A Greener Heterogeneous Scheduling Algorithm via Blending Pattern of Particle Swarm Computing Intelligence and Geometric Brownian Motion

Shaohui Li\(^1,2\), Hong Liu\(^1,2\), Bin Gong\(^3\) and Jinglian Wang\(^4\)

\(^1\)Department of Information Science and Engineering, Shandong Normal University, Jinan, 250014 China
\(^2\)Shandong key laboratory of distributed computer software new technology, Jinan, 250014 China
\(^3\)Department of Software, Shandong University, Jinan, 250101 China
\(^4\)Department of Information and Electrical Engineering, Ludong University, Yantai, 264025 China

Corresponding author: Shaohui Li (e-mail: lwjl22422@163.com).

This work was supported in part by National Natural Science Foundation of China under Grant 61702248, 61070017 and 61272094 and National High Technology Research and Development Program of China under Grant 2006AA01A113 and 2012AA01A306

ABSTRACT This paper focuses on the algorithms design of heterogeneous green scheduling for energy conservation and emission reduction in cloud computing. In essence, the real time, dynamic and complexity of heterogeneous scheduling require higher algorithm performance; however, the swarm intelligent algorithms although with some improvements, still exist big imbalances between local exploration and macro development or between route (solution) diversity and faster convergence. In this paper, a greener heterogeneous scheduling algorithm via blending pattern of particle swarm computing intelligence and geometric Brownian motion, is proposed, based on our earlier theoretical breakthroughs on G-Brownian motion and through a series of mathematical derivations or proofs; furthermore, in order for suitable for the hybrid processor architecture of the scheduling management server, the algorithm is designed in parallel with deep fusion of coarse-grained and master-slave models. A large number of experimental results are given. Compared with most newly published scheduling algorithms, there are significant advantages of the proposed algorithm on the dynamic optimization performance for consistent or semiconsistent and large inconsistent scheduling instances, although with lower improvement factors for small inconsistent instances.

INDEX TERMS Heterogeneous green scheduling, Swarm intelligence, Particle swarm optimization (PSO) algorithm, Standard Brownian motion, G-Brownian motion (geometric Brownian motion), Blending pattern

I. INTRODUCTION

At present, the epidemic is still raging around the world, and record breaking extreme weather is also frequent. As a matter of fact, energy conservation and emission reduction is a new pressing demand of cloud heterogeneous computing. Since 2016, the annual power consumption of data centers in China (about 120 billion KWH in 2016) has exceeded the annual power generation of the Three Gorges Hydropower Station (about 100 billion KWH in 2016); and there is a huge waste of energy in China's data centers, whose PUE (Power Usage Effectiveness) is generally greater than 2.2 while that of USA is also about 1.9 in the same period. Then, around the many theoretical or technical hot spots on green scheduling coordination, a large number of studies or discussions have been widely carried out\(^[1]\).

In essence, the candidate solutions of the scheduling algorithm correspond to the candidate schemes one to one, which means that the real time, dynamic and complexity of heterogeneous scheduling optimization problems require higher optimization performance, such as solution diversity or convergence speed\(^[2,3]\).

Concretely, the improvement of swarm intelligence algorithm represented by particle swarm optimization (PSO) algorithm has been systematically carried out from different dimensions, such as parameter selection or optimization\(^[2]\), and swarm topology restructuring\(^[3]\); among them, the
fused of different ideas is the most representative direction currently.

1) Inspired by thermodynamic molecular motion theory, some studies introduced concepts such as group centroid, acceleration and molecular force into PSO algorithm to transform the particle velocity and displacement [4-6].

2) Other studies refer to human adaptive learning and other mechanisms to realize various information sharing, so as to improve the convergence speed or show stronger ability [7-9].

3) Moreover, the representative achievements are the effective integration of PSO algorithm and Ito process driven by the standard Brownian motion [10].

Substantially, the PSO algorithms aforementioned in (1) or (2), are still approximately linear optimization-dynamic-patterns; the drive definition of the improved PSO algorithms in (3), is reduced to a special Markov stochastic process with constant expected drift rate and variance rate, also without the generality in the dynamic swarm intelligence simulation. All of them mean that the swarm intelligent algorithms although with some improvements, still exist big imbalances between local exploration and macro development or between route (solution) diversity and faster convergence for high dimensional multi-objective scheduling problems.

In this paper, based on our earlier theoretical breakthroughs on G-Brownian (Geometric Brownian) motion and through a series of mathematical derivations or proofs, a greener heterogeneous scheduling algorithm via blending pattern of particle swarm computing intelligence and geometric Brownian motion, i.e., PSO/RdBM, is proposed; furthermore, in order for suitable for the hybrid processor architecture of the scheduling management server, the algorithm is designed in parallel with deep fusion of coarse-grained and master-slave models.

II. RELATED WORK
The combing of related work mainly takes two threads: ① swarm intelligence and PSO algorithms, and ② the standard Brownian motion vs. G-Brownian motion.

A. SWARM INTELLIGENCE AND PSO ALGORITHMS
The PSO algorithm is one of the most popular swarm intelligence algorithms of the computational intelligence theory in recent years. It was first proposed by Kennedy and Eberhart in 1995, and basically adopts the concepts of "group" and "evolution" to search for the optimal solution in complex space through cooperation and competition among particles.

