Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2021 SCPE. Volume 22, Issues 2, pp. 227–235, DOI 10.12694:/scpe.v22i2.1899

CLOUD BASED RESOURCE SCHEDULING METHODOLOGY FOR DATA-INTENSIVE SMART CITIES AND INDUSTRIAL APPLICATIONS

SHIMING MA*, JICHANG CHEN†, YANG ZHANG‡, ANAND SRIVASTAVA § AND HARI MOHAN ¶

Abstract. For the data-intensive applications, resource planning and scheduling has become an important part for smart cities. The cloud computing techniques are being used for planning and scheduling of resources in data-intensive applications. The regular methodologies being used are adequately successful for giving the asset allotment yet they do not provide time effectiveness during task execution. This article presents an effective and time prioritization based smart resource management platform employing the Cuckoo Search based Optimized Resource Allocation (CSO-RA) methodology. The opensource JStorm platform is utilized for dynamic asset planning while using big data analytics and the outcomes of the experimentation are observed using various assessment parameters. The proposed (CSO-RA) system is compared with the current methodologies like particle swarm optimization (PSO), ant colony optimization (ACO) and genetic algorithm (GA) based optimization methodologies and the viability of the proposed framework is established. The percentage of optimality observed for CSO-RA algorithm is 97% and overall resource deployment rate of 28% is achieved using CSO-RA method which is comparatively much better than PSO, GA and ACO conventional algorithms. Feasible outcomes are obtained by using the CSO-RA methodology for cloud computing based large scale optimization-based data intensive industrial applications.

Key words: Resource planning and scheduling; smart cities; data-intensive industrial applications; Cuckoo Search based Optimized Resource Allocation; particle swarm optimization; ant colony optimization; genetic algorithm; large scale optimization.

AMS subject classifications. 68M14, 68W50

1. Introduction. The concept of cloud computing has developed as a significant part of organizations involving computational models, industrial sectors and smart cities including different business assets in the existing situation. The innovative appearance of technology has reformed the cloud computing reforms as the third innovation over the most recent couple of many years [1]. With the development of information concentrated industrial applications, cloud computing has become the most generally utilized virtualization platform for industries and smart cities [2]. The different kinds of utilizations including distributed computing are web applications, appropriated applications like online business, and many more [3,4]. This innovation has arisen as another worldview for offering a reliable support in the current data intensive environment. The essential idea of cloud computing depends on its capacity that is intended to store the client information on web as an alternative of putting away it locally. With the fast improvement in energy utilization of the web-based information, time viability has become a main consideration for improving thee cloud computing exhibition [5]. The time viability of a distributed computing-based application is influenced by correspondence limit of web information for performing multiple tasks execution at a specified time [6]. Therefore, the general capacity as well as performance of cloud computing tasks can be improved by utilizing proper task planning and execution systems.

Recently, the issue of task scheduling has gathered the researchers' attention for improving the various factors of cloud computing environment like execution cost, resource utilization, consumption of power as well as its fault tolerance capabilities [7]. The finding of compromise between accurate time scheduling as well as least energy consumption has become a major research challenge. Cloud computing require the vast number of servers connected to the internet, therefore, there is a need for task scheduler to effectively arrange the

*Nanning University, Nanning 530200, China (shimingma84@outlook.com).
†Nanning University, Nanning 530200, China (jichangchen78@gmail.com).
‡Guangxi Vocational and Technology Institute of Industry, Nanning 530001, China (yangzhang221@outlook.com).
§ABES Engineering College, Ghaziabad, India (anand8355@gmail.com).
¶KEC Engineering College, Ghaziabad, India (hari.mohan@krishnacollege.ac.in).

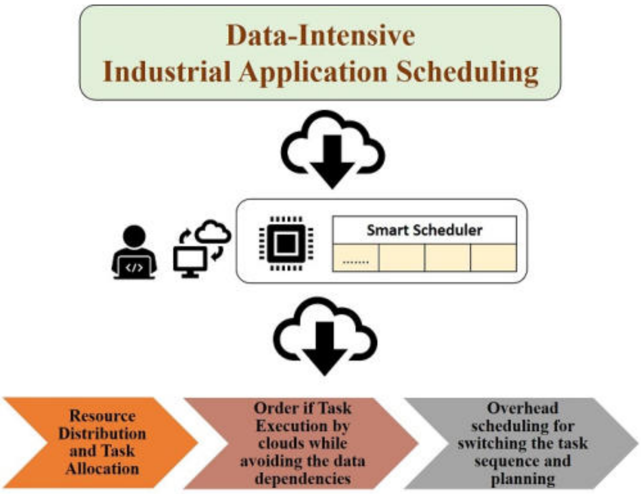

Fig. 1.1. *Concerning aspects of data-intensive industrial application scheduling*

