2022 12th International Conference on Computer and Knowledge Engineering (ICCKE) | 978-1-6654-7613-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICCKE57176.2022.9960123

An optimal workflow scheduling method in cloud-fog computing using three-objective Harris-Hawks algorithm

Ahmadreza Montazerolghaem
Department of Computer Engineering
University of Isfahan
Isfahan, Iran
a.montazerolghaem@comp.ui.ac.ir

Maryam Khosravi
Department of Computer Engineering
University of Isfahan
Isfahan, Iran
maryam.khosravi@eng.ui.ac.ir

Fatemeh Rezaee
Department of Computer Engineering
University of Isfahan
Isfahan, Iran
fatemeh.rezaee@eng.ui.ac.ir

Mohammad Reza Khayyambashi
Department of Computer Engineering
University of Isfahan
Isfahan, Iran
m.r.khayyambashi@eng.ui.ac.ir

***Abstract*— Today, the Internet of Things (IoT) use to collect data by sensors, and store and process them. As the IoT has limited processing and computing power, we are turning to integration of cloud and IoT. Cloud computing processes large data at high speed, but sending this large data requires a lot of bandwidth. Therefore, we use fog computing, which is close to IoT devices. In this case, the delay is reduced. Both cloud and fog computing are used to increasing performance of IoT. Job scheduling of IoT workflow requests based on cloud-fog computing plays a key role in responding to these requests. Job scheduling in order to reduce makespan time, is very important in realtime system. Also, one way to improve system performance is to reduce energy consumption. In this article, three-objective Harris Hawks Optimizer (HHO) scheduling algorithm is proposed in order to reduce makespan time, energy consumption and increase reliability. Also, dynamic voltage frequency scaling (DVFS) has been used to reduce energy consumption, which reduces frequency of the processor. Then HHO is compared with other algorithms such as Whale Optimization Algorithm (WOA), Firefly Algorithm (FA) and Particle Swarm Optimization (PSO) and proposed algorithm shows better performance on experimental data. The proposed method has achieved an average reliability of 83%, energy consumption of 14.95 KJ, and makespan of 272.5 seconds.**

Keywords—Internet of Things; Cloud-Fog computing; Harris hawks optimization algorithm; Workflow scheduling; DVFS

I. INTRODUCTION

The concept of IoT was first presented by Kevin Ashton [1] in 1999. Purpose of IoT is to collect environmental data through sensors and store and process them automatically [2,3]. IoT suffers from problems such as performance, security, privacy and reliability due to processing and storage power. Therefore, to solve these problems, we move towards cloud computing. Cloud computing processes huge amounts of data quickly and cheaply. To send this large data to the cloud, a lot of bandwidth is needed, and fog computing is used to solve this problem [4,5]. Like cloud computing, fog computing processes and stores data collected by sensors [6]. To reduce latency and network traffic volume in the cloud, data is temporarily processed and stored on network edge devices [7]. Fog computing has many advantages, including geographical distribution and large scale, lower operational costs, flexibility and heterogeneity and scalability, low latency [8]. In Fig. 1, cloud computing is first substrate, where cloud servers store and process large volumes of data. In next substrate, there is fog computing, which consists of fog machines. The bottom substrate contains the end devices of the IoT. Fog computing is a mediation between terminal devices and the cloud and it brings several services near to the terminal devices [8,26]. Each fog server is a virtual machine. To manage fog virtual resources, we use fog scheduling, which in addition to providing QOS, minimizes makespan, resource conception, and data transfer [18,19]. But scheduling jobs and workflow is an NP-hard problem [20]. Job scheduling in order to reduce makespan time, is very important in realtime system. Also, reducing the energy consumption improves system performance.

Therefore, in this contribution, an optimal workflow scheduling method is presented using three-objective HHO algorithm. In this schedule, we considered factors of makespan time, reliability, and energy consumption. DVFS are used to decrease energy consumption. DFVS is a method to reduce energy consumption by reducing processor frequency [21,22].

This paper is organized as follows: part II will review the previous works, part III presents system model and formulated job scheduling problem, part IV presents our method based on HHO algorithm, in part V presented evaluations and compared

978-1-6654-7613-3/22/$31.00 ©2022 IEEE

our proposed method with several Meta-Heuristic (MH) algorithms such as WOA, FA and PSO.

