IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 9, 2020, accepted June 9, 2020, date of publication June 18, 2020, date of current version June 26, 2020.
Digital Object Identifier 10.1109/ACCESS.2020.3003304

An Algorithm Incarnating Deep Integration of Hardware-Software Energy Regulation Principles for Heterogeneous Green Scheduling

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 (lwjl22422@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61702248, Grant 61070017, and Grant 61272094, in part by the National High Technology Research and Development Program of China under Grant 2006AA01A113 and Grant 2012AA01A306, and in part by the Talent Introduction Project of Ludong University under Grant LB2016015.

ABSTRACT Heterogeneous green scheduling in virtual cloud is an urgent need of human sustainable developments. However, on the one hand, there is still considerable space beyond reach of the hardware energy regulation mode; on the other hand, as the core of green software methods, meta-heuristics algorithms are still underperforming in heterogeneous scheduling, although with many achievements in homogeneous scheduling. In this paper, an efficient new meta-heuristics algorithm is presented (i.e., GHSA_di), including the co-evolutionary dynamics equation emphasizing on and taking advantage of the hardware energy-regulation principles. The experimental results show that compared with the other three meta-heuristic scheduling algorithms, GHSA_di algorithm has obvious advantages in overall performance, energy saving and scalability, for both data intensive and computing intensive instances.

INDEX TERMS Heterogeneous scheduling, green computing, meta-heuristic algorithm, energy regulation principles, deep integration.

I. INTRODUCTION

A. BACKGROUND AND MOTIVATION

Nowadays the virtual cloud, aggregating wide-area distributed homogeneous or heterogeneous clusters and other infrastructures, has been profoundly changing human life or production styles all over the world [1]. According to the studies, CO_2 produced by the industry of information communication technology (ICT), may rise to 14% of the global emissions in 2040 [2]; and other statistics show that there is a huge waste of energy in data centers, since PUE (Power Usage Effectiveness) of China's data centers is generally greater than 2.2 and that of America's is basically maintained at 1.9, where the closer the PUE value is to 1, the higher the greening degree of a data center. Therefore, no matter for environmental protection or for low-carbon economy, there are imperative requirements on the computing evolution from high performance to high efficiency [3]–[6].

The associate editor coordinating the review of this manuscript and approving it for publication was Ying Xu.

However, on the one hand, there is still considerable space beyond reach of the hardware energy regulation mode [7], [8]; on the other hand, as the core of green software methods, meta-heuristics algorithms are still underperforming in heterogeneous scheduling, although with many achievements in homogeneous scheduling [9].

Here, inspired by Darwin's natural theory or the biological immunity principles, genetic algorithms (GAs) or the artificial immune algorithms iteratively search the solution space by the meta-heuristics, with encoding/decoding the biomimetic individuals (candidate solutions) and the dynamics equation for the evolutionary mechanism [10], [11]. Especially for heterogeneous scheduling, the evolutionary dynamics equations, are constructed based on the appropriate definitions of various QoS (Quality of Service) metrics; further, green heterogeneous scheduling aims for the higher energy-efficiencies with no effect on computing performance [12].

Focusing on deep integration of hardware-software energy regulation principles, an efficient meta-heuristic scheduling algorithm, i.e., GHSA_di, is proposed in this paper; the main contributions includes (1) the co-evolutionary dynamics equation emphasizing on and taking advantage of the hardware energy- regulation principles, (2) three-dimensional encoding/decoding of the biomimetic individuals and the corresponding evolutionary mechanism, (3) a creative hierarchical parallelization algorithm-model suitable for the super hybrid systems of the scheduling server.

This research belongs to the multi-disciplinary direction of evolutionary computation, distributed artificial intelligence, green computing, heterogeneous many-core super-systems, and multi-objective optimization.

B. OUTLINING

The rest of the paper is organized as follows. Section 2 outlines the related work. In Section 3, the algorithm incarnating deep integration of hardware-software energy regulation principles for heterogeneous scheduling, i.e., GHSA_di, is proposed. Performance evaluations and the analyses of the algorithms are discussed in Section 4. Section 5 concludes the paper with a summary.

II. RELATED WORK

A. HARDWARE ENERGY REGULATION PRINCIPLES

Nowadays it has been the growing trend to use heterogeneous many-core systems in accelerating super-scale scientific computing; here, heterogeneity usually contains two aspects of signification: the design of hardware itself (such as the manufacturing engineering or cores) and the real-time running state.

Concurrently, there is another upward trend for the intellectualization of the hardware. Involving circuit or microelectronic level, there are many landmark achievements, such as DVFS (Dynamic Voltage Frequency Scaling) and DPM (Dynamic Power Management) [7]. It's smart how it does this: according to the real-time tasks or workloads, a minimum number of active components can be provided or the working frequency is reduced [8]. During the dynamic adjustment from active mode to sleep mode, the energy consumption component must be in ''idle'' state; however, research shows that the energy consumption of 8-core Xenon processor in idle state is 60% of that in full load state [13]. Furthermore, it is noteworthy that the dynamic power consumption (W) of heterogeneous processors is very different, even under the same workloads.

Generally, the power-consumption of integrated circuits is usually composed of two parts: static and dynamic power consumption; static power-consumption is generated by leakage current, and dynamic power consumption is mainly caused by the opening and the closing the capacitor. So, the measurement of cluster processors' dynamic power consumption is more complex than that of static power consumption. At present, most of the related work is roughly estimated. In [14], firstly, the conventional hardware events are classified according to the correlation with power consumption; secondly, the number of high-order hardware events is counted according to the performance counter in real time; finally, the dynamic power consumption is estimated according to experience. With the hardware and applications developments in spirals, hardware events in different areas show a trend of diversification and time varying; then, it means that the power consumption estimation mode based on hardware event count is only applicable to the homogeneous processors.

B. SOFTWARE METHODS FOR GREEN SCHEDULING

Super-scale heterogeneous real-time scheduling needs to reduce energy consumption without affecting performance. This is not only for the low-carbon economy or sustainable human development, but also for the reliability and stability of the system [3]–[6]. Many researches turn to virtual management upgrading, using Gaussian process regression method [15], multidimensional packing mode [16] or integer programming strategy [17]. With a core of multitude to one mapping between the virtual machines and the physical resources, these ways can improve the utilization rate of the hardware resources; however, these fuzzy decisions are suitable for the saving energy in the homogeneous clusters. Also to enhance the competitiveness, the infrastructure operators prefer the lower maintenance costs. Because of the volatility of the electricity price in different time zones all over the world, some explorations are about how to minimize the costs through activating and adjusting the cluster-number in each geographical area with the satisfied requirements (such as service delays, etc.) [18]. To optimize the scheduling sequence of clusters in different time zones, in [19], an online measurement and overload probability estimation model were proposed based on the principle of large deviation, and the dynamic allocation scheme of computing resources was obtained by combining the iterative method. In [20], this problem was formalized as a Markov decision process, which included service request distribution, cloud node supply, energy storage equipment management, and the transaction; then, the dynamic control strategy was designed by using Lagrange optimization and Q-learning theory to reach a compromise between battery investment and economic savings.

To improve the performance for cloud service platforms by minimizing uncertainty propagation in scheduling workflow applications that have both uncertain task execution time and data transfer time, an unceRtaintyaware Online Scheduling Algorithm (ROSA) to schedule dynamic and multiple workflows with deadlines, is developed in [21], and a novel scheduling algorithm (PRS1) that dynamically exploits proactive and reactive scheduling methods, for scheduling real-time, aperiodic, independent tasks, is addressed in [22]; ROSA [21] and PRS1 [22] can effectively improve the performance of a cloud data center.

