International Journal of Future Generation Communication and Networking
Vol. 13, No. 3, (2020), pp. 3527–3536

An Exceedingly Efficient, Enhanced And Cost-Effective Task Scheduling Algorithm For Complex Layered Factual Systems

Shantanu Lohi[1]*, Dr. Nandita Tiwari[2]
[1]**Research Scholar& Corresponding Author*, [2]*Associate Professor*
[1, 2]*Department of Computer Science & Engineering, RKDF Institute of Science& Technology, Sarvepalli RadhaKrishnan University, Bhopal, MP, India*

[*1]*shantanulohi.kits@gmail.com*,[2]*nandita_pandit@yahoo.com*

Abstract

Task scheduling algorithms vary in a large range of input and output parameters. These parameters include but are not limited to, task type, task size, method of scheduling, pre-processing of tasks, storage systems, etc. The ability to successfully execute task scheduling for any computing system is a performance measurement in any multi-processing environment, such as working of cloud or continuous scheduling and much more. Due to the availability of a large variety of algorithms, the researchers find it difficult to select a particular algorithm while designing their own task scheduling system. To ease that issue, in this paper, we have compared different task scheduling algorithms on different storage systems, and evaluated their performance in order to find out the most optimum algorithm for a given scheduling problem.Over the years, for many cloud deployments, cost-effective resource scheduling (CERS) algorithms and dynamic and integrated resource scheduling (DAIRS) algorithms were commonly used. This paper also suggests some recommendations to improve the overall task scheduling capabilities of the system. Moreover, a statistical comparison is also done so that researchers can evaluate the best possible solution for their system while selecting a task scheduler.Through this paper, we try to put forward a detail evaluation of our algorithm based on machine learning (ML) paradigm, Task scheduling process Improved Novel Algorithm.Our proposed algorithm provides a reasonable enhancement by using a standardized dataset and results would be correlations with the amount of computer effort, timing and reaction to complex tasks, which may also include the ongoing world pandemic tasks.

Keywords: *task scheduling, machine learning, machine efforts, response time, datasets*

1. INTRODUCTION

Any effective scheduling algorithm typically makes use of the CPU, processing period and aggregate output. Scheduling is performed in various ways, depending on different criteria. It can be allocated in a standalone mode to different resources at compilation time or can be dynamically assigned at runtime. In distributed system scheduling methods are essentially: amalgamated and heuristic techniques. Techniques under heuristic are categorized into static and dynamic schedules. Dynamic scheduling could be accomplished in batch or online mode. All tasks are known in advance to scheduling in static scheduling, and are linearly assigned to virtualized servers. All the tasks are scheduled promptly in dynamic scheduling, when they show up in the network. Broadly speaking, algorithms geared towards improving the resourceallocation encompass the ensuing expedient strictures: jobs or tasks needed to be executed, proportion of virtualization obtainable for execution of tasks, operational conditions such as bandwidth, available memory and number of instructions for VMs and assignment parameters such as job magnitude, memory required for job size and so forth. Such variables are quite important for work scheduling purposes and need to be addressed whenever there is new development in the domain. The heuristic and metaheuristics polynomial complexity algorithms are competent of generating imprecise or closeto optimum schedule solutions by considering a

ISSN: 2233-7857 IJFGCN

cost of optimality. While these algorithms offer good solutions, however, they require a great deal of awareness in terms of knowledge from experts and also require interventions from humans, generally as regards to encoding schemes. These issues are not the problems in using machine learning to find solutions to task scheduling. While contemplating each of these resource provisioning parameters, a somewhat complicated processing algorithm is created that attempts to optimize the design of the cloud and mitigate the queue of tasks per unit time.

In our prior work, we utilized bio-inspired techniques to achieve an optimized schedule [10]. In that study, we could demonstrate how metaheuristics algorithms like cuckoo search algorithm and firefly algorithm can be used for task scheduling process. We further our research by considering loss of optimality as a disadvantage. We tried to develop a machine learning based approach to mitigate the problems associated with metaheuristics algorithm by focusing on machine effort, delay and response time [11].