The MH algorithms such as HHO [23], Artificial Bee Colony (ABC) [24], and Grasshopper Optimization Algorithm (GOA) [25] have been used to solve various problems.

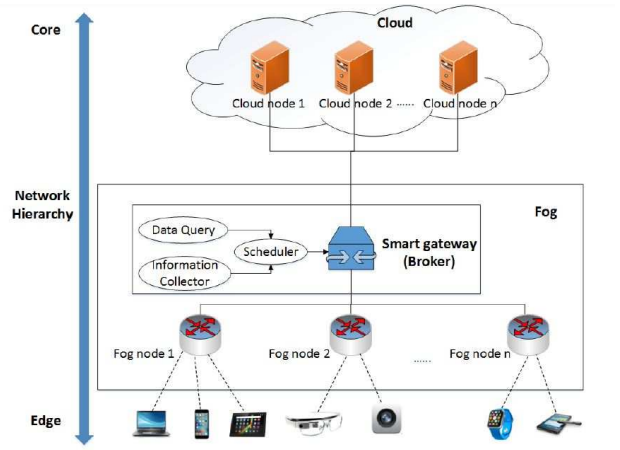

Fig. 1. Cloud-fog environment architecture[26]

II. RELATED WORK

The authors in [9] have developed a hybrid optimization algorithm called AEOSSA in cloud-fog environment. This algorithm was a combination of two algorithms, AEO and SSA, and its purpose is to reduce throughput time and makespan time. Then AEOSSA is compared with four algorithms SSA and AEO, PSO, FA on the two mentioned criteria and has shown better performance. In [10], genetic algorithm (GA) was used to minimize overall latency to schedule IoT jobs in the cloud-fog environment. The performance of GA has been better than that of Round Robin (RR), Priority-Strict Queuing (PSQ) and Wait-Fair Queuing (WFQ). techniques in reducing overall latency. The authors in [11] have created an optimal task offloading strategy for delay-sensitive and resource-sensitive programs in the cloud-fog environment, which uses the FA to reduce energy consumption and computational time. This method was compared with optimization algorithms such as PSO in terms of temperature emission, CO2 emission, energy consumption, and computational time, which has shown better performance. In [12], the Enhanced version of Multi-Verse Optimizer (EMVO) algorithm was presented for scheduling jobs in the cloud to achieve minimized makespan time and increase resources efficiency, and it was compared with two algorithms, MVO and PSO, which shown that the algorithm has a better performance. In [13], an optimization model using mixed integer programming for minimizing deadline misses of IoT tasks in the cloud environment using GA was introduced. Then the performance of GA was compared with priority scheduling and RR, which has shown better performance. In [14], a hybrid deep Q-learning task scheduling (DQTS) algorithm combining Q-learning and Artificial Neural Network (ANN) is presented for scheduling jobs in cloud. According to obtained results, the presented algorithm has less makespan and better load balance. The necessary requirements of fog computing that provides high-quality IoT services is efficient resource management. Job management and workflow are considered practical examples of resource management in fog computing environment for assigning a set of requested jobs to most suitable fog node [15-17]. None of the previous works have investigated the three parameters of Makespan time, reliability and energy consumption at the same time.

III. SYSTEM MODEL

A. System Model

Cloud-fog architecture has three substrates, as represented in Fig. 1. The edge substrate receives user requests through IoT devices. Then it sends the information to the fog substrate, which has fog nodes. Fog nodes process user requests. In this architecture, it is the cloud top substrate that provides outsourcing resources to execute workflows derived from the fog substrate. There is a smart gateway or broker in the fog substrate, that receives user requests and manages resources available in cloud and fog nodes, processing and communication costs, and the search results of data return from nodes and create most suitable workflow scheduling [26].

B. Problem Formulation

In this part, the formulation of the problem for job scheduling in cloud-fog is described.

The cloud-fog system consist of a set of M heterogeneous virtual machines denoted by $VM = \{v_1, v_2, \ldots, v_M\}$. Each v is physical processor that works in various frequency levels and voltage and is determined {SV, c, f} supply voltage, maximum processor processing capacity, and frequency level.

The workflow is set of jobs, denoted by W and model by a Directed Acyclic Graph (DAG). $J = \{J1, J2, \ldots, JN\}$ illustrates a set of N related jobs of workflow. E illustrates set of edges between jobs and is N*M.

