

## Hybridization of Metaheuristics Based Energy Efficient Scheduling Algorithm for Multi-Core Systems

J. Jean Justus<sup>1</sup>, U. Sakthi<sup>2</sup>, K. Priyadarshini<sup>3</sup>, B. Thiyaneswaran<sup>4</sup>, Masoud Alajmi<sup>5</sup>, Marwa Obayya<sup>6</sup> and Manar Ahmed Hamza<sup>7,\*</sup>

<sup>1</sup>Department of Computer Science & Engineering, St. Joseph's College of Engineering, Chennai, 600119, India

<sup>2</sup>Department of Computer Science and Engineering, Saveetha School of Engineering, SIMATS, Chennai, 602 105, India

<sup>3</sup>Department of Electronics and Communication Engineering, K. Ramakrishnan College of Engineering, Tiruchirappalli, 621112, India

<sup>4</sup>Department of Electronics and Communication Engineering, Sona College of Technology, Salem, 636 005, India

<sup>5</sup>Department of Computer Engineering, College of Computers and Information Technology, Taif University, Taif, 21944, Saudi Arabia

<sup>6</sup>Department of Biomedical Engineering, College of Engineering, Princess Nourah bint Abdulrahman University, Riyadh, 11671, Saudi Arabia

<sup>7</sup>Department of Computer and Self Development, Preparatory Year Deanship, Prince Sattam bin Abdulaziz University, AlKharj, Saudi Arabia

\*Corresponding Author: Manar Ahmed Hamza. Email: ma.hamza@psau.edu.sa

Received: 17 November 2021; Accepted: 20 December 2021

**Abstract:** The developments of multi-core systems (MCS) have considerably improved the existing technologies in the field of computer architecture. The MCS comprises several processors that are heterogeneous for resource capacities, working environments, topologies, and so on. The existing multi-core technology unlocks additional research opportunities for energy minimization by the use of effective task scheduling. At the same time, the task scheduling process is yet to be explored in the multi-core systems. This paper presents a new hybrid genetic algorithm (GA) with a krill herd (KH) based energy-efficient scheduling technique for multi-core systems (GAKH-SMCS). The goal of the GAKH-SMCS technique is to derive scheduling tasks in such a way to achieve faster completion time and minimum energy dissipation. The GAKH-SMCS model involves a multi-objective fitness function using four parameters such as makespan, processor utilization, speedup, and energy consumption to schedule tasks proficiently. The performance of the GAKH-SMCS model has been validated against two datasets namely random dataset and benchmark dataset. The experimental outcome ensured the effectiveness of the GAKH-SMCS model interms of makespan, processor utilization, speedup, and energy consumption. The overall simulation results depicted that the presented GAKH-SMCS model achieves energy efficiency by optimal task scheduling process in MCS.

**Keywords:** Task scheduling; energy efficiency; multi-core systems; fitness function; makespan

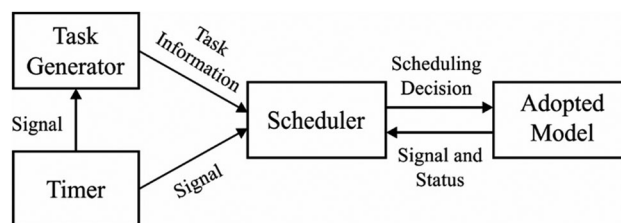


This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1 Introduction

The evolution of Multi-Core Systems (MCS) has been a major advancement for previous models of computer structure. Thus, the merits and demerits of this study pose the significance of the newly proposed method and efficiency in handling thermal dispersion and the inexistence of proficient scheduling approaches. In general, the major theme of a chip is processed in similar clock application, clock frequency, and computing voltage [1]. But the system is not processed in a similar frequency. Hence, retaining the performance symmetry between heterogeneous operating values is a significant problem and developers have managed to resolve the issues. Also, 2 plausible solutions for predefined problems namely, dynamic voltage circuitry for each core (hardware solution), and Schedule tasks between cores to activate the cores to process on similar clock frequency (software solution). The previous compensation principle is shown power exhaustion at maximum frequencies and escalates the thermal throttling. On behalf of the challenging factor, the second solution is unknown from the perception of scheduling [2]. Followed by, the applied workload is divided into cores with the aid of operating on identical cores operating on identical clock frequency for high power conservation. The new processing units offer an interface to modify voltage for optimal energy application. Hence, voltage tailoring on implementation is named Dynamic Voltage Scaling (DVS), which is a productive approach used for reducing energy consumption.

Here, dynamic clock as well as voltage modifications imply the cropping edge of energy reducing abilities in Complementary metal–oxide–semiconductor (CMOS) motherboard [3]. The correlation among frequency and voltage exhibits a base for DVS in smart processors. Hypothetically, a unique processor is a key material that provides frequent voltage levels. But, the application of continuous changing voltages is impossible due to the switching burden and support various operational levels. Hence, the advanced processing units are applicable to support a static value of discrete-level that enhances the speed among traditional and present levels. In [2], it is reported that the energy-speed curve is convex by default. Based on the statement of Jensen's inequality, the deadline limitations are satisfied and energy effective models are used for implementing tasks at ideal speed when compared with differential speed for individual operations. Fig. 1 illustrates the common process of scheduling system.



**Figure 1:** General process of scheduling

Furthermore, the simulation outcome is relevant to explore the feasibility of computing equal speed for the cores by using a processor that supports the maximum count of discrete power voltage levels. The first priority of a process denotes the concentration provides to the task while making scheduling decisions. Usually, real-time applications were not applied completely. Hence, the systems are challenging to use DVS and its scheduling methods. Under the application of the DVS model, complete maintenance has to be provided for task scheduling and numerous outcomes are attained [4].