In addition, we will analyze the impact of Use cloud containers rather than simplistic VMs and track their performance as contrasted to a non-container equivalent. Preceding resources management research typically does not take all of these issues into account and therefore our research will improve cloud efficiency compared to current algorithms such as DAIRS[6], CERS [7], D-Cloud, Differential Evolution (DE) and Enhanced DE [8]. The main goal of tasks is to achieve high computational efficiency and excellent machine throughput. Inert scheduling of tasks is meek to contrivance from the programmer's opinion, where-as, more vigorous scheduling suits true situations. Dynamic schedules lessen the rate of running the scheduler which must be charged. Amalgam algorithms are categorized as: many objective, reduction intensification or energy-conscious approach methodologies. Planning of assets for a distributed computing condition explicitly implies that the errands which are being dealt with by the scheduling system. It must be designated to appropriate handling assets of the scheduling system. Different methods and calculations are proposed for this before, and a large portion of these calculations are not completely competent for preparing the assignments in an ongoing situation. This is because of the way in which their proficiency is decreased either because of high computational multifaceted nature which brings about low reaction time, or perhaps issues of compromise in quality for task handling. This is also due to the inappropriate use of scheduling system assets which is a consequence of non-ideal and fast calculations.

A further segment of this article talks many algorithms posited by other research studies on resource allocation in which we discuss the challenges of these algorithms, augmented by an assessment between some of them. To sum up, we recommend our novel and effective machine learning algorithm called Task scheduling process Improved Novel Algorithm through various deeper inferences made especially for strategy formulation and to further raise the effectiveness of trustworthiness of our research with the aid of more convoluted computational arrangements that aren’t within the reach of this research. Our suggested methodology would be further compared to the existing CERS and DAIRS algorithms by using generic datasets and the outcomes will be correlated with the amount of machine effort, delay and responsiveness.

2. LITERATURE REVIEW

Two key parameters to be considered in a scheduling algorithm are load balancing and response time. This should also provide the users with equity and safety when delivering services in any environment like cloud or cloud based services. Establishing and putting forward an improved algorithm by integrating several significant parameters together provides an upper hand in delivering efficient services. In the cloud environment, a good programming algorithm could also be used for smarter cloud services to its clients that are regarded as a potential improvement.

We have utilized many reference papers for making an inside and out similar investigation of the methods utilized in cross breed stockpiling frameworks. They looked at execution-based estimates, SSD life expectancy, utilization of energy, reading proportion, writing proportion,

ISSN: 2233-7857 IJFGCN

IOPS, recurrence in duration and access. It is exceedingly evident from their review that none of the structures referred to in their analysis are completely prepared to combine the frameworks to achieve the most optimal performance, and all the frameworks referenced recognizes another problems associated with half breed frameworks. For example, when contrasted with Hot Data Trap and various blossom channels, the hot information recognizable proof calculations such as HFA and AMC offer greater effectiveness. Comparative similarities are produced with various approaches, and they differentiate the problems found with each of them. A comparative analysis is performed in [9], where different methods for changing the burden are considered. Procedures are separated into static, fluid, distributed, unconveyed, deterministic, probabilistic, co-usable, non-co-usable, combined and semi-incorporated procedures. Systems such as trusting the managers, cloud-inviting burden adjustment, two-stage load adjustment and stochastic slope climbing are in comparison. Every program has its own advantages and disadvantages, and requires a few changes or the other. From both of those sources [8] [9], we rendered the loading conditions calculations and evaluated that they are better for mixture stockpiling systems, and that the DAIRS [9] calculation offers the best contrasting reaction and some other cutting-edge method. Another feasible technique for storing information is defined by Lin Wu and others in [10], where object-level storage is favored over storage of information. Essentially, they convert heterogeneous information types into a solitary article, and afterward perform capacity and recovery of the equivalent. They further improve the idea by including one-sided object-based information stockpiling. The biasing of articles is done dependent on their constant access designs, and dependent on these examples the items are either put away in HDD stockpiling or SSD stockpiling. The framework can improve the composing postpone execution by over 60% and lessens the general deferral by over 25%. The framework is seen as truly steady, and should be utilized for overseeing half breed stockpiling gadgets.