At the same time, as the extension of traditional artificial intelligence, PSO algorithms have been widely applied because of its simple principle, profound background of traditional evolutionary computing and unique high-dimensional objective optimization performance [11-17].

With the deepening of application and practice, some PSO researches focus on preserving the diversity of individuals in swarm intelligence algorithms. Inspired by thermodynamic molecular motion theory, the researches [4-6] introduced concepts such as group centroid, acceleration and molecular force into PSO algorithm to transform formulas such as particle velocity and displacement. In other words, according to the distance between the particle and the center of mass, the switching between the inductive force and the repulsive force can be realized to control the flight direction of the particle, and the diversity of the population can be maintained to a certain extent.

Other studies refer to human adaptive learning and other mechanisms to realize various information sharing in swarm intelligence algorithms, and then to improve the convergence speed or show a stronger ability of later evolution compared with the original swarm intelligence algorithms [7-9].

In recent years, the representative achievement of swarm intelligence algorithm improvement is the effective fusion of standard Brownian motion or Ito Process and PSO algorithms [10]. Some experiments show that the interdisciplinarity can improve the convergence speed or maintain the swarm diversity effectively, but at the same time, it also shows the shortcomings of the algorithm, such as the lack of stability.

B. STANDARD BROWNIAN MOTION VS. G-BROWNIAN MOTION
Standard Brownian motion was first proposed by British biologist R. Brown according to the random movement of pollen on the liquid surface (1827). Later, Wiener further studied the standard Brownian motion trajectory, and theoretically gave its spatial measure definition and other accurate descriptions (1918). Then, Kiyoshi Ito established the stochastic differential equation with the interference term of the standard Brownian motion, which was widely used in the fields of economy, management and social science; for this process, local stochastic disturbance and macroscopic drift are two obvious characteristics [18-22].

On the basis of preserving the core idea of PSO algorithm, this study intends to derive the energetic particle swarm co-evolution drive equation with nonlinear expectation space and G-Brownian motion characteristics.

The team of Academician Peng, Shige from Shandong University, the cooperative unit of this paper, has made world-renowned basic theoretical researches on G-Brownian motion with their unremitting efforts and systematic theoretical advancement over the past 30 years, which are powerful and instrumental in the field of nonlinear stochastic analysis.

They include the uniqueness of solutions of backward stochastic differential equations (BSDEs) (1990), the nonlinear Feynman-Kac formula for the solution correlation between BSDEs and second-order quasilinear PDEs (1991), nonlinear expectation theory with time consistency (2006), G-Brownian motion definition (2007) [24], and G-Brownian motion numerical simulation algorithm (2019) [25].
At the same time, these preliminary works are also the valuable basis of strict theoretical derivation or proof of this topic.

Generally, in the original PSO proposed by Eberhart and Kennedy, the updated position \( x_i^{t+1} \) and velocity \( v_i^{t+1} \) of any particle, as the optimization-dynamic-equations, are defined as Equation (1) and Equation (2).

\[
\begin{align*}
    v_i^{t+1} &= \omega v_i^{t} + c_1 r_1 (pbest_i - x_i^{t}) + c_2 r_2 (gbest - x_i^{t}) \\
    x_i^{t+1} &= x_i^{t} + v_i^{t+1}
\end{align*}
\]

In Equation (1), the first part of the formula is called the memory item, where \( \omega \) represents the inertial motion, that is, the influence of the past position on the present; the second part is called self-cognition, where \( c_1 r_1 \) indicates that the direction of motion of particles comes from their experience; and the third part is called group cognition, where \( c_2 r_2 \) reflects the cooperation and information sharing between particles (see Fig. 1).

In this paper, based on the core idea of original PSO algorithm (such as memory, self-cognition and social cognition) defined as Equation (1) and Equation (2), the particle swarm evolution equation is expanded to the geometric Brownian motion model with nonlinear G-expectation, as is more generalized than the standard Brownian motion with the invariable expected drift rate or the variance rate.

Then, shown as Fig. 2, a series of strict theoretical derivation or proof is key, including ① the spatial representation of the continuous path of the particle swarm, denoted as \( x_i^{t+1} \in \Omega_T \), ② the relevant bounded Lipschitz function definition based on the space, denoted as \( \xi(X) \in \text{Lip}(\Omega_T) \), ③ the G-heat equation optimal description, denoted as \( \Gamma_H \), and ④ the nonlinear G-expectation definition, denoted as \( E_{\Gamma_H}[\xi(X)] \mid \Gamma_H \).
In other words, once \( \mathbf{X}_t^{i+1} \in \Omega_T \), \( \mathbf{x}_i(t) \in \text{Lip}(\Omega_T) \), \( \Gamma_n \), and \( \mathbf{E}[\mathbf{z}(X)] | \Gamma_n \) are obtained, then the particles are accompanied by G-Brownian motion with nonlinear G-expectation for the more intelligent optimization swarm.

The related variables and their representative meanings are shown in Table I.

