Webology (ISSN: 1735-188X)
Volume 18, Number 6, 2021

Optimization Of Tasks Scheduling In Computational Grids Using Hybrid Swarm Intelligence Algorithm

Kishor Shamrao Sakure[1] , Rajesh Kumar Boghey[2]

[1,2]Department of Computer Science & Engineering Madhyanchal Professional University, Bhopal, India.

Abstract

The scalability and reliability of computational grids are challenging tasks in a next-generation computational framework. The growing demands of distributed resources face a problem of allocation and decline the service quality of computational grids. The optimization of resources is a way to handle the problem of computational grids. This paper proposed a hybrid swarm intelligence-based task scheduling algorithm for allocating resources in computational grids. The proposed algorithm encapsulates two algorithms ant colony optimization and particle swarm optimization. The ant colony optimization algorithm handles the global request of resources, and particle swarm optimization handles the resource's local searching. The hybrid swarm intelligence algorithm scales the resource allocation process and reduces the job failure rate instead of other swarm intelligence algorithms. The proposed algorithm was simulated in MATLAB tools and tested three sizes of grid matrices. The size of the grid matrix varies from lower to higher and increase the load in respect of resource. The proposed hybrid swarm intelligence algorithm compares with the existing resource optimization algorithm. The analysis of results suggests that the proposed algorithm increase 2-3 % of job completion instead of the existing algorithm.

Keywords: - Grid, High-Performance Computing, Swarm Intelligence, Optimization, Scheduling, PSO, ACO

Introduction

The emerging technology of computer and network demand quality computational and increase the utilization of resources. The process of emergence developed the concept of grid computing that employ a group of computer resources across various places to provide the solution for a complex problem. The optimization and allocation of resources accelerate the efficiency of grid computing environments [1,2,3]. The grid system comprises a collection of heterogeneous systems and network devices that have increased the application's complexity. The heterogeneity is visible in all aspects, including functionality, input data, and transmission. There are two types of scheduling methods: static and dynamic. The scheduling performed by the compiler during compile time is known as static scheduling, and some of the constraints considered during the process are communication cost, data dependency, and synchronization. Moreover, dynamic scheduling refers to scheduling that occurs in run-time by the processor and requires much sensible inference. The management of resource allocation using methods of scheduling approach in conventional and dynamic scheduling [4,5,6]. The conventional scheduling approach applies CPU scheduling methods with the defined condition and declines the performance of computational grid computing. The dynamic scheduling of resource allocation and handling of tasks uses the swarm intelligence function to optimize the allocation and handling of tasks. The dynamic nature of the grid [12] adds more complexity to GSP, and hence it becomes essential to explore other pathways for developing a suitable technique for the GSP. The use of bio-inspired principles [7, 8] in computing has given several useful optimizations [17] techniques that can be applied to different areas. They are grouped as swarm [32] and evolutionary methods [13]. The methods whose behavior are driven by nature and natural phenomenon are classified as swarm-based algorithms. PSO [19]and Ant Colony Optimization (ACO) [9, 20] are members of the swarm family. A class of evolutionary algorithms [16] like Genetic algorithms (GA) [26] have gained substantial popularity in solving an optimization problem. These algorithms are equipped with helpful exploration and exploitation capabilities to search the solution space, making them a potential candidate for solving GSP. Although many algorithms have been developed to solve GSP, no algorithm has been proven the best by the researchers because GSP is an NP-complete problem. PSO has fewer parameters compared to GA or ACO. Therefore, it is very much suitable for combinatorial optimization problems. This paper proposed an optimized task scheduling algorithm using ACO and PSO. The hybrid algorithm of ACO and PSO decreases the job failure rate and increases job completion in computational grid environments. The proposed algorithm is scalable such that it can be applied to data sets of varying sizes. Furthermore, it performs load balancing, and resource utilization is also improved. Also, the working of algorithms is evaluated by a standard workload that gives actual results about the proposed algorithm and helps make necessary comparisons with other approaches. The arrangement of paper as in section II related work, in section III, describes the proposed methodology, and section IV describes the experimental analysis of the proposed algorithm and existing algorithms. Moreover finally, conclude with the future direction in section V.

