Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 9827259, 1 page
https://doi.org/10.1155/2023/9827259

Retraction

Retracted: Low-Power Task Scheduling Algorithm for the Multicore Processor System Based on the Genetic Algorithm

Wireless Communications and Mobile Computing

Received 13 September 2023; Accepted 13 September 2023; Published 14 September 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] X. Li, "Low-Power Task Scheduling Algorithm for the Multicore Processor System Based on the Genetic Algorithm," *Wireless Communications and Mobile Computing*, vol. 2022, Article ID 4363937, 6 pages, 2022.

Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 4363937, 6 pages
https://doi.org/10.1155/2022/4363937

Research Article

Low-Power Task Scheduling Algorithm for the Multicore Processor System Based on the Genetic Algorithm

Xianning Li

Department of Information Engineering, Guangxi Technological College of Machinery and Electricity, Nanning, Guangxi 530007, China

Correspondence should be addressed to Xianning Li; 1710411129@hbut.edu.cn

Received 26 June 2022; Revised 19 July 2022; Accepted 28 July 2022; Published 10 August 2022

Academic Editor: Aruna K K

In order to solve the periodic hard real-time tasks with dependencies on multicore processors, the author proposes a low-power task scheduling algorithm for multicore processor systems based on the genetic algorithm. This method first uses the RDAG algorithm to separate the tasks and then takes the lowest power consumption as the principle; a genetic algorithm is used to determine the task mapping. Experimental results show that based on the power consumption model of Intel PXA270, several random task sets are used for simulation experiments, which shows that this method saves 20% to 30% of the energy consumption compared with the existing methods. This method effectively shortens the completion time of tasks, improves the utilization efficiency of multicore system resources, improves the parallel computing capability of multicore systems, reduces the average response time of tasks, and improves the throughput and resource utilization of multicore systems.

1. Introduction

In recent years, research on single-chip multicore processor architecture has gradually increased, with many students using it for some connections; for example, the new server group is a multistudent connection [1]. However, if the multicore processor has not yet reached the end of the process, the operating system will not be able to use the quality of the standard varieties of the process at all if it is always used directly in the process. In addition to multicore processors, system-based scheduling technology is now one of the hotspots of research.

At present, it is generally believed that the most promising task scheduling technology is to use heuristic method to group first and then use the genetic algorithm to schedule [2]. Based on this idea, the author proposes a new scheduling algorithm suitable for multicore processor parallel systems, the algorithm first allocates tasks according to the heuristic algorithm, in order to obtain a reasonable grouping scheme based on multicore processors, and the genetic algorithm is used for scheduling [3]. The problem of finding the best assignment for a task is usually an NP-complete problem, which has been proven in the literature. Heuristics are generally considered a feasible way to find near-optimal solutions to NP-complete problems. The author proposes an iteration-based heuristic algorithm to assign tasks to processors in multicore systems. The algorithm is based on the basic idea that multiple cores in the processor communicate through shared cache or through high-speed channel internal connections, and the communication overhead is negligible, and those tasks with frequent communication are assigned to the cores on the same processor for execution. This algorithm takes full advantage of the characteristics of multicore systems, and the communication overhead during task execution is reduced [4]. But the algorithm itself only considers the task allocation of the kernel on the multicore system and does not involve the task scheduling after the task allocation. Based on the proposed ideas and the characteristics of multicore systems, the author first assigns tasks to appropriate processors and then schedules tasks to a certain core in the processor for execution. In the process of task allocation and scheduling, the task allocation model in the

literature is improved, and the idea of task scheduling based on replication is added; finally, the genetic algorithm is used to schedule tasks on each processor [5].

Using a genetic algorithm scheduling strategy, explore how to complete the balanced distribution of tasks in multicore to achieve acceleration. Algorithms exploit the elasticity of system resources, automatically search for parallel subtasks, and allocate them to the corresponding computing nodes reasonably, which improves the resource scheduling performance of multicore systems; it realizes the optimal scheduling of tasks submitted by users and achieves the goal of balancing the computing load of each processor in the system and improving the overall performance of the multicore system, as shown in Figure 1.