### Table I
THE RELATED VARIABLES AND THEIR REPRESENTATIVE MEANINGS

| Symbol | Description |
|--------|-------------|
| \( v_i(t) \) | The instantaneous velocity of Particle \( i \) at the given moment \( t+1 \) |
| \( x_i(t+1) \) | The updated displacement of Particle \( i \) at the given moment \( t+1 \) |
| \( p_{best} \) | The best position ever for Particle \( i \) until the given moment \( t+1 \) |
| \( g_{best} \) | Historical optimum location of the particle swarm |
| \( C_{b, lip}(R^{k\times d}) \) | The space of bounded and Lipschitz functions on \( R^{k\times d} \) |
| \( \Omega_T \) | The space of all \( R^d \)-valued continuous paths \( \{x_i[t] \in \mathbb{N}, t \in [0,T] \} \) starting from origin, equipped with the supremum norm and equal to \( C([0,T]; R^d) \) |
| \( \text{Lip}(\Omega_T) \) | The set of the bounded and Lipschitz functions on the space \( \Omega_T \) |
| \( \mathbf{z}(X) \) | The bounded and Lipschitz function on the space \( \Omega_T \) |
| \( \phi \) | The canonical mapping of the bounded and Lipschitz function \( (\mathbf{z}(X)) \) on the space \( \Omega_T \) |
| \( B_n \) | The Borel σ-algebra on the space \( \Omega_T \) where \( t_t \in [0,T] \) |
| \( H \) | The Hurst exponent of the time series |
| \( S(d) \) | The space of all \( d \times d \) symmetric matrices |
| \( G: S(d) \rightarrow R \) | Each given monotonic and nonlinear function, referred to as the G-heat function |
| \( \Gamma_n \) | The Gamma function defined in this paper as the G-heat function for a more generalized co-evolution of the particle swarms |
| \( \mu(t,x_i) \) | The viscosity solution of the G-heat function |
| \( x_{0}^{i}, \ldots, x_{1}^{i} \) | A dynamic nonlinear G-expectation space of the bounded and Lipschitz function(\( \mathbf{z}(X) \)) |
| \( \mathbf{E}[\mathbf{z}(X)] \) | The equation definition of geometric Brownian motion with nonlinear G-expectation for a more generalized co-evolution of the particle swarms where \( x_{0}^{i} \in \Omega_T \) and \( \mathbf{z}(X) \in \text{Lip}(\Omega_T) \) |

1) SPATIAL REPRESENTATION OF DISPLACEMENT UPDATE OF PARTICLE SWARM

By Equation (1), the PSO algorithm can be regarded as an Ito process driven by Brownian motion and drifting toward "two" attractors, which are the best position ever for Particle \( i \) until the given moment \( t+1 \) \( p_{best} \) and the historical optimum location of the particle swarm until the given moment \( t+1 \) \( g_{best} \) where \( c_{1} \) and \( c_{2} \) are the drift coefficients.

Then the spatial representation of the continuous path (displacement update) of Particle \( i \), which is denoted by the following Equation (3) and Equation (4), can be obtained.
\[ \Gamma_H = \frac{1}{2} \left( x_i^{2H} + x_j^{2H} - |x_i - x_j|^{2H} \right)_{i,j=1,...,N}. \] (8)

Then, we can get the dynamic nonlinear G-expectation space of the bounded and Lipschitz function( \( \xi(X) \)) as Equation (9).

\[
\hat{\mathbb{E}}[\xi(X)]|_{\Gamma_H} = \frac{1}{N} \left( \frac{Y}{\Gamma_H} \right)^{\hat{\nu}_{\Xi}} - \left( \frac{t}{\Gamma_H} \right)^2
\] (9)

where \( Y = (X_{1H}, X_{2H}, \ldots, X_{NH})^t \).

4) **THE ENERGIZED OPTIMIZATION DYNAMICS DRIVEN BY THE G-BROWNIAN MOTION OF PARTICLE SWARM**

To summarize briefly, the optimization-dynamic-equation of particle swarm driven by the geometric Brownian stochastic motion with nonlinear G-expectation, denoted as \( (X_{i+1}^t, \xi(X) \in \text{Lip}(\Omega_T), \Gamma_H \mathbb{E}[\xi(X)]|_{\Gamma_H} ) \), is well defined in Equation (4), Equation (7), Equation (8) and Equation (9), respectively, in order for a more generalized co-evolution of the particle swarms.

**B. ALGORITHM DESCRIPTION**

In this paper, in order for suitable for the hybrid processor architecture of the scheduling management server, the algorithm is designed in parallel with deep fusion of coarse-grained and master-slave models (see Fig. 3).

Specifically, several particles subgroups can be assigned to different nodes according to coarse-grained model.

A large number of particle velocity or displacement updates on each node adopt CPU-GPU collaborative master-slave parallel design. Here, the CPU is regarded as the primary server, and several threads on the GPU are the clients.

Here, C-CUDA language, MICH-VMI communication protocol and ParadisEO software component can be used.

Following that, the algorithm can be described as follows.