execution of various tasks involved. The major factors affecting the performance of task scheduling algorithms is the time span as well as the energy consumed as the effective task scheduler should use fewer resources as well as consume less time [8,9]. These factors are specific in designing and implementation of the task scheduling strategies and plays a vital role in improving the fault tolerance capabilities of energy efficient tasks [10]. The various concerning aspects for data intensive application scheduling are depicted in Figure 1.1.

In the current situation, the task planning and asset allotment systems ought to be advanced for exact and precise asset provisioning and adaptable execution of cloud computing for data intensive applications [11,12]. There are different components of task allotment and scheduling revealed in the previous literatures which adds to the cloud computing-based computer programming structures providing various new methodologies [13-15]. However, some of these methods are not cost effective, error prone and are computationally complex providing a scope for improvement for current research paradigms.

This article presents a powerful asset for resource planning and execution for data-intensive smart city and industrial applications. An intelligent resource management platform is proposed in this research employing the Cuckoo Search based Optimized Resource Allocation (CSO-RA) methodology. Time prioritization is considered as the fundamental important aspect for this study and optimization is achieved using the cuckoo search mechanism. This approach poses the novelty of dynamic and optimized resource allocation that can accomplish effective scheduling and planning. Open source JStorm dataset platform is utilized in this article for dynamic asset planning while using big data analytics. The proposed (CSO-RA) system is likewise compared to the current methodologies being utilized in this field like particle swarm optimization (PSO), ant colony optimization (ACO) and genetic algorithm (GA) based optimization methodologies and the consistency of the proposed framework is established. The effectual time-sensitive resource scheduling is accomplished utilizing the proposed approach and its adequacy to perform better in the real time environment is justified by the comparative exploration.

The remainder of this article is organized as: Literature survey is given in section 2 which presents the contemporary related work in this field. The material and strategies used for this exploration work are given in section 3 followed by the outcomes of experimentation in section 4. Section 5 of this article gives the conclusion of this examination work alongside with the future exploration scope.

2. Literature Survey. The information concentrated data intensive frameworks have become a functioning exploration domain because of the usage of large information in the current innovative technological world. Different researchers inspected the utilization of huge information innovation using the big data analytics in the field of medical care, biomedical and informatics [16, 17]. In the last few decades, researchers in the domain of healthcare are utilizing the enormous information-based approaches for imagining new procedures and revelations for resource allocation. Different empirical and exploratory methodologies have been reviewed by the analysts to give better alternatives for task scheduling.

An empirical methodology has been adopted by Lee, et al. [18] for providing proficient resource scheduling and achieve effectual cost effective and least energy utilization outcomes. Nosrati, et al. [19] achieved the resource allotment task by recognizing the latencies in correspondence and geological distance of the framework. The optimization procedures are adopted in this approach for limiting the losses in energy utilization. Verma, et al. [20] introduced the resource allotment method that dynamically predicts the resource scheduling and allotment strategy avoiding high computational time and cost. Xiao, et al. [21] proposed a theory of resource distribution and allocation which can easily limit the energy utilization processes by optimized planning. A PSO based provisioning technique was introduced by Netjinda, et al. [22] for cost minimization and cloud provisioning and arrangement. Ding, et al. [23] assessed various techniques involving the resource allocation methodology for QOS aware applications while taking the nature of resources and various other related factors into consideration. A dynamic resource allocation platform was established by Koch et al. [24] utilizing the maximum likelihood estimation approach for resource optimization.

An application-oriented effort for resource allocation was presented by Peng, et al. [25] for workload management and resource allocation. Fereshteh, et al. [26] established a multiple objective based PSO platform for the allocation of resources as well as to maximize the customer utility. Thein et al. [27] carried away the work of an effective cloud-based resource allocation infrastructure while sidewise managing the cost and revenue resources of the data center. Yang, et al. [28] proposed an energy management platform for dealing with the energy utilization scenario based on the task scheduling and optimization mechanism. Tafsiri, et al. [29] proposed a linear optimization-based approach for strengthening the human relations as well utilize maximum of the pricing resources using auction-based methodology. A multi-specialist framework was described by Prieta, et al. which empowers the successful assignment of resources utilizing the cloud computing approach with worldwide coordination through web [30-32]. The researchers examined various downsides of conventional unified frameworks for allocation and planning of resources and its optimization [33-36]. In the recent literature, the authors have claimed that the cloud computing frameworks have become an important aspect of future internet-based mechanisms as they can adapt to the user demand and may realize dynamic resource management. However, resource allocation and planning are sometimes difficult for expanding information intricacy due to the effect of energy utilization. There is the requirement to keep a compromise between resource allocation, distribution and execution which is the research challenge for the researchers working in this domain.

