Multi-variable Stress balancing wireless method based on the status of the machines in the cloud spaces by neural networks

Sedighe Zakeri[1]*, Karthikeyan Krishnasamy[2], B. Gomathi [3]

1 Department of Information Technology, College of Engineering and Technology, Iran,
*Corresponding author. Email: *sed.zakeri66@gmail.com*
2 Department of Information Technology, Karpagam College of Engineering, Coimbatore, 641 032, India., Email: karthiaish1966@gmail.com
3 Department of Information Technology, Hindusthan College of Engineering and Technology, Coimbatore, 641 032, India. , Email: gomathib@hindusthan.net

Abstract: Cloud computations are based on the computer networks such as Internet which presents a new pattern to provide, consume and deliver services such as infrastructure, software, ground and other resources using network. The inappropriate timing of assigning loads to the virtual machines in the computational space could lead to unbalance in the system. One of the challenging planning problems. In the cloud data centers is considering both assigning and migration (transfer) of the virtual machines with the ability of reconfiguration and the integrated features of the hosting physical machines. In this article, we introduce an integrated and dynamic timing algorithm based on the Genetic evolution algorithm. The suggested method was evaluated based on these factors and different inputs. Our suggested method is done using Java programming language and cloud-SME simulation. The results show that the execution time and the response time were improved by 12 and 1 percent respectively.

Keywords: Cloud computations, task timing, genetics algorithm, response time, virtual machine

1. Introduction

Preface Internet, since its beginning, has seen many changes, some of them have led to a change in humans life style in recent decades. One of these changes happened with the introduction of cloud computations. This technology has become popular because of its features, because all services are available for users as one service. Naturally, any change and new concept in the technology world has its own problems and complexities. Using the cloud computations is of no exception and has laid many challenges in front of the experts of this field, such as load balancing, security, assurance capability, ownership, backing up the data and portability. Considering the importance of load balancing process in cloud computations to improve users' satisfaction, we intend to assign the load to the appropriate machines as the input of cloud space
Using a load balance method based on the genetics evolution algorithm with several targets.
Cloud computations are one of the most recent changes in information technology and will become world wide with time. Considering the market, perceiving the load balancing effects in the cloud is important. The cloud computations ground is an automatic service provider ground which allows users to buy, remote control, dynamic scalability and system management. Currently, load balancing in the cloud computation systems is a challenge [1]. Load balancing is reassigning the whole load to particular vertices of a group system to use resources more efficiently, improving the response time of a task in a situation that some vertices are wide open, while others have less loads. Load balancing is a necessary solution for increasing the service level agreement and using the resources more efficiently. The considered load could be the processor load, the used memory, network delay or load [2].
Recently, multi-variable methods have been studied extensively for their wide application in various industries such as robotics, IOT and multiconductor transmission lines [3-9]. It seems that neural network has no issue but one problem of neural networks could be having to fix the depth of the network during training that prior synthesis methods suffer from, one feasible solution according to Hassantabar et al. is a

synthesis methodology that can generate compact and accurate neural networks [10-11]
There are different methods to solve this issue, Mohammadzadeh et al. suggest particle swarm optimization method [12-13] or Afshar implements Population Balance Modeling (PBM) and Initial Development Method (IDM) [14-15] on the other hand Azarang et al. describe about Laplacian-based and Nonfragile fuzzy method [16-18]. Ramtin et al. have studied Self-stabilizing algorithms [19-21]. Khiabani et al. have conducted research on design and implementation of an optimal switching controller based on State-Dependent Riccati equation [22-24]. Razzaghi has worked on nonlinear dynamics and control of an inertially actuated jumper robot [25-26], and on the following dynamic addressing by Ghayouraneh et al. who have worked on Holistic optimization technique for solving low thermal conductivity [27-31]. Many other researches have been conducted in this field by Gharib [32] in quantitative-fuzzy Controller Design for Multivariable Systems with Uncertainty. Hemmati et al. [33-35] have used metaheuristic optimization techniques for solving problems in this field.