The **PSO/RdBM** algorithm

**Step 1:** Initialize the iteration (\( \tau \)) and the subgroups, each subgroups of \( k \) particles;

**Step 2:** Randomly initialize the velocity and the position of the particle \( i \);

**Step 3:** While \( (\tau < \tau_{\text{max}}) \) and (other termination criteria are not satisfied)

**Step 4:** Do in parallel for each island */ Obtain coarse-grained model, one of parallel and distributed models */

**Step 5:** \( \tau = \tau + 1; \)

**Step 6:** For each particle in the subgroups */ Obtain master-slave model, another parallel model */

**Step 7:** Velocity update via Equation (1) and make it act on the current Particle \( i \);

**Step 8:** Apply the Equation (3) and Equation (4) to denote the spatial representation of the continuous path (displacement update) of the Particle \( i \);

**Step 9:** Apply the Equation (5), Equation (6) and Equation (7) to define the related bounded Lipschitz functions based on the space;

Apply the gamma function Equation (8)
Step 10: where \( t \in [t_{m-t},t) \) and \( i = 1, \ldots, k \) to well define the nonlinear function \( G \), which is often referred to as G-heahe equation;

Step 11: Apply the Equation (9) to get the dynamic nonlinear G-expectation space of the bounded and Lipschitz function \((\xi(X))\);

Step 12: Generate the new subgroups;

Step 13: Sort the particles in the subgroups in a decreasing order of fitness values, and save the fittest particle in the external memory;

Step 14: Perform local search strategies;

Step 15: End For

Step 16: If \( t < T_{\text{migration interval}} \) then

Step 17: Create \( \Psi_{\text{new}} \) for the current subgroups;

Step 18: Send \( \Psi_{\text{new}} \) to the neighboring subgroups;

Step 19: Receive \( \Psi_{\text{new}} \) from the neighboring subgroups;

Step 20: Construct the founding subgroups \( \Xi \);

Step 21: Select \( k \) particles into \( \Xi \);

Step 22: Replace the subgroups \( \Psi_{\text{new}} \) with \( \Psi_{\text{new}} \);

Step 23: End If

Step 24: End Do in parallel

Step 25: End While

Step 26: Output the best particle.

**IV. EXPERIMENT RESULTS AND DISCUSSION**

In this section, all the experiments have been carried out at National Supercomputing Center in Jinan, China.

**A. SIMULATOR AND SIMULATION PARAMETERS**

To ensure that the comparison between the algorithms is fair, there are not any special requirements for parameter setting between different methods; in other words, the general parameter values of the PSO algorithm are set normally.

Famous unimodal and multimodal test problems are shown in **Table II**. They are roughly divided into unimodal functions with changing variables, and multimodal functions with fixed or changing variables; and they are used to verify the static optimization performance of the different methods, where unimodal functions can assist in the global convergence validation and multimodal functions can test the ability of the local search or averting premature convergence.

| Function                                      | Dim | Range  | \( f_{\text{min}} \) | Type       |
|-----------------------------------------------|-----|--------|----------------------|------------|
| \( F_1(x) = \sum_{i=1}^{n} x_i^2 \)          | 30  | [-100,100] | 0                   | Unimodal   |
| \( F_2(x) = \sum_{i=1}^{n} |x_i| + \prod_{i=1}^{n} |x_i| \) | 30  | [-10,10]   | 0         | Unimodal   |
| \( F_3(x) = \sum_{i=1}^{n} \left( \sum_{j=1}^{n} x_j \right)^2 \) | 30  | [-100,100] | 0                   | Unimodal   |
| \( F_4(x) = \max \{ |x_i|, 1 \leq i \leq n \} \) | 30  | [-100,100] | 0                   | Unimodal   |
| \( F_5(x) = \sum_{i=1}^{n} [100(x_{i-1}^2 + (x_i - 1)^2)] \) | 30  | [-30,30] | 0                   | Unimodal   |
| \( F_6(x) = \sum_{i=1}^{n} [x_i + 0.5 \text{rand}] \) | 30  | [-100,100] | 0                   | Unimodal   |
| \( F_7(x) = \sum_{i=1}^{n} \text{sin}(|x_i|) \) | 30  | [-500,500] | -12567 | Multimodal |
| \( F_8(x) = \sum_{i=1}^{n} x_i^2 - 10\cos(2\pi x_i) + 10 \) | 30  | [-5,12.5,12] | 0                   | Multimodal |
| \( F_9(x) = -20\exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}) - \exp\left(\frac{1}{n} \sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e \) | 30  | [-32.32] | 0                   | Multimodal |
| \( F_{10}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{n}}\right) + 1 \) | 30  | [-600,600] | 0                   | Multimodal |
| \( F_{11}(x) = \prod_{i=1}^{n} |x_i| \cos\left(\frac{x_i}{\sqrt{n}}\right) + \prod_{i=1}^{n} |x_i| \sin\left(\frac{x_i}{\sqrt{n}}\right) \) | 30  | [-50,50] | 0                   | Multimodal |

**TABLE II**

FAMOUS UNIMODAL AND MULTIMODAL TEST PROBLEMS

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
\[ u(x,a,k,m) = \begin{cases} 
(k(x-a))^m & x > a \\
0 & -a \leq x \leq a \\
(k(x+a))^m & x < -a 
\end{cases} \]