While the data centers pay attention to high performance, they are urgent to seek breakthrough in effective implanting assessment criteria of energy in cloud middleware. Existing work is mainly based on approximately linear mathematical

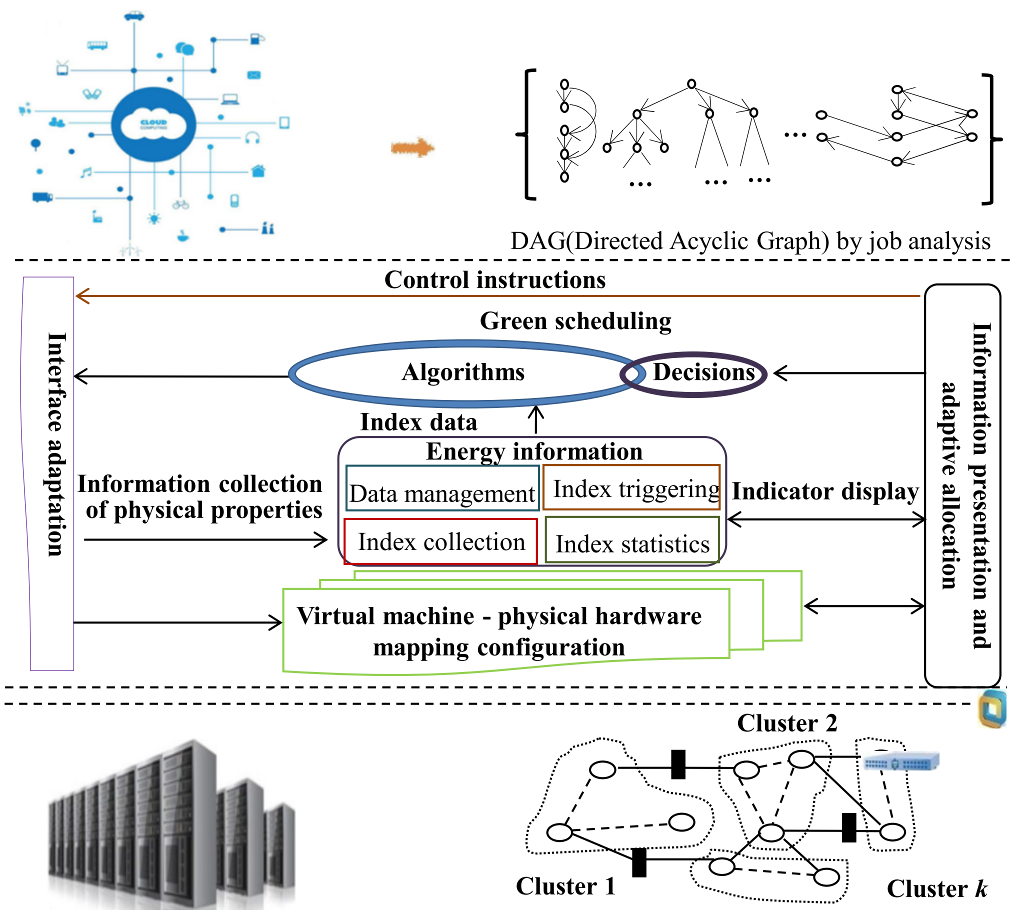

FIGURE 1. The architecture of heterogeneous scheduling in virtual cloud.

models of dynamic energy consumptions [19], [20], [23]. The share of data intensive services such as social networks, has risen sharply, which consist of a large number of stochastic tasks with Terabytes (TB) or petabytes (PB) of data; however, for data intensive applications, they often are short of the accuracy and promptness.

Moreover, the scheduling algorithm is usually heuristic. It aggregates multiple QoS indexes into one goal, and obtains feasible solutions in the decision space. Such an approach often reduces the quality of the final solution or lacks flexibility and scalability [24].

In recent years, evolutionary algorithms with better meta-heuristics have been used to solve the problem of cloud scheduling [25], [26]. In [27], the partition parameter adaptation differential evolution (PPADE) was used to deal with challenging constrained scheduling problem; with its characteristics of new mutation strategy "current-topbest/U-Ip", PPADE [27] had stable convergence and increased the power production in average best benefit by 9.06, 17.09, 35.69, 69.67∗10(8) kWh in wet year. Based on artificial immune theory, a multi-objective constraint scheduling algorithm (MOCTS-AI) was proposed in [28]; MOCTS-AI [28] added prior knowledge to the vaccine selection and population updating, so that the algorithm convergence was accelerated for the scheduling problems. For many-objective optimization problems, [29] suggested a clustering-based evolutionary algorithm, i.e., MaOEA/C; by classifying the population into a number of clusters, MaOEA/C [29] balanced the diversity and convergence. The existing research practice shows that although the speed index of scheduling server can reach the peak value, the efficiency of meta-heuristic algorithm is not high in most of the time.

III. AN ALGORITHM INCARNATING DEEP FUSION OF ENERGY REGULATION PRINCIPLES

Generally, cloud system includes user layer, middleware layer, virtual resource layer and infrastructure layer. As the core of middleware, scheduling algorithm is also the most stressed component (See **Figure. 1**).

A. THE OPTIMIZATION DYNAMICS EQUATION EMPHASIZING ON AND TAKING ADVANTAGE OF THE HARDWARE ENERGY-REGULATION PRINCIPLES

Based on experimental improvements, physical simulation and mathematical theory, a series of optimization forces in the equation are defined as follows.

Definition 1 (Dynamic Energy Consumption (in Wh)): In the light of energy heterogeneity of the different CPU processor-type (denoted by $v \in$ {1,2,3}), the dynamic energy consumption (in Wh) is the product of power values (in W) and execution time (denoted by $\Delta \boldsymbol{T}_i$). Indeed, there are great differences in dynamic power consumption (in W).

In general, the power consumption of an integrated circuit is usually composed of static and dynamic parts; because the static power is produced by the leakage current in the steady state, which can be defined as a constant, the evaluation of energy consumption among QoS indexes should pay more attention to the dynamic parts.

Since several CPU frequencies are allowed, the dynamic energy consumption (in Wh) of all nodes contains that of different DVFS levels.

The dynamic energy consumption (in Wh) of all nodes, denoted by $\boldsymbol{Dynamic\_energy}(\phi)$, *is given by* ***Eq.(1)***:

$$\boldsymbol{Dynamic\_energy}(\phi)=\sum\nolimits_{k=1}^{n_k}\sum\nolimits_{h=1}^{n_k^h}\sum\nolimits_{i=1}^{n_F^v}\{\boldsymbol{load}^v[\varsigma_{full}(\boldsymbol{F}_i^v)\ \varsigma_{idle}(\boldsymbol{F}_i^v)]+\varsigma_{idle}(\boldsymbol{F}_i^v)\}\times\Delta\boldsymbol{T}_i$$

where $\boldsymbol{n}_F^v$ *is the number of different DVFS levels for the CPU processor-type* $\boldsymbol{v} \in$ *{1,2,3},* $\boldsymbol{load}^v$ *is the CPU load at Frequency* $\boldsymbol{F}_i^v$, $\varsigma_{full}(\boldsymbol{F}_i^v)$ *is the power consumption (in W) of the CPU with full load running at Frequency* $\boldsymbol{F}_i^v$ *by manual intervention, and* $\varsigma_{idle}(\boldsymbol{F}_i^v)$ *is the power consumption (in W) of the CPU with no load running at Frequency* $\boldsymbol{F}_i^v$.

Definition 2(Response Time): One of the most important performance factors, is considered as the execution time of virtual machines (VMs); it can be evaluated based on the number of instructions (in million) that VMs have to execute, denoted by $\boldsymbol{NbInstr}^{\theta}$.

