Design and performance analysis of a new optimization algorithm based on Finite Element Analysis

Zhiqiang Ning¹,², Youshan Gao¹ and Aihong Wang¹

¹College of Mechanical Engineering, Taiyuan University of Science and Technology, Taiyuan, Shanxi, PR China
²Department of Mechanical and Electronic engineering, Shanxi Institute of Technology, Yangquan, Shanxi, PR China

Abstract
Aiming at the problem that many algorithms could not effectively balance the global search ability and local search ability, a new optimization algorithm is proposed. Inspired by Finite Element Analysis (FEA) approach, a relationship of mapping between Finite Element Analysis approach and a population-based optimization algorithm is constructed through comparing the similarities and differences of FEA node and ideal particle. In algorithm framework, the stiffness coefficient corresponds to a user-defined function of the value of an objective function to be optimized, and the node forces among individuals are defined and an attraction-repulsion rule is established among them. The FEA approach that can simulate multi-states of matter is adopted to balance the global search ability and local search ability in the novel optimization algorithm. A theoretical analysis is made for algorithm parallelism. The conditions for convergence are deduced through analyzing the algorithm based on discrete-time linear system theory. In addition, the performance of the algorithm is compared with PSO for five states which include free state, diffusion state, solid state, entirely solid state, synthesis state. The simulation results of six benchmark functions show that the algorithm is effective. The algorithm supplies a new method to solve optimization problem.

Keywords
Finite element analysis, optimization, multi-states of matter, parallelism, convergence

Corresponding author:
Youshan Gao, College of Mechanical Engineering, Taiyuan University of Science and Technology, Waliu street number 66, Wanbailin District, Taiyuan, Shanxi, 030024, PR China.
Email: tkgaoyoushan@126.com

Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (https://creativecommons.org/licenses/by-nc/4.0/) which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (https://us.sagepub.com/en-us/nam/open-access-at-sage).
Introduction

Inspired by the laws of nature and swarm intelligence in biology society, heuristic algorithm applies different strategies to search the solution space. Genetic algorithms mimic Darwinian natural selection, where “fitness” selects individuals for inheritance, selection, crossover, and adaptive mutation. Particle Swarm Optimization (PSO) which is based on the swarming behavior of birds seeking food, is the earliest example of a Swarm Intelligence algorithm. Particle Swarm Optimization (PSO) algorithm has existed premature convergence for multimodal search problems, and PSO has undergone numerous improvements. Some nature-inspired heuristics analogize the laws of physics, as do CFO, EM and APO. CFO searches for extrema by “flying” through the decision space a group of “probes” whose trajectories are deterministically governed by the gravitational analogy. EM is inspired Coulomb’s Force Law for electrostatic charges. A framework of artificial physics optimization (APO) algorithm is constructed, in which a swarm of particles are treated as physical individuals seeking for a global optimum in the problem space driven by virtual force.

These heuristics have their applying ranges, advantages and disadvantages respectively. Some heuristics are easily trapped in the local optima of highly multimodal or “deceptive” functions. Therefore, a lot of problems about heuristic algorithm remain to study further. Finite Element Analysis is one of the important foundational theories of digital technology. FEA has the features of simulating multi-states of matter, which has reference value for innovation of intelligence optimization algorithm. The primary principle of FEA is that a complex system can be divided into many comparative small and simple elements which facilitate understanding complex systems, so that a thoughtful understanding is obtained by combining the elements. In many fields finite element method is used for simulating multi-states of matter, including solid mechanics, flow simulation, and gas flow field.

Aiming at the problem that many algorithms using a single motion rule could not effectively balance the global search ability and local search ability, a novel optimization algorithm based on Finite Element Analysis which is easy to simulate multi-states of matter and to implement parallel computing is proposed. Compared with the PSO and EM, the proposed algorithm is inspired by different laws of physics. The novel algorithm has advantages of effective motion which is affected by all the other particle, while PSO only flies to the historical best particle and the current best particle. The novel algorithm which has multi- states of matter is better at fine search than EM. Therefore, compared with the PSO and EM, the proposed algorithm has advantages of higher diversity and search efficiency. The paper constructs algorithm framework and verify the effectiveness of the algorithm through experiment. A theoretical analysis is made for algorithm parallelism at main steps, and the implementation of parallelism is a further study work.

The remainder of this paper is organized as follows: Section 2 introduces the FEA approach. Section 3 describes a relationship of mapping between FEA and a population-based optimization algorithm. Section 4 describes FEAO framework.
and analyzes theoretical parallelism of FEAO algorithm. Section 5 analyzes FEAO convergence. Section 6 compares the performance of FEAO five states and PSO against six recognized benchmark functions. Section 7 presents conclusions and suggestions for future work.