2. Literature Review

With the maturity of multicore system hardware structure, the task scheduling problem of the multicore system has become the most important research direction in this field. Multicore systems are mainly composed of computing resources and communication resources; the scheduling problem of multicore systems can be summarized as the allocation of multiple computing resources and communication resources on the time axis [6]. An excellent resource scheduling algorithm can reasonably allocate the computing resources of multicore systems and effectively reduce the total execution time and total consumption of processor computing, so as to tap the potential performance of the system as much as possible. In supercomputers, there are already many solutions to related problems, for example, heuristic intelligent task scheduling based on the genetic algorithm and ant algorithm, agent technology derived from artificial intelligence, strictly defined mathematical object Petri net, and various task scheduling algorithms in multiclient multiserver mode [7]. However, the applications of the above-mentioned algorithms are mostly aimed at supercomputers, and although they have reference significance for micro-multicore systems, they cannot be directly applied. Moreover, special consideration should be given to the decomposition and scheduling of a single large task and the communication between subtasks. Genetic algorithms use simple coding technologies and procedures designed to represent complex situations and solve complex problems, and their solution can be considered the right process [8]. Genetic algorithms are used to perform a variety of scheduling tasks, using similar methods and solutions around the world to study genetic traits and determine the divisions needed to achieve the goal of equality. This shortens the time to complete tasks and increases the efficiency of the use of many basic resources [9].

With the widespread application of wireless and mobile devices, the issue of power consumption has become increasingly prominent [10]. Power consumption is not a single problem; it is related to system security and heat dissipation costs. Therefore, low power consumption, as a key design requirement of real-time embedded systems, is receiving more and more attention. For multicore energy-saving task scheduling, Nucamendi-Guillén made an in-depth discussion, pointing out the problem of real-time energy-saving scheduling in multicore systems, which can be summarized into three aspects: allocation problem, priority problem, and speed regulation problem [11]. Zhang proposed an energy-efficient task assignment method based on heterogeneous multicore processors [12]. The method reduces the total energy consumption of the system by assigning tasks to two rounds. Nakada proposed a multiprocessor energy-efficient scheduling algorithm for periodic tasks [13]. The algorithm determines the minimum processor scheduling requirements through static analysis and dynamically scales the voltage of each processor under the condition of schedulability, thereby reducing the energy consumption of the entire system. Bahrami et al. propose a multicore energy-efficient task scheduling method for streaming media applications. After the method converts the dependent task set into independent tasks, the task mapping is determined with the principle of minimum energy consumption under the constraints of hard real-time conditions [14]. Nakada et al. proposed an energy-efficient task scheduling algorithm for homogeneous multicore processors [13]. After the task is independent, the algorithm arranges task mapping according to the principle of the shortest scheduling length and then performs frequency/voltage adjustment in each core to reduce system power consumption.

These studies generally do not consider the voltage conversion energy of the tasks of each processor under different frequency/voltage modes, and the solutions obtained by some algorithms are approximate optimal solutions, and there is still room for optimization. Therefore, based on the current research status and through in-depth research, a method for energy-saving task scheduling based on the genetic algorithm is designed. In this method, the RDAG algorithm is used to separate the tasks first, and then, the genetic algorithm is used to determine the task mapping based on the principle of the lowest energy consumption [15]. Based on the IntelPXA270 power consumption model, several random task sets are used for simulation experiments, and the results show that the method is superior to the existing methods.

3. Methods

3.1. Tasks and Power Consumption

3.1.1. Task Model. The task set can be abstracted as a directed acyclic graph, denoted as $G = \{V, E, p, \mathrm{ET}, \mathrm{CT}, D\}$, where V is the set of nodes, E is the set of directed edges, p is the set of iterations between nodes, ET is the set of execution cycles of tasks, CT is the set of communication cycles between tasks, and D is the set of task deadline [16]. For example, $v_i \in V$ and $v_j \in V$ represent the two tasks of the task set, respectively. e_{ij} represents the dependency of two tasks; that is, task v_j is executed after v_i. $p(e_{ij}) = k$ (k is a constant) means that task v_j is executed in the kth round after task v_i. $E_{ti} \in \mathrm{ET}$ represents the number of cycles executed by task v_i, and $Ct_{ij} \in \mathrm{CT}$ represents the number of intercore communication cycles between task v_i and task v_j.

