Multi-Objective Task Scheduling in Cloud Computing Using an Imperialist Competitive Algorithm

Majid Habibi
Department of Computer Engineering,
Tabriz Branch, Islamic Azad University,
Tabriz, Iran

*Nima Jafari Navimipour
Department of Computer Engineering,
Tabriz Branch, Islamic Azad University,
Tabriz, Iran

***Abstract*—Cloud computing is being welcomed as a new basis to manage and provide services on the internet. One of the reasons for increased efficiency of this environment is the appropriate structure of the tasks scheduler. Since the tasks scheduling in the cloud computing environment and distributed systems is an NP-hard problem, in most cases to optimize the scheduling issues, the meta-heuristic methods inspired by nature are used rather than traditional or greedy methods. One of the most powerful meta-heuristic methods of optimization in the complex problems is an Imperialist Competitive Algorithm (ICA). Thus, in this paper, a meta-heuristic method based on ICA is provided to optimize the scheduling issue in the cloud environment. Simulation results in MATLAB environment show the amount of 0.7 percent improvement in execution time compared with a Genetic Algorithm(GA).**

Keywords—Cloud Computing; Tasks scheduling; Imperialist Competitive Algorithm

I. Introduction

Today, data centers are composed of thousands of computers that are distributed worldwide and users use the computers frequently to send e-mail, read, write, search, etc. Users perform the mentioned operations using the available browser and as a service receiver attached to the servant. Also, cloud computing [1, 2, 3, 4, 5] combines parallel concepts and computing systems to provide for computers and other machines shared resources, hardware, software and information [6, 7, 8, 9]. In the cloud, users use the provided services according to their needs. The National Institute of Standards that is the Information Technology Laboratory, provides a comprehensive definition of cloud computing: Cloud computing is a pay-per-use model for enabling access, and easy and on-demand network access to a shared resource of computing resources configurable as networks, servers, saving resources, programs, and services that can be prepared with a high-speed and be published with little effort of management or interaction of servant. There are many challenges in a cloud environment. However, task scheduling is an important challenge that has continued to remain a challenge despite many efforts made recently in the field . The services provided in cloud computing are grouped into 4 categories, including SaaS (Software as a Service) [10, 11, 12, 13, 14], IaaS (Infrastructures as a Service) [15, 16, 17, 18, 19], PaaS (Platforms as a Service) [20, 21, 22, 23, 24, 25] and EaaS (Expert as a Service) [8, 9, 26, 27, 28, 29].

In distributed systems, task scheduling should dedicate subtasks to resources that enhance system performance and that is the scheduling methods which determine the performance order of these tasks [30]. As the allocation of processor and resource to tasks is a complex issue in heterogeneous distributed systems such as cloud environment, many methods and algorithms have been provided to reduce time complexity and simultaneous function of subtasks. Some of the available challenges in the field of scheduling tasks in heterogeneous distributing systems include heterogeneous resources, total running time, runtime and productivity convergence speed in meta-heuristic methods and efficiency of scheduling method. According to the importance of this issue, in this paper, research has been conducted in the field of scheduling computing tasks on cloud computing and heterogeneous systems.

In the second section of this paper, a review and background of research have been provided. In section 3, we define the proposed method. In section 4, the results of the simulation and comparing of the proposed algorithm with other algorithms have been discussed and assessed. Finally, in the last section the conclusion and future work are presented.

II. Related Work

In this section, the function of a number of related procedures in the field of scheduling tasks on cloud systems has been studied and discussed. Sung-Soo Kim, *et al.* [31] in 2014, using optimization algorithm based on biogeography (BBO) which is a new method of meta-heuristic algorithms, optimized task scheduling in cloud computing. Several criteria have been used to assess the task scheduling algorithms and the runtime and the task circle duration have been considered as important criteria which are the main aim of this algorithm. The advantage of this algorithm is to optimize the duration of performance of functions.