The prominently applied policy for scheduling real-time process is a priority-based module which has been categorized into 2 classes namely, Fixed priority and Dynamic priority [3]. The first approach allocates a static value for the priority of all processes in a task, which has to be varied from the priorities allocated to

jobs emanated by alternate tasks in the system. The second module has no limitations where the priorities are allocated to a single process. Even though dynamic approaches are assumed theoretically, it is unpredictable if a transient burden exists [5].

This paper presents a new hybrid genetic algorithm (GA) with a krill herd (KH) based energy-efficient scheduling technique for multi-core systems (GAKH-SMCS). GA is an evolutionary algorithm using the searching approaches for the identification of nearest optimal solutions. To resolve the issue of ineffective information flow between different generations in GA, the KH algorithm is incorporated into GA by altering the GA cycle and operators, appending swarm intelligence (SI) and rousing from Krill movements. The GAKH-SMCS model involves a multi-objective fitness function using four parameters such as makespan, processor utilization, speedup, and energy consumption to schedule tasks proficiently. The performance of the GAKH-SMCS model has been validated against two datasets namely random dataset and benchmark dataset. In short, the paper contributions can be summarized as follows

- Design an energy-efficient task scheduling algorithm using a hybrid metaheuristic technique for MCS
- Hybridizing GA with KH algorithms for improving the information flow among generations in GA
- Derive a multi-objective fitness function using four variables namely
- Validate the results of the GAKH-SMCS model against random and benchmark dataset.

The remaining sections are planned as follows. Section 2 briefs the existing scheduling techniques developed for MCS. Next, the proposed GAKH-SMCS model is designed in Section 3. At last, Section 4 performs the simulation process and Section 5 concludes the work.

## 2 Existing Scheduling Approaches for MCS

Most proposals have reported task scheduling for MCS with realistic limitations. Increasing the proficiency of theoretical applications in real-time overhead is projected in [4] to schedule the periodic soft real-time jobs. In the case of periodic application, Moulik et al. [5] presented a variable energy-aware real-time scheduling device with 3 levels of hierarchical resource assignment. Initially, the scheduler determines a pair of fragments for execution and schedules the task for enabling better execution value, and manufacture the processing frequencies for each core and reduce the power utilization. Next, the extended approach [6] has considered the system-wide power application for considering the best frequency selection technique. Chwa et al. [7] applied fully-migrative frameworks of 2-type heterogeneous MCS named Hetero-Fair.

Tang et al. [8] developed a task scheduling method for combining hard periodic as well as soft aperiodic real-time jobs. Firstly, a periodic operation is scheduled by offline technique and periodic tasks are scheduled dynamically with the help of residual slack time data for a resource. Lin et al. [9] projected a processor mapping approach according to the Integer Linear Programming (ILP) mechanism for practical time streaming models. To serve the global optimization of throughput, delay, and computational cost, a global ILP approach has been deployed and suitable solutions are identified with the help of a GA. In contrast, the developed approach assumes real-time features of the method and power attributes of task requests and adverts. At this point, dynamic voltage/frequency scaling (DVFS) is applied for limiting the power application and sample scheduling methods practically with the help of the benchmark system. Hence, task scheduling models consider that energy is applied for heterogeneous MCS. Next, power reducing processor allocation for the periodic real-time process was projected [10], in which the solution for the ILP approach is retrieved using greedy methods and reduces power utilization.

A task scheduling framework is developed in [11] that assumes the implementation time for process and power utilization in the ILP scheme. Here, a task is labeled as graphical form and it can be allocated to maximum priority level with limited execution time and power application. Yu et al. [12] proposed an

ILP-based static resource allocation approach in which the voltage levels of computing elements were assumed to be the attribute in ILP problem formulation for power minimization. Zhang et al. [13] executed Shuffled Frog Leaping Algorithm (SFLA) for realistic periodic task scheduling in which the objective is to reduce the power utilization during the deadline completion. Hung et al. [14] define an energy-effective scheduling framework for the DVS model and non-DVS processing units, in which the objectives are reducing power consumption and enhance energy savings. Followed by, power consumption is classified by 2 diverse cases in which the non-DVS processing element is either based on burden or non-burden. Thus, the newly developed approaches are approximation models that depend upon ILP.

Also, Baruah et al. [15] deployed an ILP development-based dividing framework for limited-deadline processes on heterogeneous processing units. Awan et al. [16] reported task assignment on DVFS based heterogeneous MCS that reduces the total power utilization of the system considering draining energy and dynamic energy. Afterward, a task-to-core mapping approach has been depicted and evaluate the sleep condition as well as processing frequency-level of a core for the autonomous sporadic process. Lin et al. [17] presented an energy-effective task scheduling model which is operated under 2 execution models namely, Batch mode and Online mode. Initial mode, a greedy based application has been applied for reducing the overall cost in which the cost is composed of time and power utilization, computational rate, and money spent for waiting by individuals while in the second mode, a novel task has been produced typically and developed to the model. The tasks go to 2 types of online mode: Interactive and Non-interactive operations. Initially, interactive tasks require limited response time while the non-interactive process does not consider the response time. Hence, a method for online mode is employed for completing the interactive operations with limited response time and to reduce the non-interactive task implementation duration.