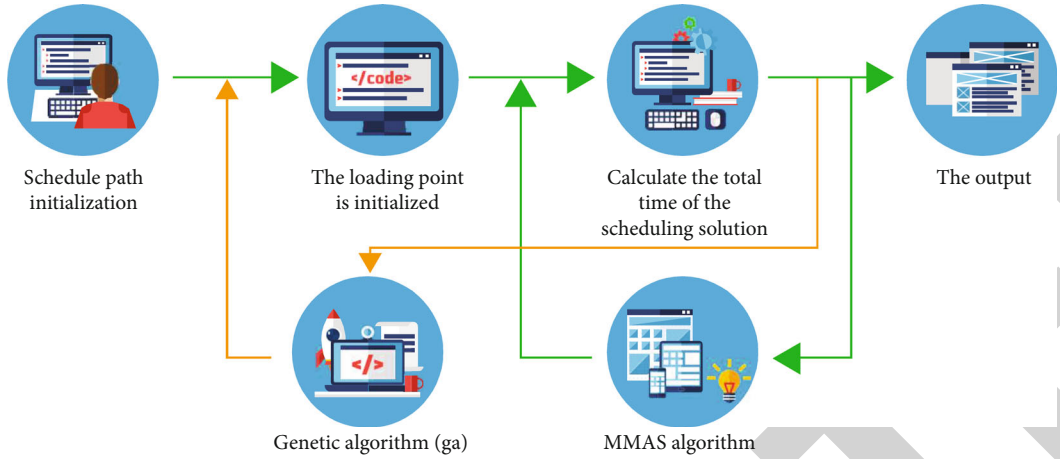

Figure 1: Multicore processor based on the genetic algorithm.

3.1.2. Power Model. For a homogeneous processor with multiple cores, it is assumed that each processor core supports m discrete frequency/voltage modes. Denote the frequency/voltage pattern of each core as $\mathrm{Core}(F, V)$, where F represents the set of frequencies and V represents the set of voltages. Among them, $\mathrm{Core}_i = (f_i, V_i)$ represents a frequency/voltage mode.

The total energy consumption is shown in

$$E_{\text{total}} = E_{\text{task}} + E_{\text{commmunication}} + E_{\text{transition}} + E_{\text{idle}}. \quad (1)$$

In formula (1), E_{task} represents the total energy consumed by all tasks, $E_{\text{communication}}$ represents the total energy consumed by intercore communication, $E_{\text{transition}}$ represents the total energy consumed by the state transition of each core, and E_{idle} represents the total energy of the trumpet when each core is idle.

The dynamic energy consumed by task v_i is shown in

$$E(v_i) = P_{\text{AC}} \times Et_i, \quad (2)$$

$$P_{\text{AC}} = C_{\text{eff}} \times V_{\text{dd}}^2 \times f, \quad (3)$$

where P_{AC} is the dynamic power consumption per cycle of the processor, Et_i is the number of execution cycles required for task v_i, C_{eff} is the average switched capacitance per cycle, V_{dd} is the power supply voltage, and f is the processor clock frequency. The conversion energy generated by the voltage conversion from $V_{\text{dd}i}$ to $V_{\text{dd}j}$ is

$$E_{ij} = C_r \times \left| V_{\text{dd}i} - V_{\text{dd}j} \right|^2, \quad (4)$$

where C_r is the power rail capacitance.

3.2. Multicore Energy-Saving Task Scheduling Algorithm

3.2.1. Algorithm Idea. There is a periodic hard real-time task set $G = \{V, E, p, \mathrm{ET}, \mathrm{CT}, D\}$ with dependencies and a homogeneous processor with n cores, where each core has a frequency/voltage mode of $\mathrm{Core}(F, V)$. The algorithm is to allocate the tasks of the task set to each core for processing at a specific processor frequency, so that the total energy consumption is minimized. The algorithm idea is shown in Figure 2. First, the RDAG algorithm proposed by some scholars makes tasks independent and then uses the genetic algorithm based on multicore energy-saving task scheduling to determine the optimal task mapping.