**Finite element analysis**

The typical finite element analysis goes through roughly four steps. With one-dimension bar structure taken as an object for study, an example is given to intuitively illustrate the process of FEA in Figure 1. The structure consists of n bar components of different geometric sizes, and the junctions of the structure is under concentrated forces. The relevant parameters of material characteristics and structure size: $E_i (i = 1, 2 \ldots n)$ —— the elastic modulus, $A_i (i = 1, 2 \ldots n)$ —— the cross sectional area of the bar, $E_i A_i / l_i$ —— stiffness coefficient of element, $u_i (i = 1, 2 \ldots n + 1)$ —— the displacement of node $i$, $P_i^{(i)}$ —— the forces on node $i$.

1. **Dividing elements and numbering node**

   At the junction of structure, $n + 1$ nodes and $n$ elements are divided. The element and node numbers are illustrated in Figure 1. Each element is separated, and the displacement and external force of each node is indicated.

2. **Element stiffness equation**

   Stiffness equation (1) of element $i$ ($i = 1, 2 \ldots n$) is described as follows:
(3) Assembling the stiffness matrix and establishing global stiffness equation (2).

\[
\begin{bmatrix}
\frac{E_i A_i}{l_i} & -\frac{E_i A_i}{l_i} \\
-\frac{E_i A_i}{l_i} & \frac{E_i A_i}{l_i}
\end{bmatrix}
\begin{bmatrix}
u_i \\
u_{i+1}
\end{bmatrix}
= \begin{bmatrix}
P_i^{(j)} \\
P_{i+1}^{(j)}
\end{bmatrix}
\]

(4) Handling boundary conditions and solving global stiffness equation

With the boundary condition \(u_{n+1} = 0\), (n + 1)th row and (n + 1)th column of global stiffness equation are both eliminated. Global stiffness equation is solved, and subsequently node displacement \([u_1, u_2, u_3, \ldots, 0]^T\) is obtained.

A relationship of mapping between FEA and a population-based optimization algorithm

To address the optimization problem, the solutions sampled from the feasible region of the problem are treated as particle which have properties, such as fitness, velocity, position. With intelligent search strategy, the optimal solution is found by the way of successive iteration velocity and position. The nodes which forces are exerted on will deform appropriately depending on stiffness equations which consist of the forces and stiffness matrix. While nodes of FEA corresponds to particle
of optimization algorithm, the paper constructs the relationship between node’s stiffness and the fitness of an objective function to be optimized. In addition, force rule of FEA corresponds to search strategy of optimization algorithm. As above, a relationship of mapping between Finite Element Analysis approach and a population-based optimization algorithm is constructed, as showed in Figure 2.

Matter can exist as a solid, liquid, or gas. Finite element method is used for simulating multi-states of matter. Adopting FEA simulation capacity, three node states which correspond to three different motion rules are designed for the novel algorithm. By setting the property of element stiffness matrix the nodes enter one of three states which include solid state, diffusion state, and free state. Three main states of FEAO algorithm draw analogies from three motion rules which effectively balance the global search ability and local search ability in optimization algorithm. (1) Solid state: solid state draws analogies from this method which can disperse the particles and improve the diversity of the population in order to avoid premature convergence. According to solid mechanics, atoms of solid states are not easily to move. Strategy of solid state turn the excessive-gathering particles among population into solid state which avoid excessive-gathering particles to get too close. The implement method of solid state is that particular negative values are assigned to off-diagonal elements of stiffness matrix. (2) Diffusion state: diffusion state is the exact opposite of solid state. In diffusion state the node motion is motivated. The implement method of diffusion state is that particular positive values are assigned to off-diagonal elements of stiffness matrix. (3) Free state: the node motion is not

Figure 2. The mapping of FEA to optimization algorithm.
related to each other. The implement method of free state is that zero values are assigned to off-diagonal elements of stiffness matrix. Three main states are implemented through off-diagonal elements of stiffness matrix as shown in Table 1.

| State       | Off-diagonal elements | Nodes motion |
|-------------|-----------------------|--------------|
| Solid state | -                     | Limit        |
| Diffusion state | +                     | Promote      |
| Free state  | 0                     | No effect    |

Handling method of solving one-dimensional problem extends to multi-dimensional problem. By the principle of FEA element stiffness matrix, the diagonal element $k_{ii}$ ($k_{jj}$) denotes the value of the force which is required to be exerted on node $i$($j$), while node $i$($j$) generates per unit displacement, with other nodes remaining 0 displacement. According to an optimization problem with n dimensions, stiffness equation (3) of element is described as follows: $i$, $j$——the node numbers; The off-diagonal element $k_{z,z+n}$ denotes value of the force which is required to be exerted on node $i$ while node $j$ generates per unit displacement in $z$th dimension, with other node remaining 0 displacement. Positive sign of $k_{z,z+n}$, negative sign of $k_{z,z+n}$ and zero of $k_{z,z+n}$ correspond to diffusion state, solid state, and free state respectively.

$$K_{ij} = \begin{bmatrix}
1 & 2 & \cdots & n & n+1 & n+2 & \cdots & 2n \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
k_{ii} & \cdots & 0 & k_{1,1+n} & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & k_{ii} & 0 & \cdots & k_{n,2n} \\
k_{1+n,1} & \cdots & 0 & k_{jj} & \cdots & 0 \\
\vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \ddots & \vdots \\
0 & \cdots & k_{2n,n} & 0 & \cdots & k_{jj}
\end{bmatrix}$$