3. Proposed Methodology. The significant goal of proposed CSO-RA methodology is to perform cloud-based resource allocation and distribution in order to improve the internet reliability of data by employing the time efficient approach. The significant target of this particular work is time prioritization therefore, an optimized cuckoo search algorithm is used for resource prediction and allocation. The proposed CSO-RA algorithm is depicted in Figure 3.1.

This algorithm is a natural-inspired optimization method that is completely based on the process of reproduction of a cuckoo bird. There are several constraints which restricts the CSO-RA algorithm like:

i. The nest is selected in a random manner by each of the cuckoo bird and it lays single egg in it.

ii. The host species may find the alien eggs in their nest and the probability of finding it is p [0,1].

iii. Only the eggs with good quality are carried over to the next generation, otherwise they are either thrown away or abandoned by the invaders.

These constraints should be accomplished in the minimum execution time slot available and they should comply with the characterized time constraints. Time prioritization is experimentally investigated using the different grouping scenarios and dynamic allocation is done to validate the proposed CSO-RA framework.

The proposed algorithm works well for resource scheduling while undertaking the real-time task execution. This methodology works well for the applications with information optimality having differing timing restraints. The execution strategy for the CSO-RA methodology includes various steps like checking the practicality of the assigned tasks then optimizing the capacity of resource storage for performing a particular task. The proposed algorithm should meet the base execution time and should comply with the characterized time restrictions.

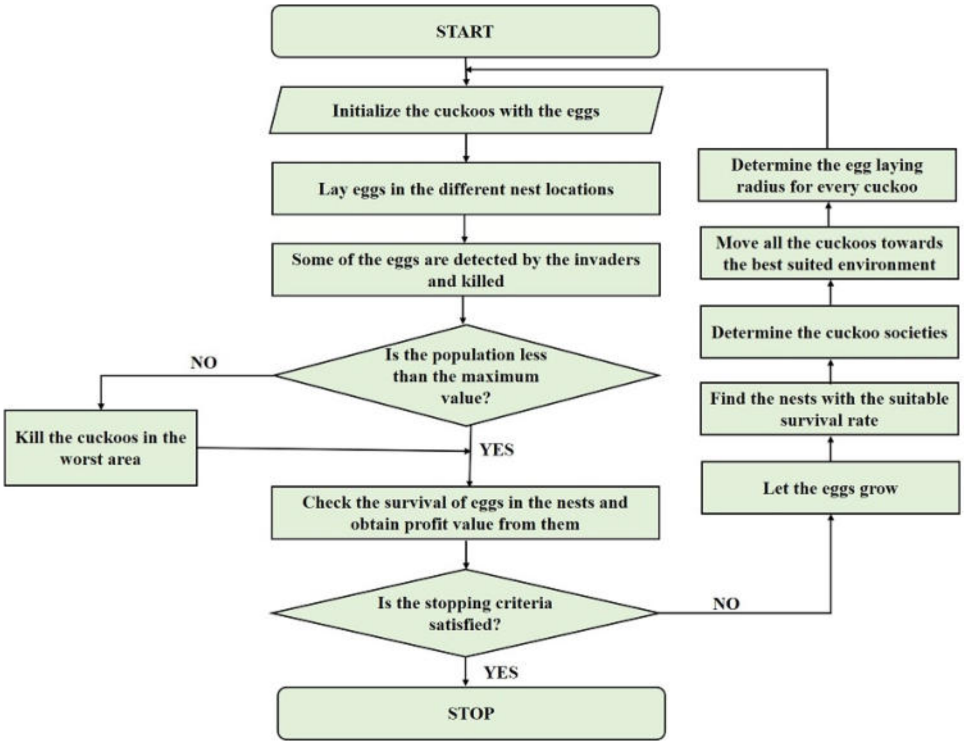

Fig. 3.1. *CSO-RA algorithm for efficient resource allocation*

Table 3.1
Experimental Simulation Environment

Resource Type	Simulation Environment Configuration
Hardware Resource	Intel ® Core 0.5-4 i5 processor
Software Resource	RAM 2-8GB, external storage: 1TB, Maven 3.2.2 Compiler and a single ethernet port

3.1. Experimental Simulation Environment. The experimental analysis of the CSO-RA algorithm is done using the different computer configurations. The computer configuration is varied in between 6 physical nodes which comprises of single master node, 4 computer nodes and 1 client node. The configuration of the system is varied in accordance to the user request and the opensource J-Storm platform is used in this experimentation for real time resource allocation and schedule planning. The entire experimental environment is tabulated in Table 3.1.