Among them, the idea of the genetic algorithm based on multicore energy-saving task scheduling is as follows.

(1) Initialization of task scheduling population

One advantage of genetic algorithm is that it can implicitly search for multiple solutions in the solution space in parallel; of course, this requires random generation of an initial population containing multiple solutions. Suppose there are m tasks in the task set, and the individual i in the population is represented as $U_i = \{v_{i1}, v_{i2}, \cdots, v_{im}, f_{i1}, f_{i2}, \cdots, f_{im}\}$. v_{ij} represents the processor core allocated to the task v_j in the task set, and f_{ij} represents the corresponding processor frequency when the task runs on the core v_{ij}. The population initialization steps are as follows: (a) randomly generate an individual; (b) execute the tasks on each core in the individual in the order f_i from large to small; (c) if the total execution time of tasks on a processor core is longer for a limited time, the individual will be eliminated; and (d) go back to step (a) until a given number of individuals are generated to form a population.

(2) Selection of the scheduling sequence

The purpose of selection is to enable the better individuals to be inherited to the next generation with a greater probability. Let the fitness function of individual i be

$$F_{it_1} = \frac{1}{Eg_i}. \quad (5)$$

Eg_i is the total energy consumed by individual i. Assuming that there are m tasks in the task set, there are n processor cores in total, and there are $S(i)$ tasks on core i, which are

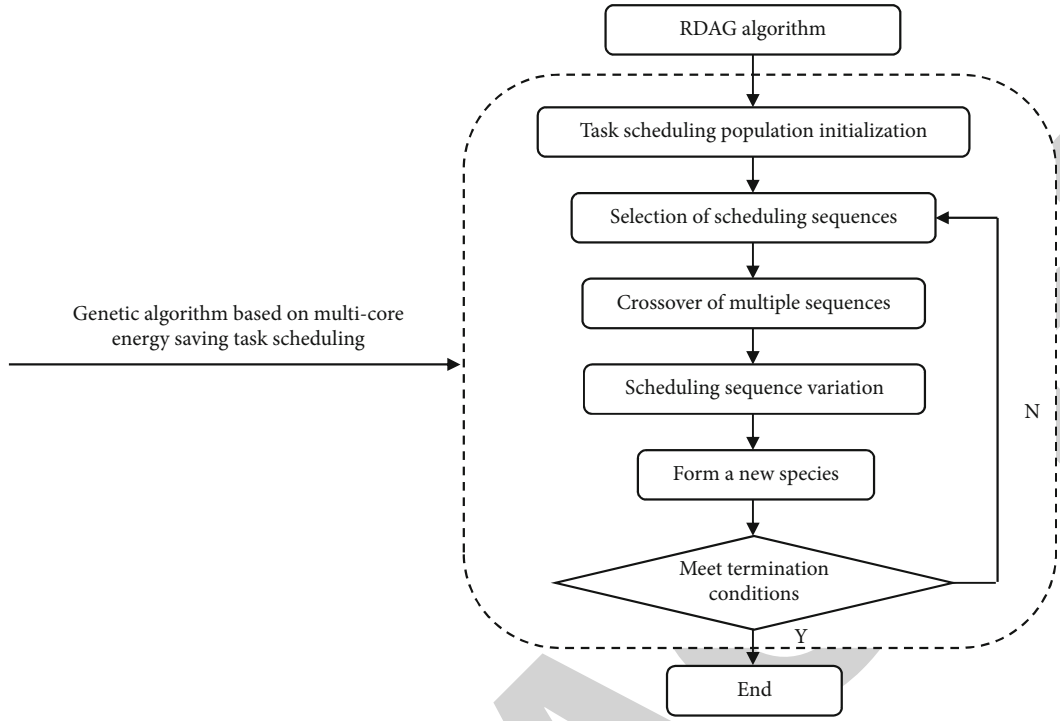

Figure 2: Algorithm idea.