An optimization based on FEA

The FEAO algorithm addresses the problem of locating the global minima of a (nonlinear) objective function $f(x)$ defined on a bounded hyperspace; that is, determine:

$$\min \{f(X) : X \in \Omega \subset \mathbb{R}^n, \ f : \Omega \subset \mathbb{R}^n \to \mathbb{R}, \}$$

where $\Omega = \{X \in \Omega | x_k^{\min} \leq x_k \leq x_k^{\max}, \ k = 1, \cdots, n\}$ is the bounded region of feasible (allowable) solutions. $n$: dimensionality of the problem. $x_k^{\max}$: upper bound in each dimension. $x_k^{\min}$: lower bound in each dimension. $f(X)$: pointer to the function being minimized. $N_{pop}$: population size.
The FEAO algorithm comprises four main procedures: (a) Initialization; (b) Global stiffness matrix Calculation; (c) Nodes force calculation; (d) Displacement and position solution. FEAO’s algorithm framework (pseudocode) appears in Table 2.

### Initializing population

In the Initialization step, \( N_{\text{pop}} \) individuals are randomly generated in range \([x_{\text{min}}, x_{\text{max}}]\). The fitness (value of the objective function) for each individual is calculated using the function pointer \( f(X) \). \( X_{\text{best}} \)— the global best position at step \( t \). \( X_i(t) = (x_{i,1}(t), x_{i,2}(t) \cdots x_{i,n}(t)) \)— position vector of node \( i \) at step \( t \). \( x_{i,k}(t) \)— position component in \( k \) dimension. \( U_i(t) = (u_{i,1}(t), u_{i,2}(t) \cdots u_{i,n}(t)) \)— displacement vector of node \( i \) at step \( t \). \( u_{i,k}(t) \)— displacement component of node \( i \) in \( k \)th dimension.

### Calculating stiffness matrix

**Calculating the diagonal elements in stiffness matrix.** While any two node \( i \) and node \( j \) constitute a bar element, the stiffness coefficient \( k_{ii} \) and \( k_{jj} \) as the diagonal elements are calculated from the objective function \( f(x) \). Clearly it is possible to define the stiffness function using different methods. Adopting mass function in the APO algorithm, \(^{15}\) the stiffness function \( k_{ii} = g(f(x_i)) \) is defined as equation

\[
G(f(x_i)) = e^{\left[\frac{f(x_{\text{best}}) - f(x_i)}{f(x_{\text{best}}) - f(x_{\text{worst}})}\right]}
\]

(4)

### Table 2. FEAO framework.

| Begin |
|-------|
| Initialize population: both displacement and position; |
| Set parameters: population size \( N_{\text{pop}} \); max_length—— the maximum number of iterations for run termination; |
| Iteration = 1; |

| While (termination criterion is not met) |
| Do |
| Evaluate all individuals using corresponding fitness function. |
| Calculate the diagonal elements in stiffness element matrix of element established by every two individuals. |
| Divide the node into three states, and calculate the off-diagonal elements in stiffness element matrix. |
| Assembly element stiffness matrix to establish global stiffness matrix \( K \). |
| Calculate the node force vector \( F \); |
| solve stiffness equation \( KS = F \) to get node displacement \( S \) and update position \( X \). |
| Evaluate all individuals using corresponding fitness function; |
| Iteration = t Iteration + 1; |
| End Do (until termination criterion is met) |

End
According to the equation \( k_{ii} = e^{f(x_{\text{best}}) - f(x_i)} \) \((i = 1, 2, \ldots n)\) and \( k_{jj} = e^{f(x_{\text{best}}) - f(x_j)} \) \((j = n + 1, 2n)\), \( k_{ii} \) and \( k_{jj} \) are assigned to the diagonal elements in stiffness matrix \( K_{ij} \) as follow by equation (5).

\[
K_{ij} = \begin{bmatrix}
1 & \cdots & n & n+1 & \cdots & 2n \\
\downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \\
0 & \cdots & 0 & k_{ii} & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & k_{ij} & 0 & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\
0 & \cdots & 0 & k_{jj} & \cdots & 0 \\
\end{bmatrix}
\]

(5)

**Calculating off-diagonal elements in stiffness matrix.** According to the equation \( c_{la_i} = \frac{f(x_{\text{max}}) - f(x_i)}{f(x_{\text{max}}) - f(x_{\text{min}})} \), the method for node state division maps the fitness values into interval \((0, 1)\) to be more reasonable in the division of states. When the condition \( c_{la_i} > \text{standard}_s \) is satisfied, nodes may enter solid state in the dimension. When the condition \( c_{la_i} \leq \text{standard}_g \) is satisfied, nodes may enter diffusion state in all dimensions. \( \text{standard}_s \) is dividing standard for solid state. And \( \text{standard}_g \) is dividing standard for diffusion state. \( \text{standard}_s \) and \( \text{standard}_g \) could be set according the function to be optimized.