### 3 Proposed GAKH Based Scheduling Technique for MCS

GA is an evolutionary algorithm using the searching approaches for the identification of nearest optimal solutions. To resolve the issue of ineffective information flow between different generations in GA, the KH algorithm is incorporated to GA by altering the GA cycle and operators, appending SI and rousing from Krill movements. The proposed GAKH-SMCS algorithm utilizes a multi-objective fitness function using four parameters for the optimal task scheduling process.

#### 3.1 Problem Statement

In this study, a pair of  $n$  tasks,  $\theta = \{\theta_1, \theta_2, \theta_3, \dots, \theta_n\}$  has to be scheduled on various  $m$  processors, which means that  $\lambda = \{\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_m\}$ . The communication channels are considered as contention free. The tasks are delivered to the processing unit and scheduled by using the scheduling approach. Then, the reported issues are formulated employing Linear Programming Problem (LPP). Assume  $\alpha_{ir}$  is a Boolean variable depicted as given below:

$$\alpha_{ir} = \begin{cases} 1, & \text{if } \theta_i \text{ is assigned to } \lambda_r \\ 0, & \text{Otherwise.} \end{cases} \quad (1)$$

Followed by, LPP is devised as shown below:

$$\text{Minimize } F_1 = \text{Max}\{FT(\theta_i) | \forall \theta_i \in \theta\}$$

$$\text{Maximize } F_2 = UT$$

$$\text{Maximize } F_3 = SP$$

Subjected to

$$\sum_{r=1}^m \alpha_{ir} = 1, \forall i, 1 \leq i \leq n \quad (2)$$

The following units are employed to define the newly projected approach:

The implementation time  $\xi T(\theta_i, \lambda_r)$  of task  $\theta_i$  on processor  $\lambda_r$  is determined by a given expression,

$$\xi T(\theta_i, \lambda_r) = \frac{\omega_i}{f_r^{op}} \quad (3)$$

where  $\omega_i$  denotes the instruction size of a task and  $\theta_i$  as well as  $f_r^{op}$  defines the processing frequency of the processor  $\lambda_r$ .

The original start time,  $AST(\theta_i, \lambda_r)$  of  $\theta_i$  on processor  $\lambda_r$  refers to the task initialized from a processor in respective time.

The releasing time of a processor  $\lambda_r$  is allocated as 0. Once the task has been executed for task  $\theta_i$ , it is measured by the given function:

$$RT(\lambda_r) = RT(\lambda_r) + \xi T(\theta_i, \lambda_r) \quad (4)$$

The finish time  $FT(\theta_i)$  refers to the total implementation time and start time of the tasks  $\theta_i$  on processor  $\lambda_r$  that is expressed by:

$$FT(\theta_i) = \xi T(\theta_i, \lambda_r) + AST(\theta_i, \lambda_r) \quad (5)$$

### 3.2 Hybridization of GA with KH Algorithm

In traditional GA, considering the absence of robust data flow among parents and children generations, the attained results are inferior in a few issues. In the newly developed GAKH technique, reasonable enhancement is accomplished by combining GA and KH methods. The major cause for presenting the change is that the GA approach does not provide optimal outcomes when compared with alternate approaches as KH lacks robust data flow and KH genetic operators are unfit in exhibiting the final outcomes. The major intension of this framework is to resolve the vulnerability of KH and GA and solve the clustering issues by KH [18]. The steps involved in newly presented approaches are given in the following.

- (1) Initialization: Develop a population randomly in a search space.
- (2) Fitness estimation: Estimating the fitness of a candidate.
- (3) Motion Operators:
  - a. Cumulative
  - b. Local
  - c. Random
- (4) Substitution
- (5) Maximize the population
- (6) Follow step 3 until reaching the termination criteria
- (7) End.

Step 3: Fitness estimation

Step 4: Under the application of GA operators and including self-organization, parent data flow, the maximum and minimum objective functions and final outcome of existing generation for the operators

and change the population development in GA and the final generations offer considerable outcomes when compared with former generations. Obviously, a chromosome resembles the krill's hierarchy by the movements defined above.

Cumulative movement. It is referred to as krill based movement. It manages to retain maximum density and go nearby group's maximum collection.

Suppose the operator  $\otimes$  implies cross-over since the parent chromosomes are assumed to be inputs and generates 2 children as result and assume the operator  $\ominus$  is a cross-over as it concerns 2 parent chromosomes as inputs and exhibits a child as simulation outcome. Operator "+" denotes the election of the maximum quality chromosome between the previous chromosomes. Additionally,  $ch_i$ ,  $ch_j$ ,  $ch_{best}$ ,  $ch_{worst}$ ,  $ch_{i,best}$ ,  $ch_{i,worst}$  means the parent chromosome 1, parent chromosome 2, best chromosome in the generation (best global), worst chromosome (worst global), optimal past of chromosome i (best local) as well as worst past of chromosome i (worst local). This movement is accomplished by considering relationships.

