the corresponding relationship among FECM, FECO and optimization problems is shown in Table I.

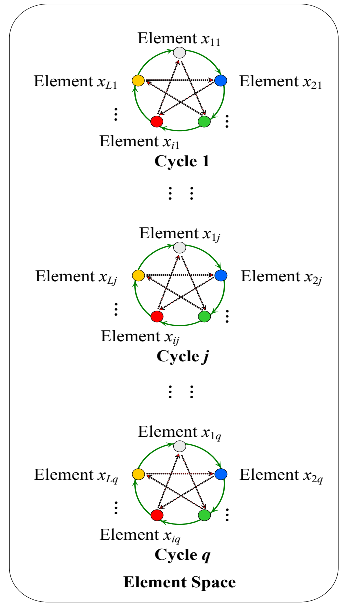

Figure 1. The structure of *Element Space*

TABLE I. Corresponding Relationship among FECM, FECO and Optimization Problems

FECM	FECO	Optimization problems
Elements (wood, fire, earth, metal, and water)	Elements $x_{ij}(k)$ $(i = 1,2,\cdots,L; j = 1,2,\cdots,q)$	Solutions x
Mass of elements $m_i(k)$ $(i = 1,2,\cdots,L)$	Mass of elements $m_{ij}(k)$ $(i = 1,2,\cdots,L; j = 1,2,\cdots,q)$	Objective function $f(x)$
Force exerted on elements $F_i(k)$ $(i = 1,2,\cdots,L)$	Force exerted on elements $F_{ij}(k)$ $(i = 1,2,\cdots,L; j = 1,2,\cdots,q)$	Variables estimating the quality of solutions

FECO is an iterated algorithm. When $k = 0$, we randomly generate $L \times q$ initial solutions $x_{ij}(0)$ ($i = 1,2,\cdots,L; j = 1,2,\cdots,q$) in the feasible region of the search space, calculate their objective function values, which are denoted by $m_{ij}(0)$, set the initial $F_{ij}(0)$ to 0. In the kth iteration, firstly evaluate objective function $m_{ij}(k)$ of $x_{ij}(k)$, then calculate $F_{ij}(k)$ by (3). The value of $F_{ij}(k)$ indicates the capability of $x_{ij}(k)$, therefore $F_{ij}(k)$ determines the update pattern of $x_{ij}(k)$ to $x_{ij}(k+1)$. The process of iterations makes good solutions reserved and inferior solutions replaced, so that the best solution could be obtained until the expected maximal iteration.

C. Update of Elements

The update of $x_{ij}(k+1)$ depends on $F_{ij}(k)$. If $F_{ij}(k) > 0$, it means that $x_{ij}(k)$ is a good solution, because the smaller $m_{ij}(k)$ is, the larger $F_{ij}(k)$ is, in order to balance all the elements. In this situation, $x_{ij}(k)$ should be reserved, so $x_{ij}(k+1)$ is:

$$x_{ij}(k+1) = x_{ij}(k) \quad if \;\; F_{ij}(k) > 0 \tag{4}$$

If $F_{ij}(k) \leq 0$, it means that $x_{ij}(k)$ is probably not a good solution, its mass $m_{ij}(k+1)$ must become thinner so the force $F_{ij}(k)$ exerted on $x_{ij}(k)$ must be negative. Normally, $x_{ij}(k)$ is a D dimension vector, denoted by $[x_{ij,1}(k)$, $x_{ij,2}(k)$, $\cdots$, $x_{ij,d}(k)$, $\cdots$, $x_{ij,D}(k)]$. In this situation, $x_{ij}(k+1)$ will be updated when $F_{ij}(k) \leq 0$:

$$\begin{cases} x_{ij,d}(k+1) = x_{ij,d}(k) + r_s\left(x_{i^*j,d}(k) - x_{ij,d}(k)\right) \; if \;\; r_m < P_m \\ x_{ij,d}(k+1) = x_{i^*j,d}(k) + r_s\left(x_{best,d} - x_{i^*j,d}(k)\right) \; if \;\; r_m \geq P_m \end{cases}$$