A bar element which consist of node \( i \) and node \( j \) enters solid state when the conditions is simultaneously satisfied as follow: (1) \( |x_{j,k} - x_{i,k}| \leq L \), \( L \) — activated distance of solid state; (2) \( c_{la_i} > \text{standard}_s \) or \( c_{la_j} > \text{standard}_s \), \( \text{standard}_s \) — dividing standard of solid state; (3) According to descending order of \( c_{la,d}(d = 1, 2 \cdots N_{\text{pop}}) \), \( c_{la_i} \) or \( c_{la_j} \) is within the top \( N \) (\( 1 < N \leq N_{\text{pop}} \)). When the bar elements satisfy all above conditions, off-diagonal elements \( k_{z,z+n} = 1, 2, \ldots n \) are assigned utilizing the equation (6). Thus, Off-diagonal element \( k_{z,z+n} = 1, 2, \ldots n \) is set in element stiffness matrix \( K_{ij} \) by equation (7).

\[
k_{z,z+n} = -1 \times \frac{T_1 \times \text{max_length}}{t_{\text{max_length}}}, z = 1, 2, \ldots n
\]

(6)

\( T_1 \) — the coefficient of solid force; \( \text{max_length} \) — the maximum number of iterations for run termination; \( t \) — iterations.

\[
K_{ij} = \begin{bmatrix}
k_{ii} & \cdots & 0 & k_{i,1+n} & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & k_{ii} & 0 & \cdots & k_{k,2n} \\
k_{i,1+n} & \cdots & 0 & k_{jj} & \cdots & 0 \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & k_{2n,n} & 0 & \cdots & k_{jj}
\end{bmatrix}
\]

(7)
A bar element which consist of node $i$ and node $j$ enters diffusion state when the conditions is simultaneously satisfied as follow: (1) $cla_i \leq \text{standard}_g$ and $cla_j \leq \text{standard}_g$ — dividing standard of diffusion state (2) According to ascending order of $cla_i(d = 1, 2 \cdots N_{\text{pop}})$, $cla_i$ and $cla_j$ are within the top $N(1 < N \leq N_{\text{pop}})$. When the bar elements satisfy all above conditions, off-diagonal elements $k_{z,z+n}(z = 1, 2 \cdots n)$ are assigned utilizing the equation (8) as follow. Thus, Off-diagonal element $k_{z,z+n}i = 1, 2 \cdots n$ is set in element stiffness matrix $K_{ij}$ utilizing equation (7).

$$k_{z,z+n}(z = 1, 2 \cdots n) = (+1) \times \frac{T_2}{\text{max}_\text{length}}$$

Where $T_2$—the coefficient of diffusion force; $\text{max}_\text{length}$— the maximum number of iterations for run termination; $t$—iterations.

**Assembling element stiffness matrix to establish global stiffness matrix $K$.**

$$K = \begin{bmatrix}
(N_{\text{pop}} - 1)k_{ii} & \cdots & 0 & k_{1,m} & \cdots & k_{1,n} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
0 & \cdots & (N_{\text{pop}} - 1)k_{pp} & 0 & \cdots & 0 \\
k_{q,1} & \cdots & 0 & (N_{\text{pop}} - 1)k_{qq} & \cdots & k_{q,r} \\
\vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
k_{n,1} & \cdots & k_{2n,n} & 0 & k_{q,r} & (N_{\text{pop}} - 1)k_{jj}
\end{bmatrix}$$

$K_{pp} = (N_{\text{pop}} - 1)k_{pp}$ is a non-zero value, node $p$ is in free state as equation (9) shows. Except the diagonal element $k_{pp} = (N_{\text{pop}} - 1)k_{qq}$ in $qth$ row, there are non-zero value in the off-diagonal elements, therefore node $q$ is in solid or diffusion state.

**Calculating the node force vector**

Based on the results of the stiffness coefficient $k_{ii}$ and $k_{jj}$, the node forces are calculated by an attraction- repulsion rule as follow.

$$P_{ij,k} = \begin{cases} 
Gk_{ii}k_{jj}r_{ij,k} \iff f(X_j) < f(X_i) \\
Gk_{ii}k_{jj}r_{ij,k} \iff f(X_j) \geq f(X_i)
\end{cases} \forall i \neq j$$

$P_{ij,k}$ is the $kth$ component of force which is exerted on node $i$ by node $j$; $r_{ij,k} = x_j - x_i$,$x_i$ and $x_j$ are the $kth$-dimension coordinate. Equation (10) shows that node $j$ attracts node $i$, and $P_{ij,k}$ is an attractive force if $f(X_j) < f(X_i)$; or node $j$ repels node $i$, $P_{ij,k}$ is a repulsive force if $f(X_j) \geq f(X_i)$.

The $kth$ component of the total force $P_{i,k}$ which is exerted on node $i$ by all other nodes is obtained by summing over all other node forces as follow.
\[ P_{i,k} = \sum_{j=1, j \neq i}^{N_{pop}} P_{ij,k} \]  
\[ P_{i,k} = \alpha \sum_{j=1, j \neq i}^{N_{pop}} P_{ij,k} \]

In this equation (11) of \( P_{i,k} \), random variable \( \alpha \) normally distributed on (0,1) is introduced.