3.1.1. Performance Assessment Parameters. There are certain assessment parameters on the basis of which the performance of the proposed CSO-RA methodology has been assessed. The different assessment parameters exploited for this study are time-efficacy, average response time and resource deployment time.

i. Time-efficacy: The parameter Time-efficacy is evaluated as the ratio of successful resource allocation to the total number of resources requested in a specific amount of time. It is expressed by Eq. (3.1).

$$T_{Eff} = \frac{RA_{successful}(T_p)}{REQ_{total}(T_p)} \tag{3.1}$$

where $RA_{successful}(T_p)$ indicates resources, which are successfully allocated and the total number of resources requested are denoted by $REQ_{total}(T_p)$ in the time period (T_p).

Table 4.1
Test load specifications for different grouping mechanisms

Grouping Mechanism	CPU core utilized	Memory usage (in GB)
Group 1	0.5	2
Group 2	1	2
Group 3	2	4
Group 4	3	4
Group 5	4	8

ii. Average Response Time: Average response time is defined as the average time taken for the successful completion of the resource allocation task. It is articulated as the difference between the total amount of allocation time to the time in which actual processing is completed. The calculation can be done by using Eq. (3.2).

$$T_{Av} = \sum_{i-0}^{n} (T_p) \frac{D_j}{n(T_p)} \tag{3.2}$$

where T_p is the $p^t h$ time slot, $n(T_p)$ indicates the total resources to be allocated in time period T_p, the initial data point is denoted by j and D_j is the time delay during the actual processing.

iii. Resource Deployment Time: The resource deployment time is computed as the average of ratio between the exact usage of the resources to the entire resource allocation in a given time slot. It is expressed by Eq. (3.3).

$$R_{Deployment}(T) = 1/n \sum_{s=1}^{n} \frac{RA_{use}(T_s)}{RA_{allocation}(T_s)} \tag{3.3}$$

where $RA_{allocation}(T_s)$ are the resources allocated during the time slot T_s and $RA_{use}(T_s)$ are the resources which are actually utilized out of the total number of n resources.

4. Results and Discussion. The experimental analysis initially includes the assessment of CSO-RA methodology for the performance assessment parameters on the basis of information gathering mechanism. The time-viability for differing resource allocation scenarios is assessed and investigated followed by the assessment and comparison of state-of-the-art techniques. The experimental computations are done in terms of time-efficacy, average response time and resource deployment time for analyzing the impact of resource allocation configuration for different grouping mechanisms. The test load specification for each of the grouping mechanism is depicted in Table 4.1.

The tabular depiction of test load specifications reveals that the CPU core utilization for group 1 is 0.5 with 2GB memory usage which increases to 1 CPU core utilization with 2GB memory usage for group 2, and 2 CPU core utilization with 4GB memory usage for group 3. These values further increase for group 4 and 5 to 3 core and 4 core with respective memory utilization of 4GB and 8GB respectively.

4.1. Result evaluation considering the varying resource configuration. These configurations are used to compute the time efficacy as well as response time for the JStorm platform which are depicted respectively in Figure 4.1 and Figure 4.2.

It is seen from figure 4.1 that the initial resource configuration corresponding to minimum core requirement with less memory storage leads to the resource intensive CPU demand. In this case, the system is not able to obtain the sufficient amount of resources for data processing at the time of peak loads. This situation may lead to excessive data accumulation. However, if we see for group 2 and 3, there are sufficient amount of resources to ensure the data processing, while improving the time- efficacy of the system.