II. Related Work

The demand for quality of computational tasks brings the attention of various research scholars and scientists in grid computing. The incremental approach of scholars achieves milestone improvements in task scheduling and resource allocation. Some significant contribution of authors describes here. In [1] author propose task scheduling at the cutting edge of Desktop Grid computing systems. Author describes the overall architecture of a Desktop Grid system as well as the BOINC middleware's computing model. Finally, author formulate and briefly describe a number of unresolved issues in Desktop Grid task scheduling. In [2] author propose a novel DNCPSO that uses non-linear inertia weight with selection and mutation operations via a directional search process to dramatically reduce make span and cost while achieving a compromising. The outcomes of simulation experiments using various real and random workflow examples show that their DNCPSO outperforms other classical and improved algorithms, demonstrating DNCPSO's effectiveness and efficiency. In [3] author propose an intelligent scheduling system from the user's perspective to reduce workflow expenditure, subject to deadlines and other execution constraints A new task execution time estimation model is developed based on virtual machine settings in real public clouds and execution data from practical workflows In [4] author gives a comprehensive overview of cloud computing fault tolerance issues, emphasizing key concepts, architectural details, and cutting-edge techniques and methods The goal is to provide insights into the current fault - tolerant approaches as well as obstacles that still need to be overcome. The survey identifies critical research instructions in this area, as well as a few promising techniques that could be used to develop efficient solutions. In [5] author identifies a comprehensive overview of cloud computing fault tolerance issues, emphasizing key concepts, architectural details, and cutting-edge techniques and methods The survey identifies critical research directions in this area, as well as a few promising techniques that could be used to develop efficient solutions. In [6] author propose a classification based on the RL-based technique employed on the first level of the taxonomy, proposals in the Model-based and Model-free categories are presented. On a second level, the proposals in the Model free category are divided into three groups. In [7] author propose a novel dynamic assignment procedure in which these two parameters are determined by the number of colluded malicious resources (CMRs) discovered thus far. As demonstrated by examples, the discussed dynamic spot-checking optimization significantly outperforms static spot checking. In [8] author gives a brief background of this emerging field, a review of current hyper-heuristic literature, and a discussion of recent hyper-heuristic frameworks Furthermore, the existing classification of selection hyper-heuristics is expanded to reflect the nature of the challenges faced in scientific literature. In [9] author find A non-dominated sorting-based particle swarm optimization approach is proposed to find an optimal schedule for workflow applications in cloud computing systems. Simulation studies and comparisons with other representative algorithms in the literature show that the discussed algorithm is promising. In [10] author utilized Grid computing is a type of distributed computing in which computing power, data storage, applications, and network resources are coordinated across dynamic and geographically dispersed organizations. Trust is a multidimensional factor that is influenced by factors such as the entity's reputation, policies, and opinions. This study discusses a trust-based scheduling approach. In [11] author proposed the various optimization techniques such as Genetic algorithm, Multi queue, Ant Colony optimization, particle swarm optimization, selective breeding, taboo search, Lion optimization techniques, and firefly algorithms for cloud scheduling problems and simulated few algorithms such as ACO, PSO, GA, LOA and compared their outcomes with firefly algorithm in order to improve the performance of cloud scheduling and workflow management in cloud. In [12] author propose a desktop grid trust model based on Dempster–Shafer theory that predicts the relative reliability of nodes based on daily computer usage behaviour based on historical data from a desktop grid platform TMDG can fully utilize the most reliable nodes for a given computation, culminating in less communication overhead and more platform computing power. In [13] author using the performance of a grid resource is modelled and evaluated SRNs, which take into account the failure-repair behaviour of its processors. As an outcome, it can be used to compute the PMF of the entire grid environment's service time for a workflow with multiple dependent programmers. In [14] author propose CMI is a novel online multi-objective auto scaler for workflows that aims to minimize make span, monetary cost, and the potential impact of errors caused by unreliable VMs. These findings provide a solid foundation for further research into other meta-heuristic methods for autoscaling workflow applications using cheap but unreliable infrastructures. In [15] author propose a set of cost and time-aware cloud workflow scheduling algorithms aimed at providing researchers with a variety of appropriate cloud workflow scheduling approaches in various scenarios. They performed a comprehensive review of various cloud workflow scheduling algorithms and classified them based on their optimization objectives and constraints. In [16] author created A balanced load by employing both the FA and the PSO heuristics. The goal is to balance the load of the entire system while also reducing the make span of a set of tasks. The outcomes of this experiment demonstrated that the discussed FA outperformed min–min scheduling, PSO, and first come, first served methods. In [17] author propose a provisioning and scheduling framework that explicitly addresses the cloud infrastructure's and workload's uncertainties and performance variability. Experiments show that the resource provisioning and scheduling plans identified by their approach effectively deal with uncertainties while also meeting the application deadline. In [18] author use Different load balancing techniques to improve cloud computing performance. Five algorithms are used to balance the load between an SG user's requests and service providers: round robin, throttled, ABC, ACO, and PSO. The simulation outcomes show that their discussed technique, HABACO, outperforms the other techniques. In [19] author examine a review of the literature on resource allocation techniques in cloud computing technology Cloud computing is a new generation technology that enables users to share resources across any communication network by utilizing the virtualization technique. In [20] author propose the ARRA, which commits advanced reservation of resources to users with the least amount of task waiting time. A simulation study reveals the algorithm's motivation in terms of load balancing. Experimental outcomes show the success of discussed algorithm in terms of resource allocation. In [21] author proposed in terms of make span and cost, three Particle Swarm Optimization (PSO)-based algorithms are compared. These algorithms were put through their paces using the same number of virtual machines (VMs) and workflows. In the same working environment, simulation experiments show that ACO-PSO outperforms basic PSO, C-PSO, and PSO-DS. In [22] author propose a dynamic resource allocation model for scheduling data-intensive applications on a hybrid mobile cloud computing environment comprised of mobile devices, cloudlets, and public cloud. The outcomes show that the discussed technique reduces the execution time for data-intensive applications by an average of 72% and mobile energy consumption by an average of 86%. In [23] author propose a model for task scheduling in cloud computing based on a HSLGSAFA. The hybrid model under consideration combines the GSA, which has been successfully scheduling tasks in the application, with the SL strategy and the FA. The developed solution is capable of ensuring user-level QoS while also increasing the credibility and economic benefit of IaaS providers. In [24] author propose a Grid computer must also deal with networks with varying topologies and geographical spread that are not primarily intended to connect to a cluster of computers. As an outcome, this research focuses on an efficient resource management scheduling and optimization approach that uses Ant colony optimization and round robin scheduling to achieve low execution intervals with low error rate probabilities. In [25] propose an improved particle swarm optimization algorithm with adaptive parameters to protect the algorithm from premature convergence to achieve a better outcome, the Cloud Sim simulator is used to simulate the operation of the algorithm, and its performance was compared to that of the classic particle swarm optimization method. In [26] author propose a statistical method for estimating the completion time of a batch of tasks in a Desktop Grid It is presented a statistical approach based on the Holt model. The outcomes of numerical experiments based on statistics from the Rake Search and LHC home volunteer computing projects are presented. In [27] author PPM approaches are used to estimate performance metrics such as execution duration, required memory, or wait times of upcoming jobs and tasks based on past performance observations. Author categories and compare the sources of performance variation, predicted performance metrics, required training data, use cases, and underlying prediction techniques. In [28] author propose a formal model for the problem of mapping data sources across fog nodes the discussed optimization problem takes both communication latency and processing time on the fog nodes into account. their findings demonstrate that the discussed heuristic can be used for optimization in the scenario under consideration. They also run a sensitivity analysis on the main heuristic parameters. In [29] author propose a probabilistic formulation of the optimization problem with the goal of minimizing the expected cost of deploying a parallel application within a given time constraint A Genetic Algorithm is used to find a suboptimal solution to the problem. In [30] author propose the provisioning of resources and the issues associated with it in today's cloud computing environment Cloud computing, scheduling, SLA, virtualization, and virtual machines are some of the terms used in this article. Distributed computing is a developing technology that provides customers with compelling administrations. In [31] author propose a thorough examination of previous and current HEMS research, taking into account various DR programmers, smart technologies, and load scheduling controllers Artificial intelligence applications for load scheduling controllers, such as artificial neural networks, fuzzy logic, and adaptive neural fuzzy inference systems, are also discussed.

