ISSN No. 0976-5697

Volume 8, No. 5, May-June 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

A Hybrid Approach for Load Balancing: Using Random Forest and Pso Approach (RFPSO)

Kirti Bala
CET Department
GNDU
Amritsar, India

Anil Kumar
CET Department
GNDU
Amritsar, India

Abstract: Cloud Computing becomes need of the hour, since cloud provides resources to machines having limited resources associated with them. Load balancing becomes critical in such environment. Proposed literature utilizes Random forest along with particle swarm optimization for balancing load across available resources in cloud machines. Random schedules are prepared by the use of random forest algorithm. These schedules are fed into cloud system for execution. If the resources are not available with current vms, PSO is applied to locate the resources. Performance analysis indicates better results in terms of Makespan, Flowtime , Time and Load balancing degree.

Keywords: Cloud Computing, Random Forest, Particle Swarm Optimization, Makespan, Flowtime, Time, Load Balancing Degree

1. INTRODUCTION

[1]Cloud Computing is a sort of Internet-based assuming that gives shared PC dealing with resources and data to PCs and diverse contraptions on demand. It is a model for enabling ubiquitous, on-demand access to a typical pool of configurable figuring resources (e.g., PC frameworks, servers, stockpiling, applications and administrations), which can be immediately provisioned and released with insignificant organization effort. [2]–[4]Distributed figuring and limit courses of action give customers and endeavors distinctive capacities to store and process their data in either restrictive, or untouchable data focuses that may be arranged far from the user–ranging in division from over a city to over the world.[5] Distributed registering relies on upon sharing of advantages for achieve judiciousness and economy of scale, similar to an utility (like the power cross section) over a power sort out. Advocates attest that [6]distributed figuring empowers associations to avoid ahead of time establishment costs (e.g., procuring servers). As well, it enables relationship to focus on their middle associations rather than contributing vitality and money on PC foundation. [7]Proponents in like manner affirm that dispersed processing empowers attempts to get their applications up and running speedier, with improved sensibility and less upkeep, and engages information advancement (IT) gatherings to more rapidly change resources for meet fluctuating and strange business request. Cloud providers generally use a "pay as you go" illustrate. This will provoke all of a sudden high charges if supervisors don't acclimate to the cloud assessing model. [8]Random backwoods are an outfit learning strategy for portrayal, backslide and diverse endeavors, that work by building up an immense number of decision trees at planning time and yielding the class that is the technique for the classes (gathering) or mean desire (backslide) of the individual trees. Self-assertive decision timberlands solution for decision trees' penchant for over fitting to their planning set. [9], [10]The primary computation for discretionary decision woods was made by Tin Kam Ho using the unpredictable subspace technique, which, in Ho's enumerating, is a way to deal with execute the "stochastic partition" approach to manage arrange proposed by Eugene Kleinberg. [11]Particle Swarm Optimization (PSO) is a heuristic headway count which has been getting energy among standard analysts starting late.[12]PSO is a nature motivated calculation which is in light of the Newton's law of gravity and the law of movement.[13] PSO is gathered under the populace based approach and is accounted for to be more instinctive. The calculation is expected to enhance the execution in the investigation and abuse capacities of a populace based calculation, in light of gravity guidelines. Be that as it may, as of late PSO has been censured for not truly in light of the particles movement . PSO is accounted for to avoid the separation between masses in its recipe, though mass and separation are both necessary parts of the law of gravity. In spite of the feedback, the calculation is as yet being investigated and acknowledged by the logical group.

[14]Boosting system lifetime is a noteworthy goal for planning and conveying a remote sensor organize (WSN). Grouping sensor hubs is a compelling topology control approach accomplishing this objective. In this paper, we display another strategy to draw out the system lifetime in light of the Improved Particle Swarm Optimization calculation, which is an enhancement technique intended to choose target hubs. The convention considers both vitality efficiency and transmission separation, and hand-off hubs are utilized to mitigate the over the top power utilization of the bunch heads. The proposed convention brings about better dispersed sensors and a very much adjusted bunching framework upgrading the system's lifetime. We contrast the proposed convention and similar conventions by differing various parameters, e.g., the quantity of hubs, the system range measure, and the position of the base station. Recreation comes about demonstrate that the proposed convention performs well against other relative conventions in different situations.[15]With the expansion of use time, the working condition of hardware will be changed, in this way the straightforward component model is hard to meet the prerequisite of exactness. As per this exceptional trademark, a 2 arrange demonstrating of a specific sort marine condenser in view of component and recorded information is displayed in this paper. Right off the bat, the working instrument of the condenser is broke down, and afterward the component model is set up. Besides, the model parameters are enhanced by utilizing the recorded information and the proposed enhanced PSO-RBF calculation, consequently the RBF