Because of the evolving patterns of small to medium-sized businesses moving into the cloud, High-speed algorithms for scheduling tasks need to emerge. Researchers across the globe have proposed numerous task scheduling strategies to optimize resource efficiency and thus improve QoS for the program. Sakellariou and Zhao [1] took up the question of manifold applications scheduling. Presented approach is to combine several applications combining into a huge job graph. Therefore, it generally prohibits for various times and random start offsets for applications. Xie et al. [2] put forth couple of heuristic algorithms that are static in nature. These are inclusive scheduling algorithms capable of performing compound applications on heterogeneous networks with mixed criticality rates. These researchers also bring together the algorithms; F-MHEFT that purposes to boost device efficiency grounded on the fairy norm whereas, D-MHEFT targets to reach the great criticality task limits. In recent times, models and methodologies of game-theoretical and reinforcement learning (RL) are commonly applicable to the multiple constraint scheduling problems [3,4]. The idea of equilibrium is believed in game theory and Multi-agent training approaches are extremely effective in solving multi-constraint and multi-objective optimization problems. For example, Duan et al. [3] suggested a sequential cooperative game algorithm for the optimization of cost and make-span thereby meeting capacity requirements for the scheduling of large-scale workflows. After these factors, we appraised the efforts in [5] that provides a small but non uncurving computational exemplar for mission arranging, this method is fairly tranquil to usage and enforce, thereby promoting low effectiveness. Research emphases upon the VM bandwidth to attain superior performance, but really doesn't try taking other specific task parameters into consideration such as influxspell, joblimit, etc. The lag in the execution of computations and number of tasks increases almost functionally, and should therefore not be included real-time and massive contexts. The analysis proposed in [6] is focused on blending of learning based on opposition and algorithm of Lion optimization. Their analysis states that the overall algorithm reduces the task execution make time by more than 5 percent, which is a substantial number considering that the number of VMs is very limited, and there is a suitable VM for performance enhancement. S. Lohi et al. analyze the use of metaheuristics along with firefly algorithm, and the operational essence of the cuckoo

ISSN: 2233-7857 IJFGCN

search algorithm. It portrays its method through scheduling experiments on assembly line jobs [9] and other ML/DL methodologies by Harish G. et.al and Ravindra J. et.al. [10,11].

The work done Tingting Wang and others in [12] proposes a hereditary calculation based methodology for improving the heap adjusting capacities of the framework. They have introduced a twofold wellness versatile calculation work spreading over time and burden adjusting hereditary calculation (JLGA) which enhances the activity traversing delay and improves the heap adjusting abilities of the framework. The proposed JLGA calculation is seen as performing better than the standard single goal hereditary calculation, and can execute errands in a superior way. A rundown of AI-based calculations with their subtleties can likewise be found in [13], wherein calculations like Particle Swarm Optimization, whole number straight program (ILP) plan, SHEFT work process booking calculation Improved Genetic Algorithm, Multiple QoS Constrained Scheduling Algorithm, DBD-CTO calculation, improved cost-based planning calculation and Heuristic based Genetic Algorithm (HGA) are thought about. All the calculations are acceptable, and work better when identified with a particular application. Focal controlled burden balancers perform better when contrasted and their conveyed partners. This case is demonstrated by the work proposed in [14]. Utilizing their methodology, a need is doled out to each virtual machine, and dependent on this need undertakings are allocated to the VM. The assignment distribution dependent on VM need is found to diminish the reaction time by 20% when contrasted and cooperative effort, dynamic reaction time and throttled reaction time calculations. The need based methodology is sponsored up by another work done in [15], wherein the need is determined just utilizing the MIPS of the virtual machine.

3. OBSERVATIONS AND PERFORMANCE EVALUATION

As observed from the above review of literature, reserving plans improve the productivity of half breed stockpiling on the cloud. Along these lines, calculations which are focused towards load adjusting in half and half stockpiling must likewise be looked into. One such method is created by Hsung-Pin Chang and others in [17]. Utilizing their plan, any multi-layered capacity engineering can use the idea of storing plan so as to adjust cloud loads. It utilizes the constant preparing and limit statuses for every one of the distributed storage frameworks and assesses with respect to which burden must be doled out to which specific stockpiling framework. Their methodology can diminish the reaction time, and improve the general framework throughput by over 20%, when contrasted and first-start things out serve and cooperative plans. Another fascinating burden adjusting approach is named as improved max-min approach [18]. It relegates undertakings that have normal execution time with assets that are slowest, while lesser execution time errands are doled out to assets which are quickest.