And the capacity of each virtual machine (for example, in terms of million instructions per second (MIPS)), denoted by $\rho_{\boldsymbol{F}_{i,v}}^{\theta,k_h}$, *relates to CPU allowed of Node* $\boldsymbol{k}_h$.

Then, response time, denoted by $\boldsymbol{Response\_time}(\phi)$, *can be expressed (in seconds) as* ***Eq. (2)***:

$$\boldsymbol{Response\_time}(\phi)=\max_{k=1}^{\boldsymbol{n_k}}\max_{h=1}^{\boldsymbol{n_h^k}}\max_{\theta=1}^{\boldsymbol{\varpi_h^k}}(\boldsymbol{NbInstr}^{\theta}/\rho_{\boldsymbol{F}_{i,v}}^{\theta,\boldsymbol{k_h}})$$

where the VM θ is on Node k_h, whose processor type based on energy heterogeneity is $\boldsymbol{v} \in$ *{1,2,3} and current working frequency is* $\boldsymbol{F}_i^v$.

Definition 3 (Resources Scalability): One of the most important QoS metrics, is the resources scalability that means a certain amount of latent capacities under a peak load. As flexibility factors, the resources scalability denoted by $\boldsymbol{Resources\_scalability}(\phi)$ *represents the available computing power without new nodes added.*

It can be evaluated based on the maximum CPU capacity allowed of Node $\boldsymbol{k}_h$, *denoted by* $\xi_v^{\boldsymbol{k}_h}$, *and the current CPU capacity allowed of Node* $\boldsymbol{k}_h$, *denoted by* $\rho_{\boldsymbol{F}_{i,v}}^{\theta,\boldsymbol{k}_h}$. *Then, the resources scalability can be expressed as* ***Eq. (3)***:

$$\boldsymbol{Resources\_scalability}(\phi)=\{\sum\nolimits_{k=1}^{\boldsymbol{v}_k}\sum\nolimits_{h=1}^{\boldsymbol{v}_h^k}(\xi_v^{\boldsymbol{k}_h}-\rho_{\boldsymbol{F}_{i,v}}^{\boldsymbol{k}_h})\}/(\sum\nolimits_{k=1}^{\boldsymbol{v}_k}\boldsymbol{v}_h^k)$$

where $\boldsymbol{n}_{\mathbf{k}}$ *is the number of clusters,* $\boldsymbol{v}_h^k$ *is the number of computing nodes in Cluster* $\boldsymbol{k}$, *the processor type of Node* $\boldsymbol{k}_h$ *based on energy heterogeneity is* $\boldsymbol{v}$ **and its current working frequency is** $\boldsymbol{F}_i^v$.

Definition 4 (Hardware Reliability): With resistance to network failure and malicious attacks, the hardware reliability is another of the most important QoS metrics.

Here, the hardware reliability, denoted by $\boldsymbol{Hardware\_reliability}(\phi)$, *is interpreted as the average VM-number on per node, or how many VMs should be transplanted if Node* $\boldsymbol{k}_h$ *fails, expressed as* ***Eq. (4)***, *where* $\boldsymbol{n}_{\mathbf{k}}$ *is the number of clusters,* $\boldsymbol{v}_h^k$ *is the number of computing nodes in Cluster* $\boldsymbol{k}$, *and* ϖ_h^k *is the number of virtual machines running on Node* $\boldsymbol{k}_h$.

$$Hardware\_reliability(\phi)=(\sum\nolimits_{k=1}^{\boldsymbol{v}_k}\sum\nolimits_{h=1}^{\boldsymbol{v}_h^k}\varpi_h^k)/(\sum\nolimits_{k=1}^{\boldsymbol{v}_k}\boldsymbol{v}_h^k)$$

Definition 5 (Service Security): As an optimization objective by schedulers, the service security for all tasks α_i($\boldsymbol{i} \in$ {1,…,$\boldsymbol{m}$}) *is expected to be maximized.*

In view of different requirements for cloud service security, the safety benefit of an independent task α_i($\boldsymbol{i} \in$ {1,…,$\boldsymbol{m}$}) *under the timing constraint is given by* ***Eq. (5)***:

$$\boldsymbol{R}_i(\phi)=\sum\nolimits_{j=1}^{q}\sigma_i^j s_i^j$$

where σ_i^j *is the weight coefficient of the* ***j****th security requirement of Task* α_i($\boldsymbol{i} \in$ {1,…,$\boldsymbol{m}$}), *since users can specify the weight coefficients to reflect priorities among* $\boldsymbol{q}$ **requirements of the service security. Simultaneously, the** *jth requirement level of Task* α_i($\boldsymbol{i} \in$ {1,…,$\boldsymbol{m}$}) *(denoted by* s_i^j*) can be provided and should be in the scope [MIN(S_i^j),MAX(S_i^j)] where MIN(S_i^j) and MAX(S_i^j) are the minimum and maximum of safety benefits for Task* α_i($\boldsymbol{i} \in$ {1,…,$\boldsymbol{m}$}), *respectively.*

Here, $\sum_{j=1}^{q}\sigma_i^j=1$ *where* $\boldsymbol{q}$ **is the number of security requirement of Task** α_i($\boldsymbol{i} \in$ {1,…,$\boldsymbol{m}$}).

Following that, the service security (denoted by $\boldsymbol{Security\_service}(\phi)$) *for all tasks* α_i($\boldsymbol{i} \in$ {1,…,$\boldsymbol{m}$}), *can be defined as* ***Eq. (6)***:

$$\boldsymbol{Security\_service}(\phi)=\sum\nolimits_{\boldsymbol{i}=1}^{m}\psi_{\boldsymbol{i}}\boldsymbol{R}_{\boldsymbol{i}}=\sum\nolimits_{\boldsymbol{i}=1}^{\boldsymbol{m}}\psi_{\boldsymbol{i}}\sum\nolimits_{\boldsymbol{j}=1}^{\boldsymbol{q}}(\sigma_{\boldsymbol{i}}^{\boldsymbol{j}}s_{\boldsymbol{i}}^{\boldsymbol{j}})$$

where $\boldsymbol{m}$ **is the number of submitted tasks,** ψ_i($\boldsymbol{i} \in$ {1,…,$\boldsymbol{m}$}) *is set to 1 if Task* α_i($\boldsymbol{i} \in$ {1,…,$\boldsymbol{m}$}) *is accepted, and* ψ_i($\boldsymbol{i} \in$ {1,…,$\boldsymbol{m}$}) *is set to 0 otherwise.*

Taken together, the definitions of the optimization force of the QoS metrics are applied to the nonlinear heterogeneous scheduling optimization, which gives fitness or affinity evaluation of chromosomes or antibodies in the following intelligent algorithms, defined as **Eq. (7)**, where Λ_i respectively represents the weight factor of the QoS indicator.

$$\begin{aligned}\Gamma(\phi) = \min_{\phi\in\Phi}[&\Lambda_1 \cdot \boldsymbol{Dynamic\_energy}(\phi)\\ &+\Lambda_2 \cdot \boldsymbol{Response\_time}(\phi)\\ &+ \Lambda_3 \cdot \boldsymbol{Hardware\_reliability}(\phi)\\ &- \Lambda_4 \cdot \boldsymbol{Resources\_scalability}(\phi)\\ &- \Lambda_5 \cdot \boldsymbol{Security\_service}(\phi)]\end{aligned}$$

Following that, in order for adding the sufficient evolutionary dynamics to GHSA_di, the first three QoS metrics: $\boldsymbol{Dynamic\_energy}(\phi)$, $\boldsymbol{Response\_time}(\phi)$, $\boldsymbol{Hardware\_reliability}(\phi)$ have to be minimized, as opposed to $\boldsymbol{Resources\_scalability}(\phi)$ and $\boldsymbol{Security\_service}(\phi)$ that have to be maximized.