According to theory of FEA, the node force vector is calculated as follow.

\[ P^e = [P_{1,1}P_{1,2} \cdots P_{1,n} \cdots P_{N_{pop},1}P_{N_{pop},2} \cdots P_{N_{pop},n}]^T \]

**Solving node displacement**

Only the diagonal element of nodes in free state is a non-zero value. Thus, displacement of the free nodes could be obtained by the direct calculation, while displacement in solid or diffusion state are calculated through solving global stiffness equation.

Global stiffness equation (13) is as follow:

\[ KS(t) = P^e \]

\[ S(t) = (s_{1,1}(t), s_{1,2}(t) \cdots s_{1,n}(t) \cdots s_{N_{pop},1}(t), s_{N_{pop},2}(t) \cdots s_{N_{pop},n}(t)) \]

\( f(X_{best}) \) is the objective function value at the position of the "best" node. According to the finite element method, the constraint condition is that the displacement is zero at the position of the “best” node. The constraint condition is imposed by crossing out the rows and columns which involve \( X_{best} \) of global stiffness equation. \( 0 \leq \omega < 1 \) is a user-specified inertia weight that determines how easily the previous displacement can be changed, with larger values of \( \omega \) corresponding to greater displacement changes.

The final FEAO step is to update position, which refers to the displacement of nodes through the decision space using the previously computed total force to calculate a “displacement,” which then is used to update the nodes’ position. The displacement and coordinates of node \( i \) at the next iteration, are updated using equations (14) and (15), respectively:

\[ u_{i,k}(t+1) = \omega u_{i,k}(t) + s_{i,k}(t) \]  
\[ x_{i,k}(t+1) = x_{i,k}(t) + u_{i,k}(t + 1) \]
Parallel analysis

The FEAO algorithm adopts the parallel finite element method. Any element stiffness matrix is only dependent on themselves, and does not involve other element. All element stiffness matrix is simultaneously calculated, while the elements of matrix are implemented with parallelism.17 The global stiffness matrix is assembled by the element stiffness matrix in accordance with the principle of “matching the number of nodes”. In the phase of assembling matrix, the processes in which all elements that have no common nodes with each other are superimposed are handled concurrently. The global stiffness equation is solved through both direct and iterative method.18 There are mainly parallel direct methods: parallel Gauss-Jordan method, parallel Gauss method, parallel multifrontal method. Iterative method mainly refers to parallel preprocessing conjugate gradient method.19–20

Convergence analysis

The paper analyzes the convergence of FEAO algorithm theoretically, adopting the method by which Zeng Jianchao et al proved the algorithm convergence of PSO21 and APO22 based on discrete-time linear system theory. Nodes which are in solid state and diffusion state not always exist in iterative process, but there are always nodes in free state. Convergence of the algorithm is proved by convergence analysis of nodes in free state. Only the diagonal element of nodes in free state is a non-zero value. Therefore, the global stiffness equations are simplified to the root equations of a degree. The solutions of the stiffness equation (13) simplified are as follow.

\[
s_{i,k}(t) = \alpha \sum_{j=1}^{N_{\text{pop}}} \frac{P_{s,i}}{(N_{\text{pop}}-1)k_{ij}}
\]

When an individual \(i\) chosen arbitrarily can prove the convergence, the result could extent to all the others. Each dimension of equations (10)–(15) is irrelevant to the other dimension. \(X_{\text{best}}\) is assumed to be constant over some number of iterations. For reducing the convergence analysis, the paper drops the dimension index \(k\) in equations (14) and (15), the update equations become:

\[
\begin{align*}
    u_i(t+1) &= \omega u_i(t) + \alpha \sum_{j=1}^{N_{\text{pop}}} \frac{P_{s,i}}{(N_{\text{pop}}-1)k_{ij}} \\
    x_i(t+1) &= x_i(t) + u_i(t+1)
\end{align*}
\]

Define \(N_i = \{j|f(X_j) < f(X_i) \forall j \in S\}\) and \(M_i = \{j|f(X_j) \geq f(X_i) \forall j \in S\}\). Where, \(S\) is the set of all individuals. The displacement update equation (18) becomes:
\[ u_i(t + 1) = \omega u_i(t) + \alpha \sum_{j=1}^{N_{\text{pop}}} \frac{P_{i,k}}{(N_{\text{pop}} - 1)k_{ji}} \]
\[ = \omega u_i(t) - \alpha \sum_{j \in N_i} G_{k_{ji}}(X_i(t) - X_j(t))/(N_{\text{pop}} - 1) \]
\[ + \alpha \sum_{j \in M_i} G_{k_{ji}}(X_i(t) - X_j(t))/(N_{\text{pop}} - 1) \]
\[ = \omega u_i(t) + (\alpha \sum_{j \in M_i} G_{k_{ji}} - \alpha \sum_{j \in N_i} G_{k_{ji}})X_i(t)/(N_{\text{pop}} - 1) \]
\[ - (\alpha \sum_{j \in M_i} G_{k_{ji}} - \alpha \sum_{j \in N_i} G_{k_{ji}})X_j(t)/(N_{\text{pop}} - 1) \]