$$(i = 1,2,\cdots,L;\; j = 1,2,\cdots,q;\; d = 1,2,\cdots,D) \tag{5}$$

where p_m is a pre-given probability, r_m is a random number in the range of [0, 1], r_s is a random number in the range of $[-p_s,\, 1+p_s]$, p_s is a scale factor. $x_{i^*j}(k)$ is the element whose $F_{ij}(k)$ is the biggest in jth cycle, i.e. $F_{i^*j}(k) \geq F_{ij}(k)$ $(i^* \in \{1,2,\cdots,L\}, i\forall\{1,2,\cdots,L\})$. $x_{i^*j}(k)$ is denoted by $[x_{i^*j,1}(k)$, $x_{i^*j,2}(k)$, $\cdots$, $x_{i^*j,d}(k)$, $\cdots$, $x_{i^*j,D}(k)]$. x_{best} is the best solution so far, $x_{best} = [x_{best,1}$, $x_{best,2}$, $\cdots$, $x_{best,d}$, $\cdots$, $x_{best,D}]$.

D. Flowchart of FECO

The flowchart of FECO is shown in Fig. 2.

The initialization includes setting parameters of L, q, $max\ iteration$, p_s, p_m, w_{gp}, w_{rp}, w_{ga}, and w_{ra}, setting or calculating the initial values of $x_{ij}(0)$, $m_{ij}(0)$, and $F_{ij}(0)$.

In the kth iteration, for the jth cycle, $F_{ij}(k)$ ($i = 1,2,\cdots,L$) is calculated based on $m_{ij}(k)$ which was derived from $x_{ij}(k)$ last iteration. $x_{ij}(k+1)$ is updated by (4) or (5) according to the value of $F_{ij}(k)$. Then $m_{ij}(k+1)$ can be calculated. After $k \geq max\ iteration$, the final optimal solution x_{best} can be given.

IV. Experiments

The FECO algorithm was coded in MATLAB R2014a and executed under a 2.6 GHz Intel Core i7-4510U, 8 GB RAM, and Windows 8.1.

A. Benchmark Functions

The benchmark functions we used to analyze the performance of FECO and to compare with some algorithms based on various mechanisms are from [17] where the detailed description of the 23 functions can be found. Functions f_1–f_{13} are high-dimensional problems ($D = 30$). The dimensions of functions f_{14}–f_{23} are 2, 3, 4, or 6 respectively. The benchmark functions are grouped into three categories [17]. Functions f_1 – f_7 are unimodal. Functions f_8–f_{13} are multimodal functions where the number of local minima increases exponentially with the problem dimension. Functions f_{14}–f_{23} are low-dimensional functions which have only a few local minima.

B. Comparison with Optimization Algorithms Based on Various Mechanisms

Because of the restrictions of pages, the parameters comparison experiments of FECO are omitted in this paper. According to the experiment results, we set parameters of FECO as: $L = 5$, $q = 20$, $p_s = 0.6$, $p_m = 0.9$, $w_{gp} = w_{rp} = w_{ga} = w_{ra} = 1$. The optimization results of 23 benchmark functions by FECO are average values of 51 independent runs.