III. Proposed Methodology

The proposed algorithm accelerates task scheduling in a grid computing environment. The scheduling algorithm encompasses particle swarm optimization (PSO) and ant colony optimization (ACO). The ACO algorithms reduce the maximum load of tasks based on the nature of similarity. The particle swarm optimization allocates the resource for the execution of jobs. The processing of ant and PSO as the global and local optimal solution for the processing of grid task scheduling[24,25].

The processing of hybrid algorithm process the different tasks as t1,t2,……………..,tn. The W is weight factor of all task assign for the grid environments. τ is the value of pheromones of ants, v1 and v2 is velocity of particle agents, c1 and c2 is constants value of particle? The allocation process of task describe here.

Begin 1 define the value of task set T={ t1,t2,…………..,tn} in PSO population

a. Define the value of velocity V1=0 and V2=0 and W=0
b. Selection of tasks as ants of given constraints function

$$F(T) = \frac{(FLt-FGt)}{LtX\ Gt}, Wi\epsilon T(t1, t2 \ldots \ldots tn) \ldots \ldots \ldots \ldots \ldots (1)$$

Here Lt is local task and Gt is global task of set of define task matrix of grid

The selected tasks assign as the ants of ACO algorithm TA={at1,at2,………..,atn} the local optimal estimation function as

$$Pbest = \begin{cases} \frac{(\tau_i)^{\alpha}\left(LI_i^{S_j}\right)^{\beta}}{\sum_{g\notin S_j}(\tau_g)^{\alpha}\left(LI_g^{S_j}\right)^{\beta}} & \text{if } i \notin S_j \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots (2) \\ 0 & \text{otherwise} \end{cases}$$

Here τ_i is phenomenon value of ants and LI is value of least interface of ants.

Begin 2. Exchange the local best to global best allocation

Put the local best task matrix for the processing of resource allocation.

1. Estimate difference of tasks as relative difference
   $RT = \frac{TSI}{Wd}$ Here Tsi is interference value of ants and Wd is sum value of PSO space.
2. The PSO space creates the allocation states of resources
   $$Allocation = \begin{cases} \frac{\max\ (RF)\ -F(s)}{\max_{h=1:}(WS)} & \text{if } T_i \in RT_j \ldots \ldots \ldots \ldots \ldots \ldots \ldots (3) \\ 0 & \text{otherwise} \end{cases}$$
3. Measure relative distance of jobs as
   $$Rd = \sum_{fd=1}^{n}\sum_{pf=1}^{m}(Ti - Tr) \ldots \ldots \ldots \ldots \ldots \ldots (4)$$
4. The value of Rd is zero resource is allocated to dedicated tasks.
5. Else the process of allocation goes into steps 2