Each job consists of several instructions per million units to executed and a collection of predecessors and successors. As shown in Fig. 2, The entry job for DAG has $Pred\ (J_{entry}) = \Phi$ and the exit job has $Succ(J_{exit}) = \Phi$. Each edge $e_{i,j} \in E$ shows the graft between J_i and J_j , which has weight $W_{i,j}$, illustrates the amount of data need to be exchanged between job J_i and J_j [22,27].

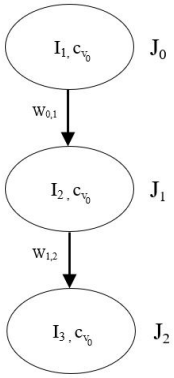

Fig. 2. DAG of workflow[27]

Each job is allocated to a *VM* with a special frequency. The execution time of job J_i allocated to $v_{r_{fy}}$ is represented by:

$$\text{ET} = \frac{I_i}{c_{v_r}} \times \frac{v_{r_{fmax}}}{v_{r_{fy}}} \quad (1)$$

In the equation: I_i−number of instructions that need to execute for J_i;
c_{v_r} −maximum capacity of v_r in MIPS;
$v_{r_{fmax}}$ −maximum frequency level for v_r;
$v_{r_{fy}}$ −frequency level f_y that has any value from $f = \{f_1, f_2, \dots\dots, f_L\}$ for v_r .

Now we use DVFS to reduce energy consumption by reducing frequency [16].

DVFS is a mechanism that decreases energy consumption by reducing the CPU frequency level of VM and uses a powerful management technique to decrease energy consumption and high reliability. To maintain reliability. It uses a powerful management technique to decrease energy consumption and high reliability [16].

IV. PROPOSED METHOD BASED ON HARRIS HAWKS OPTIMIZATION

HHO is a population-based MH algorithm. MH algorithms mimic natural phenomena [28]. MH algorithms include two phases: exploration and exploitation. To schedule jobs, we use the three-objective HHO algorithm to minimizing makespan and energy consumption and increasing reliability. A candidate solution in the optimization problem was considered for each hawk [28].

$$fitness(t) = min_{i\in 1,2,\dots,M} \sum_{k=1}^{N} \frac{Makespan_{k,i} + Energy\ Consumption_{k,i}}{Reliability_{k,i}} \quad (2)$$

For each time of t makespan calculated by:

$$\text{Makespan(t)} = max_{i\in 1,2,\dots,M} \sum_{k=1}^{N} ET_{k,i} \quad (3)$$

In the exploration phase and at the beginning of the optimization, the position of the hawks is randomly selected and the hawks are distributed in the entire search space according to equation (4). As a result, they increase the algorithm exploration power to find the global solution. Then the objective function calculates for each hawk and the hawk with the lowest value is select as the candidate and the position of the other hawks update according to the equation (4) [28].

$$X(t+1) = \begin{cases} X_{rand}(t) - r_1|X_{rand}(t) - 2r_2X(t)|, & q \geq 0.5 \\ \big(X_{rabbit}(t) - X_m(t)\big) - r_3\big(LB + r_4 - (UB - LB)\big), & q < 0.5 \end{cases} \quad (4)$$

In the equation: X(t+1) −location vector of harris hawks in the next repetition;
X_{rabbit} (t) −location of prey(rabbit);
X(t) −location vector of harris hawks, r_1 to r_4;
q−random numbers in (0,1), that update in each repetition;
L_B and U_B−upper bound and lower bound of decision variables that in this task scheduling problem are in order 1, M;
X_{rand}(t) −Choose a hawk from prevalent population at random;
X_m−average location of the hawks' prevalent population.

The average position of hawks is attained using Eq. (5) [28]:

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t) \quad (5)$$

In the equation X_i(t) shows each hawk position in repetition t and N illustrates total number of hawks.

Based on escape energy of prey, the HHO algorithm changes from exploration phase to exploitation phase and between various exploitative behaviors.

E indicates prey's escape energy. when |E| ≥1, exploration process happens, and when |E| <1 exploitation process happens in later steps. In exploitation phase, if r <0.5, luck of prey escaping will be successful, and if r≥0.5, luck of prey escaping before surprise attack will not be successful [28].