neural system show, which can mirror the info and yield qualities of the condenser, is gotten. Through the test, it is demonstrated that the model is steady with genuine dynamic reaction of the real gear. Through coordinating the set up condenser show with the marine power execution recreation demonstrate library, the dynamic reaction of every parameter of the condenser when cooling water mass stream rate and steam turbine speed lessened 20% are considered. The examination result can give reference to the streamlining outline and control procedure of the marine condenser.

Proposed writing center around using the uses of Random woods and PSO to limit Make span and Flow time related with occupation portion. Rest of the paper is sorted out as under: area 2 portrays writing review, segment 3 depicts proposed framework, segment 4 gives execution examination and results, segment 5 gives conclusion and future degree, segment 6 gives references.

2. LITERATURE SURVEY

Applications and stages for the utilization of distributed computing for administration mechanisms have been proposed and created. This paper[16] addresses the utilization of distributed computing parallel scheduling. The suggestion made in [17] describe Energy efficiency in Distributed computing offers an insignificant exertion respond in due order regarding address this issue. In any case, if conveyed processing datacenters are arranged at distant districts from plant floors where robots work, fast wide-go frameworks (WAN) are required. Another approach proposed by [18] and [19] for confinement and employment allotment using gravitational inquiry calculation and molecule swarm headway (PSO) is addressed. The condition for the updates of speed and position of particles in PSO is balanced. For that, the estimations of two variables, cycle IT and α are evaluated through an as of late made Fuzzy Inference System. Strategies recommended through [20]Support Vector Machine (SVM), a methodology in light of [21]Particle swarm advancement (PSO) to streamline the SVM parameters was proposed and was used for the grouping and division. It deals with the element extraction and choice issues which are evaluated for interleaved orthogonal repeat divisions to get to uplink systems. Immediately propose by [22] algorithm which is faster in nature, [23]Particle Swarm Optimization (PSO) with focus symmetric trimmed connection lattice and different flag characterization standard is displayed with the end goal of productive estimation Going for the issue of heterogeneous multi-UAVs' agreeable errand task when finishing assault and harm assessment undertakings. Proposes [24] set forward a half breed algorithm PSO-GA, which is a mix of the attractive energy efficient PSO and Genetic algorithm. [32] In the range of distributed computing load adjusting, the Particle Swarm Optimization (PSO) algorithm is neoteric and now applauded very, however as of late a more neoteric algorithm which conveys the classifier into load adjusting is introduced. Plus, an algorithm called red-dark tree which is going for moving forward the proficiency of asset dispatching is additionally applauded. In any case, the 3 algorithms all have diverse detriments which can't be overlooked. For instance, the dispatch effectiveness of PSO algorithm is not fulfilling; despite the fact that classifier and red-black tree algorithm enhance the effectiveness of dispatching undertakings, the execution in load adjusting is not that great, therefore the enhanced PSO algorithm is displayed. Some inquires about are intended to get the upsides of new algorithm. Above all else, the time multifaceted nature and execution for every algorithm in principle are registered; and after that genuine information which are created in analysis are given to illustrate the execution. What's more, from the examination result, it can be found that for the speed of algorithm itself PSO is the most minimal, and the enhanced PSO take care of this issue in some degree; enhanced PSO algorithm has the best execution in undertaking settling furthermore, PSO is the second one, the red-dark and Naive Bayes algorithm are much slower; PSO and enhanced PSO algorithm perform well in load adjusting, while the other two algorithms don't do well.

.

3. PROPOSED SYSTEM

Proposed literature uses Random forest algorithm in order to form distinct schedules. These schedules are then fed in to allocation system. In case resources are not available in current machines then gravity search algorithm is applied to accomplish resource availability. Overall implication of RFPSO(Random Forest Particle Swarm Optimization) is minimal Makespan.