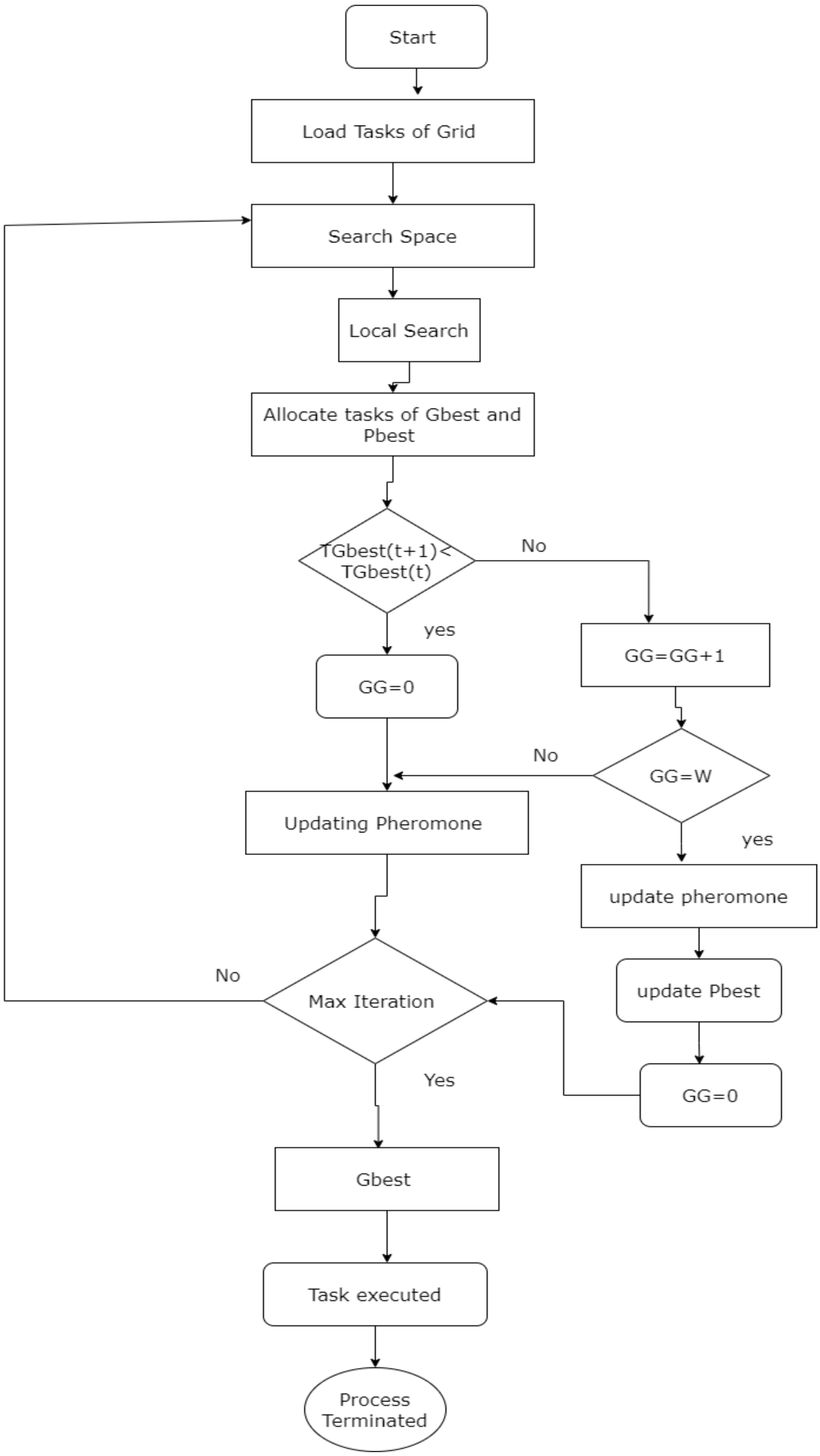


Fig: 1 Proposed model of hybrid task scheduling in computational Grid environments

IV. Experimental Analysis

To validate the Hybrid task scheduling algorithm simulated in MATLAB Software. The process of simulation undergone in different size of grid matrix as 10 X10, 20X20 and 40X 40. The performance parameters were measured as job completion and failure rate [25,27,28,31].

Job completion

Job completion is one of the most important standard metrics used to measure the performance of fault tolerant systems [5]. Job completion is defined as:

$$\text{job Completion} = \frac{\mathrm{n}}{\mathrm{T}}$$

Where n is the total number of jobs submitted and Tn is the total amount of time necessary to complete n jobs. Job completion is used to measure the ability of the grid to accommodate jobs.

Job Failure

It is the percentage of the tendency of the selected grid resources to fail and is defined as:

$$\text{failure} = \frac{\sum_{j=1}^{m} \mathrm{Pfj}}{\mathrm{m}} \times 100\%$$

Where m is the total number of grid resources and Pfj is the failure rate of resource j. Through this metric, the faulty behaviour of the system can be expected.

Table1: Result analysis of Small Job (10*10) Dataset using MF-ACO, MF-PSO and Proposed model with the help of Job Failure rate and job completion rate Using number of jobs and number of resources.

SMALL JOB (10*10)							
		MF-ACO		MF -PSO		PROPOSED	
No. Of Jobs	No. Of Resources	Job Failure Rate	Job Completion Rate	Job Failure Rate	Job Completion Rate	Job Failure Rate	Job Completion Rate
100	50	3.26	85.16	3.58	86.14	3.78	86.34
200	100	3.56	85.35	3.71	85.98	3.97	86.32
300	150	4.11	85.98	4.31	86.34	4.96	87.24
400	200	4.65	86.58	4.96	87.04	5.34	87.34
500	250	4.98	86.94	5.13	87.32	5.67	87.98
600	300	5.01	87.24	5.48	87.83	5.95	88.42
700	350	5.23	88.87	5.81	89.54	6.37	90.54
800	400	5.46	89.65	5.88	90.56	6.41	91.46
900	450	5.67	90.03	6.21	90.97	6.87	91.57
1000	500	5.94	91.24	6.78	92.07	7.04	92.78

Table 2: Result analysis of Middle Job (20*20) Dataset using MF-ACO, MF-PSO and Proposed model with the help of Job Failure rate and job completion rate Using number of jobs and number of resources.

