Independent Task Scheduling based on Meta-Heuristic Algorithm in Cloud Computing

Dina Riadh Abdulrazzaq[1] **Narjis Mezaal Shati**[2]
dinashibani@uomustansiriyah.edu.iq **dr.narjis.m.sh@uomustansiriyah.edu.iq**
Haider K. Hoomod[3]
dr.haiduh@uomustansiriyah.edu.iq

Abstract: Task scheduling is a very important topic in the context of cloud computing because it affects the quality of the offered cloud computing services. In the literature, many strategies have been presented for task scheduling handling, with the focus of the available algorithms on reducing execution time while ignoring other quality of service (QoS). This paper has presented an optimal scheduling algorithm to minimize execution time and energy consumption based on the Flower Pollination Algorithm (FPA) and Henon's chaotic map for scheduling independent tasks. First, Henon's chaotic map has been employed to generate the initial solutions, and then FPA has been used to schedule cloudlets on appropriate virtual resources. Tests have been performed on two data sets. The results of the proposed scheduling algorithm have been studied and compared with those of other existing algorithms in the literature. The results show that the proposed scheduling algorithm has a better result than the competing algorithms in terms of both execution time and energy consumption.

Keywords: Cloud computing, energy consumption, chaotic map.

1. Introduction

Cloud Computing (CC) is the ability to access on-demand resources over the Internet without direct management by the user [1]. CC deployment models which refer to the access, ownership, and scale of any cloud service or services by the users are classified into four types of cloud computing deployment models: public,

[1] Ph.D. Student, Computer Science Department, Faculty of Science, Mustansiriyah University, Baghdad, Iraq.
[2] Assist. Prof., Computer Science Department, Faculty of Science, Mustansiriyah University, Baghdad, Iraq.
Prof., Computer Science Department, Faculty of Education, Mustansiriyah University, Baghdad, Iraq.

private, hybrid, and community cloud. The public model delivers an online-based service to various consumers. The main advantage of it is that it is usually larger in size. While the disadvantage of this model is that it offers limited configurations and security protection [2], Whereas the private model, also known as "internal cloud," is built specifically for each project. This model gets around some of the issues that a public cloud has, such as data security, but it also has its own drawbacks, including being more expensive and having a smaller size [3]. The hybrid model, on the other hand, combines the benefits of public and private CC environment models via special methods that allow transmission of data and application between them [4]. Last but not least, the community cloud model which described as a dispersed system that is formed by incorporating different cloud services to address the particular requirements of an industry, business, or community. In other words, organizations with similar goals or responsibilities may share the community's infrastructure. It is typically run by a third party or a group of one or more community organizations [5].

The resources available in the cloud are limited, so a proper scheduling algorithm that allocates a task to the appropriate processing unit (virtual machine) is necessary to prevent any wastage of cloud computing resources. Due to the increasingly dynamic demands on the limited resources offered by the CC environment, it has become clear the importance of finding a suitable scheduling strategy for the purpose of having the optimal task to virtual resource allocation. Scheduling can be classified as an NP-hard problem due to many factors, including the dynamic increase in data capacity across all domains as well as the rapid increase in task size [6]. All of these factors have increased the burden of finding a suitable scheduling algorithm. Here, the meta-heuristic [7] trend emerges as a robust computational model that has the promising capability to find the optimal solution to the NP-hard problem. It is worth noting that most current scheduling studies focus on optimization of time, neglecting the other QoS, which includes energy consumption. In this paper, a resourceful scheduling algorithm regarding execution time and energy consumption based on the Flower Pollination Algorithm (FPA) and the Henon chaotic map for the solution initialization has been suggested. The rest of the paper is arranged as follows: Section 2 gives the details of the theoretical background. Section 3 deals with the proposed scheduling algorithm. The results have been deliberated in Section 4. The conclusion and future suggestions are presented in Section 5.

2. Theoretical Concepts

This section has been dedicated to clarifying the concepts related to the proposed CC scheduling algorithm, which includes:

2.1 Execution Time

The execution time of the independent task scheduling in CC is valued as defined in Eq. (1):

$$ET=\sum_{j=0}^{v}\sum_{i=0}^{c}\frac{ET_i}{VMCap_j} \quad (1)$$

Where v is the number of VMs, while c is the number of user requests, VMCap referring to CPU speed, ET is the submitted cloudlet duration. Note that the lower execution time is the best [8].