0.1 \{ \sin(2(3\pi x)) + \sum_{i=1}^{n}(x_i-1)^2[1+\sin(2(3\pi x_i+1)) \] 

\[ F_{i1}(x) = \sum_{i=1}^{n}(x_i-1)^2[1+\sin(2(3\pi x_i))]+\sum_{i=1}^{n}u(x_i,5,100,4) \]

\[ F_{i1}(x) = \left[ \frac{1}{500} \sum_{j=1}^{5} \left[ \frac{1}{\sum_{i=1}^{n}(x_i-a)^2} \right] \right]^{-1} \]

\[ F_{i2}(x) = \sum_{i=1}^{n} \left[ a_i - \frac{b_i(b_i + b_i x_i)}{b_i^2 + b_i x_i + b_i x_i} \right]^2 \]

\[ F_{i3}(x) = 4x_1^2 + 2.1x_1 + \frac{1}{3}x_1^3 + x_1x_2 - 4x_2^2 + 4x_2^3 \]

\[ F_{i3}(x) = (x_2 - \frac{5.1}{4\pi^2} x_2 + \frac{5}{x_2} - 6) + 10(1 - \frac{1}{8\pi}) \cos x_1 + 10 \]

\[ F_{i4}(x) = \frac{1}{2} \sum_{i=1}^{10} \left[ 10 + (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) x_i \right] \]

\[ F_{i5}(x) = \sum_{j=1}^{n} \exp \left[ \sum_{i=1}^{n} a_i (x_i - p_j)^2 \right] \]

\[ F_{i6}(x) = \sum_{i=1}^{n} \left[ (x-a_i)(x-a_i)^T + c_i \right]^{-1} \]

| Function | Item | BLPSO | GEPSO | SPSO | OIPS | PSO/RdIm | COM | Function | Item | BLPSO | GEPSO | SPSO | OIPS | PSO/RdIm |
|----------|------|-------|-------|------|------|--------|-----|----------|------|-------|-------|------|------|--------|
| F1       | Best | 4.22E-01 | 1.61E-07 | 1.94E-06 | 1.06E-19 | 0.00E+00 |
| F1       | Median | 9.21E-01 | 2.68E-05 | 5.19E-06 | 1.86E-15 | 0.00E+00 |
| F1       | Mean | 1.01E-00 | 4.61E-04 | 5.09E-06 | 4.81E-08 | 0.00E+00 |
| F1       | Worst | 1.94E-00 | 3.08E-03 | 9.34E-06 | 2.41E-06 | 0.00E+00 |
| F1       | STD | 3.58E-01 | 8.82E-04 | 1.82E-06 | 3.40E-07 | 0.00E+00 |
| F2       | Best | 1.72E-01 | 2.98E-03 | 1.47E-05 | 9.08E-12 | 0.00E+00 |
| F2       | Median | 3.70E-01 | 6.70E-03 | 1.24E-06 | 3.09E-11 | 0.00E+00 |
| F2       | Mean | 3.73E-01 | 6.83E-03 | 6.42E-06 | 2.97E-11 | 0.00E+00 |
| F2       | Worst | 6.54E-01 | 1.26E-04 | 7.36E-05 | 7.37E-11 | 0.00E+00 |
| F2       | STD | 1.01E-01 | 1.94E-04 | 1.64E-05 | 1.52E-11 | 0.00E+00 |
| F3       | Best | 9.07E-02 | 7.18E-02 | 2.22E-07 | 5.03E-03 | 0.00E+00 |
| F3       | Median | 3.71E-01 | 2.54E-03 | 1.07E-03 | 1.15E-02 | 0.00E+00 |
| F3       | Mean | 4.27E-01 | 2.72E-03 | 3.64E-02 | 1.20E-02 | 0.00E+00 |
| F3       | Worst | 1.27E-00 | 7.43E-03 | 9.70E-01 | 2.63E-02 | 0.00E+00 |
| F3       | STD | 2.59E-01 | 1.18E-03 | 1.56E-01 | 4.30E-03 | 0.00E+00 |
| F4       | Best | 0.13E-117 | 0.49E-219 | 1.36E-179 | 1.68E-224 | 6.68E-228 |
| F4       | Median | 1.33E+10 | 1.79E+208 | 1.77E+168 | 1.28E+169 | 6.68E+228 |
| F4       | Mean | 2.24E+126 | 4.49E+288 | 1.39E+159 | 3.09E+99 | 6.68E+228 |
| F4       | Worst | 7.13E+227 | 6.49E+298 | 1.33E+139 | 0.77E+122 | 6.68E+228 |
| F4       | STD | 1.03E+09 | 1.69E+36 | 3.36E+09 | 6.68E+09 | 0.00E+00 |
| F5       | Best | 1.78E-09 | 2.29E-28 | 0.28E-17 | 0.36E-09 | 0.00E+00 |
| F5       | Median | 2.24E-17 | 1.38E-09 | 2.19E-13 | 1.36E-17 | 0.00E+00 |
| F5       | Mean | 1.22E-17 | 1.49E-28 | 1.39E-17 | 1.62E-19 | 0.00E+00 |
| F5       | Worst | 1.22E-09 | 1.28E-17 | 0.36E-11 | 2.19E-09 | 0.00E+00 |
| F5       | STD | 1.37E-09 | 0 | 2.19E-03 | 2.36E-17 | 0.00E+00 |
| F6       | Best | 1.62E-19 | 1.57E-02 | 6.28E-19 | 3.16E-09 | 0.00E+00 |
| F6       | Median | 2.28E-13 | 1.39E-09 | 2.66E-13 | 1.33E-07 | 0.00E+00 |
| F6       | Mean | 1.73E-07 | 1.49E-17 | 1.22E-09 | 1.78E-09 | 0.00E+00 |
| F6       | Worst | 0.22E-19 | 2.28E-19 | 0.36E-09 | 2.19E-09 | 0.00E+00 |
| F6       | STD | 1.37E-09 | 0 | 0 | 3.36E-17 | 0.00E+00 |