Middle Job (20*20)							
		MF-ACO		MF -PSO		PROPOSED	
No. Of Jobs	No. Of Resources	Job Failure Rate	Job Completion Rate	Job Failure Rate	Job Completion Rate	Job Failure Rate	Job Completion Rate
100	50	4.25	83.33	4.49	83.65	4.78	84.35
200	100	4.35	84.65	4.56	84.87	4.89	85.01
300	150	4.59	85.35	4.75	85.97	4.96	86.78
400	200	5.33	85.95	5.36	86.42	5.44	86.97
500	250	5.34	86.38	5.46	86.79	5.67	87.11
600	300	5.61	86.65	5.71	86.98	5.87	87.32
700	350	5.89	87.32	5.97	87.69	6.12	87.98
800	400	6.23	87.56	6.31	87.97	6.74	88.34
900	450	6.98	88.33	7.01	88.63	7.36	88.91
1000	500	7.23	88.96	7.38	89.34	7.84	89.54

Table 3: Result analysis of large Job (40*40) Dataset using MF-ACO, MF-PSO and Proposed model with the help of Job Failure rate and job completion rate Using number of jobs and number of resources.

Large Job (40*40)							
		MF-ACO		MF -PSO		PROPOSED	
No. Of Jobs	No. Of Resources	Job Failure Rate	Job Completion Rate	Job Failure Rate	Job Completion Rate	Job Failure Rate	Job Completion Rate
100	50	3.18	89.01	3.41	89.34	3.65	89.54
200	100	3.45	89.62	3.62	89.89	3.98	90.14
300	150	3.78	90.23	3.94	90.65	4.26	90.85
400	200	4.01	90.67	4.23	90.96	4.78	91.23
500	250	4.33	91.58	4.56	91.73	4.96	91.97
600	300	4.73	91.65	4.98	91.98	5.23	92.56
700	350	4.91	92.36	5.23	92.34	5.57	92.82
800	400	5.33	92.68	5.78	92.91	5.98	93.25
900	450	5.41	93.45	5.95	93.78	6.34	93.96
1000	500	5.71	94.33	6.01	94.57	6.48	94.78

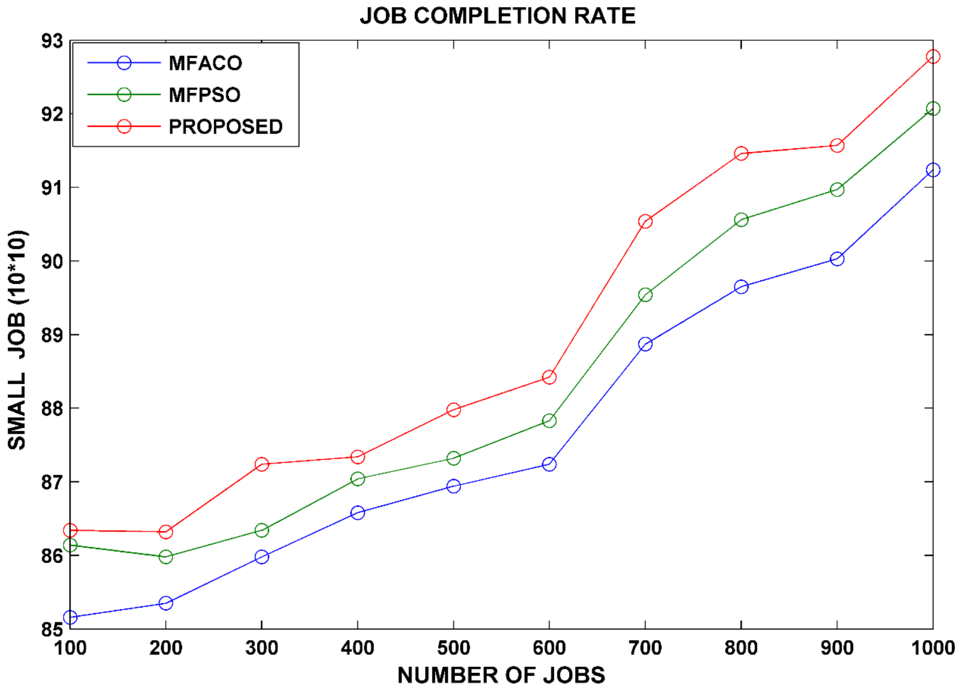
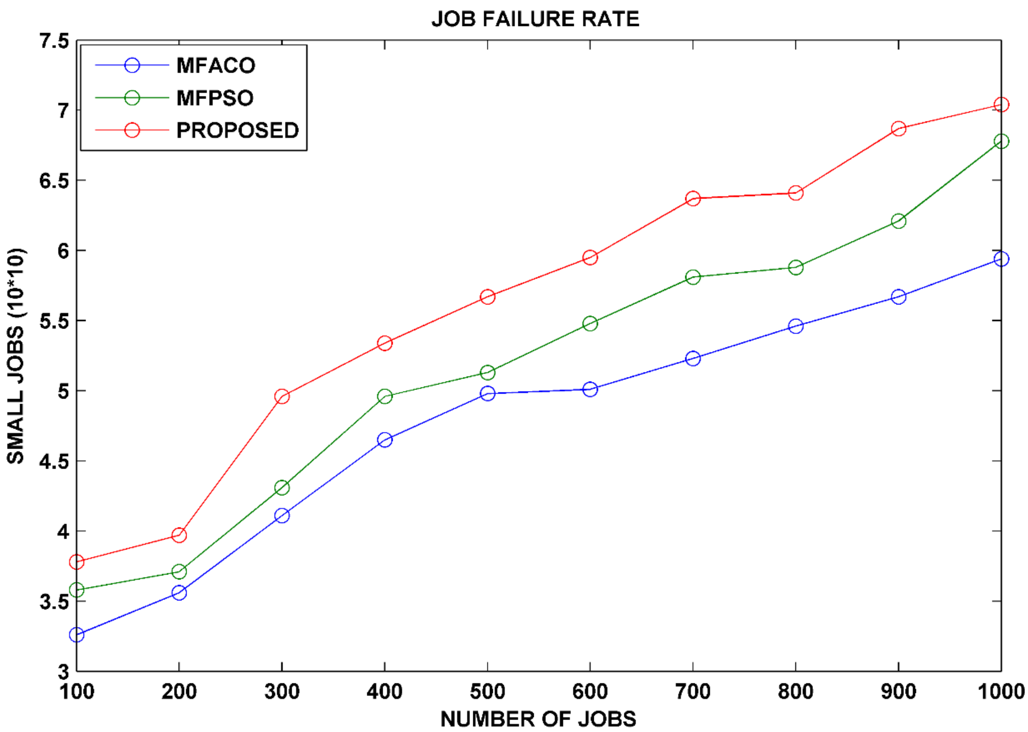