Anticipating new errands while portion of existing assignments so the cloud can viably work in future is a novel method to move toward the undertaking booking issue. The work done by Yangyang Gao and others in [19] portrays a plan which definitely takes a shot at this standard. In light of their plan, the heap balancer is made burden mindful, with the goal that it can take choices dependent on the future interest of the heap. They have built up the framework on a half breed memory structure, which settles on it a decent decision for examination. It utilizes a hub level-load-movement plot so as to play out this undertaking. The proposed plot is better as far as reaction time, and CPU use when contrasted and DRA technique by over 20%, and must be utilized while structuring a heap balancer for exceptionally thick cloud assignments. Hybridizing the heap adjusting calculations can likewise be another approach to adjust load in groups.

We can also consider the task scheduling process for the ongoing pandemic situation of covid-19. As the COVID-19 pandemic continues to grow, researchers around the world are striving to fully comprehend, minimize and stifle its spread. Main fields of work include researching the transmission of COVID-19, promoting its identification, designing alternative vaccines and therapies, and examining the socio-economic impacts of the pandemic. The objective of this consideration is not to assess the bearing of the techniques mentioned or to suggest their use, but

ISSN: 2233-7857 IJFGCN

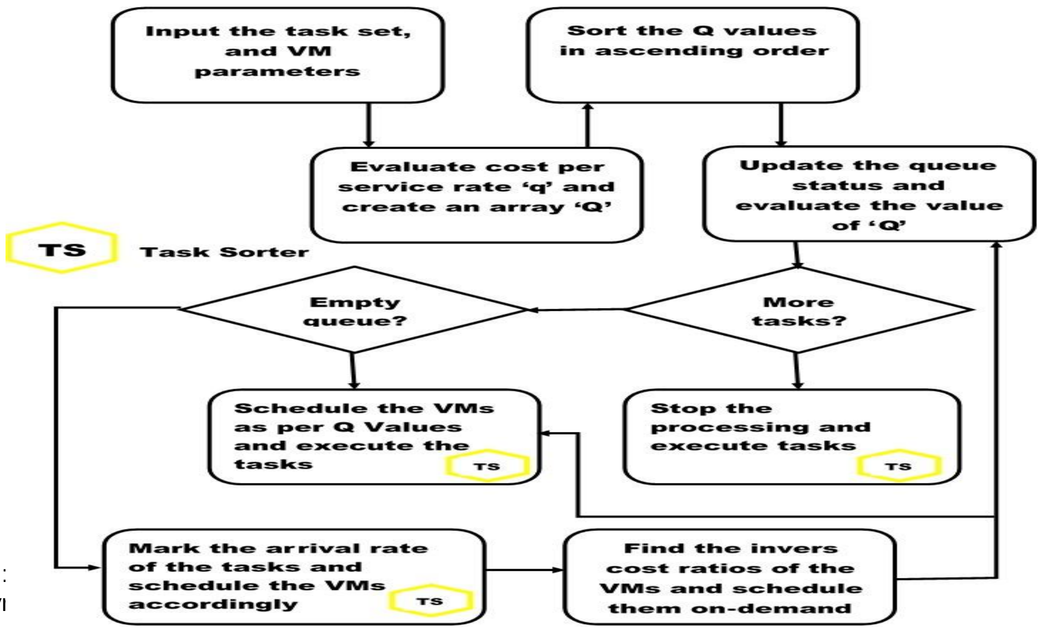
to show the reader the scope of the current applications and to provide an objective overview and a route map of how Machine Learning with task scheduling could support the global retort to the COVID-19 disease.We consider that the scope of this analysis is limited to the machine learning techniques (ML) and Artificial Intelligence (AI) and that we have also submitted conclusions as to whether certain methodologies fall within this classification.