Two task scheduling methods named Heterogeneous Earliest Finish Time (HEFT) and Critical-Path-On-a-Processor (CPOP) based on the list have been presented with limited processors to achieve high efficiency and fast timing [32]. HEFT method in every procedure selects a task that has an upward rank and maps via using the method according to processors rank that reduces the earliest start time. However,in CPOP method to indicate priority the sum of the values of upward rank and downward rank of tasks is used, as well as these two algorithms are different in choosing the processor of tasks, so that CPOP algorithm assigns tasks on the critical path to the processors that minimize the makespan of all tasks in critical path.

In [32], a scheduling algorithm called Longest Dynamic Critical Path (LDPC) is proposed with a high-efficiency. Provided algorithm is used on distributed computing systems with a limited number of processors. LDCP scheduling method is based on a list which uses a new method to select tasks for scheduling in a heterogeneous environment of distributed computing systems which enables this algorithm to schedule tasks with a better quality. In this paper, the proposed algorithm has been compared with two DLS with HEFT scheduling algorithm; the survey results show that the proposed method of mentioned algorithms is better in acceleration and time of scheduling. Also, the efficiency of the proposed algorithm with the increase of communications costs between tasks in the graphic is increased compared to other two algorithms. As a result, this algorithm provides a practical solution for scheduling programs with a parallel power with more communication costs in heterogeneous environments of computing systems.

In [33], a two-step algorithm called Hybrid Heuristic–Genetic Scheduling (H2GS) has been suggested for tasks scheduling in heterogeneous distributed computing systems. The first stage of the algorithm is based on LDCP list for scheduling with high quality and in second stage scheduling obtained from the first stage is injected to the initial population of a GA called Genetic Algorithm for Scheduling (GAS). The function of the second stage leads to a shorter schedule. The efficiency of H2GS is compared with two methods of DLS and HEFT and the results show that the proposed algorithm improves the mentioned methods. In [34], a task scheduling method in heterogeneous computing systems has been presented by employing GAs of multiple priority offers. The main idea of this method that is called Multiple priority memetic algorithm (MPQGA) is the use of simultaneous advantageous of heuristic and evolutionary algorithm and avoidance of their objections. In order to assign priorities to tasks, a GA is used and heuristic method of Earliest Finish Time (EFT) for mapping tasks to the processor is applied. MPQGA algorithm of combining processes has used a mutation and strong fit for tasks scheduling. Experimental and comparative test results conducted on the Real world graphs and random revealed that MPQGA algorithm is optimized two heuristic algorithms and a meta-heuristic method called Basic Genetic Algorithm (BGA).

Verma, *et al.* [35] used an improved GA for tasks scheduling in the cloud computing environment. The initial population in this algorithm unlike the standard GA is not chosen randomly and the results of the two algorithms are used in the production of the initial population.

Ting, *et al.* [36] have provided an optimized algorithm for tasks scheduling in the cloud computing environment. After applying the selecting operators, combination and mutation of simulated GAs are used; in this case, the new generation will be closer to the optimal solution. Meanwhile, in this method the parameter of quality of service is composed of five parameters included completion time, bandwidth, cost, distance, and reliability, which due to the type of task different values are assigned to each of these parameters.

Vey, *et al.* [37] have used GAs to minimize the time to complete tasks in a cloud computing environment. A matrix that is the anticipated time for implementation of each task from any source is used to calculate the completion time. Also, a parameter is used that is the indicator of time which finishes the allocated processor in the previous schedule; with this parameter, the workload of the source will also be used to find the optimal sequence.