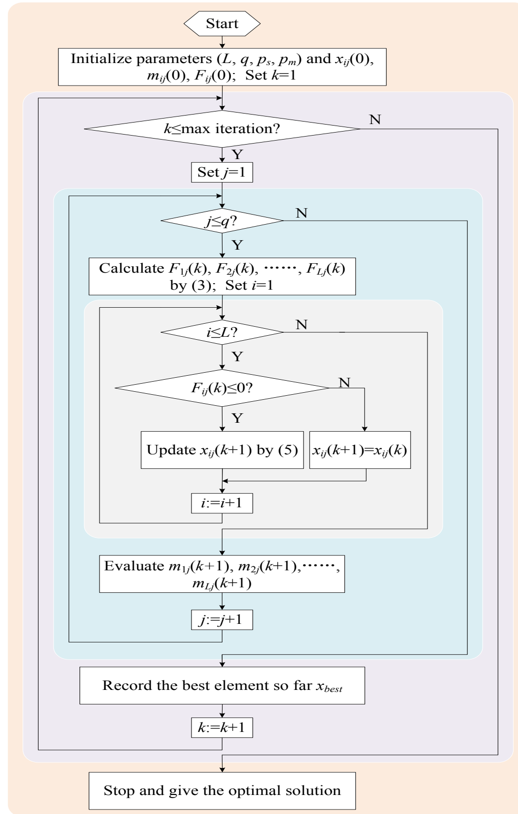

Figure 2. The flowchart of FECO algorithm

Comparison experiments are carried out for 23 benchmark functions between FECO and other optimization algorithms based on various mechanisms, which are GA, DE, Classical Evolutionary Programming (CEP) [17], Fast Evolutionary Programming (FEP) [17], Conventional Evolutionary Strategy (CES) [18], Fast Evolutionary Strategy (FES) [18], Generalized Generation Gap Model with Generic Parent-centric Recombination Operator (G3PCX) [19], PSO, Group Search Optimizer (GSO) [20], Real-Coded Biogeography-Based Optimization (RCBBO) [21], and Real-Coded Chemical Reaction Optimization (RCCRO) [22]. The mean value of minimum of these algorithms are from published references listed above.

The number of function evaluations (FEs) for the compared algorithms are given in corresponding references. The FE limits of FECO for every benchmark functions are less than or equal to the minimal FEs of other compared algorithms, in order to ensure the validity of the comparison experiments.

The comparison results among 12 algorithms are shown in Table II, the best result is highlighted for each function. The data in Table II are the mean value of minimum for every algorithm and every function. "+", "–", or "~" in parentheses indicates FECO is better than, worse than, or similar to the other 11 algorithms using Wilcoxon sign rank test with significance value 0.05.

For high-dimensional unimodal functions f_1–f_7, FECO performs visibly better than GA and DE, but visibly worse than PSO. FECO performs better than or similar to all the compared algorithms for f_6 . For high-dimensional multimodal functions with many local minima f_8–f_{13}, FECO performs visibly better than GA, CEP, CES, PSO, and G3PCX, but visibly worse than DE. It performs better for high-dimensional multimodal functions than high-dimensional unimodal functions. FECO performs better than all the compared algorithms for f_{10}. For low-dimensional multimodal functions with only a few local minima f_{14}–f_{23}, FECO performs visibly better than GA, FES, CES, PSO, DE and G3PCX, but visibly worse than GSO and RCCRO. It also performs better for low-dimensional multimodal functions than high-dimensional unimodal functions. FECO performs better than all the compared algorithms except RCCRO for f_{21}, f_{22}, and f_{23}.

We also obtained the statistical results using the Friedman test with data in Table II, the Friedman mean ranks are shown in Table III. FECO is ranked third among 12 algorithms. For all the 23 benchmark functions, FECO performs worse than RCCRO and GSO, but better than the other algorithms. Although different test methods probably lead to different statistical results, the results in Table II can evince that FECO has the ability to get good solutions for these benchmark functions.

Generally, as a novel algorithm, FECO performs good despite it isn't the best algorithm. The experiments indicate that it is a viable algorithm for solving continuous function optimization problems.

TABLE II. OPTIMIZATION RESULTS OF COMPARISON AND STATISTICAL RESULTS WITH WILCOXON SIGN RANK TEST (SIGNIFICANCE VALUE 0.05), " + ," " − ," OR " ~ " IN PARENTHESES INDICATES FECO IS BETTER THAN, WORSE THAN, OR SIMILAR TO COMPARED ALGORITHM