2.2 Energy Consumption

Energy Consumption (EC) is the term used to describe the amount of energy used by a VM. In order to improve system performance and provide consumers with better service, minimum EC is required [8,9]. The definition energy consumption (in joules) is defined as follows:

$$EC=(\sum_{i=0}^{v} ET_i \times \exists_i + (ET_i - M) \times \delta_i)\times VMCap_i \quad (2)$$

$$\exists_i = 10\text{-}8 \times VMCap2i \quad (3)$$

$$\delta_i = 0.6\times \exists_i \quad (4)$$

where

ET_i is the total execution time, v is the number of virtual machines. M is the virtual machines maximum execution time. ∃i is the consumed energy in the active state. $VMCap_i$ is the CPU speed of the virtual machine. δ_i is the consumed energy in the idle state.

2.3 Henon Chaotic Map

Nowadays, chaotic maps are widely used in various branches of computer science due to their amazing properties, such as mixing property, random behavior and non-periodicity [10]. This makes chaos theory a promising option for scheduling. The generated chaotic series have used to represent have used to represent the sequencing order of cloudlets execution [11]. Among the various maps, Henon map which is a 2D map have examined in this paper. It is defined as:

$$xhk+1 = 1 - \alpha * xhk2 + \beta * yhk \quad (5)$$

$$yhk+1 = xhk \quad (6)$$

Where $x$hk and yhk are the initial values, while $x$hk+1 and yhk+1 are the next generated values. α and β have the values 1.4 and 0.3 respectively.

2.4 Flower Pollination Algorithm

FPA is an effective meta-heuristic algorithm which mimics the reproduction process of flowering plants. It has a local and global pollination steps which mimics the exploitations and exploration searching techniques, respectively [12,13].
The local pollination is denoted as defined in Eq. 7:

$$s_i^{t+1} = s_i^t + \partial.(s_j^t - s_k^t) \quad (7)$$

where s_i^t represents the ith flower (solution) at the current iteration, while s_j^t *and* s_k^t are randomly selected flowers of the same initial solutions and ∂ is a random number in the range [0, 1].
While, the the global pollination step, can be mathematically represented as defined in Eq.8:

$$s_i^{t+1} = s_i^t + \gamma.L(b^* - s_i^t) \quad (8)$$

$$L = \frac{\partial\tau(\partial)\sin\gamma\pi/2}{\pi s^{1+\partial}} \quad (9)$$

where b* is the best flower (solution) gained so far, γ is a scaling factor used to control the step size, and the γ a random number in range of [0.1]. L is derived from Levy distribution [14]. Note that a switch probability p in [0, 1] is used to control the type of flower pollination (local or global) at each iteration.

3. Proposed Scheduling Algorithm

The entire task of the proposed scheduling algorithm, which has denoted as FPA_TS, can be summarized in the following steps:

1. Instantiate the data center.
2. Construct a VM list and list them with the data center. Note that each VM has different characteristics.
3. Initiate the data center broker.
4. Submit the VMs list to the data center broker.
5. Read a sequence of cloudlets from the dataset. Note that cloudlets are non-preemptive and independent.
6. Formulate initial solutions.
7. Formulate mathematical model for optimized execution time and energy consumption of the scheduling problem in the context of CC.
8. Propose a CC scheduler based on FPA.

9. Send the optimal solution obtained to the data center broker, which is responsible for managing and scheduling the cloudlets on available VMs in the data center.

10. Launch the simulation.

11. Examining the simulation's outcomes.

These steps are involved in two main basic stages: the initialization stage and the optimization stage. The general structure of the presented scheduler is depicted in Figure 1.