Description of algorithms hybridized is listed as under

1. Random Forest Algorithm

[25], [26]This algorithm is used in order to form schedule of the jobs presented. Schedules are formed by randomly placing jobs within the batch. Jobs are listed in terms of processes. The processes are divided into tasks for parallel allocation. The batches contained different scheduling orders presented to virtual machines within the cloud.

$$Task_i = Jobs_i/C$$

Equation 1: Task Division Criteria

One schedule is prepared, Tasks as batch is submitted for execution. In case jobs are executed then next job order is processed otherwise particle swarm optimization is followed.

2. PSO

The [27], [28]particle swarm streamlining is utilized to find assets inside the cloud. Assets can be available anyplace inside the cloud so as to find them, particles as introductory populaces are circulated. The PBEST and GBEST go about as ceasing criteria. PBEST if is best is supplanted with GBEST. The PSO stops as required assets are found.

4. METHODOLOGY

The related work on these scheduling techniques has shown that the use of Random forest has shown a low convergence rate to the true global minimum even at high numbers of dimensions. Particle swarm optimization algorithm has been widely accepted as a global optimization algorithm of current interest for distributed optimization and control. Particle swarm optimization is limited to initial set of particles, wrongly selected particles tends to poor results. In order to overcome these constrains a new hybrid Particle swarm optimization and Random forest algorithm for cloud

computing environment will be proposed to enhance the energy consumption rate further

Detailed progress of proposed literature is accomplished through following methodology.

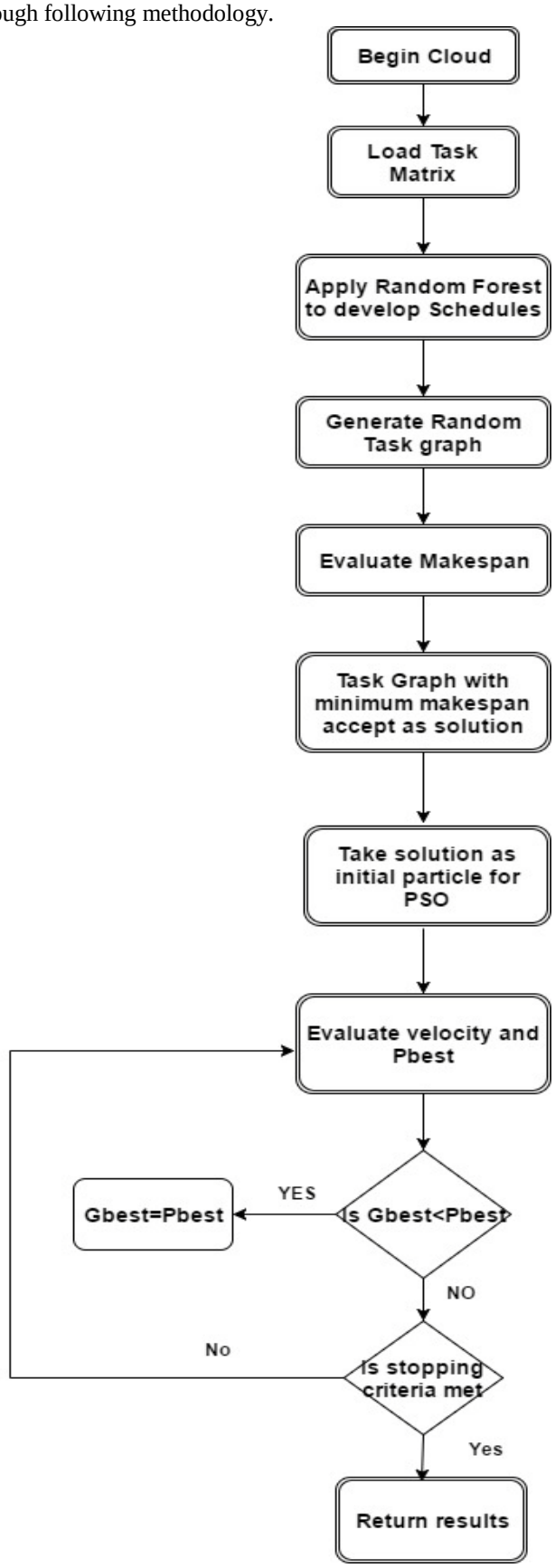

Figure 1. Proposed Methodology

A.) PARAMETERS CONSIDERED FOR EVALUATION

There are number of parameters considered for evaluation. Prime objectives considered for evaluation is Makespan, Flowtime and time consumption and load balancing degree Units

1) Makespan

[29]Makespan is the total time consumed in executing all the jobs present within the system. In other words Makespan is the finish time of last job executed.

$$Makespan = \sum finished_i / Total\_jobs \quad (1)$$

Equation 1: Makespan evaluation formula

In eq.1 $finished_i$ is the total finish time of the last job..