Another feature of the dynamics equation is the compromise coefficients. Here, the compromise coefficients can be tailored due to preference for relevant indicators.

B. THREE-DIMENSIONAL ENCODING/DECODING OF THE BIONIC INDIVIDUALS

A scheduling candidate scheme mapping among the tasks $\{\boldsymbol{Xi}^{\mathbf{r}}(\boldsymbol{i} \in \{1,\ldots, \boldsymbol{m}\}, \boldsymbol{r} \in \mathrm{R}^+)\}$, the virtual machines $\{\boldsymbol{Yi}^{\mathbf{r}}(\boldsymbol{i} \in \{1,\ldots, \boldsymbol{m}\}, \boldsymbol{r} \in \mathrm{R}^+)\}$ and the nodes $\{\boldsymbol{Zi}^{\mathbf{r}}(\boldsymbol{i} \in \{1,\ldots, \boldsymbol{m}\}, \boldsymbol{r} \in \mathrm{R}^+)\}$, is regarded as a biomimetic individual $\boldsymbol{Ch}_{\mathbf{r}}(\boldsymbol{i} \in \{\boldsymbol{r} \in \mathrm{R}^+)$.

Specifically, the gene feature $\{\boldsymbol{Gi}^{\mathbf{r}}(\boldsymbol{i} \in \{1,\ldots, \boldsymbol{m}\}, \boldsymbol{r} \in \mathrm{R}^+)\}$ of $\boldsymbol{Ch}_{\mathbf{r}}(\boldsymbol{i} \in \{\boldsymbol{r} \in \mathrm{R}^+)$ is expressed as the three-dimensional encoding, which represents the random task $\boldsymbol{Xi}^{\mathbf{r}}(\boldsymbol{i} \in \{1,\ldots, \boldsymbol{m}\}, \boldsymbol{r} \in \mathrm{R}^+)$ is assigned to the virtual machine $\boldsymbol{Yi}^{\mathbf{r}}(\boldsymbol{i} \in \{1,\ldots, \boldsymbol{m}\}, \boldsymbol{r} \in \mathrm{R}^+)$ of the computing node $\boldsymbol{Zi}^{\mathbf{r}}(\boldsymbol{i} \in \{1,\ldots, \boldsymbol{m}\}, \boldsymbol{r} \in \mathrm{R}^+)$.

So the biomimetic individual $\boldsymbol{Ch}_{\mathbf{r}}(\boldsymbol{i} \in \{\boldsymbol{r} \in \mathrm{R}^+)$ is encoded as three-dimensional matrices (see **Eq.(8)**).

$$Ch_r = \begin{bmatrix} \boldsymbol{X}_1^r & \boldsymbol{Y}_1^r & \boldsymbol{Z}_1^r \\ \boldsymbol{X}_2^r & \boldsymbol{Y}_2^r & \boldsymbol{Z}_2^r \\ \vdots & \vdots & \vdots \\ \boldsymbol{X}_i^r & \boldsymbol{Y}_i^r & \boldsymbol{Z}_i^r \\ \vdots & \vdots & \vdots \\ \boldsymbol{X}_m^r & \boldsymbol{Y}_m^r & \boldsymbol{Z}_m^r \end{bmatrix}$$

In decoding rules, emphases are put on the two situations of assigning different tasks to the same virtual machine as follows.

(1) If the logic depths are different, the depth sorting principle is followed to avoid long waiting and even deadlock between tasks.

(2) If the logic depths are same, the ranking principle of coupling strength is followed to shorten the critical-path length for the optimal effect.

C. EVOLUTIONARY OPERATOR-DEFINITIONS

In general, genome evolution simulation based on three dimensional matrices encoding, includes the definition of intelligent operators, such as individual selection, crossover, mutation and clone.

Clone operators play an important role in the diversity and approximation of green heterogeneous scheduling candidate solutions. In GHSA_di algorithm, the clone operation $\Gamma_{\mathbf{C}}$ of the bionic population $\boldsymbol{Ch} = \{\boldsymbol{Ch}_1, \boldsymbol{Ch}_2,\ldots, \boldsymbol{Ch}_\epsilon,\ldots, \boldsymbol{Ch}_{\boldsymbol{\theta}}\}$ can be defined as **Eq.(9)**.

$$\begin{aligned}&Ch^*(\iota)\\ &= \Gamma_C\{Ch_1(\iota), Ch_2(\iota), \ldots, Ch_\epsilon(\iota), \ldots, Ch_{\theta(\iota)}(\iota)\}\\ &= \Gamma_C(Ch_1(\iota)) + \ldots + \Gamma_C(Ch_\epsilon(\iota)) + \ldots + \Gamma_C(Ch_{\theta(\iota)}(\iota))\\ &= \{Ch_1^1(\iota), Ch_1^2(\iota), \ldots, Ch_1^\epsilon(\iota), \ldots, Ch_1^{31}(\iota)\}\\ &\quad + \ldots + \{Ch_{\theta(\iota)}^1(\iota), Ch_{\theta(\iota)}^2(\iota), \ldots, Ch_{\theta(\iota)}^\epsilon(\iota), \ldots, Ch_{\theta(\iota)}^{3\theta(\iota)}(\iota)\}\end{aligned}$$

Here, $\Gamma_C(Ch_i(\iota)) = \{Ch_i^1(\iota), Ch_i^2(\iota),\ldots, Ch_i^\epsilon(\iota),\ldots, Ch_i^{3i}(\iota)\}$, $i = 1,2,\ldots, \theta(\iota)$; $\boldsymbol{3i}\in[1, m_c]$ is the adjustable parameter, indicating clone probability, where $\boldsymbol{3i} = 1$ means no clone operation on $\boldsymbol{Ch}_{\mathbf{i}}(\iota)$ and $\boldsymbol{m}_{\mathbf{c}}$ is the upper limit of clone probability.

In GHSA_di algorithm, by using the same clone probability 3 for each bionic individual, the size of feasible non dominated solution set in the optimization process is almost doubled, the diversity of individuals is maintained and the group convergence is accelerated, defined as **Eq.(10)**.

$$\begin{aligned}Ch^*(\iota) = \{Ch_1^1(\iota), Ch_1^2(\iota), \ldots, Ch_1^3(\iota)\} + \ldots + \{Ch_{\theta(\iota)}^1(\iota),\\ Ch_{\theta(\iota)}^2(\iota), \ldots, Ch_{\theta(\iota)}^3(\iota)\}\end{aligned}$$

In contrast to clone, the selection operation divides the population into non inferior solution or inferior solution, and only non inferior solution can enter the next generation.