We acknowledge that many of the publications referenced are still pre-printed at the time of this assessment; considering the rapid nature of the crisis, we have tried to be thorough in our analysis, but their true scientific rationality should always be evaluated by the research community through peer-reviewed evaluation and other quality assurance mechanisms.Yan et.al. [17] suggest an innovative method focused on the characteristics of patients with medical data and blood tests to help in allotment of doctors to the patients and physicians classify high-risk patients as soon as possible, This would dramatically increase the treatment plan of all patients and the mortality of terminally ill individuals.Data were collected from 375 patients, the authors identified three main clinical indicators of more than 300 clinical characteristics, including clinical heuristics to predict patient mortality.

A complementary research designed to predict whether current COVID-19 patients will need long-term hospitalization or not based on a U-Net derivative, trained on semi-automatic CT imaging data (Qi et al., 2020). This means that after an initial diagnosis has been made, it may also be possible to use ML to forecast to what degree patients ' conditions that crumble and would require long-time in-patient care. In this scenario, use of task scheduling is pertinent and usable to a larger extent.A far reaching examination of these domain procedures is given in [11], wherein load adjusting calculations are isolated into general, compositional or computerized reasoning based. Calculations like min-min, task gathering based, center division-based, cooperative effort, most limited anticipated postponement, and so forth go under the general classification. These calculations mull over the center VM parameters and the undertaking parameters just, and dispose of any design issues related with them. Ordinarily, these calculations are the least mind boggling, and are appropriate for little to medium measured assignment portion. Calculations like L3B, which comprises of asset the executive module and bundle the executives module; parceling plans; layering plans; queueing plans like DAIRS; multi-criteria-dynamic, and so forth go under the engineering based burden adjusting plans. Artificial intelligence based burden adjusting plans are unbounded in number, in light of the fact that there are an inconclusive number of AI calculations to play out any improvement task.

Taking into consideration many studies and works that we referred during the process of this research; we utilize our novel machine learning algorithm, Task scheduling process Improved Novel Algorithm (TspINA) whose block diagram is described below.

Fig. 1: Block Diagram: Proposed Methodology



ISSN:

- As aninput is given, the VM parameters can be set.
- In this case, it shall evaluate the cost per service rate and then creation of an array would take place. Here, to sort the tasks, a task sorter method would be used for greater simplicity.
- Task execution would be done with respect to the availability of the VMs and schedule on demand.

Table 1. Allotment of doctors for treating patients: efficiency scenario

Number of Patients	Number of Doctors	Efficiency of Allotment
100	10	95
200	10	93
300	10	91.7
400	10	90.6
500	10	88.6
600	10	85.6

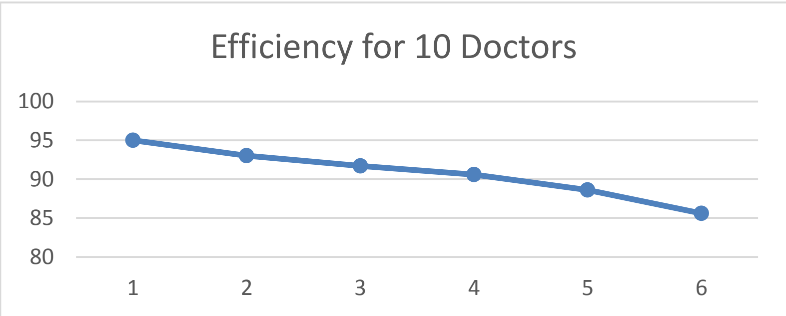
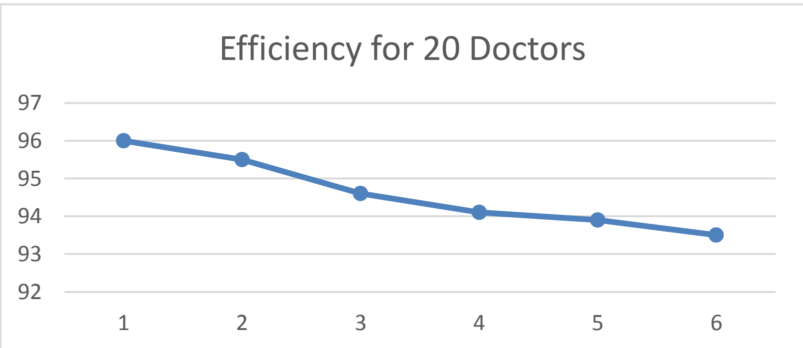