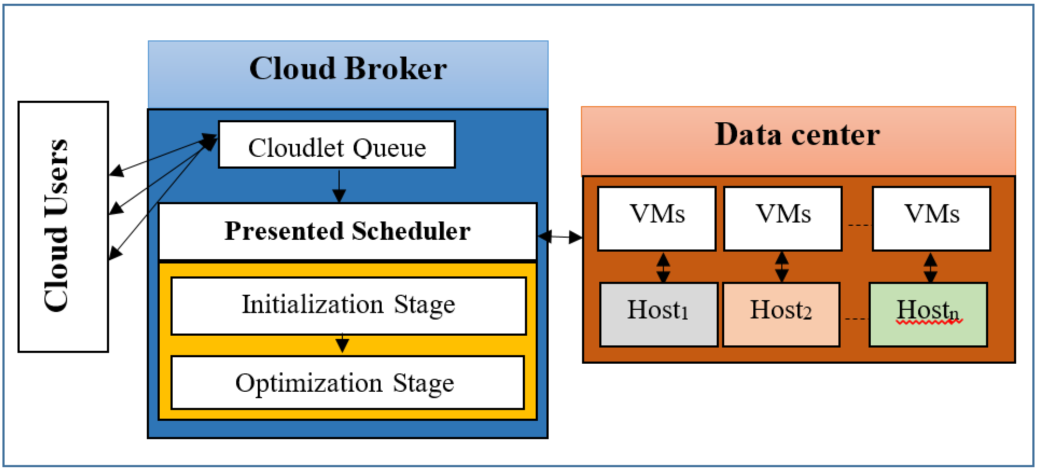


Figure 1: The presented scheduler component

3.1 The Iinitialization Stage

The first step of the presented scheduling algorithm is to prepare initial solutions (IS). Henon's map has used to prepare initial cloudlets sequence of size n and each solution contains c cloudlets. First, generate a series of c × n/2 chaotic values of the 2D Henon map (xh, yh), and then the c × n values of the continuous initial solutions are generated by intersecting the chaotic values xh and yh. The SPV rule [15] has used to convert continuous values of IS into discrete values. Finally, the actual initial solution allocation vector which represent the mapping of the generated cloudlet sequences to the offered virtualized resources have applicable as defined in Eq. 10.

Si = cloudlet number *mod* the number of the offered VM (10)

Figure 2 shows the steps for preparing the initial solution.

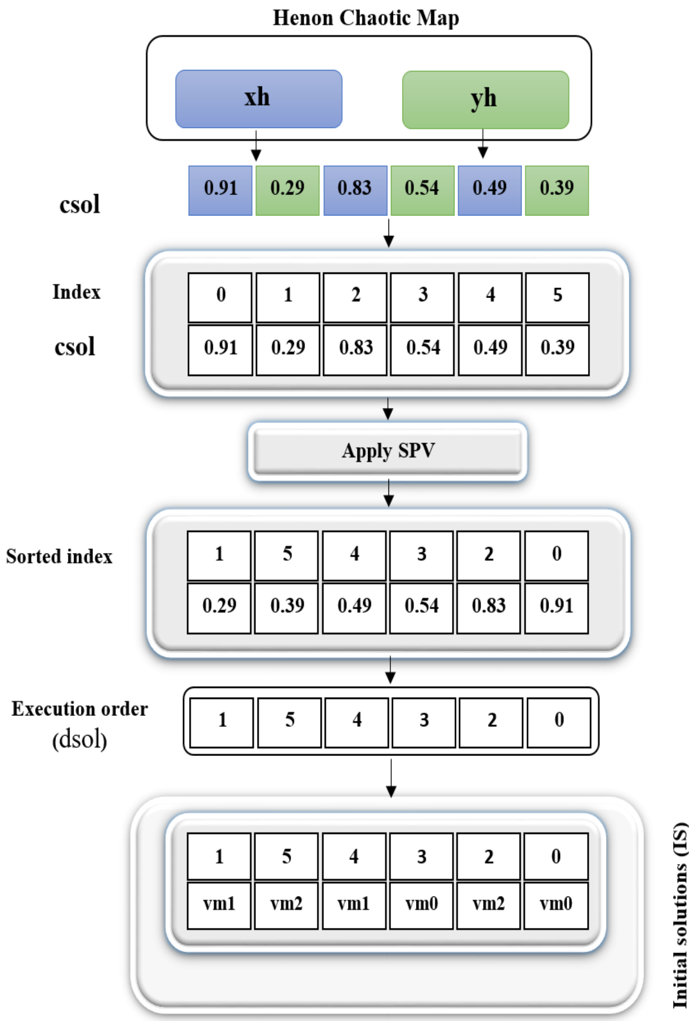


Figure 1: The Initialization stage

Algorithm 1 illustrate the details of the initialization stage.