Considering the response time depicted in Figure 4.2, it is seen that the average response time is improving for all the grouping mechanisms. The gradual increase in response time enables the accurate processing of the

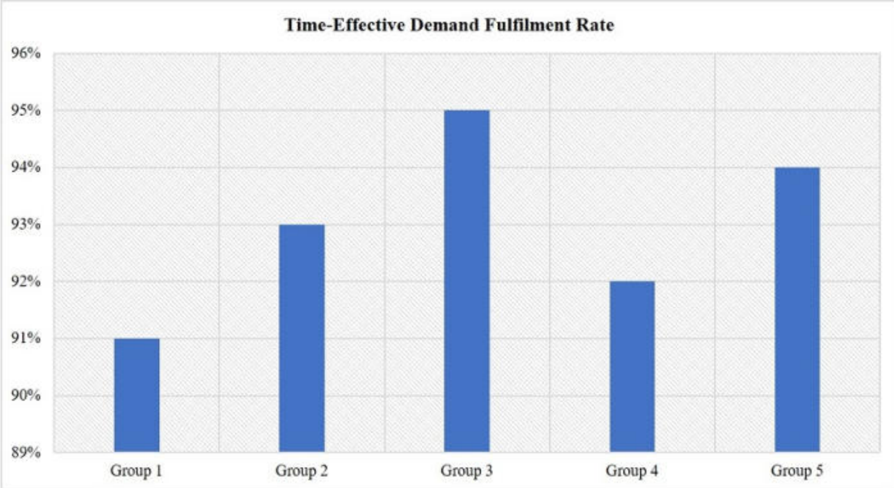

Fig. 4.1. *Time efficacy for test load specifications under different grouping mechanisms*

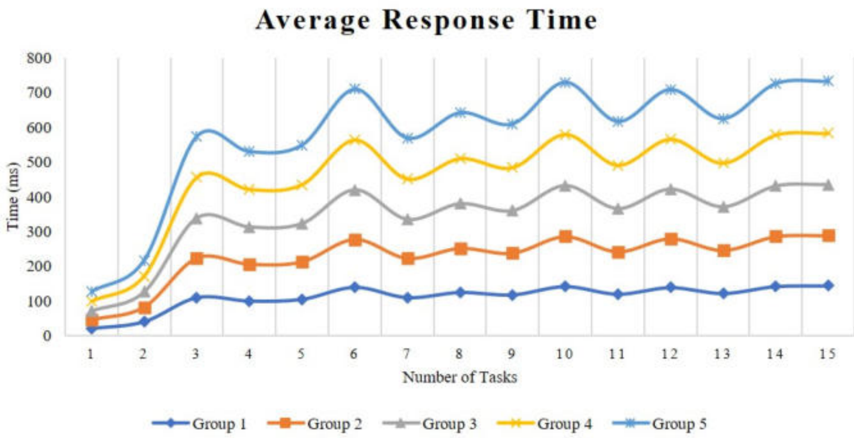

Fig. 4.2. *Response time for test load specifications under different grouping mechanisms*

data while significantly reducing the overall execution time. A comparative analysis is also drawn in this article in terms of resource deployment time for varying load specifications under different grouping mechanisms. This comparison is drawn in Figure 4.3.

The dynamic allocation accomplished in this article using the CSO-RA methodology has reduced the large queue of data for different component being deployed in the scheduling and planning mechanism. The resource deployment time increases during the peak load time and it in-turn minimizes the dwell time. All these assessment parameters also provide the information about the optimized reduction in complete processing time that also minimizes the overall time elapsed for the flow of data during the cloud computing process.

4.2. Comparative analysis with the state-of-the-art techniques. To assess the viability of the proposed CSO-RA methodology for task scheduling and planning, a comparative analysis is done with the three state of the art technique in terms of overall resource deployment rate and optimality rate. The compared techniques were particle swarm optimization (PSO), genetic algorithm (GA) and Ant colony optimization (ACO) approach. The comparison of overall resource deployment rate is done in Figure 4.4.

The comparison is drawn for different conventional approaches in terms of resource deployment rate for varying number of tasks. The proposed CSO-RA methodology provides stable performance comparative to the

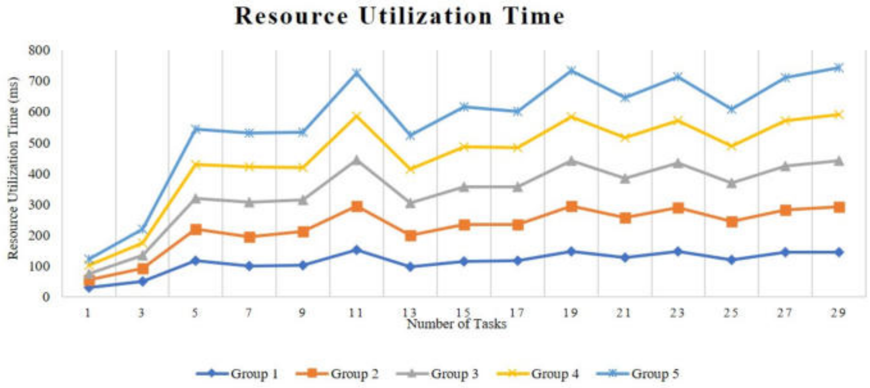

FIG. 4.3. *Resource deployment time for test load specifications under different grouping mechanisms*