Chart 1 : Allotment of doctors for treating patients: efficiency scenario

Table 2: Allotment of doctors for treating patients: efficiency scenario

Number of Patients	Number of Doctors	Efficiency of Allotment
100	20	96
200	20	95.5
300	20	94.6
400	20	94.1
500	20	93.9
	20	93.5



ISSN: 2233-7857 IJFGCN

Chart 2 : **Allotment of doctors for treating patients: efficiency scenario**

Table 3: Allotment of doctors for treating patients: efficiency scenario

Number of Patients	Number of Doctors	Efficiency of Allotment
100	40	97.9
200	40	96.8
300	40	96.1
400	40	95.7
500	40	95.2
600	40	95.1

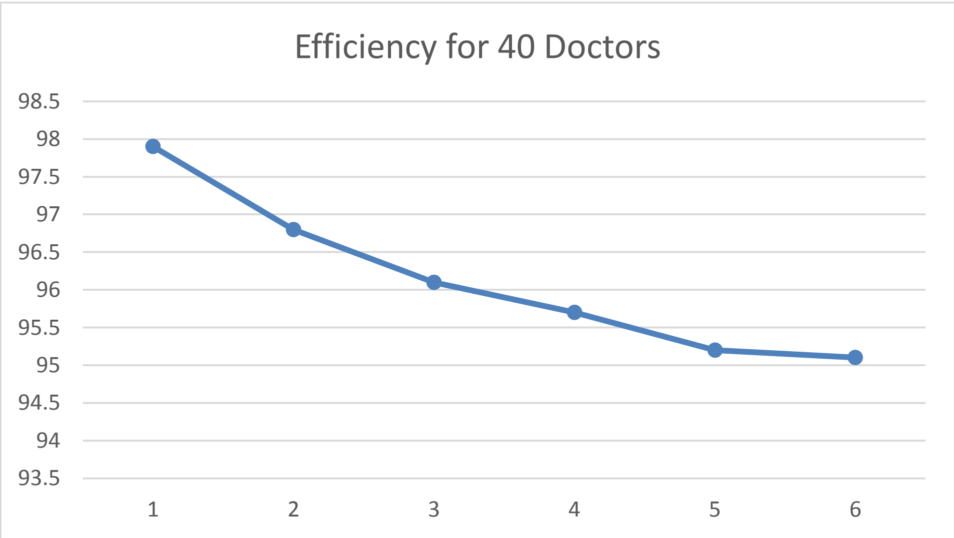


Chart 3 : Allotment of doctors for treating patients: efficiency scenario

4. RESULT ANALYSIS

We receive the notion and acquaintance of our recommended algorithmic system by utilizing it for various datasets:

- In how many cycles of execution each or more of the machines will run to accomplish all or most of the tasks.
- Stintnecessary for the structure to gauge the implementationcomponent for task mapping is delay in executing the algorithm.
- The response stint is the structure's time needed to complete all the tasks.

We found variations in parameters by calculating the mean of all three parameters based on these comparisons. That supports our results and the approaches that we have suggested.From the high points during our experimentations, it is apparent that in comparison with the results obtained from the CERS and DAIRS algorithms, the effects generated by our recommended technique are promising and encouraging.

ISSN: 2233-7857 IJFGCN

5. INFERENCE & UPCOMING WORK

In our research, we have thought over on few areas in which our work can be utilized. These are; a) taking different kind of datasets like medical decision support system, financial systems, b) in order to further enhance the efficiency of the algorithm, the influence of the parameters must be explored, the mathematical analysis on the results be developed, new analytical methods developed and criteria for assessing and suggesting more accuracy, c) improving the process and using it in business or industry systems like Cloud oriented systems, IoT systems, industrial manufacturing programs, conformance of probabilistic theories of economics, health and societal computations and so on. This work can be further enhanced by comparing the performance of mutually dependent tasks. Further work is required in order to understand and extend the concept in the planning of machine learning tasks and to evaluate better outcomes. In keeping with world-class datasets and files, we aim to find more experiments to reinforce our conceptual structure and algorithms.These experiments can also target data sets from Covid-19 to generate other results. We look forward to using our work to produce more rational results to support the use of resources by planning work on the basis of our machine learning algorithm.