In cross-over operator is used on 2 chromosomes i and j utilizing Eq. (6).

$$x_{i,j} = \begin{cases} x_{of\ spring1,2}, & if\ rand_{i,j} < Cr \\ x_{i,j} & else \end{cases} \quad (6)$$

Where rand value of considered from [0, 1] and Cr denotes the cross-over value. The cross-over operator is employed on 3 chromosomes i, j, and k interms of Eq. (7).

$$x_{i,j,k} = \begin{cases} x_{ofspring1,2,3}, & if\ rand_{i,j,k} < Cr \\ x_{i,j,k} & else \end{cases} \quad (7)$$

where rand is among [0, 1] and Cr defines the cross-over rate.

$$X = (ch_i \otimes ch_j) + (ch_{best} \otimes ch_{worst}) \quad (8)$$

$$Y = (ch_i \otimes ch_j \otimes ch_{best}) + (ch_i \otimes ch_j \otimes ch_{worst}) \quad (9)$$

$$A = (ch_i \ominus ch_{best}) + (ch_i \ominus ch_{worst}) \quad (10)$$

$$B = (ch_j \ominus ch_{best}) + (ch_j \ominus ch_{worst}) \quad (11)$$

$$K = (ch_i \otimes ch_{i,best}) + (ch_j \otimes ch_{j,best}) \quad (12)$$

$$Z = (ch_i \otimes ch_{i,worst}) + (ch_j \otimes ch_{j,worst}) \quad (13)$$

$$N = \alpha X + \beta Y + \gamma A + \delta B + \eta K + \vartheta Z \quad (14)$$

where N defines the cumulative movement as well as binary coefficients implies the selection and lack of item selection.

The local movement is evolved by the foraging behavior of krill. Here, krill from a former place of food seeks food. Operator mutation is  $\oplus$  considered as chromosomes as inputs and generates a novel chromosome by applying Eq. (16).

$$ch_{i,j} = \begin{cases} ch_{r,j} & if\ rand_{i,j} < Mu \\ ch_{i,j} & else \end{cases} \quad (15)$$

$$F = ch_i \oplus ch_{best} \oplus ch_{i,best} \quad (16)$$

where  $Mu$  denotes the mutation value.

Random diffusion is developed by the random movement of krill. During this approach, the presented approach every generation generates  $D^{max}$  novel chromosomes and includes them to the generation. Followed by, chromosomes are decided depends on quality and population size. The significant point is a value of random movements while the evolutionary model of method depends upon the given function reduces by using Eq. (17).

$$D_i = D^{max} \left( 1 - \frac{1}{I_{max}} \right) \tag{17}$$

where  $i$  denotes the generation value and  $I_{max}$  defines the numerous generations. The score of  $i$  is a significant parameter and depends upon the performance. Fig. 2 illustrates the flowchart of KH model.

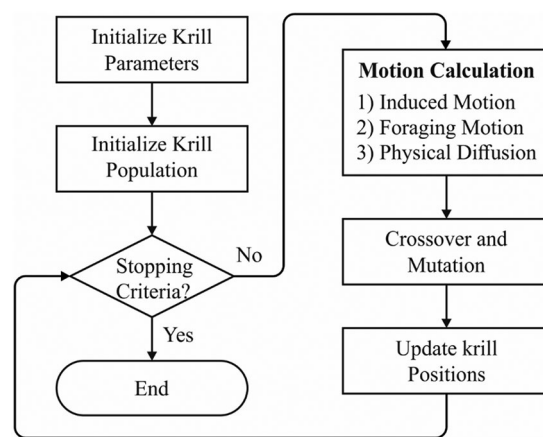


Figure 2: The flowchart of KH algorithm

### 3.3 Application of GAKH Algorithm for Task Scheduling in MCS

The projected model named GAKH based scheduling in MCS is composed of the objectives given in the following.

#### 3.3.1 Representing Chromosomes

Chromosomes must offer better solutions to the applied problems. In this model, they are defined as a string of arbitrarily created processor value. The basic population is developed by arbitrarily produced  $P_{SIZE}$  of chromosome, in which  $P_{SIZE}$  implies a population size. Consider the system with a collection of tasks  $\theta = \{\theta_1, \theta_2, \theta_3, \dots, \theta_{10}\}$  and 3 processors  $\lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ . It is monitored that 2nd position of a gene is 1, which refers that  $\theta_2$  is allocated to  $\lambda_1$ . Thus,  $\theta_1, \theta_3$ , and  $\theta_5$  is allocated to  $\lambda_2, \lambda_3$ , and  $\lambda_2$  correspondingly.

#### 3.3.2 Fitness Function

For deriving the fitness score, 3 objectives have been applied namely, reduction of makespan, increment in processor application, and improvisation of speedup value. In this model, a chromosome is estimated using a fitness score and identifies an optimal solution for the applied problem.

Reduction of Makespan: A task  $\theta_i \in \theta$  has to be allocated to the processors and the execution is completed within a limited duration. Hence, the primary objective is defined in the following:

Objective 1( $O_1$ ) : Minimize( $Mks$ ) (18)

where  $Mks = Max (FT(\theta_1), FT(\theta_2), FT(\theta_3) \dots FT(\theta_n))$ .

Increment of Processor application: It is considered as the ratio among average time where the processors are engaged by complete system schedule time and determined as given below:

$$UT = \frac{1}{Mks} \times \frac{\sum_{r=1}^m \xi T(\lambda_r)}{m} \quad (19)$$

Then, the second objective is recommended as:

$$\text{Objective 2}(O_2) : \text{Maximize } (UT) \quad (20)$$

Improvisation of Speedup: A speedup factor is considered as the ratio among sequential implementation time by makespan as depicted in the following:

$$SP = \frac{\sum_{i=1}^n \xi T(\theta_i, \lambda_r)}{Mks} \quad (21)$$

$$\text{Objective 3}(O_3) : \text{Maximize } (SP) \quad (22)$$

Reduction of energy consumption (EC): It is defined as the amount of energy spent for communication purposes.