With the following definitions:

\[ G_{N_i} = \frac{\alpha \sum_{j \in N_i} G_{k_{ji}}}{N_{\text{pop}} - 1} \]  
(20)

\[ G_{M_i} = \frac{\alpha \sum_{j \in M_i} G_{k_{ji}}}{N_{\text{pop}} - 1} \]  
(21)

\[ G_{NM_i} = G_{N_i} - G_{M_i} \]  
(22)

\[ F_{G_i} = \alpha \left( \sum_{j \in N_i} G_{k_{ji}} - \sum_{j \in M_i} G_{k_{ji}} \right) X_i(t) \]  
(23)

Equation (19) becomes:

\[ u_i(t + 1) = \omega u_i(t) - G_{NM_i}X_i(t) + F_{G_i} \]  
(24)

Substituting equation (24) into equation (18) yields the following 2nd order difference equation.

\[ X_i(t + 1) - (1 + \omega - G_{NM_i})X_i(t) + \omega X(t - 1) = F_{G_i} \]  
(25)

Through property analysis of equation (25), the convergence of the RV sequence \( \{E[X_i(t)]\} \) may be proved, \( E \{X_i(t)\} \)——the expectation of random variable \( X_i(t) \). Applying the expected value operator to equation (25), the equation (25) become:

\[ EX_i(t + 1) - (1 + \omega - E(G_{NM_i}))EX_i(t) + \omega EX(t - 1) = E(F_{G_i}) \]  
(26)

where

\[ \varphi_i = E(G_{NM_i}) = \frac{1}{2} G \left( \sum_{j \in N_i} k_{ji} - \sum_{j \in M_i} k_{ji} \right) \]  
(27)

\[ \Theta_i = E(F_{G_i}) = \frac{1}{2} G \left( \sum_{j \in N_i} k_{ji} EX_j(t) - \sum_{j \in M_i} k_{ji} EX_j(t) \right) \frac{1}{N_{\text{pop}} - 1} \]  
(28)
The characteristic equation of equation (25) is:

\[ \lambda^2 - (1 + \omega - \varphi_i)\lambda + \omega = 0 \quad (29) \]

Theorem 1. If and only if \( 0 \leq \omega < 1 \) and \( 0 < \varphi_i < 2(1 + \omega) \) converges to \( X_{\text{best}} \).

Proof. The condition of \( \{EX_i(t)\} \) convergence is that the absolute values of both eigenvalues are less than 1.

\[ \left| 1 + \omega - \varphi_i \pm \sqrt{(1 + \omega - \varphi_i)^2 - 4\omega} \right| < 1 \quad (30) \]

There are three cases that depend on the value of the discriminant \( (1 + \omega - \varphi_i)^2 - 4\omega \) as follows.

1. \( (1 + \omega - \varphi_i)^2 = 4\omega \). The eigenvalues are real and equal.
2. \( (1 + \omega - \varphi_i)^2 < 4\omega \), both eigenvalues are real, but different.
3. \( (1 + \omega - \varphi_i)^2 > 4\omega \), the two eigenvalues are complex conjugates with equal magnitudes.

By summarizing three cases, the convergence condition for the sequence of random variables \( \{EX_i(t)\} \) is obtained as follows.

\[ 0 \leq \omega < 1 \quad \text{and} \quad 0 < \varphi_i < 2(1 + \omega) \quad (31) \]

When \( \{EX_i(t)\} \) is convergent, with the assumption that limit of convergence is \( X^a \), substituting \( \lim_{t \to \infty} EX_i(t) = X^a \) into equation (26) yields

\[ \varphi_i X^a = \theta_i \quad (32) \]

Substituting equations (27) and (28) into equation (32) yields

\[ \frac{1}{2} G \left( \sum_{j \in N_i} k_{ij} X^a - \sum_{j \in M_i} k_{ij} X^a - \sum_{j \in N_i} k_{ij} EX_i(t) + \sum_{j \in M_i} k_{ij} EX_i(t) \right) = \frac{1}{2} G \left( \sum_{j \in N_i} k_{ij} EX_i(t) - \sum_{j \in M_i} k_{ij} X_j(t) \right) \]

which may be written

\[ \Rightarrow \sum_{j \in N_i} k_{ij} X^a - \sum_{j \in M_i} k_{ij} X^a - \sum_{j \in N_i} k_{ij} EX_i(t) + \sum_{j \in M_i} k_{ij} EX_i(t) = 0 \]

\[ \Rightarrow \sum_{j \in N_i} k_{ij} (X^a - EX_i(t)) - \sum_{j \in M_i} k_{ij} (X^a - EX_i(t)) = 0 \]

\[ \Rightarrow k_{\text{best}} (X^a - EX_i(t)) + \sum_{j \notin N_i \text{best}} k_{ij} (X^a - EX_i(t)) = \sum_{j \notin M_i} k_{ij} (X^a - EX_i(t)) = 0 \quad (33) \]

By inspection, equation (33) is satisfied if and only \( EX_i(t) = X^a = X_{\text{best}} \). Substituting equation (27) into equation (31) yields
According to equation (34), the algorithm converges by setting an appropriately small enough value for $G$.