**Table III:** Statistical values of the methods for the 30-variable unimodal/multimodal benchmarks functions (F1 - F6)
| Function | Item | BLPSO[4] | GEPSO[3] | SPSP[3] | OIPSP[10] | PSO/Rd[3][4] |
|----------|------|---------|---------|---------|-----------|-------------|
| $F_0$    | Best | 6.66E-09 | 4.49E-22 | 3.26E-09 | 1.36E-11  | 0.00E+00    |
|          | Median| 2.79E-03 | 1.77E-19 | 2.66E-11 | 1.09E-09  | 0.00E+00    |
|          | Mean  | 1.13E-07 | 1.49E-28 | 1.36E-09 | 1.66E-09  | 0.00E+00    |
|          | Worst | 5.02E-13 | 4.28E-13 | 1.36E-10 | 2.22E-13  | 0.00E+00    |
|          | STD   | 2.56E-11 | 1.79E-09 | 0        | 1.36E-09  | 0           |
| $F_6$    | Best | 1.57E-16 | 1.33E-09 | 6.49E-13 | 3.06E-08  | 0.00E+00    |
|          | Median| 2.28E-11 | 1.39E-07 | 2.66E-12 | 1.33E-11  | 0.00E+00    |
|          | Mean  | 1.72E-07 | 1.46E-17 | 1.22E-09 | 1.78E-09  | 0.00E+00    |
|          | Worst | 0.26E-19 | 2.13E-19 | 4.94E-09 | 3.17E-10  | 0.00E+00    |
|          | STD   | 0.28E-06 | 1.07E-09 | 0        | 3.66E-17  | 0           |
| $F_{10}$ | Best | 1.66E-09 | 1.49E-22 | 2.28E-19 | 3.66E-09  | 0.00E+00    |
|          | Median| 2.96E-13 | 1.37E-09 | 2.66E-13 | 1.36E-09  | 0.00E+00    |
|          | Mean  | 1.13E-07 | 1.49E-28 | 1.36E-09 | 1.68E-09  | 0.00E+00    |
|          | Worst | 5.23E-19 | 2.28E-19 | 3.66E-09 | 2.19E-19  | 0.00E+00    |
|          | STD   | 3.37E-09 | 0        | 0        | 1.36E-17  | 0           |
| $F_{16}$ | Best | 1.162288 | 1.032288 | 0.999288 | 1.00322   | 0.999098    |
|          | Median| 1.232288 | 1.173288 | 1.153288 | 1.153288  | 0.999098    |
|          | Mean  | 1.149288 | 1.153288 | 1.153288 | 1.114288  | 0.999098    |
|          | Worst | 1.363288 | 1.263288 | 1.093288 | 1.029197  | 0.999098    |
|          | STD   | 2.96E-13 | 1.37E-09 | 2.66E-13 | 1.36E-09  | 0.000307    |
| $F_{20}$ | Best | -1.031628 | -1.031627 | -1.031627 | -1.031627 | -1.031628   |
|          | Median| -1.031627 | -1.031627 | -1.031627 | -1.031627 | -1.031628   |
|          | Mean  | -1.031627 | -1.031627 | -1.031627 | -1.031627 | -1.031628   |
|          | Worst | -1.031628 | -1.031628 | -1.031628 | -1.031628 | -1.031628   |
|          | STD   | 1.36E-09 | 2.19E-19 | 0.22E-09 | 1.28E-07  | 2.22E-16    |
| $F_{26}$ | Best | 0.397849 | 0.397817 | 0.397817 | 0.398017  | 0.397875    |
|          | Median| 0.398028 | 0.397999 | 0.397977 | 0.398019  | 0.397875    |
|          | Mean  | 0.398028 | 0.397999 | 0.397977 | 0.398019  | 0.397875    |
|          | Worst | 0.398028 | 0.397999 | 0.397977 | 0.398019  | 0.397875    |
|          | STD   | 1.13E-17 | 1.49E-28 | 1.36E-09 | 1.68E-09  | 0           |
| $F_{30}$ | Best | 3.149228 | 3.153228 | 3.153228 | 3.144228  | 3.002817    |
|          | Median| 3.232228 | 3.173228 | 3.153218 | 3.153228  | 3.002817    |
|          | Mean  | 3.162828 | 3.032288 | 3.002288 | 3.002288  | 3.002817    |
|          | Worst | 3.632228 | 3.632228 | 3.693228 | 3.693228  | 3.096173    |
|          | STD   | 5.22E-19 | 2.28E-19 | 3.69E-09 | 2.19E-19  | 1.68E-17    |
| $F_{36}$ | Best | -3.322995 | -3.322995 | -3.322995 | -3.322995 | -3.322417   |
|          | Median| -3.281995 | -3.322995 | -3.326195 | -3.322995 | -3.322417   |
|          | Mean  | -3.261777 | -3.291995 | -3.37206  | -3.322995 | -3.322417   |
|          | Worst | -3.300796 | -3.33697 | -3.321995 | -3.321978 | -3.322417   |
|          | STD   | 5.17E-19 | 2.22E-19 | 2.19E-17 | 2.36E-06  | 0.060607    |
| $F_{42}$ | Best | -10.536299 | -10.536297 | -10.536302 | -10.536287 | -10.536337  |
|          | Median| -10.536301 | -10.535977 | -10.536287 | -10.535966 | -10.536337  |
|          | Mean  | -10.536309 | -10.536299 | -10.536289 | -10.536299 | -10.536337  |
|          | Worst | -10.536607 | -10.536677 | -10.496228 | -10.549517 | -10.536337  |
|          | STD   | 6.68E-8  | 5.49E-07 | 7.07E-06 | 1.36E-06  | 7.79E-22    |