Algorithm 1: Cloudlet sequence generation

Input: Number of initial solutions (nIs), Henon map parameter, number of cloudlet (c).

Output: cloudlet sequence (iSol)

Begin

1. Parameter initialization
   Construct a one dimensional real-values vector temp1 of size nIs/2×n/2
   Construct a one dimensional real-values vector temp2 of size nIs/2×n/2
   temp1[0]←initial value of xh
   Temp2[0]← initial value of yh
   Construct a one dimensional real-values vector csol of size nIs/2
   Construct a one dimensional real-values vector osol of size nIs/2
2. Generate cloudlets executing sequence

```
For i←1 to nIs/2×n/2 do
    | temp1[i]← 1-1.4× temp1[i-1]^2 +0.3 ( temp2[i-1]) // Equation 5
    | temp2[i]← temp1[i-1]// Equation 6
End for
For k←0 to nIs/2×n do
    | if k mod 2 =0
    |     | csol[k]←temp1[k]
    | Else
    |     | csol[k]←temp2[k]
    | End
End for
```

3. Split one-dimensional vector csol into two dimensional array csol2d of size nIs/2 × n
4. Construct dsol by applying SPV rule to each row (solution) of csol2d
5. Formulate iSol accordint to Eq.10.
6. Return iSol.

End algorithm

3.2 Optimization Stage

With the completion of the initialization phase, the presented scheduling algorithm moves to the next and final phase, which is the optimization phase which deals with producing the best mapping of user demand and virtual machines in terms of execution time and power consumption using FPA. Algorithm 2 demonstrate the main steps of optimization stage.

Algorithm 2: FPA_TS

Input: max iteration(maxItr)
Output: optimal solution

Begin

Initialize FPA parameter (p)

Initialize the initial solution (IS) using algorithm 1

1. Calculate the execution time of each solution in initial solutions (ET) as defined in Eq.1.
2. Sort the ET in ascending order.
3. gB← top solution
4. While i < maxItr do
5. For each s in IS do
6. Generate random number rand in range [0,1]
7. If rand < p
8. Perform global pollination
Create a new cloudlet sequence by using Equation 8
9. Else
10. Perform local pollination
Create a new cloudlet sequence by using Equation 7
11. End for
12. Apply SPV rule to digitize the newly cloudlet sequences
13. Allocate the cloudlet to the appropriate VM to formulate (nSol) as defined in Eq. 10.
14. Compute the ET of each solution of nSol
15. IS←nSol
16. Update gB
17. i←i+1
18. End while
19. Return gB

End algorithm

4. Result Discussion

To prove the effectiveness of the presented algorithm, an extensive examination has been conducted among the presented FPA_TS and two literature references, namely MCT-PSO [16], which have introduced a scheduling algorithm based on the integration between particle swarm optimization (PSO) and different heuristic algorithms. And the CETSA [17] scheduling algorithm, which proposed independent task scheduling to reduce cost, time, and energy consumption.
Cloudsim platform has been used to conducted the experiment under the hardware environment with “2.40 GHz processor, Intel Core(TM) i7-13700H, and 16 GB RAM”. Moreover, the NASA iPSC [18] dataset, which contains around 42264 cloudlets, and the GoCJ [19] dataset, which contains around 10450 cloudlets, have been taken into account. Implementation details of cloudsim setup have illustrated in Table 1.