III. The Proposed Method

One of the weaknesses of the ICA is its rapid convergence to the local optimum points. To improve this problem and increase the scanning ability of algorithm, a process similar to the fixed mixing process which is used in GAs is applied in the policy of absorption. This means that after applying the absorption policies on the empire to increase the seeking range of issue a uniform recombination is used in the production of the new position of colonies. Here, one of the parent recombination operators is the position of the colonizer and the position of colonized. In order to prevent random seek and mutation to inappropriate responses, a condition is used in the body of algorithms that is when the resulted position is better compared to the previous position of each colony, the current position is replaced. Formula (1) shows how this process functions in which Colonyik is the coloniali of empirek and Imperialistk shows colonial empire k. Also, α is a coincidence brother that is composed of zero and one.

$$\text{Colony}_{\text{k}} = a(Colony_{ik}) + (1-a)*(imperialist_k) \quad (1)$$

In Figure 1, an overall stage of the proposed algorithm for pseudo-code has been shown.

```
Input:npop(Population-size),problem-size,ep,$\alpha, \beta, pr$
For i=1 to npop do
    $C_{iposition}$ ← RandomPosition(problem-size)
    If i<=ep then
        EmpriresPopulation ← $C_{iposition}$
    Else
        $C_w$← GetWorstSolution(EmpiresPopulation)
        If Cost($C_{iposition}$) < Cost($C_{wiposition}$) then
              Replace(EmpiresPopulation,$C_i$,$C_w$)
        Else
            $C_{iempire}$ ← ssignAnEmpire(EmpiresPopulation)
        End
    End
    Populaton ← $C_i$
End
EvaluatePopulaton(Population)
EvaluateEmpiresPopulation(EmpiresPopulation,Population)
ImperialisticCompetition(EmpiresPoplution,Population)
EliminiateWeakestEmpire(EmpiresPoplution,Population)
```

```
End
EvalutePopulation(Population)
BestSol ← GetBestSolution(Population)
Return BestSol
```

Fig. 1. The proposed algorithm for pseudo-code

IV. Simulation Results

In order to assess the efficiency, the proposed method is compared with GA regarding the time to complete the tasks and the productivity of resources via using MATLAB. In order to assess the efficiency of the proposed algorithm, the data in Table 1 is used and Table 2 shows the parameters of the ICA.

TABLE I. SIMULATION PARAMETERS

Parameter	Values
Number of work	60~30
Virtual machine	12~6
The time of tasks	6000~1000 (MI)
Tasks size	200~50 (MB)
Bandwidth	500~100 (Mbps)
Processor speed	500~100 (MIPS)

TABLE II. IMPERIALIST COMPETITIVE PARAMETERS

Parameter	Values
Maximum frequency	100
Population size	50
Number of empires	10
(The absorption coefficient) β	2
The possibility of revolution	0.1
Rate of revolution	0.05
(Colonies the cost factor) δ	0.1
(Selection pressure) η	10

Due to the nature of meta-heuristic algorithm and random initial positions, each algorithm has been implemented 10 times on average and the obtained average results are considered as a final answer and criteria for comparison. Figure 2 shows the comparison of the proposed method with GA in terms of standard completion time, regardless of the distribution time.

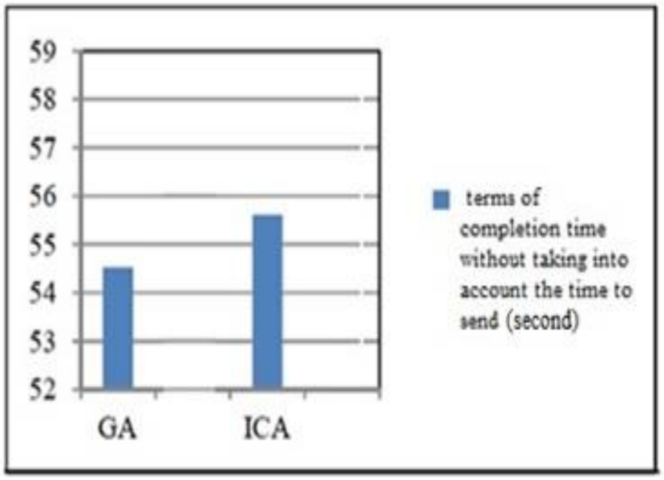

Fig. 2. Chart of methods comparison in terms of completion time without taking into account the time to send

The comparison chart of ICA with a GA is presented in a cloud environment, based on the funding function formula (1) is also shown in Figure 3, Time to send tasks from the scheduler to sources is considered according to bandwidth parameters, in the proposed cost function.