When r ≥ 0.5 and |E| ≥ 0.5, Harris hawks' position in soft besiege calculated by Eq. (6,7):

$$X(t+1) = \Delta X(t) - E|2(1 - r_5X_{rabbit}(t) - X(t)| \quad (6)$$

$$\Delta X(t) = X_{rabbit}(t) - X(t) \quad (7)$$

In the equation: ΔX(t)− difference between rabbit position vector and current location in repetition t;
r_5− a random number in (0,1) [28].

When r ≥ 0.5 and |E| < 0.5, Harris hawks' position in hard besiege calculated by Eq. (8):

$$X(t+1) = X_{rabbit}(t) - E|\Delta X(t)| \quad (8)$$

The more intelligent way is when still |E| ≥ 0.5 but r < 0.5 a soft besiege is constructed before surprise attack.

HHO supposed that the harris hawks can assess their next movement by Eq. (9):

$$Y = X_{rabbit}(t) - E|JX_{rabbit}(t) - X(t)| \quad (9)$$

This mean harris hawks start making abrupt, irregular, and rapid dives towards the prey using Eq. (10):

$$Z = Y + S \times LF(D) \quad (10)$$

In the equation: S−random vector by size 1 × D;
D−dimension of problem;
LF−levy fly function.

LF can performed by Eq. (11,12) [28]:

$$LF(x) = 0.01 \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \quad (11)$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}} \right)^{\frac{1}{\beta}} \quad (12)$$

In the equation: u, v−random values in (0,1);
($\beta = 1.5$, a constant)

In soft besiege, Eq. (13) is used to update location of the hawks:

$$X(t+1) = \begin{cases} Y & if\ F(Y) < F(X(t)) \\ Z & if\ F(Z) < F(X(t)) \end{cases} \quad (13)$$

When r < 0.5 and E < 0.5, before surprise jump, a hard besiege is performed by hawks. In this case, a situation similar to soft besiege is formed and hawks are trying to reduce their average distance with prey Eq. (13) [28]. Therefore, this process is repeated until stopping condition is maintained. Stopping condition in this research is to reach the 100th iteration. Then comes output of algorithm which is an optimum job scheduling.

V. EVALUATION

In this article, two workflow datasets including Montage and LIGO with sizes of 25, 50, 100, 1000 jobs (KB) taken from [29] are used as the tested dataset. Also, all programming has been done with MATLAB 2018b software on Windows 64-bit operating system and 5-core system.

From the criteria of fitness function, reliability, makespan time and energy consumption to evaluate HHO and compare it with FA [30], PSO [31], and WOA [32] has been used.

In TABLE I, fitness function value obtained by the algorithms for the two datasets used in this article (Montage and LIGO with different number of jobs) is shown. In this article, four PSO, FA, HHO and WOA algorithms are compared under the same conditions (same dataset, same fitness function, same number of iterations, same number of population members). Considering that our problem is a minimization problem, therefore, the lower fitness function value obtained by an algorithm, the algorithm provides better solutions.

The four related algorithms obtain the optimization process to solve problems by using two phases of exploration and exploitation. Structure of four related algorithms is different from each other and each algorithm uses different operators to complete the two phases to perform optimization. Therefore, according to difference in structure of algorithms, output of each algorithm is definitely different from other algorithms.

TABLE I. FITNESS FUNCTION RESULTS FROM ALGORITHMS FOR MONTAGE AND LIGO DATASETS (KB).

Algorithm				
	Montage25	***Montage50***	***Montage100***	***Montage1000***
PSO	0.483	0.478	0.456	0.478
FA	0.412	0.425	0.435	0.445
WOA	0.398	0.380	0.395	0.411
HHO	0.322	0.342	0.350	0.389
Algorithm				
	LIGO25	***LIGO50***	***LIGO100***	***LIGO1000***
PSO	0.476	0.477	0.456	0.467
FA	0.438	0.446	0.435	0.427
WOA	0.411	0.410	0.395	0.389
HHO	0.398	0.382	0.350	0.365

TABLE II. RELIABILITY (%) RESULTS FROM ALGORITHMS FOR FOR MONTAGE AND LIGO DATASETS (KB).