Table 1: Cloudsim implementation details

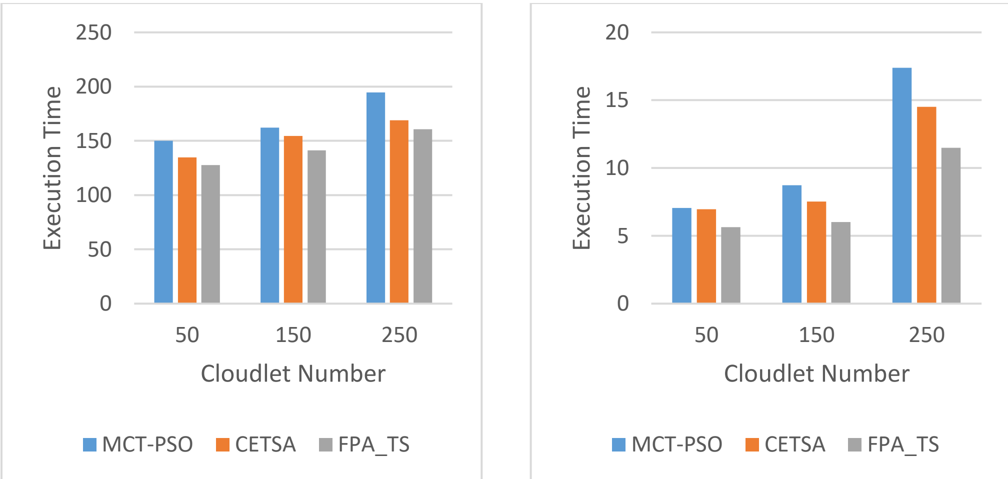
Entity	Parameter	Values
Vms	number	25
	Bandwidth	1024 Mbps
	Host number	3
	VMM	Xen
	RAM	100-2000 MB
	CPU	2000 MIPS
Data center	number	1
Cloudlets	number	50-250
	length	100-10000

Table 2 presents the numerical values of execution time for the presented FPA_TS algorithm and the compared ones. However, the visual representation of Table 2 is introduced in Fig. 3. The findings have demonstrated that it can be said that for both datasets, when the task number is small, the available resources are enough in the cloud, so all the compared scheduling algorithms have a better computation time, and the computation time increases as the number of tasks increases. Also, the FPA_TS algorithm outperformed the compared algorithms for all tested cases regarding computation time for the two datasets. This is due to the ability of the presented scheduling algorithm to maintain a good equilibrium between local and global search throughout the entire search process.
Regarding Energy consumption metric, Table 3 presented the numerical values of the proposed FPA_TS algorithm and the comparative ones. Yet, Fig. 4 offer the depiction of Table 3. It is noticeable that the presented FPA_TS algorithm have the

lower power consumption among compared algorithms. This is due the ability of the proposed FPA_TS algorithm to choose the most appropriate virtual source to assign to the incoming request.

Table 2: Execution time evaluation

dataset	Case	MCT-PSO	CETSA	FPA_TS
NASA	50	150.1197	134.7670	127.776
	150	162.1249	154.6448	141.1798
	250	194.5824	169.0450	160.6481
GOCJ	50	7.0614	6.9514	5.6372
	150	8.7372	7.5222	6.0192
	250	17.3778	14.5198	11.4806

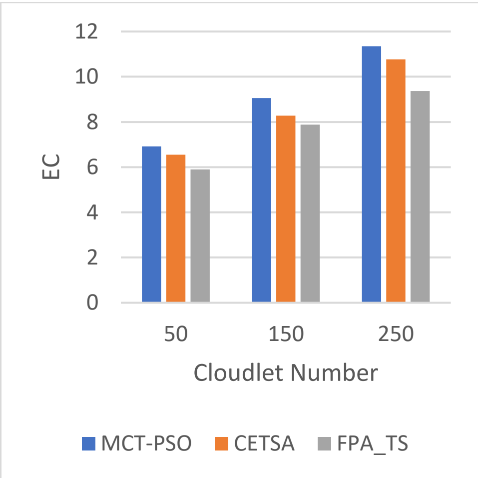


(A) (B)

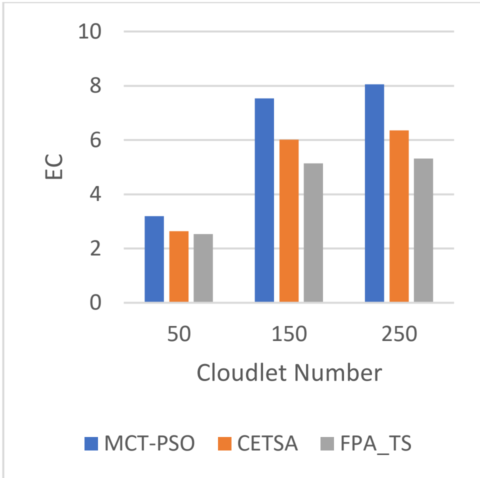
Figure 3: execution time (A) for NASA dataset (B) for GoCJ dataset