$$EC = E_T + E_R$$

$$\text{Objective 4}(O_4) : \text{Minimize } (EC) \quad (23)$$

Here, the Weighted Sum Approach (WSA) has been applied for computing Fitness Function (FF). WSA is defined as a traditional application used to optimize the MCS by task scheduling problem, in which the objective is allocated with weight [19]. The main aim of this framework is to select the solution with a reduced fitness score. Assume the parameter weight as  $\beta_1, \beta_2, \beta_3, \beta_4$  correspondingly, in which  $\sum_{j=1}^4 \beta_j = 1$ . The FF of the newly developed process is shown below:

$$\text{Fitness} = \beta_1 \times Mks + \beta_2 \times (1 - UT) + \beta_3 \times (1 - SP) + \beta_4 \times EC \quad (24)$$

The major objective is to reduce fitness value. In a crossover, 2 parent chromosomes combine and generate 2 child chromosomes. There are maximum crossover operations such as 1-point, 2-point, hybrid, and so on. Hence, the crossover process is processed after mutation, in which child chromosomes undergo mutation for a better solution. Under the mutation process, the gene position is selected arbitrarily and the value is interchanged by the alternate valuable score. In the case of the selection process, optimal chromosomes are decided based on fitness measures.

#### 4 Performance Validation

To examine the performance of the GAKH-SMCS model, a series of simulations were takes place on Intel i7 3.4 GHz CPU with 4 GB of RAM. To select the parents depending upon the fitness value, Roulette Wheel Selection is used. Also, the weighted sum approach (WSM) is used to determine the fitness of the presented model. The results are examined using a random and benchmark dataset [20]. A comparative analysis with Priority-aware Genetic Algorithm (PGA), hybrid heuristicgenetic algorithm with adaptive parameter (HGAAP), GA based scheduling (GA-SC), and Genetic Algorithm in Heterogeneous Distributed Computing System (GAHDCS).



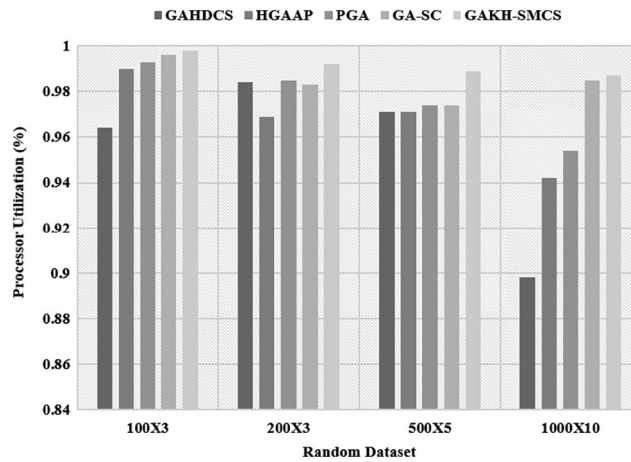
#### 4.1 Analysis of Scheduling Results on Random Dataset

The GAKH-SMCS model is tested against 4 arbitrarily created datasets such as Set 1: 100 tasks with 3 processors ( $100 \times 3$ ), Set 2: 200 tasks with 3 processors ( $200 \times 3$ ), Set 3: 500 tasks with 5 processors ( $500 \times 5$ ), and Set 4: 1000 tasks with 10 processors ( $1000 \times 10$ ). In this case, the task sizes are arbitrarily created and utilized the weights of 0.5, 0.3, 0.2 to determine the fitness value. Tab. 1 and Figs. 3, 4 showcases the results obtained by the GAKH-SMCS model on the applied random dataset for distinct measures.

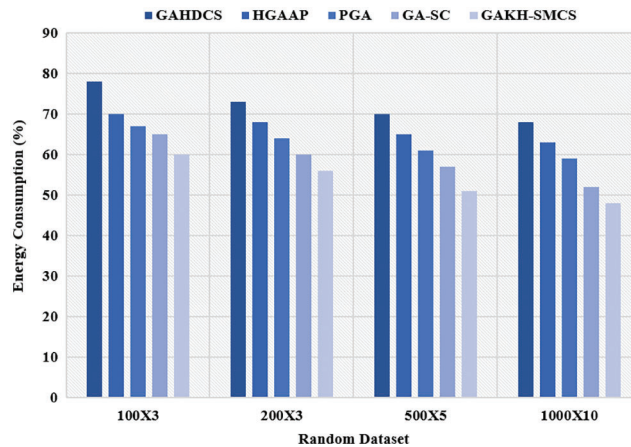
**Table 1:** Results analysis on random dataset in terms of makespan, utilization, speedup ratio, and energy consumption

Makespan					
Methods	GAHDCS	HGAAP	PGA	GA-SC	GAKH-SMCS
Set-1 (100X3)	285.563	280.799	299.855	261.743	255.511
Set-2 (200X3)	576.166	576.166	590.458	557.110	546.166
Set-3 (500X5)	933.466	938.230	1000.162	933.466	913.454
Set-4 (1000X10)	966.814	962.050	976.342	962.050	936.801
Processor Utilization					
Methods	GAHDCS	HGAAP	PGA	GA-SC	GAKH-SMCS
Set-1 (100X3)	0.964	0.990	0.993	0.996	0.998
Set-2 (200X3)	0.984	0.969	0.985	0.983	0.992
Set-3 (500X5)	0.971	0.971	0.974	0.974	0.989
Set-4 (1000X10)	0.898	0.942	0.954	0.985	0.987
Speedup Ratio					
Methods	GAHDCS	HGAAP	PGA	GA-SC	GAKH-SMCS
Set-1 (100X3)	2.827	2.753	2.902	2.903	2.932
Set-2 (200X3)	2.827	2.790	2.865	3.124	3.458
Set-3 (500X5)	4.755	4.792	4.792	4.903	4.986
Set-4 (1000X10)	8.906	9.314	9.499	9.610	9.873
Energy Consumption (%)					
Methods	GAHDCS	HGAAP	PGA	GA-SC	GAKH-SMCS
Set-1 (100X3)	78.00	70.00	67.00	65.00	60.00
Set-2 (200X3)	73.00	68.00	64.00	60.00	56.00
Set-3 (500X5)	70.00	65.00	61.00	57.00	51.00
Set-4 (1000X10)	68.00	63.00	59.00	52.00	48.00