### Numerical experiments

In this section, FEAO’s performance is compared with PSO’s through the six recognized benchmark functions listed in Table 3. The Table 3 includes the search domain, the known minimum function value.

FEAO algorithm runs were made with the following empirically determined parameters: inertia weight $\omega = 0.9 - \frac{t}{\text{MAXITER}} \times 0.5$, where $t$ is iteration number and MAXITER is the maximum number of iterations for run termination, MAXITER = 6000; constant $G = 0.29$, problem dimensionality $n = 30$; population size $N_{\text{pop}} = 20$; activated distance of solid state $L = 1.0e-05$; Solid coefficient $T_1 = 5$, diffusion coefficient $T_2 = 3$; diffusion threshold standard $g = 1$; solid threshold standard $s = 0.7$; $C_1$ —— the terminal iteration of solid state; $C_2$ —— the start iteration of diffusion state; $N$ —— the maximum number of nodes in solid or diffusion state.

Each numerical experiment was repeated 50 times. The recorded performance data were the best function value (BEST), the average best function value (Mean), the max function value (MAX), and its standard deviation (STD). The PSO test results were selected from the references. The performance of FEAO algorithm is compared with PSO for five types which include free state, diffusion state, solid state, entirely solid state, synthesis state.

| Function     | $n$ | $N_{\text{pop}}$ | $[x_{\min}, x_{\max}]$ | Known optimum |
|--------------|----|------------------|-------------------------|---------------|
| Tablet       | 30 | 20               | [-100,100]              | 0.0           |
| Quadric      | 30 | 20               | [-100,100]              | 0.0           |
| Rosenbrock   | 30 | 20               | [-50,50]                | 0.0           |
| Griewank     | 30 | 20               | [-300,300]              | 0.0           |
| Rastrigin    | 30 | 20               | [-5.12,5.12]            | 0.0           |
| Schaffer’s f7| 30 | 20               | [-100,100]              | 0.0           |

\[ 0 \leq \omega < 1 \text{ and } 0 < G(\sum_{j \in N_i} k_{ij} - \sum_{j \in M_i} k_{ij}) < 4(1 + \omega)(N_{\text{pop}} - 1) \quad (34) \]
Table 4 shows the comparative results. The results of testing Tablet function illustrate that except entirely solid state the other diffusion state, solid state and synthesis state of the FEAO has higher performance than the free state, and
synthesis state obtains the best performance outperforming PSO. For Rosenbrock function, the numeral experiment indicates that the other four states have the advantage over free state, and synthesis state is the best exceeding PSO. For Quadric function, through simulations, compared with PSO and free state, the other four states enhance greatly the performance. For Griewank function, the performance of free state, diffusion state, entirely solid state, and synthesis state are

Figure 3. Dynamic performance for Tablet.

Figure 4. Dynamic performance for Rosenbrock.
similar to PSO, and solid state is the best exceeding the others. By the Rastrigin and Schaffer’s f7 test functions, the searching ability of the solid state and entirely solid state is more efficient, and the performance in all states to test Schaffer’s f7 are better than PSO.
The paper compared the dynamic performance of five states which are sampled at an interval of 20 iterations as shown in Figures 3–8. The results manifest that the diffusion state, solid state and synthesis state have advantages over free state. Because the force is proportional to the distance of nodes, at the end of the iteration the distance of two nodes is too close to generate large enough force for escaping from local minima. Therefore, the dynamic curve of the free state levels off as show. Each state has the capabilities and limitations. Solid state is well-suited for dealing

![Figure 7](image1.png)

**Figure 7.** Dynamic performance for Rastrigin.

![Figure 8](image2.png)

**Figure 8.** Dynamic performance for Schaffer’s f7.

The paper compared the dynamic performance of five states which are sampled at an interval of 20 iterations as shown in Figures 3–8. The results manifest that the diffusion state, solid state and synthesis state have advantages over free state. Because the force is proportional to the distance of nodes, at the end of the iteration the distance of two nodes is too close to generate large enough force for escaping from local minima. Therefore, the dynamic curve of the free state levels off as show. Each state has the capabilities and limitations. Solid state is well-suited for dealing
with the Griewank, Rastrigin, Schaffer’s f7 functions, while synthesis and diffusion states are fitter for solving the problem of Tablet, Rosenbrock, Quadric. Solid state can improve the diversity of the population, and diffusion state enhances node force to improve search performance.