Algorithm				
	Montage25	***Montage50***	***Montage100***	***Montage1000***
HHO	83	82	84	83
WOA	78	77	79	79
FA	75	75	75	76
PSO	72	71	72	74
Algorithm				
	LIGO25	***LIGO50***	***LIGO100***	***LIGO1000***
HHO	82	82	83	84
WOA	78	77	79	81
FA	74	74	76	78
PSO	71	70	72	75

According to TABLE I, HHO obtained a lower value of fitness function than other algorithms, then WOA obtained the lowest value of fitness function, then FA and then PSO. That is, PSO has obtained the worst performance in terms of obtaining fitness function value.

TABLE II shows reliability value obtained for different algorithms on two datasets. According to what we described in previous paragraphs, the reliability value obtained by HHO is higher than other three algorithms, and reliability value obtained by PSO is the lowest.

Fig. 3. displays reliability graph for Montage dataset and Fig. 4. displays reliability graph for LIGO dataset from TABLE II. According to obtained results, HHO has an average reliability of 83%. Meanwhile, average reliability for WOA, FA, and PSO is 78.75, 75.5, and 72, respectively.

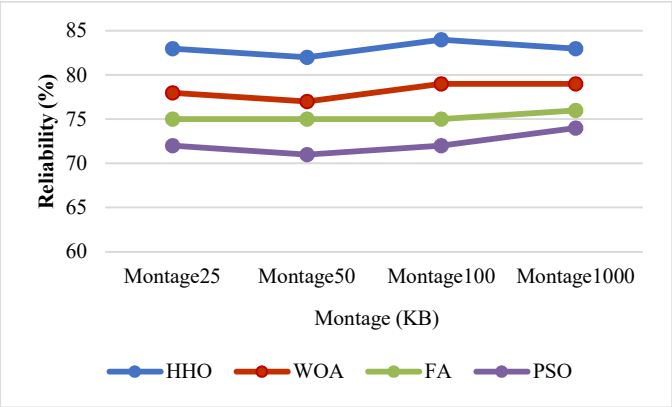

Fig . 3. Reliability (%) plot for the Montage dataset

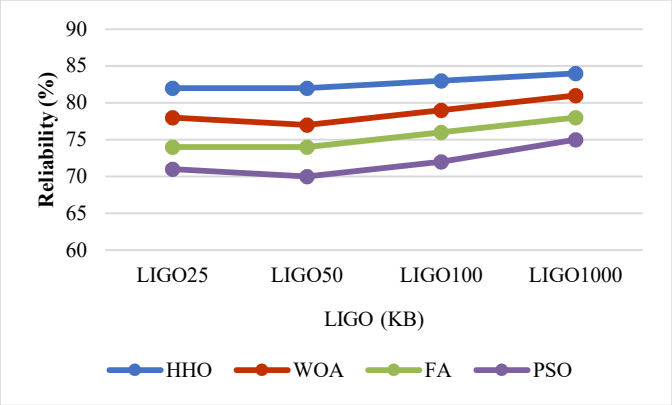

Fig. 4. Reliability (%) plot for the LIGO dataset

TABLE III shows the amount of energy used to execute different workflows by different algorithms. As can be seen from TABLE III, HHO consumed the least amount of energy and PSO consumed the most amount of energy. It should be noted that with the increase in the size of workflows (increasing the number of jobs in each workflow), the amount of energy used to execute them increases. After HHO, WOA consumed the least energy, followed by FA.

Fig.5. shows energy consumption graph from TABLE III for the Montage dataset and in Fig. 6. energy consumption graph for LIGO dataset. According to obtained results, HHO has an average energy consumption of 14.95 kJ. While average energy consumption for WOA, FA and PSO is equal to 16.6, 17.9 and 19.4, respectively.

In TABLE IV, makespan time value obtained by different algorithms for two different datasets with different number of jobs is given.

As shown in TABLE IV, the lowest value of makespan time is obtained by HHO, then WOA, then FA and then PSO.

Also, by increasing the number of jobs in each workflow, amount of makespan time obtained by each algorithm has also increased.

TABLE III. ENERGY CONSUMPTION (KJ) RESULTS FROM ALGORITHMS FOR MONTAGE AND LIGO DATASETS (KB).