An analysis of makespan results indicated that the GAHDCS model has depicted poor performance by obtaining maximum makespan over the compared methods. Besides, the HGAAP and PGA models have resulted in a slightly better makespan over the GAHDCS model. Though the GA-SC model has showcased competitive performance, the presented GAKH-SMCS model has outperformed the other methods by achieving minimum makespan. For instance, on the applied set-4 ( $1000 \times 10$ ), the presented GAKH-SMCS model reached a minimum makespan of 936.801 whereas the GAHDCS, HGAAP, PGA, and GA-SC models have attained higher makespan of 285.563, 280.799, 299.855, and 261.743 respectively.



**Figure 3:** Processor utilization analysis on random dataset



**Figure 4:** Energy efficiency analysis on random dataset

On determining results to processor utilization, it is evident that the GAHDCS model is the worst performer over the compared methods by offering the least processor utilization whereas the HGAAP, PGA, and GA-SC models have exhibited slightly better utilization over the GAHDCS. However, the presented GAKH-SMCS model has demonstrated supreme results by obtaining maximum processor utilization. For instance, on the applied Set-4 ( $1000 \times 10$ ) dataset, the presented GAKH-SMCS model has depicted higher processor utilization of 0.998 whereas the GAHDCS, HGAAP, PGA, and GA-SC models have demonstrated lower processor utilization of 0.898, 0.942, 0.954, and 0.985 respectively.

The analysis of power consumption results depicted that the GAHDCS method has demonstrated inferior performance by accomplishing high power consumption over the traditional approaches. Followed by, the HGAAP and PGA methodologies have exhibited considerable power consumption than the GAHDCS model. Even though the GA-SC technique has implied a competing function, the proposed GAKH-SMCS framework has surpassed the alternate models by accomplishing lower power utilization. For sample, on the given set-4 ( $1000 \times 10$ ), the newly GAKH-SMCS deployed technology has attained the least power consumption of 48% while the GAHDCS, HGAAP, PGA, and GA-SC methodologies have accomplished maximum power consumption of 68%, 63%, 59%, and 52% correspondingly.

On examining the final outcome utilizing the speedup ratio, it is clear that the GAHDCS method has identified inferior performers when compared with other modes by exhibiting a lower speedup ratio while the HGAAP, PGA, and GA-SC methodologies have showcased moderate utilization over the GAHDCS. Therefore, the developed GAKH-SMCS framework has depicted qualified outcomes by attaining a high speedup ratio. For example, on the given Set-4 ( $1000 \times 10$ ) dataset, the projected GAKH-SMCS framework has illustrated a maximum speedup ratio of 0.998 while the GAHDCS, HGAAP, PGA, and GA-SC frameworks have showcased limited speedup ratio of 8.906, 9.314, 9.499, and 9.610 respectively.

#### 4.2 Analysis of Scheduling Results on the Benchmark Dataset

In this study, 2 datasets 256 tasks on 8 processors ( $256 \times 8$ ) and 512 tasks on 16 processors ( $512 \times 16$ ) were assumed to verify the process [20]. The samples are represented as  $u\_x\_yyzz$  where  $u$  implies samples produced by the uniform distribution,  $x$  signifies the consistency,  $yy$  indicates heterogeneity, and  $zz$  depicts the heterogeneity. Here, the performance is validated interms of distinct evaluation measures.

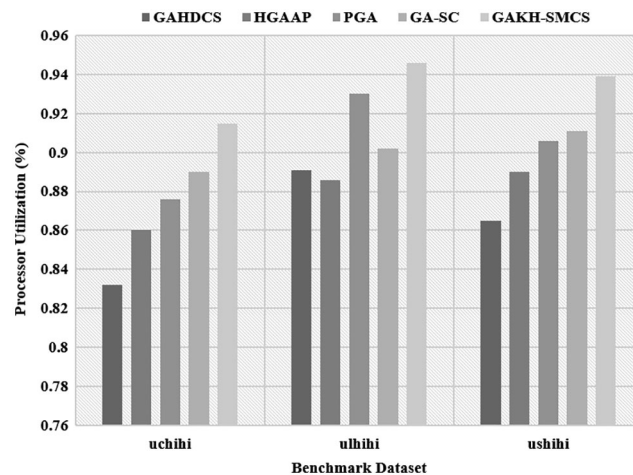
Tab. 2 and Figs. 5, 6 implies the results attained by the GAKH-SMCS method on the given benchmark dataset employing various metrics. An analysis of makespan results pointed out that the GAHDCS approach has illustrated inferior performance by gaining high makespan over the traditional models. On the other hand, the HGAAP and PGA approaches have concluded with acceptable makespan over the GAHDCS scheme. Even though the GA-SC framework has showcased competing performance, the projected GAKH-SMCS model has performed well with the alternate models by accomplishing the least makespan. For sample, on the given ushihi dataset, the newly GAKH-SMCS approach attained a low makespan of 24768088 and the GAHDCS, HGAAP, PGA, and GA-SC models have obtained optimal makespan of 27234317, 25499658, 27147584, and 26106788 respectively.