In this paper, we intend to present a method for load balancing in the cloud computation spaces based on the line length of virtual machine's transactions, the average waiting time, the average response time, the number of failed transactions, the used memory and the level of processor. In the suggested method, you choose a virtual machine to assign the load after calculating the variables of the multi target genetics algorithm.

2. Framework of Method

There are many researches done in the field of timing algorithms. Most of them focus on the load balancing for the traditional web servers. One of the challenging planning problems in the cloud data centers is considering both assigning and migration (transfer) of the virtual machines with the ability of reconfiguration and the integrated features of the hosting physical machines. Despite the traditional load balancing algorithms which only are appliable to the physical servers considering the different efficiency factors and other factors such as CPU load, memory and bandwidth, integrated network for both physical and virtual machines, in this article we create an integrated to balance the load on each server and introduce an integrated and dynamic timing algorithm based on the genetics evolution algorithm which corrects the CPU, memory and bandwidth for both physical and virtual machines. The efficiency average of CPU in a singular server called i is shown with CPU_i^U. To achieve this factor, we take the average of the CPU efficiency during the observance period.

$$CPU_u^A = \frac{\sum_i^N \left(CPU_i^U\right) CPU_i^n}{\sum_i^N CPU_i^n} \qquad (1)$$

N shows the whole number of physical servers in a cloud data center. Similarly, the memory usage average, the bandwidth of server i and all memories and all the bandwidth in a cloud data center could be defined as NET^u_i, $NEM^u_{i,}$ NET_u^A and NEM^A_u respectively.

The integrated load unbalance level (ILB_i) of the server i. variance is a factor for showing how separate are the numbers in a set statistically.

And then, the created variance for the load balance of the average unbalance level of a cloud data center is calculated:

$$IBL_{avg}^{CDC} = \frac{IBL_{cpu} + IBL_{mem} + IBL_{net}}{N} \qquad (2)$$

3. The genetics algorithm to assign resources

Due to the importance of the locating virtual machines in the cloud data centers problem, there have been many researches in recent yeas in this field. In this article, we have provided the multi-target genetics algorithm on the timing problems of the independent tasks. The advantage of this method is using an algorithm such as genetics which is very strong for exhaustive search and is convergent to the optimized answer with a quick and random search in the problem space. Therefore, we obtain good results from it. the main idea of this algorithm is transferring the hierarchical features using Genomes.

3.1. Chromosome display

A schematic planning design could be stated by a chromosome which is coded like vertex-task [4]. The

length of chromosome (the number of genomes) is equal to the number of virtual machines. The value of a genome equals the number of physical machines which its task matches the execution. Considering the below chromosome:
{1,3,2,4,3,1,4,2,3,1,3,2,4,1}
We can conclude that the first virtual machine is executed in the first cloud data center, and the second virtual machine is executed in the third cloud data center. Node$_i$ shows the ith physical machine which shows the related virtual machines.

3.2. The initial value of the population

Simple GAs create the initial population randomly and persons have low performances. In this algorithm, greedy algorithm is used. We create the initial population via the best choice based on current situation. Here, we assume that the virtual machine is executed enough and there are no priorities.

$$Node_1: \{1,6,10\}, Node_2: \{3,8,12\}, Node_3: \{2,5,9,11\}, Node_4: \{4,7,13\},$$

3.3. Compatibility function

In natural world, compatibility is the ability of reproduction. The genetics algorithm imitates the principal of the most compatible nature survival to do the searching process in which the compatibility function is the quantitative factor of persons in the population [36]. The operators include selection, intersection and mutation are all based on compatibility. The persons performance reflects the convergence of the algorithm and the search of the best relative solution directly. More compatibility means better performance an vice versa. More compatible persons may reproduce while other might get extinct. The compatibility function is the driver force of the genetics algorithm.
Load balancing is distributing the requests between several virtual machines which provide the same services for the users. The load balancing is done clearly, which means that the process is hidden from the users and users don't know about the load balancing and which machine in the cloud answers them. in this article we define a compatibility function based on several targets. The efficiency goals include these:

1) Less usage time
2) Less variance for the load of data centers and physical machines
3) Response time
4) The remaining memory

The suggested method must create a suitable balance between these goals and also, on the other hands, these goals may have different priorities based on different applications. Therefore, to achieve this goal, we must give the user this privilege to create an appropriate balance between these factors and to do so, the user assigns weights to these goals. Therefore, the eligibility function of the suggested method is as below[37]:

$$Fitness = \alpha \times ILB + \beta \times total\ Time + \varphi \times WT + \rho \times Mem_{ava} \quad (3)$$

In which α and β are different weight coefficients for each efficiency goals and $\alpha+\beta+\varphi+\rho=1$

3.4. Selection strategy

Although using the spin selection algorithm, the intersection and mutation operation we can make sure of population's variety, the better performing person may get destroyed with these actions. To prevent this, the experts are copies to the next generation without other operators. In this article, expertism is used to allow some better organisms of the current generation to be transferred to the next generation without change. Theories have proved that the genetics algorithm has a good convergence with the expertism selection algorithm.

4. Implementation environment

To implement the suggested method, we have used the cloudSim 3.0.3 cloud computation simulator software with Java programming language. This simulation was done in the computer with a core i7 processor and 8 GB main memory and on windows 10.

The size of the population was 40 and the number of repetitions were 30. The goal of optimization is obtaining an optimized operational series which lead to the minimum cost. Any of these costs were obtained after 10 repetition is each of the executional conditions.

5. Comparing the whole execution time of the suggested method with other algorithms in different states

In this part, the suggested method based on the multi-target genetics hyper-innovative algorithm with some methods of task timings in cloud computation space. Since this article is the first one to use multi target genetics algorithm to time and write the tasks in the distributed space, there are no similar methods to do the comparison based on them. therefore, we use other researches in this field which have used other hyper-innovative and evolution algorithms for timing to compare and analyze the results of the suggested method. We use the algorithms in [38], [39] and [40] to do the comparison.

Table 1: comparing the execution time of different methods

Number of tasks	**The whole execution time of**			
	the suggested method	the method in [38]	the method in [39]	the method in [40]
200	35	40	43	51
400	80	90	98	102
600	150	170	185	192
800	195	230	254	265
1000	285	320	361	382
1200	370	440	489	503
1400	540	590	623	645
1600	620	680	730	751
1800	735	800	840	878
2000	810	880	921	953