	GA	FEP	CEP	FES	CES	PSO	GSO	DE	G3PCX	RCBBO	RCCRO	FECO
f_1	3.171E+00 (+)	5.700E-04 (+)	2.200E-04 (+)	2.500E-04 (+)	3.400E-05 (+)	3.693E-37 (-)	1.948E-08 (+)	6.576E-06 (+)	**6.404E-79** (-)	1.390E-03 (+)	6.427E-07 (+)	6.602E-16
f_2	5.771E-01 (+)	8.100E-03 (+)	2.600E-03 (+)	6.000E-02 (+)	2.100E-02 (+)	**2.917E-24** (-)	3.704E-05 (+)	2.894E-04 (+)	2.803E+01 (+)	7.990E-02 (+)	2.196E-03 (+)	6.514E-12
f_3	9.750E+03 (+)	1.600E-02 (-)	5.000E-02 (-)	1.400E-03 (-)	1.300E-04 (-)	1.198E-03 (-)	5.783E+00 (-)	1.212E+04 (+)	**1.064E-76** (-)	2.270E+01 (-)	2.966E-07 (-)	5.460E+02
f_4	7.961E+00 (+)	3.000E-01 (-)	2.000E+00 (-)	**5.500E-03** (-)	3.500E-01 (-)	4.123E-01 (-)	1.078E-01 (-)	5.790E+00 (+)	4.543E+01 (+)	3.090E-02 (-)	9.318E-03 (-)	3.546E+00
f_5	3.386E+02 (+)	5.060E+00 (-)	6.170E+00 (-)	3.328E+01 (-)	6.690E+00 (-)	3.736E+01 (-)	4.984E+01 (~)	9.338E+01 (+)	**3.091E+00** (-)	5.540E+01 (~)	2.706E+01 (-)	5.968E+01
f_6	3.697E+00 (+)	**0.000E+00** (~)	5.778E+02 (+)	**0.000E+00** (~)	4.112E+02 (+)	1.460E-01 (+)	1.600E-02 (+)	**0.000E+00** (~)	9.462E+01 (+)	**0.000E+00** (~)	**0.000E+00** (~)	**0.000E+00**
f_7	1.045E-01 (+)	7.600E-03 (-)	1.800E-02 (+)	1.200E-02 (~)	3.000E-02 (+)	9.902E-03 (-)	7.377E-02 (+)	3.967E-02 (+)	9.797E-01 (+)	1.750E-02 (+)	**5.405E-03** (-)	1.244E-02
f_8	-1.257E+04 (-)	-1.255E+04 (-)	-7.917E+03 (+)	-1.256E+04 (-)	-7.550E+03 (+)	-9.660E+03 (+)	**-1.257E+04** (-)	-1.257E+04 (-)	-2.577E+03 (+)	-1.257E+04 (-)	-1.257E+04 (-)	-1.169E+04
f_9	6.509E-01 (-)	4.600E-02 (-)	8.900E+01 (+)	1.600E-01 (-)	7.082E+01 (+)	2.079E+01 (+)	1.018E+00 (-)	**7.261E-05** (-)	1.740E+02 (+)	2.620E-02 (-)	9.077E-04 (-)	1.112E+01
f_{10}	8.678E-01 (+)	1.800E-02 (+)	9.200E+00 (+)	1.200E-02 (+)	9.070E+00 (+)	1.340E-03 (+)	2.655E-05 (+)	7.136E-04 (+)	1.352E+01 (+)	2.510E-02 (+)	1.944E-03 (+)	**7.630E-09**
f_{11}	1.004E+00 (+)	1.600E-02 (+)	8.600E-02 (+)	3.700E-02 (+)	3.800E-01 (+)	2.323E-01 (+)	3.079E-02 (+)	**9.054E-05** (-)	1.127E-02 (+)	8.490E-02 (+)	1.117E-02 (+)	4.808E-04
f_{12}	4.357E-02 (+)	9.200E-06 (-)	1.760E+00 (+)	2.800E-02 (+)	1.180E+00 (+)	3.950E-02 (+)	**2.765E-11** (-)	1.886E-07 (-)	4.593E+00 (+)	3.280E-05 (-)	2.074E-02 (+)	2.033E-03