Fig 2: Comparative analysis of small job (10*10) Dataset using MF-ACO, MF-PSO and Proposed model with the help of job completion rate using number of jobs. Here we observe that the job completion rate of that proposed is better than other two techniques MF-ACO, MF-PSO.



.

Fig 3: Comparative analysis of small job (10*10) Dataset using MF-ACO, MF-PSO and Proposed model with the help of job failure rate using number of jobs. Here we observe that the job failure rate of that proposed is better than other two techniques MF-ACO, MF-PSO.

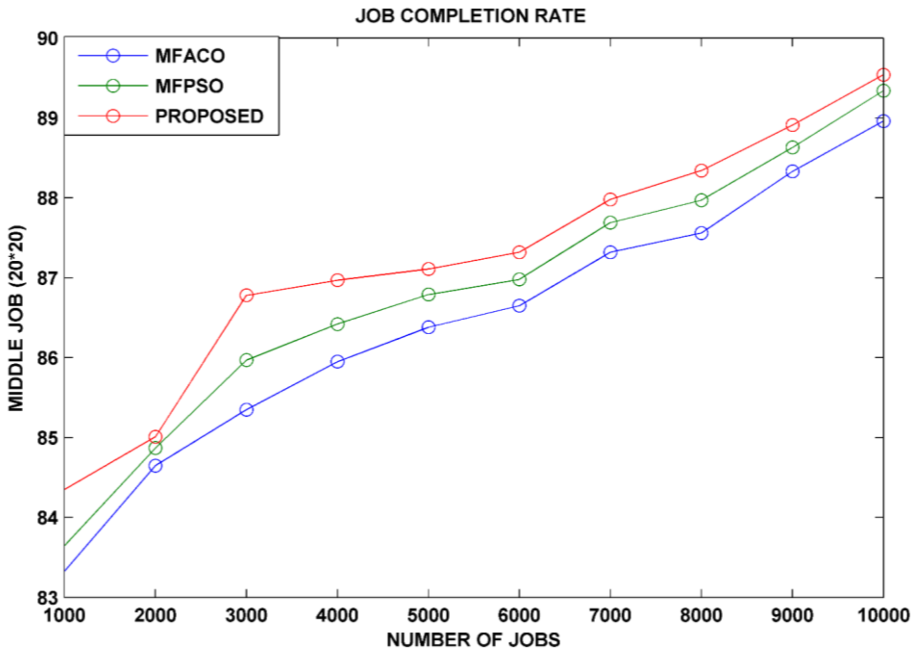


Fig 4: Comparative analysis of middle job (20*20) Dataset using MF-ACO, MF-PSO and Proposed model with the help of job completion rate using number of jobs. Here we observe that the job completion rate of that proposed is better than other two techniques MF-ACO, MF-PSO.

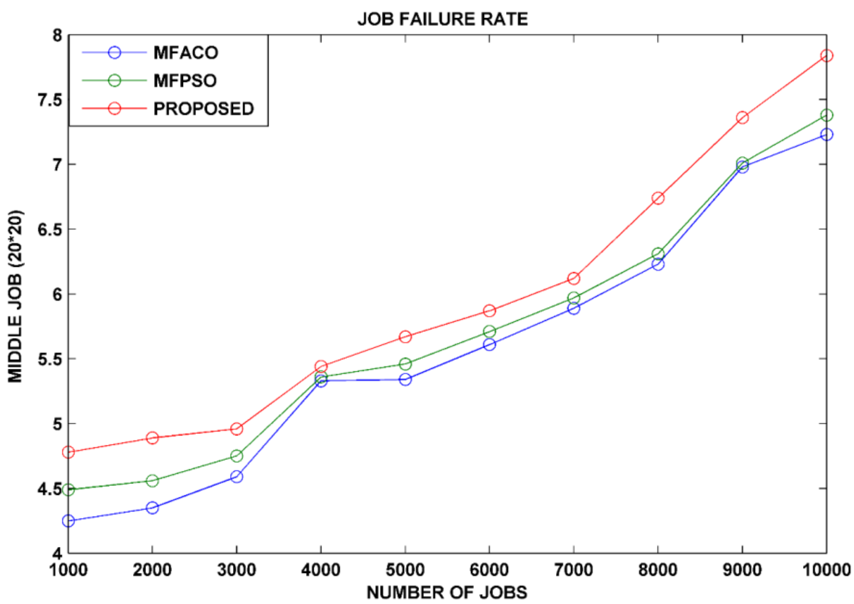


Fig 5: Comparative analysis of middle job (20*20) Dataset using MF-ACO, MF-PSO and Proposed model with the help of job failure rate using number of jobs. Here we observe that the job failure rate of that proposed is better than other two techniques MF-ACO, MF-PSO.