Table 3: EC evaluation

dataset	Case	MCT-PSO	CETSA	FPA_TS
NASA	50	6.9166	6.5541	5.90118
	150	9.0578	8.2747	7.8862
	250	11.3518	10.7693	9.3627
GOCJ	50	3.1920	2.6357	2.5315
	150	7.5382	6.0099	5.1417
	250	8.0529	6.3605	5.3113



(A)



(B)

Figure 4: EC (A) for NASA dataset (B) for GoCJ dataset

5. Conclusions

In this paper, an optimization scheduling algorithm implemented using Cloudsim has been supported. The proposed scheduling algorithm has been denoted to reduce execution time and EC. The scheduling algorithm consist mainly of two steps; the first step concerns the formulation of initial solutions. While optimization is the focus of the second step, both phases aid in allocating cloudlets effectively to the available virtual resources. The results have been calculated by comparing the FPA_TS, MCT-PSO and CETSA scheduling algorithms. the experimental results indicate that the FPA_TS scheduling algorithm has the best performance as compared with MCT-PSO and CETSA, depending on the metrics EC and execution time. For future work, Adapt the currently offered FPA_TS scheduling algorithm to work in edge and fog computing scenarios. Extend the introduced into Multi-objective optimization in order to optimize more quality of service parameters.

12. References

[1] Mahmood HA, Hashem SH. Network intrusion detection system (NIDS) in cloud environment based on hidden Naïve Bayes multiclass classifier. Al-Mustansiriyah Journal of Science. 2018;28(2):134-42.

[2] F.H. Ali, R. N. Jawad RN, W. A. Hussein, "Solving Bi-Criteria and Bi-Objectives of Total Tardiness Jobs Times and Range of Lateness Problems Using New Techniques," Al-Mustansiriyah Journal of Science, vol 33, No. 3, pp 27-35, 2022.

[3] H. B. Patel and N. Kansara, "Cloud Computing Deployment Models: A Comparative Study," International Journal of Innovative Research in Computer Science & Technology (IJIRCST),2021.

[4] Z. K.Tavbulatova, K. Zhigalov, S. Y. Kuznetsova, A. M. Patrusova," Types of cloud deployment," In Journal of Physics: Conference Series Vol. 1582, No. 1, pp. 012085,. IOP Publishing,2020.

[5] S. Gupta, A. Gupta, G. Shankar, "Cloud Computing: Services, Deployment Models and Security Challenges," In proceeding of the 2nd International Conference on Smart Electronics and Communication (ICOSEC), pp. 414-418, IEEE, 2021.

[6] U. K. Jena, P. K. Das, and M. R. Kabat ," Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment," Journal of King Saud University-Computer and Information Sciences, Vol. 34, No. 6, pp. 2332-42, 2022 .

[7] M. Haris, and S. Zubair ," Mantaray modified multi-objective Harris hawk optimization algorithm expedites optimal load balancing in cloud computing," Journal of King Saud University-Computer and Information Sciences, Vol. 34, No. 10 ,pp. 9696-709,2022.

[8] K. Mishra, and S. K. Majhi , "A binary Bird Swarm Optimization based load balancing algorithm for cloud computing environment," Open Computer Science, Vol. 11, No.1 , pp. 146-60,2021.

[9] S.S Hassan, Z. Iqbal, M.J. Ikram, and M. Ishaque,” Henon chaotic map-based image encryption scheme using bit-level circular shift,” J. Theor. Appl. Inf. Technol, Vol. 100, No. 6, pp.1960-1973, 2022.

[10] Qi, Fei, S. Huang, T. Li, H. Yang, and X. Kang "2D henon-chebyshev chaotic map for image encryption," In 2019 IEEE 21st International Conference on High Performance Computing and Communications. pp. 774-781. IEEE, 2019.

[11] Yousif SF, Ali FH, Alshaikhli KF. Using Local Search Methods for Solving Two Multi-Criteria Machine Scheduling Problems. Al-Mustansiriyah Journal of Science. 2023 Dec 30;34(4):96-103.