Algorithm				
	Montage25	***Montage50***	***Montage100***	***Montage1000***
HHO	7.12	12.21	16.25	24.23
WOA	8.35	13.85	18.64	25.56
FA	9.50	15.10	19.98	27.04
PSO	11.2	17.02	21.03	28.37
Algorithm				
	LIGO25	***LIGO50***	***LIGO100***	***LIGO1000***
HHO	10.34	12.25	14.37	25.12
WOA	11.62	13.47	15.64	27.36
FA	12.75	14.56	16.76	28.97
PSO	13.89	15.73	17.78	30.46

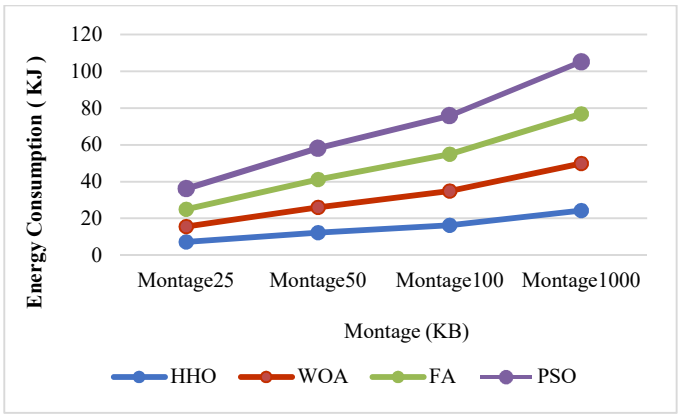

Fig. 5. Energy Consumption graph for the Montage dataset

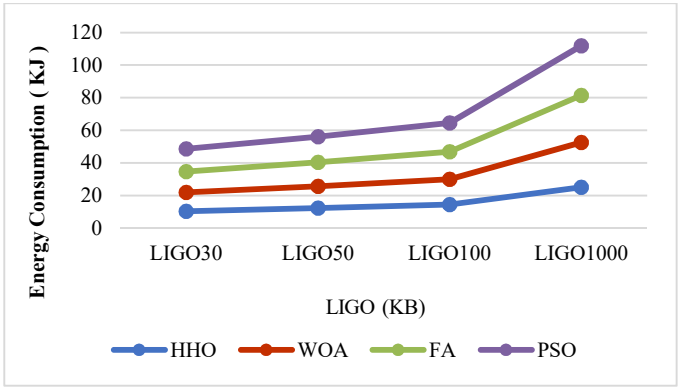

Fig. 6. Energy Consumption chart for the LIGO dataset

TABLE IV. MAKESPAN TIME (S) RESULTS FROM ALGORITHMS FOR MONTAGE AND LIGO DATASET (KB).

Algorithm				
	Montage25	***Montage50***	***Montage100***	***Montage1000***
HHO	120	200	270	500
WOA	150	225	290	525
FA	175	240	315	550
PSO	195	260	340	570
Algorithm				
	LIGO25	***LIGO50***	***LIGO100***	***LIGO1000***
HHO	125	185	250	520
WOA	145	200	275	550
FA	160	215	290	585
PSO	180	235	320	600

Fig. 7. shows the makespan time plot from TABLE IV for the Montage dataset. The results show HHO has an average makespan time of 272.5 seconds. While average makespan time for WOA, FA and PSO is equal to 297.5, 320 and 341.25 respectively.

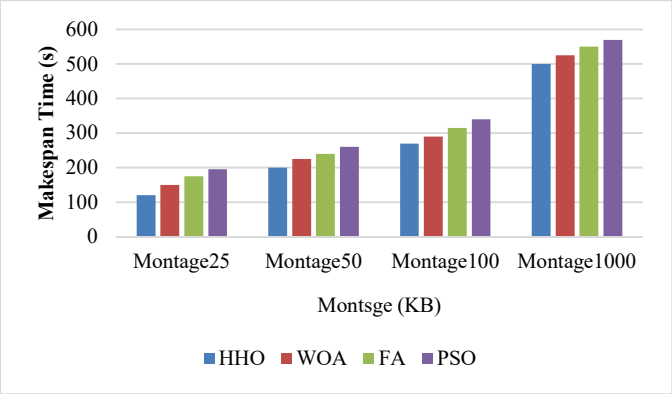

Fig. 7. Makespan Time chart for the Montage dataset

Fig. 8. shows Makespan Time plot from TABLE IV for LIGO dataset.