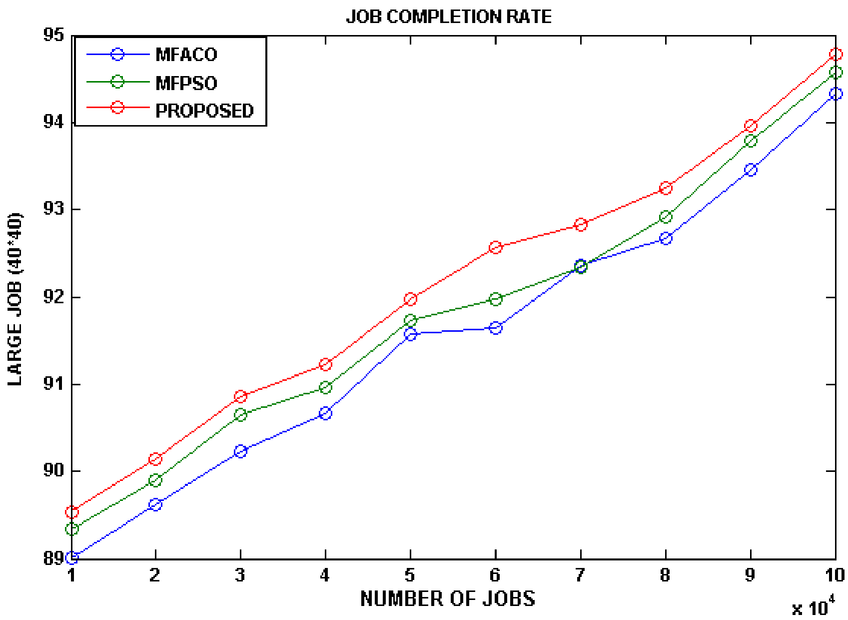


Fig 6: Comparative analysis of large job (40*40) Dataset using MF-ACO, MF-PSO and Proposed model with the help of job completion rate using number of jobs. Here we observe that the job completion rate of that proposed is better than other two techniques MF-ACO, MF-PSO.

Fig 7: Comparative analysis of large job (40*40) Dataset using MF-ACO, MF-PSO and Proposed model with the help of job failure rate using number of jobs. Here we observe

that the job failure rate of that proposed is better than other two techniques MF-ACO, MF-PSO.

V. Conclusion & Future Work

The performance of a hybrid task scheduling algorithm overcomes the limitation of job failure and job compilation in grid computing. The proposed algorithm compared with other meta-heuristic functions MF-ACO and PSO for task scheduling in grid computing. The hybrid algorithm combines the concept of the relative difference of tasks based on weight factors with the existing PSO algorithm to overcome the existing drawback of the classical PSO algorithm and make them more suitable for such a heterogeneous and dynamic environment. The objective set for this work is to present an efficient task scheduling mechanism that can reduce job failure, a minor degree of imbalance, a better performance ratio with the reduction in consumption of the system's resources. The effectiveness of the proposed algorithm is evaluated with the help of several simulations with the varying value of the parameters like the number of tasks, size of tasks and number of sizes of grid etc. As a result, the proposed hybrid-based task scheduling mechanism can significantly reduce job failure and be better effective. In future, we will try to extend the functionality of our proposed algorithm by including more features of underlying computing resources in a grid computing environment.

References

[1]. Ivashko, Evgeny, Ilya Chernov, and Natalia Nikitina. "A survey of desktop grid scheduling." IEEE Transactions on Parallel and Distributed Systems 29, no. 12 (2018): 2882-2895.

[2]. Xie, Ying, Yuanwei Zhu, Yeguo Wang, Yongliang Cheng, Rongbin Xu, Abubakar Sadiq Sani, Dong Yuan, and Yun Yang. "A novel directional and non-local-convergent particle swarm optimization-based workflow scheduling in cloud–edge environment." Future Generation Computer Systems 97 (2019): 361-378.

[3]. Jia, Ya-Hui, Wei-Neng Chen, Huaqiang Yuan, Tianlong Gu, Huaxiang Zhang, Ying Gao, and Jun Zhang. "An intelligent cloud workflow scheduling system with time estimation and adaptive ant colony optimization." IEEE Transactions on Systems, Man, and Cybernetics: Systems (2018).

[4]. Kumari, Priti, and Parmeet Kaur. "A survey of fault tolerance in cloud computing." Journal of King Saud University-Computer and Information Sciences 33, no. 10 (2021): 1159-1176.

[5]. Abdulredha, Mohammed Najm, A. Attea Bara'a, and Adnan Jumaa Jabir. "Heuristic and meta-heuristic optimization models for task scheduling in cloud-fog systems: a review." Iraqi Journal for Electrical and Electronic Engineering 16, no. 2 (2020).

[6]. Garí, Yisel, David A. Monge, Elina Pacini, Cristian Mateos, and Carlos García Garino. "Reinforcement learning-based application autoscaling in the cloud: A survey." Engineering Applications of Artificial Intelligence 102 (2021): 104288.

[7]. Levitin, Gregory, Liudong Xing, Barry W. Johnson, and Yuanshun Dai. "Optimization of dynamic spot-checking for collusion tolerance in grid computing." Future Generation Computer Systems 86 (2018): 30-38.

[8]. Drake, John H., Ahmed Kheiri, Ender Özcan, and Edmund K. Burke. "Recent advances in selection hyper-heuristics." European Journal of Operational Research 285, no. 2 (2020): 405-428.