f_{13}	1.681E-01 (+)	1.600E-04 (~)	1.400E+00 (+)	4.700E-05 (~)	1.390E+00 (+)	5.052E-02 (-)	4.695E-05 (~)	9.519E-07 (~)	2.349E+01 (+)	3.720E-04 (~)	**7.048E-07** (~)	1.342E-01
f_{14}	9.989E-01 (-)	1.220E+00 (+)	1.660E+00 (+)	1.200E+00 (+)	2.160E+00 (+)	1.024E+00 (-)	**9.980E-01** (-)	1.576E+00 (+)	1.231E+01 (+)	9.980E-01 (-)	**9.980E-01** (-)	1.038E+00
f_{15}	7.088E-03 (+)	5.000E-04 (-)	4.700E-04 (-)	9.700E-04 (+)	1.200E-03 (+)	3.807E-04 (-)	**3.771E-04** (-)	5.372E-04 (-)	5.332E-04 (-)	7.860E-04 (+)	5.555E-04 (~)	5.759E-04
f_{16}	-1.030E+00 (-)	-1.030E+00 (-)	-1.030E+00 (-)	-1.032E+00 (-)	-1.032E+00 (-)	-1.016E+00 (~)	**-1.032E+00** (-)	-1.019E+00 (~)	-4.928E-01 (+)	-1.031E+00 (-)	-1.032E+00 (-)	-1.016E+00
f_{17}	4.040E-01 (+)	3.980E-01 (-)	3.980E-01 (-)	3.980E-01 (-)	3.980E-01 (-)	4.040E-01 (+)	**3.979E-01** (-)	3.995E-01 (+)	5.560E+01 (+)	3.984E-01 (-)	**3.979E-01** (-)	3.981E-01
f_{18}	7.503E+00 (+)	3.020E+00 (+)	**3.000E+00** (-)	**3.000E+00** (-)	**3.000E+00** (-)	3.005E+00 (+)	**3.000E+00** (-)	3.479E+00 (+)	8.670E+00 (+)	3.010E+00 (+)	3.001E+00 (-)	3.000E+00
f_{19}	-3.862E+00 (-)	-3.860E+00 (-)	-3.860E+00 (-)	-3.860E+00 (-)	-3.860E+00 (-)	-3.858E+00 (-)	-3.863E+00 (-)	-3.862E+00 (-)	-3.598E+00 (+)	-3.862E+00 (-)	**-3.863E+00** (-)	-3.863E+00
f_{20}	-3.263E+00 (+)	-3.270E+00 (~)	-3.280E+00 (~)	-3.230E+00 (+)	-3.240E+00 (+)	-3.185E+00 (+)	-3.270E+00 (~)	-3.316E+00 (-)	-1.980E-01 (+)	-3.317E+00 (-)	**-3.319E+00** (-)	-3.289E+00
f_{21}	-5.165E+00 (+)	-5.520E+00 (+)	-6.860E+00 (+)	-5.540E+00 (+)	-6.960E+00 (+)	-7.544E+00 (+)	-6.090E+00 (+)	-8.739E+00 (+)	-7.476E-01 (+)	-5.513E+00 (+)	**-1.011E+01** (-)	-1.002E+01
f_{22}	-5.443E+00 (+)	-5.520E+00 (+)	-8.270E+00 (+)	-6.760E+00 (+)	-8.310E+00 (+)	-8.355E+00 (+)	-6.555E+00 (+)	-9.199E+00 (+)	-9.468E-01 (+)	-6.800E+00 (+)	**-1.035E+01** (~)	-9.996E+00
f_{23}	-4.911E+00 (+)	-6.570E+00 (+)	-9.100E+00 (+)	-7.630E+00 (+)	-8.500E+00 (+)	-8.944E+00 (+)	-7.402E+00 (+)	-9.229E+00 (+)	-1.130E+00 (+)	-7.285E+00 (+)	**-1.048E+01** (~)	-9.897E+00