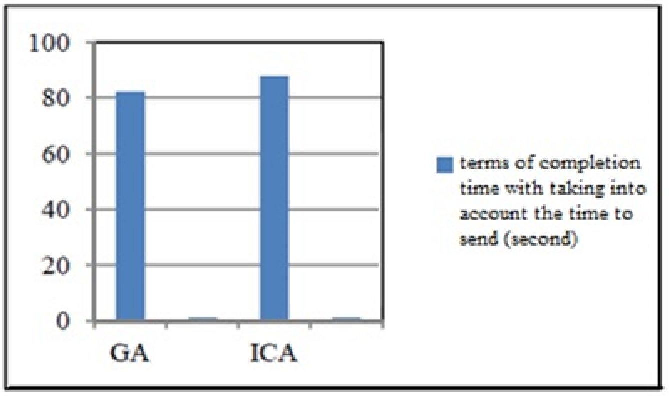

Fig. 3. The chart of comparing algorithms based on the cost function

Figure 4 also shows the efficiency of sources in the ICA comparing to the GA. The results showed an improvement in performance of the proposed algorithm comparing to the other methods proposed for tasks scheduling in the cloud environment such as GAs, particle swarm, and the standard ICA.

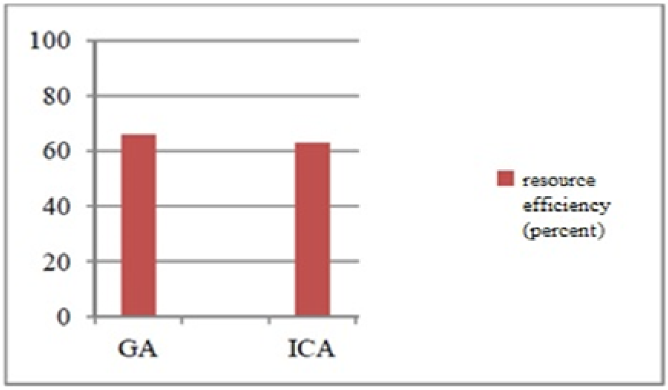

Fig. 4. The chart of comparing algorithms based on resource efficiency

In this section, an improved version of the ICA is proposed for tasks scheduling in a cloud environment. In the most proposed algorithms for mapping tasks to resources in the cloud environment, only the time of performing tasks is noted and features of sources bandwidth and time is not considered as an effective parameter in producing the final answer. In this paper, a cost function aware of bandwidth is used for optimal scheduling and enhanced efficiency of tasks scheduling resources in the cloud computing environment. As well as in order to avoid algorithm to be trapped in the local optimization and enhancing scanning ability of algorithm, after the policy of absorption a process similar to the uniform mixture of GAs has been applied to empires. Results of implementation and comparison with other proposed methods for tasks scheduling in a cloud environment, such as GA, particle swarm, and competition algorithm also represent achieving better responses and suitable colonialism standard tasks scheduling.

V. Conclusion

In most of the algorithms provided for mapping the tasks to resources in cloud environment, only attention is given to the time of performing tasks and bandwidth features of resources and time of sending tasks are not considered as effective parameters in producing the final answer. In this article for optimal timing and increasing resource efficiency, a cost function aware of bandwidth is used for tasks scheduling in cloud computing environment. Also, in order to avoid algorithms to be trapped in local optimization and increasing the scanning ability of algorithms, the same process with uniform recombination available in GAs has been applied to empires after the policy of absorption. The results of issue implementation and its comparison with other proposed methods for tasks scheduling in the cloud environment, such as GAs and standard ICA represents achieving better response and more convenient tasks scheduling. In this paper, improved version of ICA for tasks scheduling in the cloud environment is proposed. For this reason, further studies in tasks scheduling in the cloud computing environment, such as MPQMA meta-heuristic methods is used to improve tasks scheduling.