[9]. Ajeena Beegom, A. S., and M. S. Rajasree. "Non-dominated sorting based PSO algorithm for workflow task scheduling in cloud computing systems." Journal of Intelligent & Fuzzy Systems 37, no. 5 (2019): 6801-6813.

[10]. Kumar, E. Saravana, and K. Vengatesan. "Trust based resource selection with optimization technique." Cluster Computing 22, no. 1 (2019): 207-213.

[11]. Seetha, Hari Seetha. "Nature Inspired Optimization Techniques for Cloud Scheduling Problem." (2018).

[12]. Shang, Yizi, and Ling Shang. "Trust model for reliable node allocation based on daily computer usage behavior." Concurrency and Computation: Practice and Experience 30, no. 6 (2018): e4346.

[13]. Entezari-Maleki, Reza, Kishor S. Trivedi, Leonel Sousa, and Ali Movaghar. "Performability-based workflow scheduling in grids." The Computer Journal 61, no. 10 (2018): 1479-1495.

[14]. Monge, David A., Elina Pacini, Cristian Mateos, Enrique Alba, and Carlos García Garino. "CMI: An online multi-objective genetic autoscaler for scientific and engineering workflows in cloud infrastructures with unreliable virtual machines." Journal of Network and Computer Applications 149 (2020): 102464.

[15]. Aruna, M., D. Bhanu, and S. Karthik. "An improved load balanced metaheuristic scheduling in cloud." Cluster Computing 22, no. 5 (2019): 10873-10881.

[16]. Bandali, Mehdi, Alireza Hassanzadeh, Masoume Ghashghaie, and Omid Hashemipour. "Accepted Manuscript Journal of Circuits, Systems, and Computers."

[17]. Calzarossa, Maria Carla, Marco L. Della Vedova, and Daniele Tessera. "A methodological framework for cloud resource provisioning and scheduling of data parallel applications under uncertainty." Future generation computer systems 93 (2019): 212-223.

[18]. Zahoor, Saman, Sakeena Javaid, Nadeem Javaid, Mahmood Ashraf, Farruh Ishmanov, and Muhammad Khalil Afzal. "Cloud–fog–based smart grid model for efficient resource management." Sustainability 10, no. 6 (2018): 2079.

[19]. Leoba, Anto, Kombou Victor, and Fokam Poka Arsene. "An Economy-Based Preemption Resource Allocation in Cloud Computing." Available at SSRN 3310368 (2018).

[20]. Sheikh, Sophiya. "Load balancing on Computational Grid Using Advanced Reservation of Resources."

[21]. Farid, Mazen, Rohaya Latip, Masnida Hussin, Nor Asilah Wati Abdul Hamid, and AV Senthil Kumar. "COMPARATIVE ANALYSIS OF PSO-DERIVED WORKFLOW SCHEDULING ALGORITHMS IN CLOUD COMPUTING BASED ON QOS REQUIREMENTS."

[22]. Alkhalaileh, Mohammad, Rodrigo N. Calheiros, Quang Vinh Nguyen, and Bahman Javadi. "Dynamic resource allocation in hybrid mobile cloud computing for

data-intensive applications." In International Conference on Green, Pervasive, and Cloud Computing, pp. 176-191. Springer, Cham, 2019.

[23]. Neelima, P., and A. Rama Mohan Reddy. "An efficient hybridization algorithm based task scheduling in cloud environment." Journal of Circuits, Systems and Computers 27, no. 02 (2018): 1850018.

[24]. Aggarwal, Anuj, Rajesh Verma, and Ajit Singh. "An efficient approach for resource allocations using hybrid scheduling and optimization in distributed system." Int. J. Educ. Manag. Eng.(IJEME) 8, no. 3 (2018): 33-42.

[25]. Owoseni, Alaba T., Olatunde Iyaniwura, and Lukman Adebayo Ogundele. "ADAPTIVE PARTICLE SWARM OPTIMIZATION SCHEDULING MODEL FOR JOBS ON CLOUD." Annals. Computer Science Series 17, no. 1 (2019).

[26]. Ivashko, Evgeny, and Valentina Litovchenko. "Batch of tasks completion time estimation in a desktop grid." In Russian Supercomputing Days, pp. 500-510. Springer, Cham, 2018.

[27]. Witt, Carl, Marc Bux, Wladislaw Gusew, and Ulf Leser. "Predictive performance modeling for distributed batch processing using black box monitoring and machine learning." Information Systems 82 (2019): 33-52.

[28]. Canali, Claudia, and Riccardo Lancellotti. "A Fog Computing Service Placement for Smart Cities based on Genetic Algorithms." In CLOSER, pp. 81-89. 2019.

[29]. Calzarossa, Maria Carla, Luisa Massari, Giuseppe Nebbione, Marco L. Della Vedova, and Daniele Tessera. "Tuning Genetic Algorithms for resource provisioning and scheduling in uncertain cloud environments: Challenges and findings." In 2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP), pp. 174-180. IEEE, 2019.

[30]. Aggarwal, Ritu. "Resource provisioning and resource allocation in cloud computing environment." International Journal of Scientific Research in Computer Science, Engineering and Information Technology 3 (2018).

[31]. Shareef, Hussain, Maytham S. Ahmed, Azah Mohamed, and Eslam Al Hassan. "Review on home energy management system considering demand responses, smart technologies, and intelligent controllers." Ieee Access 6 (2018): 24498-24509.

[32]. Bhavani, B. H., and H. S. Guruprasad. "Resource provisioning techniques in cloud computing environment: a survey." International Journal of Research in Computer and Communication Technology 3, no. 3 (2014): 395-401.