[12] A.P. Singh, A. Kaur,” Flower pollination algorithm for feature analysis of kyoto 2006+ data set,” Journal of Information and Optimization Sciences,Vol. 40 , No. 2 , pp.467-78, 2019.

[13] M. I. Latiffi, M. R. Yaakub, I. S. Ahmad,” Flower Pollination Algorithm for Feature Selection in Tweets Sentiment Analysis,” International Journal of Advanced Computer Science and Applications, Vol 13, No. 5, 2022.

[14] K. Venkatasalam, P. Rajendran, M. Thangavel,” Improving the accuracy of feature selection in big data mining using accelerated flower pollination (AFP) Algorithm,” Journal of medical systems. Vol. 43, 2019.

[15] M. Liebenlito, N. Inayah, A. N. Rahmah, and A. Widiatmoko, “Modified Firefly Algorithm using Smallest Position Value for Job-Shop Scheduling Problems”, 2020.

[16] S. Alsaidy, A. Abbood, and M. Sahib, "Heuristic initialization of PSO task scheduling algorithm in cloud computing". Journal of King Saud University-Computer and Information Sciences, Vol. 34(6), pp.2370-2382, 2022.

[17] N. Mansouri, and R. Ghafari, "Cost-Efficient Task Scheduling Algorithm for Reducing Energy Consumption and Makespan of Cloud Computing”, Computer and Knowledge Engineering, 5(1), pp.1-12, 2022.

[18] M. Abd Elaziz, S. Xiong , K. P. Jayasena, L. Li ,” Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution,” Knowledge-Based Systems, Vol. 169, pp.39-52, 2019.

[19] A. Hussain, and M. Aleem, “GoCJ: Google cloud jobs dataset for distributed and cloud computing infrastructures,” Data, Vol. 3, No. 4, p.38, 2018.

جدولة المهام المستقلة بالاعتماد على الخوارزمية الكشفية الفوقية في الحوسبة السحابية

دينا رياض عبد الرزاق[1] نرجس مزعل شاتي[2]
dinashibani@uomustansiriyah.edu.iq dr.narjis.m.sh@uomustansiriyah.edu.iq

حيدر كاظم حمود[3]
dr.haiduh@uomustansiriyah.edu.iq

المستخلص: تعد جدولة المهام موضوعًا مهمًا جدًا في سياق الحوسبة السحابية لأنها تؤثر على جودة خدمات الحوسبة السحابية المقدمة. في الأدبيات، تم تقديم العديد من الاستراتيجيات للتعامل مع جدولة المهام، مع تركيز الخوارزميات المتاحة على تقليل وقت التنفيذ مع تجاهل جودة الخدمة الأخرى. قدمت هذه الورقة خوارزمية جدولة مثالية لتقليل وقت التنفيذ واستهلاك الطاقة بالاعتماد على خوارزمية تلقيح الزهور FPA وخريطة هينون الفوضوية لجدولة المهام المستقلة. حيث تم استخدام خريطة Henon الفوضوية لإنشاء الحلول الأولية، ثم تم استخدام FPA لجدولة السحابات على الموارد الافتراضية المناسبة. تم إجراء الاختبارات على مجموعتين من البيانات. تمت دراسة نتائج خوارزمية الجدولة المقترحة ومقارنتها مع نتائج الخوارزميات الأخرى الموجودة في الأدبيات. أظهرت النتائج أن خوارزمية الجدولة المقترحة لها نتيجة أفضل من الخوارزميات المنافسة من حيث زمن التنفيذ واستهلاك الطاقة.

الكلمات المفتاحية: الحوسبة السحابية، استهلاك الطاقة، الخرائط الفوضوية.

[1] طالبة دكتوراه؛ قسم علوم الحاسوب - الجامعة المستنصرية - كلية العلوم – بغداد - العراق
[2] استاذ مساعد دكتور؛ قسم علوم الحاسوب - الجامعة المستنصرية - كلية العلوم – بغداد – العراق
[3] استاذ دكتور؛ قسم علوم الحاسوب - الجامعة المستنصرية - كلية التربية – بغداد - العراق