arranged in descending order of execution frequency. ET represents the set of execution cycles of tasks, and CT represents the set of communication cycles between tasks. The number of intercore communication cycles between task v_i and task v_j is shown as

$$Ct_{ij} = \begin{cases} Ct_{ij} \\ 0 \end{cases}. \tag{6}$$

Formula (7) can be obtained from formulas (1), (2), (3), (4), (5), and (6):

$$Eg_i = \&\sum_{k=2}^{m} P_k \times Et_k + \sum_{k=1}^{m}\sum_{l=1}^{m} Ct_{kl} \times P_c + \sum_{k=1}^{m}\sum_{l=1}^{s(k)+1} C_r \times \left| V_{\mathrm{dd}l} - V_{\mathrm{dd}(l+1)} \right|^2 + \sum_{k=1}^{n} P_{\mathrm{idlle}} \times t_{\mathrm{idlle}k}, \tag{7}$$

where P_k represents the processor power when executing task k; P_c represents the intercore communication power; $V_{\mathrm{dd}k}$ represents the processor voltage when executing the k th task on the core; P_{idle} is the processor power when idle; and $t_{\mathrm{idle}(k)}$ indicates the idle time of the kth processor.

The roulette method is used for selection, so that the better individual has a higher probability of being selected. The way is as follows: (a) calculate the sum of the fitness functions of all individuals in the current population sum; (b) calculate an accumulated value sequence of the fitness functions: Seq = $\{s_1, s_2, \cdots, s_k\}$, where k represents the number of individuals and $s_i = \sum_{l=1}^{i} F_{it}$; and (c) randomly generate a number x, $0 \le x <$ Sum; let $s_0 = 0$; if $s_i - 1 \le x < s_i$; then, individual i is selected.

Table 1: IntelPXA270 power consumption model frequency and corresponding voltage and power consumption.

Frequency (MHz)	Voltage (V)	Power consumption (mW)
13 (idle)	0.85	44.2
104	0.90	115.0
208	1.15	279.0
312	1.25	390.0
416	1.35	570.0
520	1.45	747.0
624	1.55	925.0

Table 2: Task set characteristics.

Task set name	Number of nodes	Number of sides
Task set 1	20	30
Task set 2	30	45
Task set 3	40	60
Task set 4	50	75
Task set 5	60	88
Task set 6	70	100

(3) Crossover and mutation of scheduling sequences

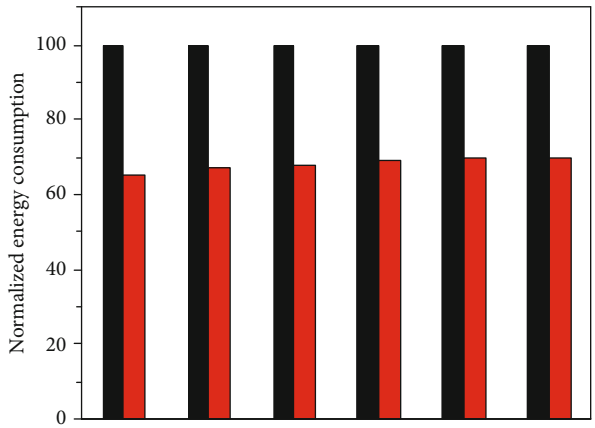

Figure 3: Comparison of average power consumption of two-core processors.

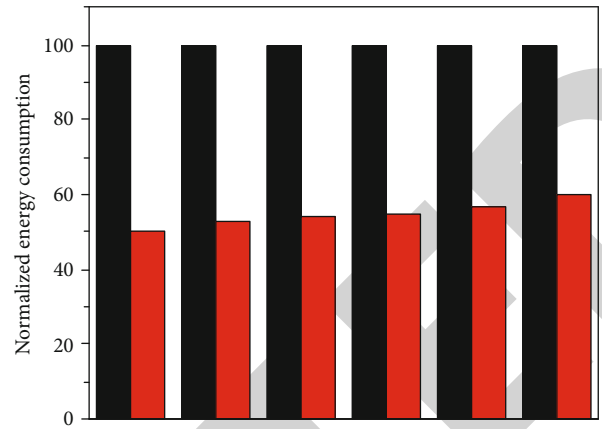

Figure 4: Energy consumption comparison.