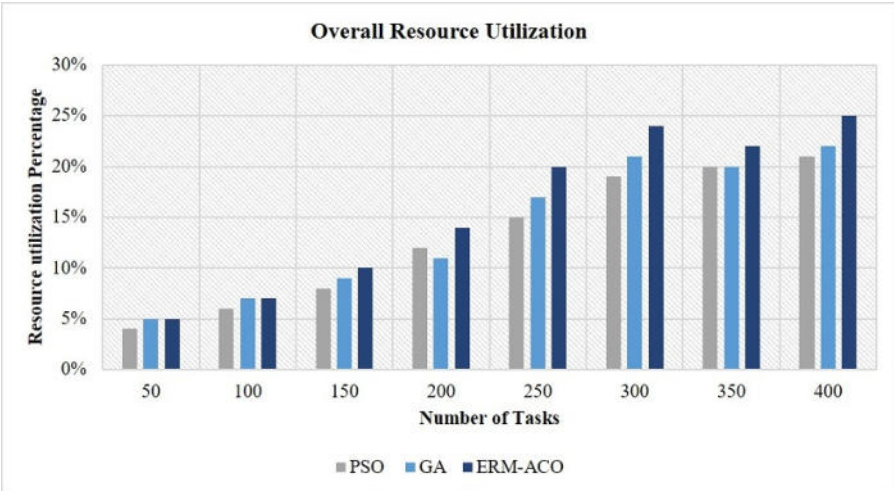

FIG. 4.4. *Comparison of overall resource deployment rate with state-of-the-art technique*

other PSO, GA and ACO algorithms. The performance of PSO, GA and ACO methods initially provides an upward trend however, a sharp fall is noticed toward the end when the number of tasks increases. However, a better gradual increase is noticed in the performance of CSO-RA algorithm when the increased number of tasks are involved.

A comparison for all these methods has also been done in terms of optimality rate for different quality of service and it is indicated in Figure 4.5.

The maximum optimality rate of 97% is achieved for the proposed CSO-RA technique which is far better than other algorithms providing 95%, 89% and 85% of optimality rate for ACO, PSO and GA algorithms respectively. It is observed that dynamic optimality value gradually improves with the increasing number of quality services and the best results are obtained for the proposed CSO-RA methodology.

As the number of task increases, the proposed CSO-RA algorithm performs much better comparative to the conventional approaches. All the resource deployment and allocation constraints are fully satisfied by CSO-RA methodology that provides the feasible outcomes for cloud computing based large scale optimization problems related to data intensive applications.

5. Conclusion. This article proposes an effective and time prioritization based CSO-RA methodology has been proposed in this article that provides a competent solution for data intensive applications. The opensource JStorm platform is utilized for the real time experimental analysis of the study and the outcomes of