References

[1] M. C. Chu-Carroll, "Code in the Cloud," Programming Google App Engine, Pragmatic Bookshelf, 306 pages, 2011.

[2] S. Asghari and N. J. Navimipour, "Review and Comparison of Meta-Heuristic Algorithms for Service Composition in Cloud Computing", Majlesi Journal of Multimedia Processing, Vol. 4, No. 4, 2016.

[3] M. Chiregi, N. Jafari Navimipour, "Trusted services identification in the cloud environment using the topological metrics", Karbala International Journal of Modern Science, 2016. (in press)

[4] M. Chiregi, N. J. Navimipour, "A new method for trust and reputation evaluation in the cloud environments using the recommendations of opinion leaders' entities and removing the effect of troll entities", Computers in Human Behavior, Vol. 60, pp. 280-292, 2016.

[5] B. Keshanchi, N. Jafari Navimipour, "Priority-based task scheduling in the cloud systems using a memetic algorithm", Journal of Circuits, Systems and Computers, 2016. (in press)

[6] B. A. Milani and N. J. Navimipour, "A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions", Journal of Network and Computer Applications, Vol. 64, pp. 229-238, 2016.

[7] N. J. Navimipour, "Task scheduling in the Cloud Environments based on an Artificial Bee Colony Algorithm", International Conference on Image Processing, Production and Computer Science (ICIPCS'2015), Istanbul,Turkey, pp. 38-44, 2015.

[8] N. J. Navimipour, F. S. Milani, "Task scheduling in the cloud computing based on the cuckoo search algorithm", International Journal of Modeling and Optimization, Vol. 5, No. 1, p. 44-47, 2015.

[9] N. J. Navimipour, A. M. Rahmani, A. H. Navin, M. Hosseinzadeh, "Expert Cloud: A Cloud-based framework to share the knowledge and skills of human resources", Computers in Human Behavior, Vol. 46, pp. 57-74, 2015.

[10] J. Lin, D. Fu, J. Zhu, "What is Cloud Computing?", IT as a Service, Vol. 11, No. 2, pp. 10-13, 2009.

[11] P. Buxmann, T. Hess, S. Lehmann, "Software as a Service", Wirtschaftsinformatik, Vol. 50, pp. 500-503, 2008.

[12] V. Choudhary, "Software as a service: Implications for investment in software development",40th Annual Hawaii International Conference on in System Sciences(HICSS 2007), pp. 209a-209a, 2007.

[13] M. Almorsy, J. Grundy, A. S. Ibrahim, "Adaptable, model-driven security engineering for SaaS cloud-based applications", Automated Software Engineering, Vol. 21, pp. 187-224, 2014.

[14] Z. Zeng, B. Veeravalli, "Optimal metadata replications and request balancing strategy on cloud data centers", Journal of Parallel and Distributed Computing, Vol. 74, pp. 2934-2940, 2014.

[15] S. Bhardwaj, L. Jain, S. Jain, "Cloud computing: A study of infrastructure as a service (IAAS)", International Journal of engineering and information Technology, Vol. 2, pp. 60-63, 2010.

[16] A. Khajeh-Hosseini, D. Greenwood, I. Sommerville, "Cloud migration: A case study of migrating an enterprise it system to iaas", 3rd International Conference on in Cloud Computing (CLOUD), pp. 450-457, 2010.

[17] A. Nathani, S. Chaudhary, G. Somani, "Policy based resource allocation in IaaS cloud", Future Generation Computer Systems, Vol. 28, pp. 94-103, 2012.

[18] A. Iosup, R. Prodan, D. Epema, "Iaas cloud benchmarking: approaches, challenges, and experience", in Cloud Computing for Data-Intensive Applications, ed: Springer, pp. 83-104, 2014.

[19] W. Wang, B. Liang, B. Li, "Revenue maximization with dynamic auctions in iaas cloud markets", IEEE/ACM 21st International Symposium on in Quality of Service (IWQoS), pp. 1-6, 2013.