**Table 2:** Results analysis on benchmark dataset in terms of makespan, utilization, speedup ratio, and energy consumption

Makespan			
Methods	uchihi	ulhihi	ushihi
GAHDCS	25673124	25326192	27234317
HGAAP	24892527	24632328	25499658
PGA	24458862	27234317	27147584
GA-SC	24198664	24979260	26106788
GAKH-SMCS	24084653	24174687	24768088
Processor Utilization			
Methods	uchihi	ulhihi	ushihi
GAHDCS	0.832	0.891	0.865
HGAAP	0.860	0.886	0.890
PGA	0.876	0.930	0.906
GA-SC	0.890	0.902	0.911
GAKH-SMCS	0.915	0.946	0.939

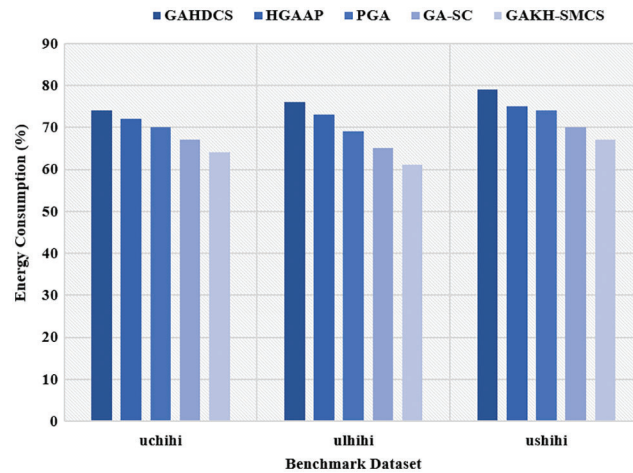
(Continued)

Table 2 (continued)			
Speedup Ratio			
Methods	uchihi	ulhihi	ushihi
GAHDCS	6.605	7.084	6.765
HGAAP	6.765	7.020	7.020
PGA	6.924	7.339	7.147
GA-SC	7.084	7.147	7.211
GAKH-SMCS	7.264	7.893	7.682
Energy Consumption (%)			
Methods	uchihi	ulhihi	ushihi
GAHDCS	74.00	76.00	79.00
HGAAP	72.00	73.00	75.00
PGA	70.00	69.00	74.00
GA-SC	67.00	65.00	70.00
GAKH-SMCS	64.00	61.00	67.00



**Figure 5:** Processor utilization analysis on benchmark dataset

On examining the final outcome utilizing processor utilization, it is clear that the GAHDCS approach has been meant to be the worst performer over the earlier models by generating minimum processor utilization while the HGAAP, PGA, and GA-SC models have depicted moderate application when compared with GAHDCS. Therefore, the projected GAKH-SMCS framework has depicted qualified results by gaining higher processor utilization. For sample, on the applied ushihi dataset, the projected GAKH-SMCS approach has showcased maximum processor utilization of 0.939 whereas the GAHDCS, HGAAP, PGA, and GA-SC models have demonstrated least processor utilization of 0.865, 0.890, 0.906, and 0.911 respectively.



**Figure 6:** Energy efficiency analysis on benchmark dataset

An examination of energy consumption results pointed out that the GAHDCS method has illustrated inferior performance by achieving high energy consumption over the classical models. Also, the HGAAP and PGA frameworks have attained moderate energy consumption over the GAHDCS scheme. Although the GA-SC technique has implied competing performance, the proposed GAKH-SMCS model has surpassed the alternate methods by accomplishing lower energy consumption. For example, on the applied ushihi dataset, the presented GAKH-SMCS model reached a limited energy consumption of 67% whereas the GAHDCS, HGAAP, PGA, and GA-SC models have attained maximum energy consumption of 78%, 75%, 74%, and 70% respectively.

On examining results for speedup ratio, it is apparent that the GAHDCS method has been identified to be the poor performer over the classical models by providing minimum speedup ratio while the HGAAP, PGA, and GA-SC methodologies have depicted acceptable consumption over the GAHDCS. Thus, the projected GAKH-SMCS framework has illustrated qualified results by gaining a high speedup ratio. For sample, on the applied ushihi dataset, the presented GAKH-SMCS technology has shown a maximal speedup ratio of 7.682 whereas the GAHDCS, HGAAP, PGA, and GA-SC technologies have illustrated least speedup ratio of 6.765, 7.020, 7.147, and 7.211 respectively.

## 5 Conclusion

This paper has designed an energy-efficient task scheduling algorithm using GAKH-SMCS. The goal of the GAKH-SMCS technique is to derive scheduling tasks in such a way to achieve faster completion time and minimum energy dissipation. The GAKH algorithm is introduced to resolve the problem of ineffective information flow between different generations in GA by altering the GA cycle and operators, appending SI, and rousing from Krill movements. The GAKH-SMCS model comprises a multi-objective fitness function by the use of four parameters such as makespan, processor utilization, speedup, and energy consumption to schedule tasks competently. To validate the performance of the GAKH-SMCS model, a series of experiments were conducted on the random dataset and benchmark dataset. The experimental results depicted that the presented GAKH-SMCS model achieves energy efficiency by optimal task scheduling process in MCS. As a part of the future scope, the performance of the GAKH-SMCS model can be improved using optimal data aggregation techniques.