For each bionic individual $Ch^\#(\iota) \in Ch^{**}(\iota)$, if $Ch^\#(\iota)$ satisfies **Eq. (11)**, it is called non inferior solution, otherwise it is inferior solution.

$$\begin{aligned}\neg\exists\, \delta_\kappa^{\varpi\#}(\iota) \neq \delta^\#(\iota)(\kappa = 1, 2, \cdots, \theta; \varpi = 1, 2, \cdots, \mathbf{3})\\ \in \delta^{**}(\iota) : (\forall i \in \{1, \cdots, m\} : f_i(\delta^\#(\iota)) \geq f_i(\delta_\kappa^{\varpi\#}(\iota)))\end{aligned}$$

In GHSA_di algorithm, the selection operation $\Gamma_{\mathbf{S}}$ of the bionic population $\boldsymbol{Ch}=\{\boldsymbol{Ch}_1, \boldsymbol{Ch}_2,\ldots, \boldsymbol{Ch}_\epsilon,\ldots, \boldsymbol{Ch}_{\boldsymbol{\theta}}\}$ can be defined as **Eq.(12)**.

$$\begin{aligned}Ch^{***}(\iota) &= \Gamma_S(Ch^{**}(\iota))\\ &= \Gamma_S(\{Ch_1^{1\#}(\iota), Ch_1^{2\#}(\iota), \ldots, Ch_1^{3\#}(\iota)\} + \ldots\\ &\quad +\{Ch_{\theta(\iota)}^{1\#}(\iota), Ch_{\theta(\iota)}^{2\#}(\iota), \ldots, Ch_{\theta(\iota)}^{3\#}(\iota)\})\\ &= \Gamma_S(\{Ch_1^{1\#}(\iota), Ch_1^{2\#}(\iota), \ldots, Ch_1^{3\#}(\iota), \ldots, Ch_{\theta(\iota)}^{1\#}(\iota),\\ &\quad Ch_{\theta(\iota)}^{2\#}(\iota), \ldots, Ch_{\theta(\iota)}^{3\#}(\iota)\})\\ &= \{Ch_{1\#}(\iota), Ch_{2\#}(\iota), \ldots, Ch_\epsilon^\#(\iota), \ldots, Ch_{\theta s(\iota)}^\#(\iota)\}\end{aligned}$$

Here, $Ch_\epsilon^\#(\iota)(\epsilon = 1,\ldots, \theta\boldsymbol{s}(\iota))$ signifies the non inferior individual of the bionic population $Ch^{**}(\iota)$; $\theta\boldsymbol{s}(\iota)$ is the number of the non inferior individuals.

Generally, co-evolutionary system selects feasible solution set according to the constraint deviation value of contemporary individuals, and then selects feasible non inferior solution set according to the objective function value of individuals.

In comparison, according to the evolutionary dynamic information matrix defined in **Section 3.A**, the GHSA_di algorithm can directly select non inferior bionic individuals in the population, which greatly improves the efficiency of the algorithm.

At the same time, different strategies of gene crossover and mutation can help to maintain the diversity of population, and to cooperate or exchange information among bionic individuals.

In GHSA_di algorithm, the gene operation $\Gamma_{\mathbf{G}}$ of the bionic population $\boldsymbol{Ch}=\{\boldsymbol{Ch}_1, \boldsymbol{Ch}_2,\ldots, \boldsymbol{Ch}_\epsilon,\ldots, \boldsymbol{Ch}_{\boldsymbol{\theta}}\}$ can be defined as **Eq.(13)**.

$$
\begin{aligned}
&Ch^{**}(\iota)\\
&= \Gamma_G(Ch^*(\iota))\\
&= \Gamma_G(\{Ch_1^1(\iota), Ch_1^2(\iota), \ldots, Ch_1^3(\iota)\} + \ldots + \{Ch_{\theta(\iota)}^1(\iota),\\
&\quad Ch_{\theta(\iota)}^2(\iota), \ldots, Ch_{\theta(\iota)}^3(\iota)\})\\
&= \{\Gamma_G(Ch_1^1(\iota)) + \Gamma_G(Ch_1^2(\iota)) + \ldots + \Gamma_G(Ch_1^3(\iota))\} + \ldots\\
&\quad +\{\Gamma_G(Ch_{\theta(\iota)}^1(\iota)), \Gamma_G(Ch_{\theta(\iota)}^2(\iota)) + \ldots + \Gamma_G(Ch_{\theta(\iota)}^3(\iota))\}\\
&= \{Ch_1^{1\#}(\iota), Ch_1^{2\#}(\iota), \ldots, Ch_1^{3\#}(\iota)\} + \ldots + \{Ch_{\theta(\iota)}^{1\#}(\iota),\\
&\quad Ch_{\theta(\iota)}^{2\#}(\iota), \ldots, Ch_{\theta(\iota)}^{3\#}(\iota)\}
\end{aligned}
$$

In general, co-evolutionary systems simulate SBX cross over or polynomial mutation operators. In GHSA_di algorithm, the selection of crossover or mutation points can also be based on the evolutionary dynamic information matrix defined in **Section 3.A**.

D. A MULTI-LEVEL PARALLEL ALGORITHM-DESIGN

Currently, there are two kinds of parallelism; one is the concurrency of the inherent evolutionary mechanism in the meta-heuristics algorithms, and the other is the high-speed parallelism of the high-performance computers originated from its parallel and distributed hierarchies.

Specially aimed at energy-aware heterogeneous scheduling with such optimization characteristics as supersized, strong constrictions, multi-objective, and non-linear, and oriented to the server running algorithms with CPU-GPU cooperative hybrid architecture, a creative hierarchical parallelization design, integrating an unconventional master slave model with the coarse grained model, is proposed in GHSA_di algorithm.

Firstly, according to the coarse grained model, a large number of subgroups of the biomimetic individuals (i.e. candidate solution set) are placed on different nodes (Step 3), with independently performing the related evolutionary operations; and in every migration cycle, each subgroup will exchange several individuals (Step 12∼ Step 19), in order to introduce better individuals and enrich the diversity of population, respectively.

Secondly, on each node, the evolutionary operators, such as crossover and mutation, are implemented by CPU, and GPU can extensively evaluate the genome fitness(Step 5∼ Step 11); here, CPU is regarded as the main server, and a number of threads executing on GPU are clients; then they make up an unconventional master-slave model.

The GHSA_di Algorithm

Step 1: **Initialize** the iteration (ι) and the subpopulation $\Xi(\iota)=\{\boldsymbol{Ch}_1(\iota), \boldsymbol{Ch}_2(\iota),\ldots, \boldsymbol{Ch}_\epsilon(\iota),\ldots, \boldsymbol{Ch}_{\boldsymbol{\theta}}(\iota)\}$, each subpopulation of Θ individuals;
Step 2: **While** ($\iota < \iota_{\mathbf{max}}$) and (other termination criteria are not satisfied)
Step 3: **Do in parallel** for each island /∗ Obtain coarse-grained model, one of parallel and distributed models ∗/
Step 4: $\iota = \iota+1$;
Step 5: **Do in parallel**/∗ Obtain master-slave model, another parallel model ∗/
Step 6: **Evaluate** genome fitness based on the dynamic equation (**Eq.(7)**) in the current subpopulation: $\Gamma(\boldsymbol{Ch}_r(\tau))(r \in \boldsymbol{R}^+, \boldsymbol{Ch}_r(\tau) \in \Xi(\tau))$;
Step 7: **Sort** the individuals fitness: $\Gamma(\boldsymbol{Ch}_r(\tau))(r \in \boldsymbol{R}^+, \boldsymbol{Ch}_r(\tau) \in \Xi(\tau))$, and **save** the fittest individual $\boldsymbol{Ch}_{\mathbf{elite}}(\iota)$ in the external memory;
Step 8: **Perform** local search strategies to ensure the Cesaro average convergence of the irreducible aperiodic Markov chain of the population sequence;
Step 9: **Perform** clonal operation;
Step 10: **Perform** gene operations, such as crossover and mutation defined as **Section 3.C**;
Step 11: **End Do in parallel**
Step 12: **If** $\iota = \tau$(migration interval) **then**
Step 13: **Create** Ψ_δ for the current subpopulation;
Step 14: **Send** Ψ_δ to the neighboring subpopulation;
Step 15: **Receive** Ψ_δ from the neighboring subpopulation;
Step 16: **Construct** the founding subpopulation Ξ;
Step 17: **Select** Θ individuals into Ξ;
Step 18: **Replace** the subpopulation Ψ_δ with Ψ_δ^τ;
Step 19: **End If**
Step 20: **End Do in parallel**
Step 21: **End While**
Step 22: **Output** the best individual.