2) Flowtime

[30]It is the time taken by individual job to be executed on machine. Maximum Flowtime can be evaluated for determining Makespan of the jobs.

$$Flowtime_i = Finished_i \quad (2)$$

Equation 2: Individual Finish time of jobs indicating Flowtime

In eq. 2 $Finished_i$ is the finish time of individual jobs.

3) Time Consumption

Time consumption is total execution time required to execute the jobs presented to the system.

$$Time_{consumption} = Finish_i - Start_i \quad (3)$$

Equation 3: Time Consumption equation

In eq. 3 Time consumption of individual jobs is given when start time of individual jobs is subtracted from finish time of individual jobs

4) Load Balancing Degree

[31]Load balancing degree indicates amount of load on individual machines within cloud. Load balancing degree is evaluated as under

$$Load_{BD} = \sum \frac{Makespan_i}{VM_{Capacity}} \quad (4)$$

Equation 4: Load Balancing Degree evaluation

In eq 4. Makespan of whole schedule is used.

Note that in proposed methodology, VM capacity is considered to be 300. Lesser the degree more efficient the approach is.

5.PERFORMANCE ANALYSIS AND RESULTS

As contrasted and the current methodologies the outcome gotten with RFPSO is better. Makespan , Flowtime and time utilization is fundamentally decreased. This is expounded with the assistance of result segment. Similar examination is exhibited to demonstrate worth of the review.

1.Parameter Makespan

Table I. Makespan Evaluation of of Random forest, Improved PSO[32] and Random Forest PSO

Schedule	***Random Forest***	***Improved PSO***	***Random Forest PSO***
Schedule 1	2057	1985	1828
Schedule 2	2375	2105	1918
Schedule 3	2286	2002	1898
Schedule 4	2155	2014	1858
Schedule 5	2315	2145	1858

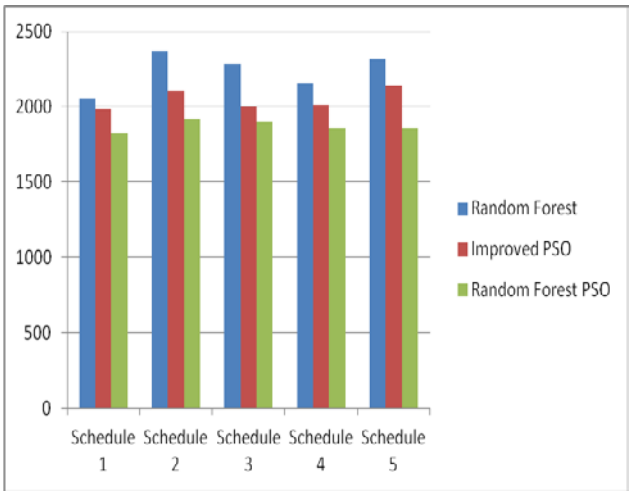

Figure 2. Plot of Makespan of existing and proposed technique Acknowledgment

Amongst the three algorithms, the makespan value for Random Forest PSO is the lowest i.e.1858 in schedule 5 or in other words. Minimum makespan is the most desirable requirement for efficient scheduling mechanisms.

2. Parameter Flowtime

Table II. : Flowtime comparison of Random forest, Improved PSO[32] and Random Forest PSO

Schedule	***Random Forest***	***Improved PSO***	***Random Forest PSO***
Schedule 1	411	271	187
Schedule 2	475	345	209
Schedule 3	457	391	206
Schedule 4	431	340	201
Schedule 5	463	355	207

From the above stated three algorithms, the value of flowtime is the lowest in case of Random Forest PSO i.e. the average flowtime for the schedules is compared. Accordingly Random Forest PSO is best among these algorithms as it possess minimum flowtime.

.