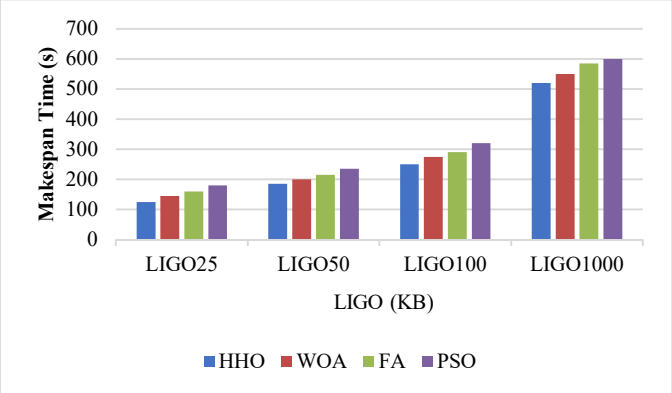

Fig. 8. Makespan Time diagram for the LIGO dataset

Considering that HHO obtained better results in terms of makespan time, reliability, fitness function value and energy consumption value, we conclude that HHO has better performance than other algorithms. As mentioned earlier, this difference in value of results obtained by algorithms is related to structure of algorithms. The structure of HHO is such that it can get better results than PSO, FA and WOA.

VI. CONCLUSION

In this article, the optimization algorithm of Harris Hawks was used as an optimization algorithm to optimize workflow scheduling problem in cloud-fog environment. The three criteria of energy consumption, reliability and makespan time were used as main criteria to determine fitness function, and final fitness function was obtained from combination of these three criteria. Finally, Harris Hawks algorithm was able to determine the best and most optimal virtual machines for workflows by using the minimization of fitness function, so that the lowest energy consumption, highest reliability and the lowest amount of makespan time are compared to firefly and whale algorithms and obtain a swarm of particles.

REFERENCES

[1] K. Ashton, “That ‘Internet of Things’ Thing,” RFiD J., 2009, pp. 49–86

[2] Giang, N.; Kim, S.; Kim, D.; Jung, M.; Kastner, W. Extending the EPCIS with Building Automation Systems: A New Information System for the Internet of Things. In Proceedings of the 2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, Birmingham, UK, 2–4 July 2014; pp. 364–369.

[3] Atlam, H.F.; Alenezi, A.; Hussein, R.K.; Wills, G.B. Validation of an Adaptive Risk-Based Access Control Model for the Internet of Things. Int. J. Comput. Netw. Inf. Secur. 2018, 10, 26–35. [CrossRef] performance, security, privacy and reliability

[4] Atlam, H.F.; Alenezi, A.; Alharthi, A.; Walters, R.; Wills, G. Integration of cloud computing with internet of things: challenges and open issues. In Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Exeter, UK, 21–23 June 2017; pp. 670–675.

[5] Ai, Y.; Peng, M.; Zhang, K. Edge cloud computing technologies for internet of things: A primer.Digit. Commun. Netw. 2017, in press.

[6] Peter, N. FOG Computing and Its Real Time Applications. Int. J. Emerg. Technol. Adv. Eng. 2015, 5, 266–269.

[7] Wen, Z.; Yang, R.; Garraghan, P.; Lin, T.; Xu, J.; Rovatsos, M. Fog orchestration for internet of things services. IEEE Internet Comput. 2017, 21, 16–24.

[8] H. F. Atlam, R. J. Walters, and G. B. Wills, "Fog computing and the internet of things: A review," big data and cognitive computing, vol. 2, no. 2, p. 10, 2018.

[9] Abd Elaziz, M., Abualigah, L. and Attiya, I., 2021. Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments. Future Generation Computer Systems, 124, pp.142-154.

[10] Aburukba, R.O., AliKarrar, M., Landolsi, T. and El-Fakih, K., 2020. Scheduling Internet of Things requests to minimize latency in hybrid Fog–Cloud computing. Future Generation Computer Systems, 111, pp.539-551.

[11] Adhikari, M. and Gianey, H., 2019. Energy efficient offloading strategy in fog-cloud environment for IoT applications. Internet of Things, 6, p.100053.

[12] Shukri, S.E., Al-Sayyed, R., Hudaib, A. and Mirjalili, S., 2021. Enhanced multi-verse optimizer for task scheduling in cloud computing environments. Expert Systems with Applications, 168, p.114230.

[13] Aburukba, R.O., Landolsi, T. and Omer, D., 2021. A heuristic scheduling approach for fog-cloud computing environment with stationary IoT devices. Journal of Network and Computer Applications, 180, p.102994.