E. THE COMPLEXITY ANALYSIS OF THE ALGORITHM

Let's assume in each evolution generation, the size of the population ***FeaNonPop*** and ***ModNonPop*** is θ, and the cloning multiples, the variable dimension, constraint dimension, and the objective function dimension are, respectively, $\boldsymbol{3}, \boldsymbol{£}, \hbar, \boldsymbol{m}$.

Complexity of cloning operation of the population ***FeaNonPop*** or ***ModNonPop***: $O(\boldsymbol{3}\theta)$.

Complexity of crossover operation of the population ***FeaNonPop*** or ***ModNonPop***: $O(\boldsymbol{£3}\theta/2)$.

Complexity of mutation operation of the population ***FeaNonPop*** or ***ModNonPop***: $O(\boldsymbol{£3}\theta)$.

Complexity of calculating the genetic affinity value of the population ***FeaNonPop*** or ***ModNonPop***: $O(\boldsymbol{£3}\theta)$.

Complexity of selecting the non-dominant solution set: $O((\boldsymbol{m}+1)(\boldsymbol{3}+1)\ \theta+\ \theta+\boldsymbol{m}(\boldsymbol{3}+1)^2\ \theta^2+(\boldsymbol{m}+1)(\boldsymbol{3}+1)\ \theta \log_2((\boldsymbol{3}+1)\ \theta))$.

Then, the time complexity of GHSA_di algorithm is polynomial time.

IV. EXPERIMENT RESULTS AND DISCUSSION

The competitive advantage of evaluation is that all the experiments have been carried out at National Supercomputing Center in Jinan, China. The cloud platform using the supercomputer Sunway TaihuLight, has advanced cold-pool micro-module rooms based on container technologies, daily service evaluation scenarios for various data or computing intensive applications, a highly visualized display environment and a sound network infrastructure.

A. SIMULATOR AND SIMULATION PARAMETERS

In the course of the experiment, 200 clusters with three common nodes based on energy heterogeneity ($v \in R+$) are used. The optimal resource utilization of CPU/GPU or the disks in the clusters, as we know, entirely depends on the heterogeneous characteristics of hardware. Then, the relevant parameter values are as follows.

(1) In further detail below, disk-optim and processor-optim represent the usage range of CPU/GPU and hard disk at maximum energy utilization, respectively. {***disk-optim***:[0.75,0.8], ***processor-optim***:[0.8,0.9], ***v***:1}, means that for the nodes with energy heterogeneity ***v***=1, there is the highest energy efficiency when the disk and CPU/GPU utilization is respectively within certain range [75%, 80%] and [80%, 90%].

(2) In the same manner, there are {***disk-optim***:[0.6,0.65], ***processor-optim***:[0.6,0.7], ***v***:2} and {***disk-optim***:[0.45, 0.5], ***processor-optim***:[0.4,0.5], ***v***:3}.

(3) In the following sections, ***processor-init*** and ***disk-init*** represent the initial utilization of CPU/GPU and hard disk at the outset of the experiment, respectively; there exist ***processor-init***∈[0.1, 0.4] and ***disk-init***∈ [0.1, 0.4].

B. OVERALL PERFORMANCE COMPARISONS

Firstly, for the nonlinear heterogeneous scheduling problem, the overall performance of GHSA_di is tested, compared with very recently published PPADE [27], MOCTS-AI [28] and MaOEA/C [29].

Observation indexes include: security value (See **Eq. (6)**), energy consumption (See **Eq.(1)**), guarantee ratio, and overall system performance (OSP).

Here, guarantee ratio is measured as the proportion of the number of schedulable tasks submitted due to the unsatisfied requirements (such as service delays, load constraints, etc.) and OSP is defined as the product of the normalized security value (See **Eq. (6)**) and guarantee ratio.

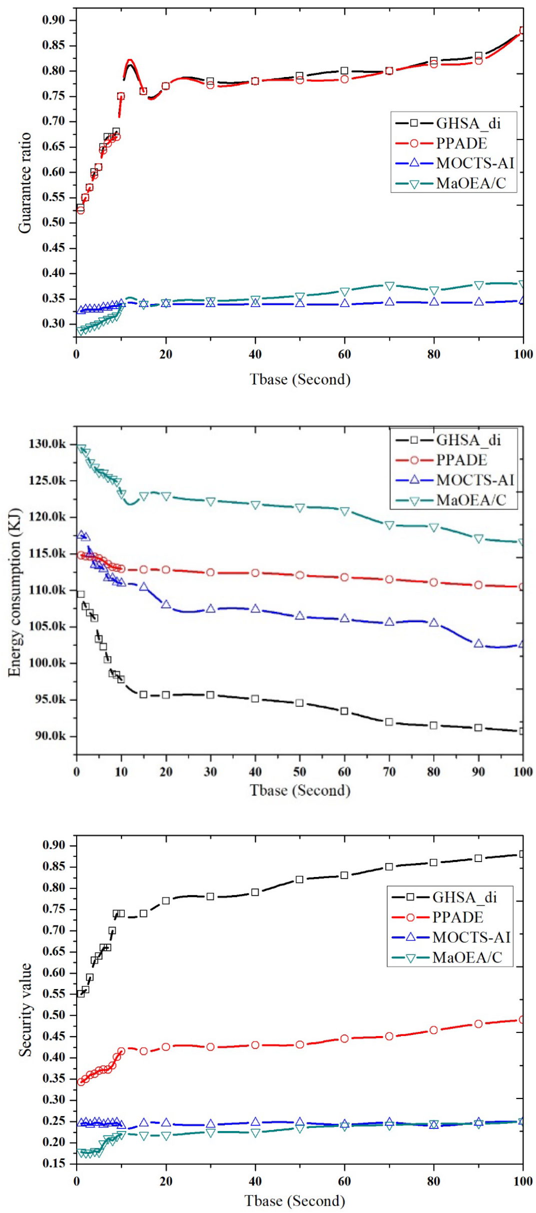

FIGURE 2. **Performance comparison between four meta-heuristics heterogeneous scheduling algorithms.**

Figure. 2 plots the performance impacts on GHSA_di, MaOEA/C [29], MOCTS-AI [28] and PPADE [27], with the deadline base (Tbase) increasing from 1 to 100 seconds.

Shown as **Figure. 2**, with the deadline base (Tbase) increasing from 1 to 100 seconds, GHSA_di is greatly superior to PPADE [27], MOCTS-AI [28]and MaOEA/C [29], in term of saving energy, security and overall system performance (OSP).