**Funding Statement:** This work was supported by Taif University Researchers Supporting Program (Project Number: TURSP-2020/195), Taif University, Saudi Arabia. Princess Nourah bint Abdulrahman University

Researchers Supporting Project number (PNURSP2022R203), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] X. Huang, K. L. Li and R. Li, "A energy efficient scheduling base on dynamic voltage and frequency scaling for multi-core embedded real-time system," in *Algorithms and Architectures for Parallel Processing, Lecture Notes in Computer Science book series*, Springer, Switzerland, vol. 5574, pp. 137–145, 2009.
- [2] T. Ishihara and H. Yasuura, "Voltage scheduling problem for dynamically variable voltage processors," in *Proc. of the 1998 Int. Symp. on Low Power Electronics and Design-ISLPED '98*, Monterey, California, United States, pp. 197–202, 1998.
- [3] J. W. S. Liu, *Real Time Systems*. Upper Saddle River: Prentice Hall, 2000.
- [4] J. M. Calandrino, D. Baumberger, T. Li, S. Hahn and J. H. Anderson, "Soft real-time scheduling on performance asymmetric multicore platforms," in *13th IEEE Real Time and Embedded Technology and Applications Symp. (RTAS'07)*, Bellevue, WA, USA, pp. 101–112, 2007.
- [5] S. Moulik, R. Devaraj and A. Sarkar, "HEART: A heterogeneous energy-aware real-time scheduler," in *2019 32nd Int. Conf. on VLSI Design and 2019 18th Int. Conf. on Embedded Systems (VLSID)*, Delhi, NCR, India, pp. 476–481, 2019.
- [6] S. Moulik, R. Devaraj and A. Sarkar, "HEALERS: a heterogeneous energy-aware low-overhead real-time scheduler," *IET Computers & Digital Techniques*, vol. 13, no. 6, pp. 470–480, 2019.
- [7] H. S. Chwa, J. Seo, J. Lee and I. Shin, "Optimal real-time scheduling on two-type heterogeneous multicore platforms," in *2015 IEEE Real-Time Systems Symp.*, San Antonio, Texas, pp. 119–129, 2015.
- [8] H. Tang, P. Ramanathan and K. Compton, "Combining hard periodic and soft aperiodic real-time task scheduling on heterogeneous compute resources," in *2011 Int. Conf. on Parallel Processing*, Taipei City, Taiwan, pp. 753–762, 2011.
- [9] J. Lin, A. Srivatsa, A. Gerstlauer and B. L. Evans, "Heterogeneous multiprocessor mapping for real-time streaming systems," in *2011 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Czech Republic, pp. 1605–1608, 2011.
- [10] J. Chen, A. Schranzhofer and L. Thiele, "Energy minimization for periodic real-time tasks on heterogeneous processing units," in *2009 IEEE Int. Symp. on Parallel & Distributed Processing*, Rome, pp. 1–12, 2009.
- [11] L. Wenjing and W. Lisheng, "Energy-considered scheduling algorithm based on heterogeneous multi-core processor," in *2011 Int. Conf. on Mechatronic Science, Electric Engineering and Computer (MEC)*, Jilin, China, pp. 1151–1154, 2011.
- [12] Y. Yu and V. K. Prasanna, "Power-aware resource allocation for independent tasks in heterogeneous real-time systems," in *Proc. Ninth Int. Conf. on Parallel and Distributed Systems, 2002*, Taiwan, China, pp. 341–348, 2002.
- [13] W. Zhang, E. Bai, H. He and A. Cheng, "Solving energy-aware real-time tasks scheduling problem with shuffled frog leaping algorithm on heterogeneous platforms," *Sensors*, vol. 15, no. 6, pp. 13778–13804, 2015.
- [14] C. Hung, J. Chen and T. Kuo, "Energy-efficient real-time task scheduling for a dvs system with a non-dvs processing element," in *2006 27th IEEE Int. Real-Time Systems Symp. (RTSS'06)*, Rio de Janeiro, pp. 303–312, 2006.
- [15] S. K. Baruah, V. Bonifaci, R. Bruni and A. M. Spaccamela, "ILP models for the allocation of recurrent workloads upon heterogeneous multiprocessors," *Journal of Scheduling*, vol. 22, no. 2, pp. 195–209, 2019.
- [16] M. A. Awan, P. M. Yomsi, G. Nelissen and S. M. Petters, "Energy-aware task mapping onto heterogeneous platforms using DVFS and sleep states," *Real-Time Systems*, vol. 52, no. 4, pp. 450–485, 2016.
- [17] C. C. Lin, Y. C. Syu, C. J. Chang, J. J. Wu, P. Liu *et al.*, "Energy-efficient task scheduling for multi-core platforms with per-core DVFS," *Journal of Parallel and Distributed Computing*, vol. 86, no. 4, pp. 71–81, 2015.

- [18] M. Akbari and H. Izadkhah, "Hybrid of genetic algorithm and krill herd for software clustering problem," in *2019 5th Conf. on Knowledge Based Engineering and Innovation (KBEI)*, Tehran, Iran, pp. 565–570, 2019.
- [19] A. Bose, T. Biswas and P. Kuila, "A novel genetic algorithm based scheduling for multi-core systems," in *Smart Innovations in Communication and Computational Sciences*, Singapore: Springer, pp. 45–54, 2019.
- [20] T. D. Braun, H. J. Siegel, N. Beck, L. L. Bölöni, M. Maheswaran *et al.*, "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, no. 6, pp. 810–837, 2001.