The purpose of crossover and mutation is to increase the diversity of the population. The single-point crossover method used is as follows: (1) two individuals are obtained by selection, (2) an intersection is randomly selected, and (3) the parts after the intersection of the two individuals are crossed. $U_i = \{v_{i1}, v_{i2}, \cdots, v_{im}, f_{i1}, f_{i2}, \cdots, f_{im}\}$ and $U_j = \{v_{j1}, v_{j2}, \cdots, v_{jm}, f_{j1}, f_{j2}, \cdots, f_{jm}\}$ are two individuals. Suppose the intersection point is k; then, the sequences after k are exchanged.

Randomly select an element from the individual to mutate; there are two cases: if the selected element represents a processor core, the mutated element also represents a processor core; if the element represents the processor frequency, the mutated element also represents the processor frequency. The k-th element of the individual U_i represents a processor core, and the element mutates to become an element representing another processor core.

3.2.2. Algorithm Description. The input of the genetic algorithm based on multicore energy-saving task scheduling is the independent task set G, the number of processor cores n, the task set deadline D, the frequency/voltage mode Core(F, V) of each core, and the different frequencies of each core. With power, communication power between cores, the number of individuals in each generation of the algorithm N, the maximum genetic number Gn, the cut-off condition q, the crossover probability P_1, and the mutation probability P_2, the output is the final scheduling sequence S. $S = [L_0, L_1, \cdots, L_{n-1}]$, where L_i represents the scheduling sequence of core i [17].

4. Results and Discussion

In order to verify the effectiveness of the algorithm, the above algorithm is compared with the methods in the literature. In the experiment, the number of execution cycles of each task is randomly generated between 10 and 50, and the number of communication cycles between tasks is randomly generated between 1 and 5. Input the number of tasks and the number of edges, and randomly generate a task set represented by a directed acyclic graph [18]. The processor adopts the power consumption model of IntelPXA270, the mode transition time of this model can be ignored, and its frequency/voltage mode and corresponding power consumption are shown in Table 1. Set the system bus frequency to be fixed at 208 MHz and the power rail capacitance C_r to be 5 pF. Under this model, 6 task sets are randomly generated, and their characteristics are shown in Table 2. Set the genetic algorithm crossover probability and mutation probability to 0.9 and 0.03, respectively. The two-core and four-core are compared, respectively, and the power consumption comparison chart is obtained, as shown in Figures 3 and 4.

With the development of many basic embedded systems, energy consumption has become a hotspot for hotspot research. According to the current study, the author worked on the shortcomings of various energy-saving algorithms and developed an energy-saving method. The process is divided into two stages: the first stage separates the tasks using the RDAG algorithm and the second stage assigns tasks to each process using a genetic procedure [19]. Experiments show that this method achieves a relatively good energy-saving effect.

5. Conclusion

Aiming at the existing micro-multicore system, the author takes the genetic algorithm as the research method and takes the mapping process from multitask to multicore system as the research core; it realizes the optimal allocation of system resources, effectively shortens the completion time of tasks, and improves the utilization efficiency of multicore system resources; parallel computing capability of the multicore system is improved, the average response time of the task is reduced, and the throughput of the multicore system and the resource utilization rate of the system are improved.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that he has no conflicts of interest.

Acknowledgments

This study was financially supported by the Basic Ability Improvement Project of Middle-Aged and Young Teachers in Guangxi (2022KY1074).

References

[1] B. Zimmer, P. Raina, S. G. Tell, Y. Zhang, and N. Pinckney, "A 0.32-128 tops, scalable multi-chip-module-based deep neural network inference accelerator with ground-referenced signaling in 16 nm," *IEEE Journal of Solid-State Circuits*, vol. 99, pp. 1–13, 2020.

[2] X. Huang, R. Yu, D. Ye, L. Shu, and S. Xie, "Efficient workload allocation and user-centric utility maximization for task scheduling in collaborative vehicular edge computing," *IEEE Transactions on Vehicular Technology*, vol. 70, pp. 3773–3787, 2021.