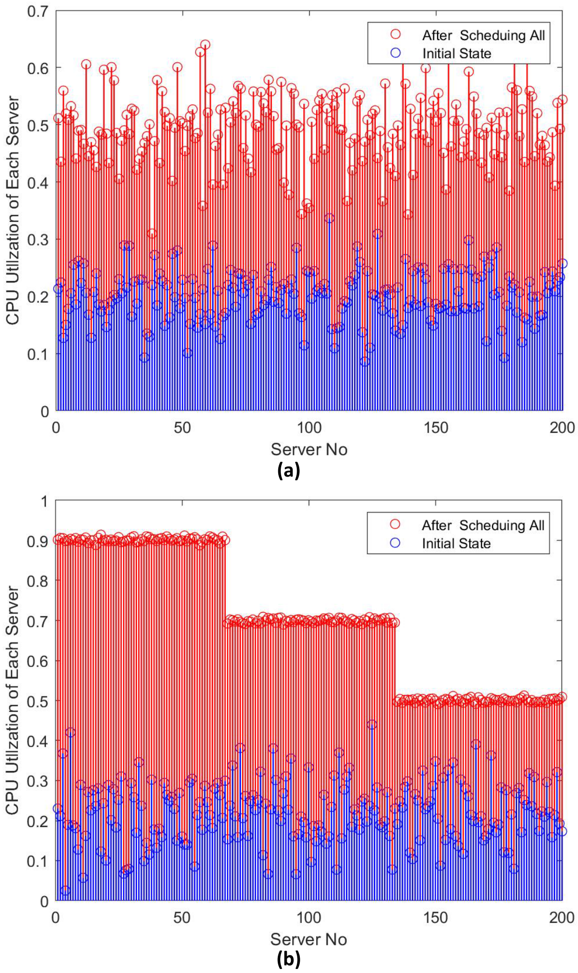

FIGURE 3. Comparison of CPU utilization after scheduling intensive tasks by the different algorithms.

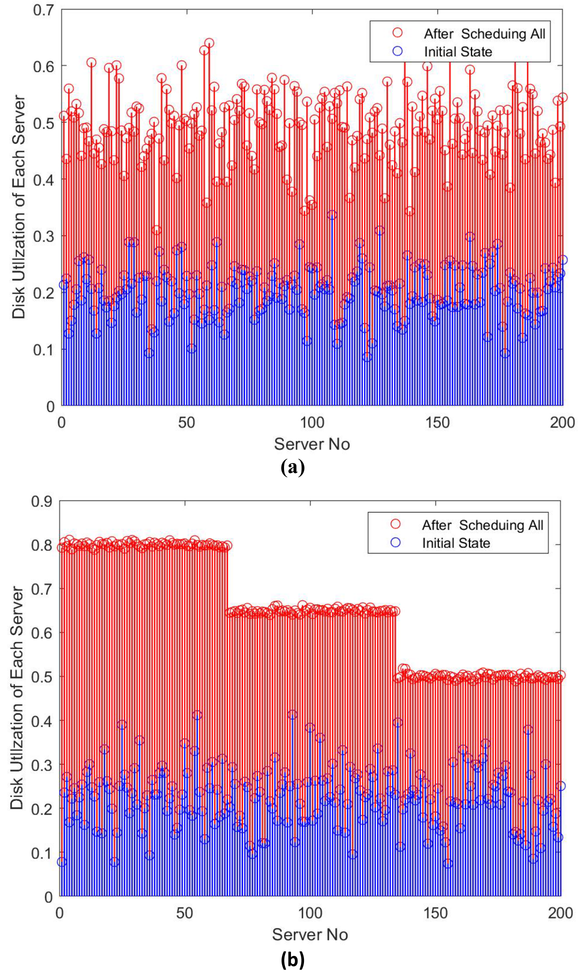

FIGURE 4. Comparison of hard disk utilization after scheduling intensive tasks by the different algorithms.

C. A IMPACT OF INTEGRATION OF ENERGY REGULATION PRINCIPLES ON THE SCHEDULING ALGORITHM

In this subsection, the impact of artificial fusion-intelligence is given, compared with PPADE [27] that shows the approximate guarantee ratio of GHSA_di through the overall performance investigates in the previous subsection.

For intensive tests, each application case is divided into 20000 tasks (***m***=20000), and the number of the virtual machines is 5000.

The utilization changes of CPU/GPU in 200 clusters by different methods: PPADE [27] and GHSA_di, are shown in **Figure. 3**.

As we can see from **Figure. 3**(a), after scheduling computing intensive tasks through PPADE [27], there is 'no visible difference' in the CPU/GPU utilization of 200 clusters although with three common nodes based on energy heterogeneity.

Shown as **Figure. 3**(b), after scheduling through GHSA_di, the CPU/GPU utilization rates of 200 clusters are approximating 0.9, 0.7 and 0.5. As mentioned earlier, the theoretical optimal values of CPU/GPU utilization are {***processor-optim***:[0.8,0.9], ***v***:1}, {***processor-optim***:[0.6, 0.7], ***v***:2} and {***processor-optim***:[0.4,0.5], ***v***:3}. Then, the CPU/GPU final utilization rates are in the scope of the theoretical optimal values; it demonstrates that GHSA_di has the advantages of reasonable deployment due to deep integration of hardware-software energy regulation principles.

For the utilization changes of disks in 200 clusters by different methods: PPADE [27] and GHSA_di, the similar results are shown in **Figure. 4**.

V. CONCLUSION

The super-scale multi-objective optimization problem under strong constraint conditions, such as green heterogeneous scheduling, narrows the bottlenecks of the evolutionary algorithms, such as insufficient dynamics, biomimetic genomes without enough diversity and too slow Cesaro convergence.

An efficient new meta-heuristic scheduling algorithm, i.e., GHSA_di is proposed, including the co-evolutionary dynamics equation emphasizing on and taking advantage of the hardware energy-regulation principles.

The experiment is divided into two parts. Firstly, for the heterogeneous scheduling problem, the overall performance of GHSA_di is tested, compared with very recently published PPADE [27], MOCTS-AI [28] and MaOEA/C [29]; secondly, we evaluate the influence of deep integration of hardware-software energy regulation principles on the evolutionary scheduling algorithm.

Extensive simulator and simulation experiments highlight that GHSA_di has the foreknowledge ability of the dynamic energy feedback after matching deployments, although there are great differences in dynamic power consumption (in W) between the heterogeneous processors even with the same working load. More importantly, they demonstrate that GHSA_di, can not only tighten hardware-software coupling relationships, but also make green space extending in breadth and depth.

REFERENCES

[1] X. Li, P. Garraghan, X. Jiang, Z. Wu, and J. Xu, "Holistic virtual machine scheduling in cloud datacenters towards minimizing total energy," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 6, pp. 1317–1331, Jun. 2018.

[2] L. Belkhir and A. Elmeligi, "Assessing ICT global emissions footprint: Trends to 2040 & recommendations," *J. Cleaner Prod.*, vol. 117, no. 3, pp. 448–463, 2018.

[3] X. Xu, Q. Zhang, S. Maneas, S. Sotiriadis, C. Gavan, and N. Bessis, "VMSAGE: A virtual machine scheduling algorithm based on the gravitational effect for green cloud computing," *Simul. Model. Pract. Theory*, vol. 93, pp. 7–13, May 2019.

[4] J. Chase and D. Niyato, "Joint optimization of resource provisioning in cloud computing," *IEEE Trans. Services Comput.*, vol. 10, no. 3, pp. 396–409, May 2017.

[5] J. Bi, H. Yuan, W. Tan, M. C. Zhou, Y. Fan, J. Zhang, and J. Li, "Application-aware dynamic fine-grained resource provisioning in a virtualized cloud data center," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1172–1183, Apr. 2017.

[6] H. Yuan, J. Bi, M. Zhou, Q. Liu, and A. C. Ammari, "Biobjective task scheduling for distributed green data centers," *IEEE Trans. Autom. Sci. Eng.*, early access, Jan. 7, 2020, doi: 10.1109/TASE.2019.2958979.

[7] J. Guerreiro, A. Ilic, N. Roma, and P. Tomas, "Modeling and decoupling the GPU power consumption for cross-domain DVFS," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 11, pp. 2494–2506, Nov. 2019.