**TABLE IV**

**STATISTICAL VALUES OF THE METHODS FOR THE 30-VALEassic UNIMODAL/MULTIMODAL BENCHMARKS FUNCTIONS ($F_1$-$F_{36}$)**

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.312628, IEEE Access

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
B. EXPERIMENTAL RESULTS AND ANALYSIS

In this subsection, the multi-objective optimization performance comparison between the different methods including the static and dynamic optimization performance results analysis, is given.

1) STATIC OPTIMIZATION PERFORMANCE RESULTS ANALYSIS
The unimodal or multimodal statistical optimization results of BLPSO[4], GEPSO[8], SPSO[9], OJPSO[10], and PSO/RdBM^G, are shown as Table III and Table IV.

From Table III and Table IV, it can be found that the quality of the solutions obtained by the PSO/RdBM^G approach is higher than BLPSO[4], GEPSO[8], SPSO[9], and OJPSO[10].

Shown as Fig. 4, the 3-D shapes of some selected functions with obvious differences in the response surface, the convergence curves of five methods (BLPSO[4], GEPSO[8], SPSO[9], OJPSO[10], and PSO/RdBM^G), and 2-D or 3-D convergence details of the PSO/RdBM^G algorithm, are given.

Further, from Fig. 4, it can be seen that for the unimodal or multimodal functions, the PSO/RdBM^G algorithm can quickly seek out the globally optimal solutions at the early co-evolutionary stage, while the other four methods (BLPSO[4], GEPSO[8], SPSO[9], and OJPSO[10]) fail to make it.

2) DYNAMIC OPTIMIZATION PERFORMANCE RESULTS ANALYSIS

Fig. 5 summarizes the averaged efficiency improvements of the solution of PSO/RdBM^G over that of BLPSO[4], GEPSO[8], SPSO[9], and OJPSO[10], for each dimension and heterogeneity model.

Shown as Fig. 6, for semiconsistent instances or for consistent instances, the averaged efficiency preponderance of the PSO/RdBM^G approach over the other four methods (BLPSO[4], GEPSO[8], SPSO[9], and OJPSO[10]) is obvious. Lower improvement factors are obtained for small inconsistent instances, but for large inconsistent instances, the advantage significantly increase.

V. CONCLUSION

As the extension of traditional artificial intelligence, PSO algorithms representing the swarm intelligence algorithms of the computational intelligence theory, have been widely used in heterogeneous multimodal optimization.

Although with some improvements, the PSO algorithms are mostly linear optimization-dynamic-models; and the PSO algorithm integrated with Ito process driven by the standard Brownian motion, is reduced to a special Markov stochastic process with constant expected drift rate and variance rate, also without the generality in the dynamic swarm intelligence simulation.

In this paper, based on the core idea of original PSO algorithm (such as memory, self-cognition and social cognition) and the strict theories such as nonlinear stochastic analysis, a greener heterogeneous scheduling algorithm via blending pattern of particle swarm computing intelligence and geometric Brownian motion, is proposed, to achieve the balance of "local exploration and macro development" or "route diversity and faster convergence".

In other words, a key series of strict theoretical derivation or proof is obtained, including the spatial representation of the continuous path of the particle swarm, denoted as \( x_{t+1} \in \Omega_T \) and defined in Equation (4), the relevant bounded Lipschitz function definition based on the space, denoted as \( \xi(X) \in Lip(\Omega_T) \) and defined in Equation (7), the G-heat equation optimal description, denoted as \( \Gamma_H \) and defined in Equation (8), and the nonlinear G-expectation definition, denoted as \( E[\xi(X)] | \Gamma_H \) and defined in Equation (9).

Then, the evaluation indicators can be divided into two categories: static and dynamic optimization performance. A large number of experimental results are given. Compared with most newly published scheduling algorithms (BLPSO[4], GEPSO[8], SPSO[9], and OJPSO[10]), there are significant advantages of the proposed algorithm on the dynamic optimization performance for consistent or semiconsistent and large inconsistent scheduling instances, although with lower improvement factors for small inconsistent instances.

ACKNOWLEDGMENT

This work has been supported by National Natural Science Foundation of China under Grant 61702248, 61070017 and
REFERENCES

[1] M. R. Bonyadi et al., “Particle swarm optimization for single objective continuous space problems: A review,” Evolutionary Computation, vol. 25, n. 1, pp. 1-54, 2017.

[2] V. Bohaienko et al., “Identification of fractional water transport model with psi-Caputo derivatives using particle swarm optimization algorithm,” Applied Mathematics and Computation, vol. 390, Article-Number 125665, 2021.

[3] A. P. Piotrowski et al., “Population size in particle swarm optimization,” Swarm and Evolutionary Computation, vol. 58, Article-Number 100718, 2020.