Conclusions and future work

Based on Finite Element Analysis, a novel optimization is proposed. The FEA approach that can simulate multi-states of matter is adopted to balance the global search ability and local search ability in the novel optimization algorithms. The condition for FEAO convergence is deduced through analyzing FEAO algorithm based on discrete-time linear system theory. The simulation results of six typical optimization test functions show that FEAO algorithm is effective. The paper analyzes parallelism at main steps by theoretical analysis. Thus, a further study of FEAO algorithm is needed to focus on realization of parallel/distributed computation.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This project is supported by National Natural Science Foundation of China (Grant No. 51875381), National Natural Youth Science Foundation of China (Grant No. 51905369), and administration committee of Yangquan development zone (No.17-015).

ORCID iD

Youshan Gao https://orcid.org/0000-0003-2364-4768

References

1. Holland J. Adaptation in natural and artificial systems. Cambridge, MA: MIT press, 1992.
2. Eberhart R and Kennedy J. New optimizer using particle swarm theory. In: MHS’95 Proceedings of the sixth international symposium on micro machine and human science, Nagoya, Japan, 4–6 October 1995, pp. 39–43. Piscataway, NJ: IEEE.
3. Eberhart R and Shi Y. Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the 2001 congress on evolutionary computation, Seoul, South Korea, 27–30 May 2001, pp. 94–100. Piscataway, NJ: IEEE.
4. Shen X, Li Y, Wang W, et al. A dynamic adaptive particle swarm optimization for knapsack problem. In: Proceedings of Wcica 2006: sixth world congress on intelligent control and automation, Dalian, China, 21–23 June 2006, pp. 3183–3187. IEEE.
5. Formato R. Central force optimization: a new nature inspired computational framework for multidimensional search and optimization. In: Krasnogor N, Nicosia G, Pavone M, et al. (eds) Nature inspired cooperative strategies for optimization. Berlin: Springer, 2008, 221–238.
6. Formato R. Improved CFO algorithm for antenna optimization. Prog Electromagn Res 2010; 19: 405–425.
7. Formato R. Central force optimization: a new deterministic gradient-like optimization metaheuristic. OPSEARCH 2009; 46(1): 263–282.
8. Birbil S and Fang RS. An electromagnetism-like mechanism for global optimization. J Global Optim 2003; 25(3):
9. Xie LP, Zeng JC and Cui ZH. On mass effects to artificial physics optimization algorithm for global optimization problems. Int J Innov Comput Applic 2010; 2(2): 69–76.
10. Bathe KJ. Finite element procedures. Watertown, MA: Klaus-Jürgen Bathe, 2014.
11. Noor AK. Parallel processing in finite element structural analysis. Eng Comput 1988; 3: 225–241.
12. Kohei F, Takuma Y, Tsuyoshi L, et al. Acceleration of element by element kernel in unstructured implicit low-order finite-element earthquake simulation using open ACC on Pascal GPUs. In: Third workshop on accelerator programming using directives, Salt Lake City, UT, USA, 14 November 2016, pp. 1–12. IEEE.
13. Chung ET, Efendiev Y and Lee CS. Mixed generalized multiscale finite element methods and applications. Multiscale Model Sim 2015; 13(1): 338–366.
14. Zeng P. Fundamentals of finite element analysis. Beijing: Tsinghua University Press, 2008.
15. Xie LP and Zeng JC. The performance analysis of artificial physics optimization algorithm driven by different virtual forces. ICIC-EL 2010; 4(1): 239–244.
16. Maenhout B and Vanhoucke M. An electromagnetic meta-heuristic for the nurse scheduling problem. J Heuristics 2007; 13(4): 359–385.
17. Wu D, Yan X, Tang R, et al. Parallel realization of element by element analysis of eddy current field based on graphic processing unit. Appl Comput Electrom Soc J 2018; 33(2): 168–171.
18. Hsieh KT. Parallelization of MAP3D based on element-by-element Jacobi preconditioned conjugate gradient method. IEEE Trans Magn 2003; 39(1): 139–147.
19. Meijerink JA and Vander Vorst HA. An iterative solution method for linear systems of which the coefficient matrix is a symmetric M-matrix. Math Comput 1977; 137(31): 148–162.
20. Li TW and Wu T. Parallel computing of electromagnetic field based on domain decomposition method. In: Preprints of the 5th international conference on electric utility deregulation and restructuring and power technologies. Changsha, China, 26–29 November 2015, pp. 1583–1586. IEEE.
21. Cui ZH and Zeng JC. A new model of particle swarm optimization and its theoretical analysis. J Comput Res Develop 2006; 43(1): 96–100.
22. Xie LP, Zeng JC and Formato RA. Convergence analysis and performance of the extended artificial physics optimization algorithm. Appl Math Comput 2011; 218(8): 4000–4011.
23. He R, Wang Y, Qing W, et al. An improved particle swarm optimization based on self-adaptive escape velocity. J Softw 2005; 16(12): 2036-2044.
Author biographies

Zhiqiang Ning is a Doctoral Student of Taiyuan University of Science and Technology. He is a Lecturer of Shanxi Institute of Technology. His research direction is mechanical design and optimization.

Youshan Gao is a Professor of Taiyuan University of Science and Technology. He is a doctoral tutor. His research direction is energy recovery of construction machinery.

Aihong Wang is a Professor of Taiyuan University of Science and Technology. Her research direction is hoisting machinery.