TABLE III. STATISTICAL RESULTS OF COMPARISON WITH FRIEDMAN TEST (SIGNIFICANCE VALUE 0.05), $p-value$ IS 2.7550E-10

	GA	FEP	CEP	FES	CES	PSO	GSO	DE	G3PCX	RCBBO	RCCRO	FECO
Friedman mean rank	9.3478	5.8696	7.2391	6.0435	7.5870	6.2609	5.1739	5.6304	9.8261	6.3478	3.3478	5.3261
Final rank	11	5	9	6	10	7	2	4	12	8	1	3

V. CONCLUSION

The conception and basic structure of the Five-elements Cycle Optimization (FECO) algorithm is proposed in this paper. This algorithm is inspired by the theory of generation and restriction among the five elements of wood, fire, earth, metal, and water. Experiments have corroborated the feasibility and suitability of FECO for solving continuous optimization functions.

There are a lot of possibilities to expand the basic FECO.

1) There are more kinds of relationships between five elements except generation and restriction on the basis of the Chinese Five-elements Theory. The Five-elements Cycle Model (FECM) is expected to extend to a series of models.

2) The framework of FECO and details of implementation could have manifold forms. For example, we can design multilayer circles in Element Space, change the grouping mode of elements, or improve the formula of elements updates.

3) FECO can be combined with various optimization algorithms to construct more optimization methods.

REFERENCES

[1] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, “Optimization by simulated annealing,” *Science*, vol. 220, no. 4598, pp. 671–680, May 1983.

[2] L. J. Fogel, A. J. Owens, and M. J. Walsh, Artificial Intelligence Through Simulated Evolution. New York: Wiley, 1966.

[3] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA: Addison-Wesley, 1989.

[4] I. Rechenberg, *Evolutions Strategie—Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution*. Berlin, Germany: Frommann-Holzboog, 1973.

[5] F. W. Glover, "Future paths for integer programming and links to artificial intelligence," *Computers and Operations Research*, vol. 13, no. 5, pp. 533–549, 1986.

[6] M. Dorigo, T. St¨utzle, *Ant Colony Optimization*. Cambridge, MA: MIT Press, 2004.

[7] J. Kennedy, R. Eberhart, *Swarm Intelligence*. San Francisco, CA: Morgan Kaufmann, 2001.

[8] R. Storn, K. Price, “Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, no. 11, pp. 341-359, 1997.

[9] V. Cutello, G. Nicosia, "An immunological approach to combinatorial optimization problems," *Lecture Notes in Computer Science*, Springer, vol. 2527, pp. 361–370, 2002.

[10] R. G. Reynolds, “An introduction to cultural algorithms,” in *Proceedings of the Third Annual Conference on Evolutionary Programming*, New Jersey, 1994, pp. 131-139.

[11] P. Moscato, “On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms,” California Institute of Technology,Tech. Rep. Caltech Concurrent Computation Program, Report 826, Pasadena, California, USA, 1989.

[12] L. Xu, "BYY learning system and theory for parameter estimation, data smoothing based regularization and model selection," *Neural, Parallel and Scientific Computations*, vol. 8, no. 1, pp. 55–82, 2000.

[13] W. Zhang, “YinYang bipolar lattices and L-sets for bipolar knowledge fusion, visualization, and decision,” *Int. J. Inf. Technol. Decision Making*, vol. 4, no. 4, pp. 621–645, Dec. 2005.

[14] S. C. Tam, H. K. Tam, L. M. Tam, T. Zhang, “A new optimization method, the algorithm of changes, for Bin Packing Problem,” in *Proceedings of the IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA 2010)*, University of Hunan, Changsha, China, September 23-26, 2010.

[15] R. Zhao, “Yin-Yang optimization for traveling salesman problem,” in *22nd European Conference on Operational Research EURO XXII*, Prague, Czech Republic, July 08-11, 2007.

[16] M. Liu, “Five-elements Cycle Optimization Algorithm for The Travelling Salesman Problem,” in *the 18th International Conference on Advanced Robotics (ICAR)*, Hong Kong, July 10-12, 2017.

[17] X. Yao, Y. Liu, and G. Lin, “Evolutionary programming made faster,” *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Aug. 1999.

[18] X. Yao and Y. Liu, “Fast evolution strategies,” in *Proc. 6th Int. Conf. Evol. Program*. VI, 1997, pp. 151–162.

[19] K. Deb, A. Anand, and D. Joshi, “A computationally efficient evolutionary algorithm for real-parameter optimization,” *Evol. Comput.*, vol. 10, no. 4, pp. 371–395, Dec. 2002.

[20] S. He, Q. H. Wu, and J. R. Saunders, “Group search optimizer: An optimization algorithm inspired by animal searching behavior,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 973–990, Aug. 2009.

[21] W. Gong, Z. Cai, C. X. Ling, and H. Li, “A real-coded biogeography-based optimization with mutation,” *Appl. Math. Comput*., vol. 216, no.9, pp. 2749–2758, Jul. 2010.

[22] Albert Y. S. Lam, Victor O. K. Li, and James J. Q. Yu, “Real-Coded Chemical Reaction Optimization,” *IEEE Trans. Evol. Comput.*, vol. 16, no. 3, pp. 339–353, Jun. 2012.