[4] X. Chen et al., “Biogeography-based learning particle swarm optimization for combined heat and power economic dispatch problem,” Knowledge-Based Systems, vol. 208, Article-Number 106463, 2020.

[5] B. Wei et al., “Multiple adaptive strategies based particle swarm optimization algorithm,” Swarm and Evolutionary Computation, vol. 57, Article-Number 100731, 2020.

[6] R. Santos et al., “A rotationally invariant semi-autonomous particle swarm optimizer with directional diversity,” Swarm and Evolutionary Computation, vol. 56, Article-Number 100700, 2020.

[7] A. Darwish et al., “An optimized model based on convolutional neural networks and orthogonal learning particle swarm optimization algorithm for plant diseases diagnosis,” Swarm and Evolutionary Computation, vol. 52, Article-Number 100616, 2020.

[8] D. Sedighizadeh et al., “GEPSO: A new generalized particle swarm optimization algorithm,” Mathematics and Computers in Simulation, vol. 179, n. 1, pp. 194-212, 2021.

[9] D. F. Hu et al., “Probabilistic convergence analysis of the stochastic particle swarm optimization model without the stagnation assumption,” Information Sciences, vol. 547, n. 2, pp. 996-1007, 2021.

[10] W. Q. Song et al., “Multifractional Brownian motion and quantum-behaved particle swarm optimization for short term power load forecasting: An integrated approach,” Energy, vol. 194, Article-Number 116847, 2020.

[11] Vimal Kumar Pathak and Amit Kumar Singh, “Form error evaluation of noncontact scan data using constriction factor particle swarm optimization,” Journal of Advanced Manufacturing Systems, vol. 16, n. 3, pp. 205-226, 2017.

[12] Vimal Kumar Pathak, Ramanpreet Singh and Swati Gangwar, “Optimization of three-dimensional scanning process conditions using preference selection index and metaheuristic method,” Measurement, vol. 146, n. 11, pp. 653-667, 2019.

[13] Swati Gangwar and Vimal Kumar Pathak, “Dry sliding wear characteristics evaluation and prediction of vacuum casted marble dust (MD) reinforced ZA-27 alloy composites using hybrid improved bat algorithm and ANN,” Materials Today Communications, vol. 25, n. 12, 101615, 2020.

[14] Vimal Kumar Pathak and Ashish Kumar Srivastava, “A novel upgraded bat algorithm based on cuckoo search and Sugeno inertia weight for large scale and constrained engineering design optimization problems,” Engineering with Computers, https://doi.org/10.1007/s00366-020-01127-3, 2020.

[15] Ashish Goyal, Nipun Gautam and Vimal Kumar Pathak, “An adaptive neuro-fuzzy and NSGA-II-based hybrid approach for modelling and multi-objective optimization of WEDM quality characteristics during machining titanium alloy,” Neural Computing and Applications, https://doi.org/10.1007/s00521-021-06261-7.

[16] Ramanpreet Singh, Kumar Gaurav and Vimal Kumar Pathak, “Best-Worst-Play (BWP): A metaphor-less Optimization Algorithm,” Journal of Physics: Conference Series, vol. 1455, 012007, 2019.

[17] Ahmad Rezaee Jordehi and Jasonrita Jasni, “Approaches for FACTS optimization problem in power systems,” in IEEE International Power Engineering and Optimization Conference, Melaka, Malaysia, 2012, https://doi.org/10.1109/PEOCO.2012.6230889.

[18] A. Shahnazi-Pour et al., “Numerical simulation of the Hurst index of solutions of fractional stochastic dynamical systems driven by fractional Brownian motion,” Journal of Computational and Applied Mathematics, vol. 386, Article-Number 113210, 2021.

[19] Z. W. Yang et al., “Strong convergence analysis for Volterra integro-differential equations with fractional Brownian motions,” Journal of Computational and Applied Mathematics, vol. 383, Article-Number 113156, 2021.

[20] D. Betsakos et al., “On the probability of fast exits and long stays of a planar Brownian motion in simply connected domains,” Journal of Mathematical Analysis and Applications, vol. 493, Article-Number 124454, 2021.

[21] A. Das et al., “Quantum Brownian motion: Drude and Ohmic baths as continuum limits of the Rubin model,” Physical Review E, vol. 102, Article-Number 062130, 2020.

[22] D. Banos et al., “Strong existence and higher order frechet differentiability of stochastic flows of fractional Brownian motion driven SDEs with singular drift,” Journal of Dynamics and Differential Equations, vol. 32, n. 4, pp. 1819-1866, 2020.

[23] Y. Ren et al., “Robust stability and boundedness of stochastic differential equations with delay driven by G-Brownian motion,” International Journal of Control, vol. 93, n. 12, pp. 2886-2895, 2020.

[24] S. Peng, “G-Expectation, G-Brownian Motion and Related Stochastic Calculus of Itô’s type,” The Abel Symposium 2005, Abel Symposia, Springer, pp. 541–567, 2007.

[25] S. Peng, “Nonlinear expectations and stochastic calculus under uncertainty,” Springer, 2019.

[26] T. Braun, “A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems,” Journal of Parallel and Distributed Computing, vol. 61, n. 6, pp. 810-837, 2001.