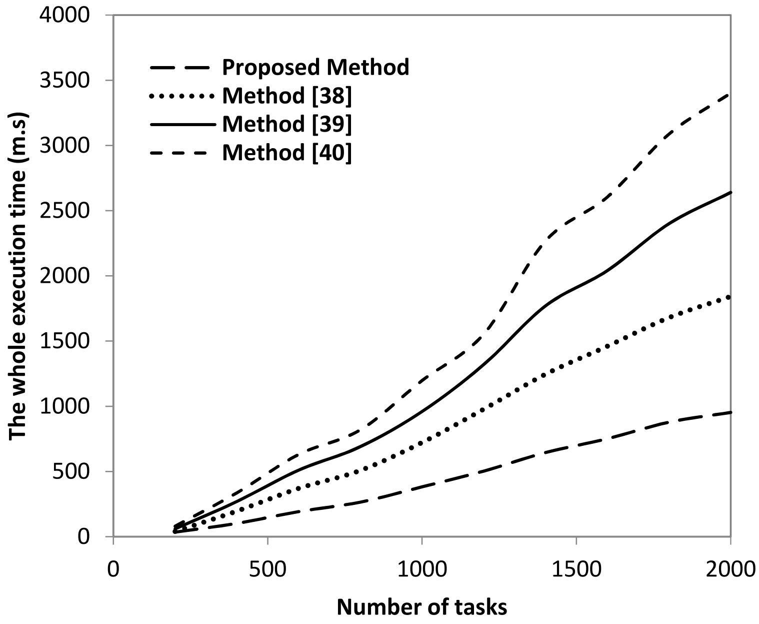

Figure 1: comparing the execution time of the different methods

The suggested method optimized the load using the multi-target genetics algorithm with dynamically transferring the local work load from a machine to another machine in the remote vertex or a less-used machine and by doing so, it reduced the whole execution time. This operation maximized the user satisfaction, minimizes the response time and increase the efficiency of resources and reduce the works' trace. The execution time was compared in this part. In next parts, the rest of efficiency goals will be analyzed.

6. Comparing based on the response time factors

The response time is the time distance between the execution request and the beginning of execution by the processor in the host cloud data centers. This factor is a standard factor in the field of timing. In this part, we will have a comparison between the suggested method and the researches done based on the response time. In fact, this factor represents the justice ratio via answering when the resources are appropriately used. To compare the suggested method with the past researches, we consider some states based on the number of different tasks. Table 2 shows the results clearly.

Table 2: the comparison of response time of different methods

Number of tasks	The whole execution time of			
	the suggested method	the method in [38]	the method in [39]	the method in [40]
200	1.6	1.7	2.0	2.2
400	1.8	1.9	2.1	2.3
600	1.9	2.1	2.2	2.5
800	2.2	2.4	2.6	2.8
1000	2.5	2.8	2.9	3.2
1200	2.8	3.1	3.4	3.5
1400	3.1	3.3	3.8	3.9
1600	3.6	4.0	4.2	4.4
1800	3.9	4.1	4.4	4.6
2000	3.5	4.4	4.7	4.8

doi:10.20944/preprints202005.0174.v1

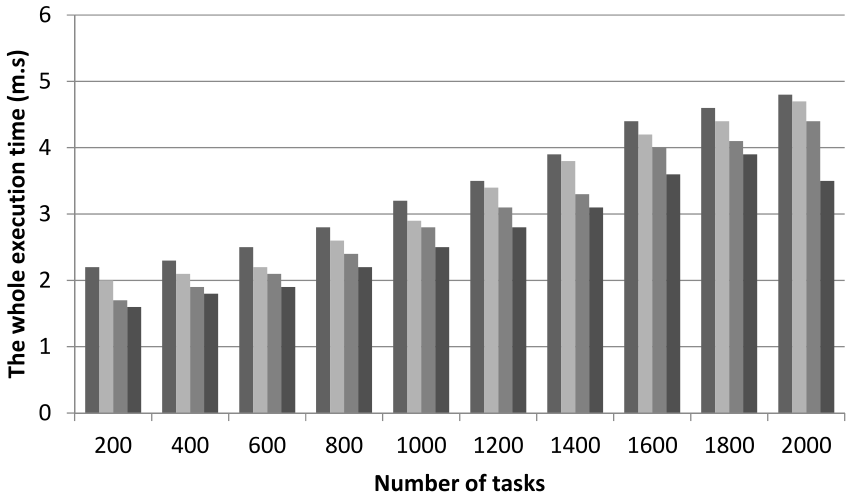

Figure 2: the comparison of response time of different methods

The suggested method clearly improves the tasks response time compared to past researches. The suggested method balances the load for distributed computation systems while saving the costs and is efficient for creating balance in the input tasks.

7. Evaluating the load balance of processors based on the number of hosts

Efficiency states the ratio of execution time of a computation resource in the cloud computation system to the whole time. In fact, this factor shows the appropriate use of resources. The value of this factor is a number between 0 and 1. The closer is this number to 1, the more efficiency and when we get far from 1 towards 0, it shows the grade computations resources waste. We must pay attention that the efficiency level must be active in all processors with almost equal ratios to have justice in distributing the tasks between the processors. To compare the suggested method to the other methods, the execution will be done based on the change of the number of hosts and the efficiency of the computational resources will be evaluated. The number of hosts used on this article is as same as the compared previous research which is 50. By changing the number of hosts, the load of any host changes, but we must pay attention that the suggested method must be able to create an appropriate load balance between the cloud hosts and distribute a suitable number of tasks between them. the standard deviation of the efficiency variable of the different processors will be used to show the suitable balance. Table 3 shows the efficiency of the suggested method in different states.