[20] P. Mell and T. Grance, "Draft NIST working definition of cloud computing", Referenced on June. 3rd, Vol. 15, 2009.

[21] M. Miller, J. Lei, "Cloud computing", M. Beijing: Machine Industry Publication, Vol. 4, 2009.

[22] S. Eludiora, O. Abiona, A. Oluwatope, A. Oluwaranti, C. Onime, L. Kehinde, "A user identity management protocol for cloud computing paradigm," Int'l J. of Communications, Network and System Sciences, Vol. 4, pp. 152-163, 2011.

[23] H. Dinesha and V. Agrawal, "Multi-level authentication technique for accessing cloud services", International Conference on, in Computing, Communication and Applications (ICCCA), pp. 1-4, 2012.

[24] D. Zeginis, F. D'Andria, S. Bocconi, J. Gorronogoitia Cruz, O. Collell Martin, P. Gouvas, et al., "A user-centric multi-PaaS application management solution for hybrid multi-cloud scenarios", Scalable Computing: Practice and Experience, Vol. 14, 2013.

[25] M. Sellami, S. Yangui, M. Mohamed, S. Tata, "PaaS-independent Provisioning and Management of Applications in the Cloud," Sixth International Conference on in Cloud Computing (CLOUD), IEEE, pp. 693-700, 2013.

[26] N. J. Navimipour, A. H. Navin, A. M. Rahmani, M. Hosseinzadeh, "Behavioral modeling and automated verification of a Cloud-based framework to share the knowledge and skills of human resources", Computers in Industry,Vol.68, pp. 65-77, 2015.

[27] N. J. Navimipour, "A formal approach for the specification and verification of a trustworthy human resource discovery mechanism in the Expert Cloud", Expert Systems with Applications, Vol.42, pp. 6112–6131, 2015.

[28] M. Oussalah, D. Professor Ali Hessami, N. Jafari Navimipour, A. Masoud Rahmani, A. Habibizad Navin, M. Hosseinzadeh, "Job scheduling in the Expert Cloud based on genetic algorithms", Kybernetes, Vol. 43, pp. 1262-1275, 2014.

[29] M. Ashouraie, N. Jafari Navimipour, M. Ramage, and P. Wong, "Priority-based task scheduling on heterogeneous resources in the Expert Cloud", Kybernetes, Vol. 44, 2015.

[30] M. A. Khan, "Scheduling for heterogeneous systems using constrained critical paths", Parallel Computing, Vol. 38, pp. 175-193, 2012.

[31] S. S. Kim, J.-H. Byeon, H. Yu, H. Liu, "Biogeography-based optimization for optimal job scheduling in cloud computing," Applied Mathematics and Computation, Vol. 247, pp. 266-280, 2014.

[32] M. I. Daoud, N. Kharma, "A high performance algorithm for static task scheduling in heterogeneous distributed computing systems", Journal of Parallel and distributed computing, Vol. 68, pp. 399-409, 2008.

[33] M. I. Daoud, N. Kharma, "A hybrid heuristic–genetic algorithm for task scheduling in heterogeneous processor networks," Journal of Parallel and Distributed Computing, Vol. 71, pp. 1518-1531, 2011.

[34] Y. Xu, K. Li, T. T. Khac, M. Qiu, "A multiple priority queueing genetic algorithm for task scheduling on heterogeneous computing systems", 9th International Conference on in High Performance Computing and

Communication, 14th International Conference on Embedded Software and Systems (HPCC-ICESS), IEEE, Liverpool, pp. 639-646, 2012.

[35] S. Kaur, A. Verma, "An efficient approach to genetic algorithm for task scheduling in cloud computing environment", International Journal of Information Technology and Computer Science (IJITCS), Vol. 4, pp. 74-79, 2012.

[36] G. Guo-ning, H. Ting-lei, G. Shuai, "Genetic simulated annealing algorithm for task scheduling based on cloud computing environment", International Conference on Intelligent Computing and Integrated Systems, Guilin, pp. 60-63, 2010.