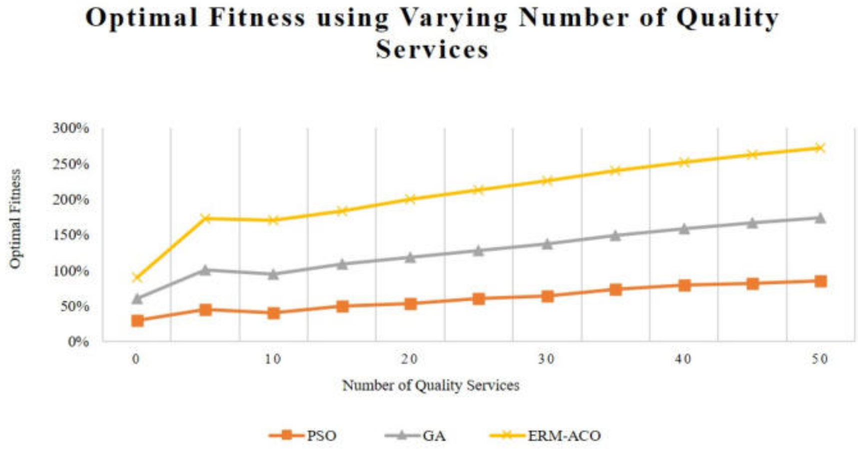

Fig. 4.5. *Optimality rate for different state-of-the-art techniques*

the experimentation are observed in terms of different parameters like time-efficacy rate, average response time, resource deployment rate and optimality value. The proposed CSO-RA methodology when compared to the presents overall resource deployment rate of 28% which is comparatively much better than the 21%, 22% and 25% respective rates obtained by PSO, GA and ACO conventional algorithms. The percentage of optimality observed for CSO-RA algorithm is 97% which is much better than 95% 89% and 85% optimality fitness values obtained for ACO, GA and PSO respectively. The CSO-RA methodology provides feasible outcomes for cloud computing based large scale optimization problems related to data intensive applications. The future perspective of this work is to extend its competence for other grouping mechanisms as well as varying cloud platforms.

REFERENCES

[1] Greenberg, A., Hamilton, J., Maltz, D. A., and Patel, P., *The cost of a cloud: research problems in data center networks*, 2008.

[2] Guo, C., Wu, H., Tan, K., Shi, L., Zhang, Y., and Lu, S., *Dcell: a scalable and fault-tolerant network structure for data centers*, In Proceedings of the ACM SIGCOMM 2008 conference on Data communication, 75-86, 2008.

[3] Wang, G., and Ng, T. E., *The impact of virtualization on network performance of amazon ec2 data center*, In 2010 Proceedings IEEE INFOCOM, 1-9, 2010.

[4] Ijaz, S., Munir, E. U., Anwar, W., and Nasir, W., *Efficient scheduling strategy for task graphs in heterogeneous computing environment*, International Arab Journal of Information Technology, 10(5), 486-492, 2013.

[5] Gobalakrishnan, N., and Arun, C., *A new multi-objective optimal programming model for task scheduling using genetic gray wolf optimization in cloud computing*, The Computer Journal, 61(10), 1523-1536, 2018.

[6] Jena, R. K., *Task scheduling in cloud environment: A multi-objective ABC framework*,Journal of Information and Optimization Sciences, 38(1), 1-19, 2017.

[7] Srichandan, S., Kumar, T. A., and Bibhudatta, S., *Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm*, Future Computing and Informatics Journal, 3(2), 210-230, 2018.

[8] Natesan, G., and Chokkalingam, A., *Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm*, ICT Express, 5(2), 110-114, 2019.

[9] Adhikari, M., Nandy, S., and Amgoth, T., *Meta heuristic-based task deployment mechanism for load balancing in IaaS cloud*, Journal of Network and Computer Applications, 128, 64-77, 2019.

[10] Mansouri, N., Zade, B. M. H., and Javidi, M. M., *Hybrid task scheduling strategy for cloud computing by modified particle swarm optimization and fuzzy theory*, Computers & Industrial Engineering, 130, 597-633, 2019.

[11] Chen, Z. G., Du, K. J., Zhan, Z. H., and Zhang, J., *Deadline constrained cloud computing resources scheduling for cost optimization based on dynamic objective genetic algorithm*, In 2015 IEEE Congress on Evolutionary Computation (CEC), 708-714, 2015.

[12] Hu, H., Li, Z., Hu, H., Chen, J., Ge, J., Li, C., and Chang, V., *Multi-objective scheduling for scientific workflow in multicloud environment*, Journal of Network and Computer Applications, 114, 108-122, 2018.

[13] Casas, I., Taheri, J., Ranjan, R., Wang, L., and Zomaya, A. Y., *GA-ETI: An enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments*, Journal of computational science, 26, 318-331, 2018.

[14] Luo, J., Wu, M., Gopukumar, D., and Zhao, Y., *Big data application in biomedical research and health care: a literature*