Table 3: analyzing the level and standard deviation of processors efficiency

The number of hosts	The efficiency percentage	Standard deviation
10	0.99	0
20	0.97	0.02
30	0.96	0.02
40	0.94	0.03
50	0.91	0.03

The presented result shows the appropriate efficiency of the suggested method. Furthermore, further study of table 3 shows that the standard deviation of the processors is very low and the suggested method is able to create almost equal efficiency between the processors and manage the dynamic conditions of the problem.

8. Conclusion

The suggested method of this article was simulated and different comparisons were carried out to evaluate the performance of the suggested method comparing to other algorithms. The results show that the suggested algorithm performs better than the algorithms in past researches. We compared our suggested algorithm with other existing techniques. The results show that the suggested algorithm reduces the extra costs. This method balances the load for uneven distribution computation systems and has appropriate efficiency to create

doi:10.20944/preprints202005.0174.v1

balance in the tasks.

References:

[1] Barie A, Sasinsky,(2019), “the complete reference of cloud computation”, PLoS One Journal. Vol. 9, no. 11, pp, 63-66.

[2] Liu, F & Liu, Z. (2012). “Distributed load balancing allocation of virtual machine in cloud data center”, IEEE 3rd International conference on Software Engineering and Service Science (ICSESS). 3(1). Pp.20-23.

[3] Zakershahrak, Mehrdad, Akshay Sonawane, Ze Gong, and Yu Zhang. "Interactive Plan Explicability in Human-Robot Teaming." In 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), pp. 1012-1017. IEEE, 2018.

[4] Zakershahrak, Mehrdad, Ze Gong, Nikhillesh Sadassivam, Akkamahadevi Hanni and Yu Zhang. "Online Explanation Generation for Human-Robot Teaming." arXiv preprint arXiv:1903.06418 (2019).

[5] Qazani, M. R. C., Asadi, H., & Nahavandi, S. (2019). High-Fidelity Hexarot Simulation-Based Motion Platform Using Fuzzy Incremental Controller and Model Predictive Control-Based Motion Cueing Algorithm. IEEE Systems Journal.

[6] Qazani, M. R. C., Asadi, H., & Nahavandi, S. (2019). A New Gantry-Tau-Based Mechanism Using Spherical Wrist and Model Predictive Control-Based Motion Cueing Algorithm. Robotica, 1-22.

[7] Qazani, M. R. C., Pedrammehr, S., Abdi, H., & Nahavandi, S. (2019). Performance Evaluation and Calibration of Gantry-Tau Parallel Mechanism. Iranian Journal of Science and Technology, Transactions of Mechanical Engineering, 1-15.

[8] Pour, M. S., Bou-Harb, E., Varma, K., Neshenko, N., Pados, D. A., & Choo, K. K. R. (2019). Comprehending the IoT cyber threat landscape: A data dimensionality reduction technique to infer and characterize Internet-scale IoT probing campaigns. Digital Investigation, 28, S40-S49.

[9] Ghadimi, A., & Asadi, S. (2018). Modelling of composite right/left-handed active multiconductor transmission lines (AMCTL) in time domain. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields, 31(1), e2257

[10] Hassantabar, S., Wang, Z., & Jha, N. K. (2019). SCANN: Synthesis of compact and accurate neural networks. arXiv preprint arXiv:1904.09090.