[3] S. Lim, M. J. Kim, and W. A. Chang, "A genetic algorithm (GA) approach to the portfolio design based on market movements and asset valuations," *IEEE Access*, vol. 8, pp. 140234–140249, 2020.

[4] P. Luo, F. R. Yu, J. Chen, J. Li, and V. Leung, "A novel adaptive gradient compression scheme: reducing the communication overhead for distributed deep learning in the Internet of things," *IEEE Internet of Things Journal*, vol. 8, pp. 11476–11486, 2021.

[5] G. Kielanski and B. V. Houdt, "On the asymptotic insensitivity of the supermarket model in processor sharing systems," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 5, no. 2, pp. 1–28, 2021.

[6] W. Wen, Y. Cui, T. Quek, F. C. Zheng, and S. Jin, "Joint optimal software caching, computation offloading and communications resource allocation for mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7879–7894, 2020.

[7] M. Alsolamy and G. Taha, "The use of the ant algorithm in the audit planning of multi-branch organizations," *iBusiness*, vol. 13, no. 4, pp. 187–209, 2021.

[8] M. E. Gee, A. Dempsey, and J. E. Myers, "Caesarean section: techniques and complications," *Obstetrics, Gynaecology and Reproductive Medicine*, vol. 30, no. 4, pp. 97–103, 2020.

[9] W. Rong, B. Luo, and Z. Liu, "P-12.4: investigation of microlens to improve the efficiency of micro-LED display system," *SID Symposium Digest of Technical Papers*, vol. 52, no. S1, pp. 605–608, 2021.

[10] Y. Hasegawa and T. Kanazawa, "Game theoretic analysis of incentive-based power consumption reduction problems with for-profit or nonprofit aggregator," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E103.A, no. 2, pp. 390–397, 2020.

[11] S. Nucamendi-Guillén, D. Flores-Díaz, E. Olivares-Benitez, and A. Mendoza, "A memetic algorithm for the cumulative capacitated vehicle routing problem including priority indexes," *Applied Sciences*, vol. 10, no. 11, p. 3943, 2020.

[12] Z. Zhang, "A computing allocation strategy for internet of things' resources based on edge computing," *International Journal of Distributed Sensor Networks*, vol. 17, 2021.

[13] T. Nakada, H. Yanagihashi, K. Imai, H. Ueki, and H. Nakamura, "An energy-efficient task scheduling for near real-time systems on heterogeneous multicore processors," *IEICE Transactions on Information and Systems*, vol. E103.D, no. 2, pp. 329–338, 2020.

[14] F. Bahrami, B. Ranjbar, N. Rohbani, and A. Ejlali, "PVMC: task mapping and scheduling under process variation heterogeneity in mixed-criticality Systems," *Computing*, vol. PP(99), pp. 1–1, 2021.

[15] M. Fan and A. Sharma, "Design and implementation of construction cost prediction model based on SVM and LSSVM in industries 4.0," *International Journal of Intelligent Computing and Cybernetics*, vol. 14, no. 2, pp. 145–157, 2021.

[16] J. Jayakumar, S. Chacko, and P. Ajay, "Conceptual implementation of artificial intelligent based E-mobility controller in smart city environment," *Wireless Communications and Mobile Computing*, vol. 2021, 8 pages, 2021.

[17] G. Veselov, A. Tselykh, A. Sharma, and R. Huang, "Applications of artificial intelligence in evolution of smart cities and societies," *Informatica (Slovenia)*, vol. 45, no. 5, p. 603, 2021.

[18] P. Ajay, B. Nagaraj, R. Arun Kumar, R. Huang, and P. Ananthi, "Unsupervised hyperspectral microscopic image segmentation ung deep embedded clustering algorithm," *Scanning*, Article ID 1200860, 9 pages, 2022.

[19] Q. Zhang, "Relay vibration protection simulation experimental platform based on signal reconstruction of MATLAB software," *Nonlinear Engineering*, vol. 10, no. 1, pp. 461–468, 2021.