review, Biomedical informatics insights, 8, BII-S31559, 2016.
[15] Min-Allah, N., Qureshi, M. B., Jan, F., Alrashed, S., and Taheri, J., *Deployment of real-time systems in the cloud environment*, BThe Journal of Supercomputing, 1-22, 2020.
[16] Hu, Y., Wang, H., and Ma, W., *Intelligent cloud workflow management and scheduling method for big data applications*, Journal of Cloud Computing, 9(1), 1-13, 2020.
[17] Qureshi, M. S., Qureshi, M. B., Fayaz, M., Zakarya, M., Aslam, S., and Shah, A., *Time and Cost Efficient Cloud Resource Aallocation for Real-Time Data-Intensive Smart Systems*, Energies, 13(21), 5706, 2020.
[18] Lee, Y. C., and Zomaya, A. Y., *Energy efficient utilization of resources in cloud computing systems*, The Journal of Supercomputing, 60(2), 268-280, 2012.
[19] Nosrati, M., and Karimi, R., *Energy efficient and latency optimized media resource allocation*, International Journal of Web Information Systems, 2016.
[20] Verma, M., Gangadharan, G. R., Narendra, N. C., Vadlamani, R., Inamdar, V., Ramachandran, L., ... and Buyya, R., *Dynamic resource demand prediction and allocation in multi-tenant service clouds*, Concurrency and Computation: Practice and Experience, 28(17), 4429-4442, 2016.
[21] Xiao, Z., Song, W., and Chen, Q., *Dynamic resource allocation using virtual machines for cloud computing environment*, CIEEE transactions on parallel and distributed systems, 24(6), 1107-1117, 2012.
[22] Netjinda, N., Sirinaovakul, B., and Achalakul, T., *Cost optimization in cloud provisioning using particle swarm optimization*, In 2012 9th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 1-4, 2012.
[23] Ding, S., Xia, C., Cai, Q., Zhou, K., and Yang, S., *QoS-aware resource matching and recommendation for cloud computing systems*, Applied Mathematics and Computation, 247, 941-950, 2014.
[24] Koch Fernando, Assuncao Marcos D., Cardonha Carlos, Netto Marco A.S., *Optimising resource costs of cloud computing for education*, ELSEVIER Future Generation Computer Systems, 55, 473-479, 2016.
[25] Peng Jun-Jie, Zhi Xiao-Fei, Xie Xiao-Lan, *Application type-based resource allocation strategy in cloud environment*, ELSEVIER Microprocessors and Microsystems, 1-7, 2016.
[26] Fereshteh Sheikholeslami, Nima Jafari Navimipour, *Service allocation in the cloud environments using multi-objective particle swarm optimization algorithm based on crowding distance*, ELSEVIER Swarm and Evolutionary Computation, 35, 53-64, 2017.
[27] Thein Thandar, Myo Myint Myat, Parvin Sazia, Gawanmeh Amjad, *Reinforcement learning based methodology for energy-efficient resource allocation in cloud data centers*, Journal of King Saud University -Computer and Information Sciences, 2018.
[28] Yang, J., Jiang, B., Lv, Z., and Choo, K. K. R., *A task scheduling algorithm considering game theory designed for energy management in cloud computing*, Future Generation computer systems, 105, 985-992, 2020.
[29] Tafsiri Seyedeh Aso, Yousefi Saleh, *Combinatorial double auction-based resource allocation mechanism in cloud computing market*, Journal of Systems and Software, 137, 322-334, 2018.
[30] De la Prieta, F., Rodríguez-González, S., Chamoso, P., Corchado, J. M., and Bajo, J., *Survey of agentbased cloud computing applications*, Future Generation Computer Systems, 100, 223–236, 2019.
[31] Tseng, L., Wu, Y., Pan, H., Aloqaily, M., and Boukerche, A., *Reliable broadcast with trusted nodes: Energy reduction, resilience, and speed*, Computer Networks, 182, 107486, 2020.
[32] Ali, F., Bouachir, O., Ozkasap, O., and Aloqaily, M., *SynergyChain: Blockchain-assisted Adaptive Cyberphysical P2P Energy Trading*, IEEE Transactions on Industrial Informatics, 2020.
[33] Masud, M., Gaba, G. S., Alqahtani, S., Muhammad, G., Gupta, B. B., Kumar, P., and Ghoneim, A., *A lightweight and robust secure key establishment protocol for internet of medical things in COVID-19 patients care*, IEEE Internet of Things Journal, 2020.
[34] Masud, M., Gaba, G. S., Choudhary, K., Hossain, M. S., Alhamid, M. F., and Muhammad, G., *Lightweight and anonymity-preserving user authentication scheme for IoT-based healthcare*, IEEE Internet of Things Journal, 2021.
[35] Masud, M., Alazab, M., Choudhary, K., and Gaba, G. S., *3P-SAKE: privacy-preserving and physically secured authenticated key establishment protocol for wireless industrial networks*, Computer Communications, 175, 82-90, 2021.
[36] Masud, M., Gaba, G. S., Choudhary, K., Alroobaea, R., and Hossain, M. S., *A robust and lightweight secure access scheme for cloud based E-healthcare services*, Peer-to-peer Networking and Applications, 1-15, 2021.

Edited by: Pradeep Kumar Singh
Received: May 30, 2021
Accepted: Sep 20, 2021