[11] Hassantabar, S., Dai, X., & Jha, N. K. (2019). STEERAGE: Synthesis of Neural Networks Using Architecture Search and Grow-and-Prune Methods. arXiv preprint arXiv:1912.05831

[12] Mohammadzadeh, A., Ghoddoosian, A., & Noori-Damghani, M. (2011). Balancing of the flexible rotors with particle swarm optimization method. International Review of Mechanical Engineering, 5(3), 490-496.

[13] Damghani, Mohammad Nouri, and Arash Mohammadzadeh Gonabadi. “Numerical Study of Energy Absorption in Aluminum Foam Sandwich Panel Structures Using Drop Hammer Test.” Journal of Sandwich Structures & Materials, vol. 21, no. 1, Jan. 2019, pp. 3–18, doi:10.1177/1099636216685315.

[14] Afshar, A., Hosseini, M. S., & Behzadfar, E. (2014). Numerical study of the agglomerates dispersion behavior in shear and elongational flow fields in viscous media using Population Balance Modeling (PBM). Scientia Iranica. Transaction C, Chemistry, Chemical Engineering, 21(6), 2107.

[15] Thompson, D. S., Meng, D., Afshar, A., Bassou, R., Zong, J., Bonaccurso, E., ... & Vercillo, V. (2018). Initial Development of a Model to Predict Impact Ice Adhesion Stress. In 2018 Atmospheric and Space Environments Conference (p. 3344).

[16] Azarang, A., Kamaei, S., Miri, M., & Asemani, M. H. (2016). A new fractional-order chaotic system and its synchronization via Lyapunov and improved Laplacian-based method. Optik, 127(24), 11717-11731.

[17] Azarang, A., Ranjbar, J., Mohseni, H., & Andy, M. A. (2017). Output feedback synchronization of a novel chaotic system and its application in secure communication. International Journal of Computer Science and Network Security, 17, 72-77.

[18] Azarang, A., Miri, M., Kamaei, S., & Asemani, M. H. (2018). Nonfragile fuzzy output feedback synchronization of a new chaotic system: design and implementation. Journal of Computational and Nonlinear Dynamics, 13(1), 011008.

[19] Ramtin, A., Hakami, V., & Dehghan, M. (2013, December). A Perturbation-Proof Self-stabilizing Algorithm for Constructing Virtual Backbones in Wireless Ad-Hoc Networks. In International Symposium on Computer. Networks and Distributed Systems (pp. 66-76). Springer, Cham

[20] Ramtin, A., Hakami, V., & Dehghan, M. (2014, September). A self-stabilizing clustering algorithm with fault-containment feature for wireless sensor networks. In 7'th International Symposium on Telecommunications (IST'2014) (pp. 735-739). IEEE.

[21] Ramtin, A., Hakami, V., & Dehghan, M. (2014, May). Self-stabilizing algorithms of constructing virtual backbone in selfish wireless ad-hoc networks. In 2014 22nd Iranian Conference on Electrical Engineering (ICEE) (pp. 914- 919). IEEE.

[22] Khiabani, A. G., & Heydari, A. (2019). Design and implementation of an optimal switching controller for uninterruptible powersupply inverters using adaptive dynamic programming. IET Power Electronics

[23] Khiabani, A. G., & Heydari, A. (2018, September). Optimal switching of voltage source inverters using approximate dynamic programming. In ASME 2018 Dynamic Systems and Control Conference (pp. V001T01A006-V001T01A006). American Society of Mechanical Engineers.

[24] Babazadeh, R., & Khiabani, A. G. (2018, October). Nonlinear Observer Design for RC Battery Model for Estimating State of Charge & State of Health Based on State-Dependent Riccati Equation. In 2018 IEEE Electrical Power and Energy Conference (EPEC) (pp. 1-5). IEEE.

[25] Razzaghi, P., Al Khatib, E., & Bakhtiari, S. (2019). Sliding mode and SDRE control laws on a tethered satellite system to de-orbit space debris. Advances in Space Research, 64(1), 18-27.