[14] Tong, Z., Chen, H., Deng, X., Li, K. and Li, K., 2020. A scheduling scheme in the cloud computing environment using deep Q-learning. Information Sciences, 512, pp.1170-1191.

[15] L. Yin, J. Luo, and H. Luo, "Tasks scheduling and resource allocation in fog computing based on containers for smart manufacturing," IEEE Transactions on Industrial Informatics, vol. 14, pp. 4712-4721, 2018.

[16] O. H. Ahmed, J. Lu, A. M. Ahmed, A. M. Rahmani, M. Hosseinzadeh, and M. Masdari, "Scheduling of Scientific Workflows in Multi-Fog Environments Using Markov Models and a Hybrid Salp Swarm Algorithm," IEEE Access, vol. 8, pp. 189404-189422, 2020.

[17] D. Tychalas and H. Karatza, "A scheduling algorithm for a fog computing system with bag-oftasks jobs: Simulation and performance evaluation," Simulation Modelling Practice and Theory, vol. 98, p. 101982, 2020.

[18] J. C. Guevara and N. L. da Fonseca, "Task scheduling in cloud-fog computing systems," Peer-to-Peer Networking and Applications, vol. 14, pp. 962-977, 2021.

[19] A. Arunarani, D. Manjula, and V. Sugumaran, "Task scheduling techniques in cloud computing: A literature survey," Future Generation Computer Systems, vol. 91, pp. 407-415, 2019.

[20] J. Wan, B. Chen, S. Wang, M. Xia, D. Li, and C. Liu, "Fog computing for energy-aware load balancing and scheduling in smart factory," IEEE Transactions on Industrial Informatics, vol. 14, pp. 4548-4556, 2018.

[21] Ahmed, O.H., Lu, J., Xu, Q., Ahmed, A.M., Rahmani, A.M. and Hosseinzadeh, M., 2021. Using differential evolution and Moth–Flame optimization for scientific workflow scheduling in fog computing. Applied Soft Computing, 112, p.107744.

[22] Hassan, H.A., Salem, S.A. and Saad, E.M., 2020. A smart energy and reliability aware scheduling algorithm for workflow execution in DVFS-enabled cloud environment. Future Generation Computer Systems, 112, pp.431-448.

[23] Raziani, S.; Salehnia, T.; Ahmadi, M. Selecting of the best features for the knn classification method by Harris Hawk algorithm. In Proceedings of the 8th International Conference on New Strategies in Engineering, Information Science and Technology in the Next Century, Dubai, United Arab Emirates (UAE), 2021; Available online: https://civilica.com/doc/1196573/ (accessed on 16 March 2022).

[24] Salehnia, T., Fath, A., 2021, Fault tolerance in LWT-SVD based image watermarking systems using three module redundancy technique. Expert Systems with Applications. 179, 115058.

[25] Salehnia, T.; Izadi, S.; Ahmadi, M. Multilevel image thresholding using GOA, WOA and MFO for image segmentation. In Proceedings of the 8th International Conference on New Strategies in Engineering, Information Science and Technology in the Next Century, Dubai, United Arab Emirates (UAE), 2021; Available online: https://civilica.com/doc/1196572/ (accessed on 16 March 2022).

[26] Pham, X.Q. and Huh, E.N., 2016, October. Towards task scheduling in a cloud-fog computing system. In 2016 18th Asia-Pacific network operations and management symposium (APNOMS) (pp. 1-4). IEEE.

[27] Guevara, J.C. and da Fonseca, N.L., 2021. Task scheduling in cloud-fog computing systems. Peer-to-Peer Networking and Applications, 14(2), pp.962-977.

[28] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M. and Chen, H., 2019. Harris hawks optimization: Algorithm and applications. Future generation computer systems, 97, pp.849-872.

[29] https://confluence.pegasus.isi.edu/display/pegasus/Deprecated+Workflow+Generator.

[30] X.-S. Yang, “Firefly algorithms for multimodal optimization”, Foundations and Applications, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 169–178, 2009.

[31] J. Kennedy, R. Eberhart, “Particle swarm optimization”, International Conference on Neural Networks, vol. 4, pp. 1942–1948, 1995.

[32] S. Mirjalili and A. Lewis, “The whale optimization algorithm,” Advances in engineering software, vol. 95, pp. 51-67, 2016.