REFERENCES

[1] H. Zhao and R. Sakellariou, ‘‘Scheduling multiple DAGs onto heterogeneous systems,’’ in Proc. 20th IEEE Int. Parallel Distrib. Process. Symp., Apr. 2006, p. 14.

[2] G. Xie, G. Zeng, L. Liu, R. Li, and K. Li, ‘‘High performance real-time scheduling of multiple mixed-criticality functions in heterogeneous distributed embedded systems,’’ J. Syst. Archit., vol. 70, pp. 3–14, Oct. 2016.

[3] R. Duan, R. Prodan, and X. Li, ‘‘Multi-objective game theoretic scheduling of bag-of-tasks workflows on hybrid clouds,’’ IEEE Trans. Cloud Comput., vol. 2, no. 1, pp. 29–42, 2014.

[4] D.P. Bertsekas, ‘‘Feature-based aggregation and deep reinforcement learning: A survey and some new implementations,’’ IEEE/ACM Trans. Audio, Speech, Language Process., pp. 1–31, 2018.

[5] R. Lowe, Y. Wu, A. Tamar, J. Harb, O. P. Abbeel, and I. Mordatch, ‘‘Multi-agent actor-critic for mixed cooperative-competitive environments,’’ in Advances in Neural Information Processing Systems 30, I. Guyon et al., Eds. Red Hook, NY, USA: Curran Associates, 2017, pp. 6379–6390.

[6] W. Tian, Y. Zhao, Y. Zhong, M. Xu and C. Jing, "A dynamic and integrated load-balancing scheduling algorithm for Cloud datacenters," *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, Beijing, 2011, pp. 311-315, doi: 10.1109/CCIS.2011.6045081.

[7] D. Zhong and Z. Li, "The study of scheduling function optimization in CERS based on speech enhancement algorithm," 2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet), Yichang, 2012, pp. 3652-3656, doi: 10.1109/CECNet.2012.6202151.

[8] J. Li, J. Liu and J. Wang, "An Improved Differential Evolution Task Scheduling Algorithm Based on Cloud Computing," 2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Wuxi, 2018, pp. 30-35, doi: 10.1109/DCABES.2018.00018.

[7] Lohi Shantanu A., Jaiswal A.A., Gorewar H.V., Lohi Snehal, Analytical Assessment of Nature- Inspired Metaheuristic Algorithms to Elucidate Assembly Line Task Scheduling

ISSN: 2233-7857 IJFGCN

Problem. In: Advances in Intelligent Systems and Computing, vol. 933, pp. 217-228. Springer, Singapore, 2019

[8] Lohi Shantanu A., Nandita Tiwari, Assessment of suitability of metaheuristics and machine learning for task scheduling process: A review of aptness in heavy task environments. In: Smart Innovations, Systems and Technologies, vol. 182, Springer,Singapore, 2020.

[9] S.A. Lohi, N. Tiwari et.al. “Analysis and review of effectiveness of metaheuristics in task scheduling process with delineating machine learning as suitable alternative,”2020 International Conference on Innovative Trends in Information Technology (ICITIIT), Indian Institute of Informaion Technology, Kottayam, Kerala,India, 2020, pp. 1-6, IEEE Xplore, doi: 10.1109/ICITIIT49094.2020.9071559.

[10] Gorewar H.V.,Tiwari N., “A review on Indian UID based automated electoral roll generation mechanism shunning duplication and redundancy”, In: Yu-Dong Zhang, Tomonoby Senjyu, Chakchai So-In, Amit Joshi, Smart Trends in Computing and Communications: Proceedings of SmartCom, Advances in Intelligent Systems and Computing, vol. 182. Springer, Singapore, ISBN 978-981-15-5223-6. https://link.springer.com/chapter/ 10.1007/978-981-15-5224-3_42, 2020.