[26] Razzaghi, P., Al Khatib, E., & Hurmuzlu, Y. (2019). Nonlinear dynamics and control of an inertially actuated jumper robot. Nonlinear Dynamics, 1-16.

[27] Avval, A. G., & El-Ghazaly, S. M. (2018, April). Holistic optimization technique for solving low thermal conductivity of sapphire substrates in high frequency devices. In 2018 Texas Symposium on Wireless and Microwave Circuits and Systems (WMCS) (pp. 1-4). IEEE.

[28] A. G. Avval, E. Larique, S. M. El-Ghazaly, "Heterojunction field effect transistors" in Reference Module in Materials Science and Materials Engineering, Elsevier, 2018.

[29] Ghayouraneh, S., El-Ghazaly, S. M., & Rankin, J. M. (2018). Dynamic Addressing for On-Demand Mobility. In 2018 Aviation Technology, Integration, and Operations Conference (p. 4152).

[30] Ghayouraneh, S., Rankin, J. M., & El-Ghazaly, S. M. (2019). Practical Considerations in Traffic Flow Systems for ODM Vehicles. In 2019 Integrated Communications, Navigation and Surveillance Conference (ICNS) (pp. 1-7). IEEE.

[31] Ghayouraneh, S., El-Ghazaly, S., & Rankin, J. (2019). Investigation of Traffic Flow for ODM Vehicles in Dynamic ICAO Addressing Process. In AIAA Aviation 2019 Forum (p. 3627).

[32] Gharib, M. R., & Daneshvar, A. (2019). Quantitative-fuzzy Controller Design for Multivariable Systems with Uncertainty. International Journal of Control, Automation and Systems, 17(6), 1515-1523.

[33] Hemmati, R., & Rahideh, A. (2017). Optimal design of slotless tubular linear brushless PM machines using metaheuristic optimization techniques. Journal of Intelligent & Fuzzy Systems, 32(1), 351-362.

[34] Hemmati, R., Wu, F., & El-Refaie, A. (2019, May). Survey of Insulation Systems in Electrical Machines. In 2019 IEEE International Electric Machines & Drives Conference (IEMDC) (pp. 2069-2076). IEEE.

[35] Nikoobakht, A., Aghaei, J., Fallahzadeh-Abarghouei, H., & Hemmati, R. (2019). Flexible Co-Scheduling of Integrated Electrical and Gas Energy Networks under Continuous and Discrete Uncertainties. Energy.

[36] Hemmati, R., & Rahideh, A. (2017). Optimal design of slotless tubular linear brushless PM machines using metaheuristic optimization techniques. Journal of Intelligent & Fuzzy Systems, 32(1), 351-362.

[37] Hemmati, R., Wu, F., & El-Refaie, A. (2019, May). Survey of Insulation Systems in Electrical Machines. In 2019 IEEE International Electric Machines & Drives Conference (IEMDC) (pp. 2069-2076). IEEE.

[38] Shahmiri A, “a review on genetics algorithm”, Jam newspaper, Click attachment, No. 209, Tehran, 2009

[39] Kansal, N. J., & Chana, I. (2012). Cloud load balancing techniques: A step towards green computing. *IJCSI International Journal of Computer Science Issues*, *9*(1), 238-246.

[40] Shafi’i M. A., Latiff M. S., Gaddafi A., Madni S. H., Secure Scientific Applications Scheduling Technique for Cloud Computing Environment Using Global League Championship Algorithm, PLoS One Journal. Vol. 6, no. 11, 2016.

[41] P. Kumar, A. Verma, Scheduling using improved genetic algorithm in cloud computing for independent tasks, ICACCI '12 Proceedings of the International Conference on Advances in Computing, Communications and Informatics, Pages 137-142, 2012.

[42] M. A. Tawfeek, A. El-Sisi, A. E. Keshk, F. A. Torkey, Cloud task scheduling based on ant-colony optimization, Computer Engineering & Systems (ICCES), 2013 8th International Conference on, 2015.