[8] S. Pan and P. K. T. Mok, "A 10-MHz hysteretic-controlled buck converter with single On/Off reference tracking using turning-point prediction for DVFS application," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 66, no. 11, pp. 4502–4515, Nov. 2019.

[9] Z. Ye, S. Mistry, A. Bouguettaya, and H. Dong, "Long-term QoS-aware cloud service composition using multivariate time series analysis," *IEEE Trans. Services Comput.*, vol. 9, no. 3, pp. 382–393, May 2016.

[10] Y.-J. Song, X. Ma, X.-J. Li, L.-N. Xing, and P. Wang, "Learning-guided nondominated sorting genetic algorithm II for multi-objective satellite range scheduling problem," *Swarm Evol. Comput.*, vol. 49, pp. 194–205, Sep. 2019.

[11] H. Liu, B. Liu, H. Zhang, L. Li, X. Qin, and G. Zhang, "Crowd evacuation simulation approach based on navigation knowledge and two-layer control mechanism," *Inf. Sci.*, vols. 436–437, pp. 247–267, Apr. 2018.

[12] S.-Y. Hsieh, C.-T. Chen, C.-H. Chen, T.-H. Yen, H.-C. Hsiao, and R. Buyya, "Novel scheduling algorithms for efficient deployment of MapReduce applications in heterogeneous computing environments," *IEEE Trans. Cloud Comput.*, vol. 6, no. 4, pp. 1080–1095, Oct. 2018.

[13] J. Guerreiro, A. Ilic, N. Roma, and P. Tomás, "DVFS-aware application classification to improve GPGPUs energy efficiency," *Parallel Comput.*, vol. 83, pp. 93–117, Apr. 2019.

[14] J. Sahni and P. Vidyarthi, "A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 2–18, Jan. 2018.

[15] M. B. Karimi, A. Isazadeh, and A. M. Rahmani, "QoS-aware service composition in cloud computing using data mining techniques and genetic algorithm," *J. Supercomput.*, vol. 73, no. 4, pp. 1387–1415, Apr. 2017.

[16] D. B. Prats, J. L. Berral, and D. Carrera, "Automatic generation of workload profiles using unsupervised learning pipelines," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 142–155, Mar. 2018.

[17] Y. Xiong, S. Huang, M. Wu, J. She, and K. Jiang, "A Johnson's-Rule-based genetic algorithm for two-stage-task scheduling problem in data-centers of cloud computing," *IEEE Trans. Cloud Comput.*, vol. 7, no. 3, pp. 597–610, Jul. 2019.

[18] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, "A double deep Q-learning model for energy-efficient edge scheduling," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 739–749, Sep. 2019.

[19] D. Cheng, J. Rao, Y. Guo, C. Jiang, and X. Zhou, "Improving performance of heterogeneous MapReduce clusters with adaptive task tuning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 3, pp. 774–786, Mar. 2017.

[20] Z. Shao, D. Pi, W. Shao, and P. Yuan, "An efficient discrete invasive weed optimization for blocking flow-shop scheduling problem," *Eng. Appl. Artif. Intell.*, vol. 78, pp. 124–141, Feb. 2019.

[21] H. Chen, X. Zhu, G. Liu, and W. Pedrycz, "Uncertainty-aware online scheduling for real-time workflows in cloud service environment," *IEEE Trans. Services Comput.*, early access, Aug. 21, 2018.

[22] H. Chen, X. Zhu, H. Guo, J. Zhu, X. Qin, and J. Wu, "Towards energy-efficient scheduling for real-time tasks under uncertain cloud computing environment," *J. Syst. Softw.*, vol. 99, pp. 20–35, Jan. 2015.

[23] Q. Zhang, M. Lin, L. T. Yang, Z. Chen, S. U. Khan, and P. Li, "A double deep Q-Learning model for energy-efficient edge scheduling," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 739–749, Sep. 2019.

[24] F. Abazari, M. Analoui, H. Takabi, and S. Fu, "MOWS: Multi-objective workflow scheduling in cloud computing based on heuristic algorithm," *Simul. Model. Pract. Theory*, vol. 93, pp. 119–132, May 2019.

[25] Q. Wu, F. Ishikawa, Q. Zhu, and Y. Xia, "Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters," *IEEE Trans. Services Comput.*, vol. 12, no. 4, pp. 550–563, Jul. 2019.

[26] X. Xiao, W. Zheng, Y. Xia, X. Sun, Q. Peng, and Y. Guo, "A workload-aware VM consolidation method based on coalitional game for energy-saving in cloud," *IEEE Access*, vol. 7, pp. 80421–80430, 2019.

[27] Z. He, J. Zhou, H. Qin, B. Jia, and C. Lu, "Long-term joint scheduling of hydropower station group in the upper reaches of the yangtze river using partition parameter adaptation differential evolution," *Eng. Appl. Artif. Intell.*, vol. 81, pp. 1–13, May 2019.

[28] Y. Xie and J. Wu, "Multi-objective constraint task scheduling algorithm for multi-core processors," *Cluster Comput.*, vol. 22, no. 3, pp. 953–964, Sep. 2019.

[29] Q. Lin, S. Liu, K. C. Wong, M. Gong, C. A. C. Coello, J. Chen, and J. Zhang, "A clustering-based evolutionary algorithm for many-objective optimization problems," *IEEE Trans. Evol. Comput.*, vol. 23, no. 3, pp. 391–405, Jun. 2019.

SHAOHUI LI received the B.S. and M.S. degrees in computer science and technology from Shandong Normal University, Jinan, China, in 2002 and 2006, respectively.

He is currently a Teacher with the School of Information Science and Engineering, Shandong Normal University. Since 2009, he has devoted himself to the researches on the multi-disciplinary direction of green computing, heterogeneous multicore super-systems, cloud scheduling middleware, and distributed artificial intelligence. His work has been supported by the National Natural Science Foundation of China and the National High Technology Research and Development Program of China.

HONG LIU received the Ph.D. degree from the Institute of Computer Science, Chinese Academy of Sciences, China, in 1998.

She is currently a Professor and a Doctoral Supervisor with the School of Information Science and Engineering, Shandong Normal University, Jinan, China. She is also the Director of the Key Laboratory of Distributed Computer Software New Technology of Shandong Province, the Information Management Engineering Technology Center of Shandong Province, and the Computer Science and Technology Project with national characteristics. She is mainly involved in the theory and application research of swarm intelligence algorithm and cooperative computer multi-agent system. She has successively presided over six projects supported by the National Natural Science Foundation of China and nine provincial and ministerial projects. She has published more than 200 articles in SCI and EI retrieval and other famous academic journals and international conferences.

BIN GONG received the B.S. and M.S. degrees in computer science and technology from Shandong University, Jinan, China, in 1986 and 1989, respectively, and the Ph.D. degree in computer system architecture from Shandong University.

He is currently a Professor, a Doctoral Supervisor, and the Director of the High Performance Computing Center, Shandong University. He is one of the investigators who have devoted themselves to the researches on supercomputer architecture and high performance computing, since the 1980s in China. His work has been supported by the National High Technology Research and Development Program of China (863 Program).

JINGLIAN WANG received the B.S. and M.S. degrees in computer science and technology from Shandong Normal University, Jinan, China, in 2002 and 2006, respectively, and the Ph.D. degree in computer system architecture from Shandong University, Jinan, in 2015.

She is currently an Associate Professor with the College of Information and Electrical Engineering, Ludong University, Yantai, China. She is mainly involved in the multi-disciplinary direction of green computing, heterogeneous multicore super-systems, cloud scheduling middleware, and distributed artificial intelligence. Her work has been supported by the National Science Foundation for Young Scholars of China and the Talent Introduction Project of Ludong University.