[11] Jogekar R. N.,Tiwari N. “A review of deep learning techniques for identification and diagnosis of plant leaf disease”, In: Yu-Dong Zhang, Tomonoby Senjyu, Chakchai So-In, Amit Joshi, Smart Trends in Computing and Communications: Proceedings of SmartCom, Advances in Intelligent Systems and Computing, vol. 182. Springer, Singapore, ISBN 978-981-15-5223-6.URL https://link.springer.com/chapter/ 10.1007/978-981-15-5224-3_43, 2020.

[12] L. Ran, X. Shi and M. Shang, "SLAs-Aware Online Task Scheduling Based on Deep Reinforcement Learning Method in Cloud Environment," 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 2019, pp. 1518-1525, doi: 10.1109/HPCC/SmartCity/DSS.2019.00209.

[13] X. Shi, J. Dong, S. M. Djouadi, Y. Feng, X. Ma and Y. Wang, "Papmsc: power-aware performance management approach for virtualized web servers via stochastic control", Journal of Grid Computing, vol. 14, no. 1, pp. 171-191, 2016.

[14] M. C. Silva Filho, R. L. Oliveira, C. C. Monteiro, P. R. Inácio and M. M. Freire, "Cloudsim plus: a cloud computing simulation framework pursuing software engineering principles for improved modularity extensibility and correctness", 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 400-406, 2017.

[15] M. Kuo, H. Chang and D. Chang, "A Disk State-Aware Task Scheduler with Energy Efficient Prefetching and Caching," 2013 International Conference on Cloud and Green Computing, Karlsruhe, 2013, pp. 42-47, doi: 10.1109/CGC.2013.15.

[16] L. Zohrati, M. Nooraei Abadeh and E. Kazemi, "Flexible approach to schedule tasks in cloud-computing environments," in IET Software, vol. 12, no. 6, pp. 474-479, 12 2018, doi: 10.1049/iet-sen.2017.0008.

[19] Yangyang Gao, Fei Song and Yajuan Qin, "A novel real-time traffic identification mechanism," 2011 International Conference on Advanced Intelligence and Awareness Internet (AIAI 2011), Shenzhen, 2011, pp. 72-75, doi: 10.1049/cp.2011.1430.

[20] J. E. Bogess and A. Mukheeth, "Evolving engineering mission schedules: a machine-learning approach to scheduling," Proceedings of the IEEE 1997 National Aerospace and Electronics Conference. NAECON 1997, Dayton, OH, USA, 1997, pp. 749-756 vol.2, doi: 10.1109/NAECON.1997.622724.

[21] Z. Cao, H. Zhang, Y. Cao and B. Liu, "A Deep Reinforcement Learning Approach to Multi-Component Job Scheduling in Edge Computing," 2019 15th International

ISSN: 2233-7857 IJFGCN

Conference on Mobile Ad-Hoc and Sensor Networks (MSN), Shenzhen, China, 2019, pp. 19-24, doi: 10.1109/MSN48538.2019.00018.

Authors

Shantanu Anandrao Lohi is the corresponding author and he is currently pursuing Ph.D. from the Dept. of Computer Science and Engineering at RKDF Institute of Science and Engineering at Sarvepalli RadhaKrishnan University, Bhopal, MP, India. He completed his master's degree in Computer Science and Engineering from RTM Nagpur University. His research areas include, Artificial Intelligence, Machine Learning and its applications in task scheduling and development of educational theories, Cognitive Computational Science and its applications in teacher-student relationship and employability factors, Deep Learning, Metaheuristics, Software Engineering, Software Planning and Management and Engineering Education.

Dr. Nandita Tiwari is an Associate Professor at the Dept. of Computer Science and Engineering at RKDF Institute of Science and Engineering at Sarvepalli RadhaKrishnan University, Bhopal, MP, India. She completed her master's degree in Computer Science and Engineering from Rajiv Gandhi Technical University, Bhopal, MP, India and obtained her Ph.D. degree in Computer Science and Engineering from Sarvepalli RadhaKrishnan University, Bhopal, MP, India. She has vast experience of more than fourteen years in teaching and research. Her research interest includes Artificial Intelligence, Machine Learning and its applications, Deep Learning, Metaheuristics, Software Engineering, Mobile Ad-hoc Networks, Computer Security.

ISSN: 2233-7857 IJFGCN