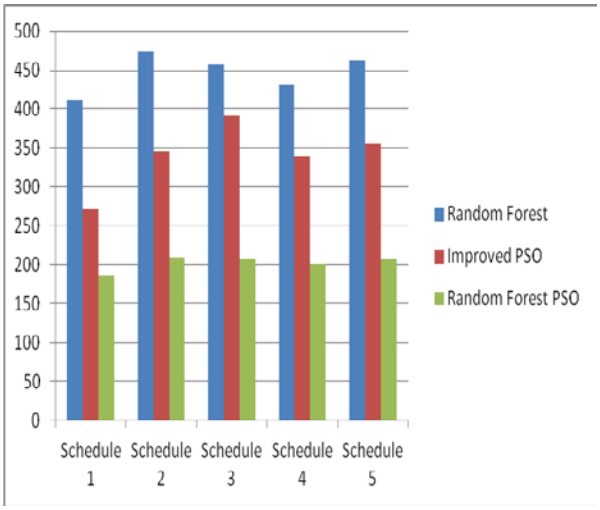

Figure 3. Plots of Flowtime

3. Parameter Time Consumption

Table III. Parameter Time Consumption

Schedule	***Random Forest***	***Improved PSO***	***Random Forest PSO***
Schedule 1-5	0.5	0.3	0.1

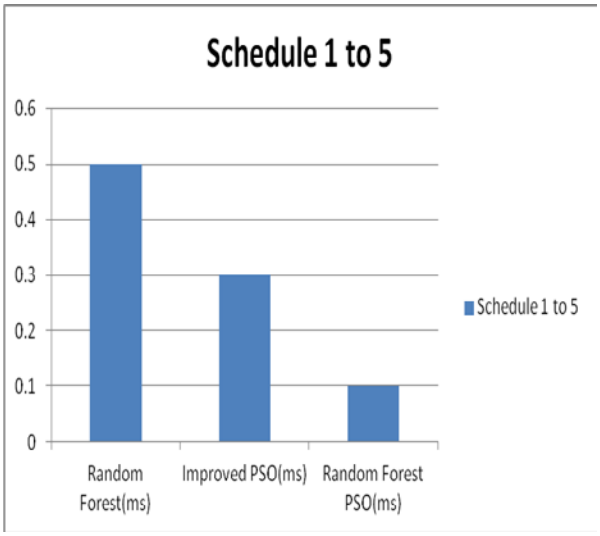

Figure 4. Time Consumption

Analysis The total time required for executing the schedules is lowest in case of Random Forest PSO. Flow time is required to be minimum for an efficient scheduling policy.

4. Parameter Load Balancing degree

Schedule	***Random Forest***	***Improved PSO***	***Random Forest PSO***
Schedule 1-5	7.66	7.11	6.36

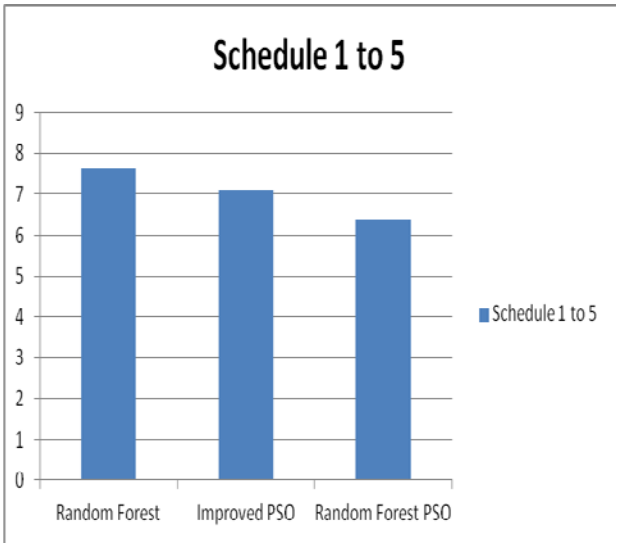

Figure 5. : Load balancing degree

Analysis: The fact that can be anticipated from the above graphs is that the load balancing degree is the lowest in case of Random Forest PSO algorithm i.e. 6.36 for schedules 1 to 5. This means that load is more likely to be equally balanced in this technique.

Overall result evolution indicates that proposed technique is better as compared to existing literature with PSO.

6. CONCLUSION AND FUTURE WORK

Resource allocation and effectively using them is key in cloud computing environment since cost associated with the resources is on the basis of pay per use. Scheduling is used for effective allocation. Proposed literature RFPSO provide effective mechanism for reducing Makespan, Flowtime, Time and load balancing degree. There exist a tradeoff between capacity and load balancing degree. In other words higher the load balancing degree lesser will be capacity of virtual machines. Results and performance analysis indicates proposed system perform better as compared to random forest algorithm.

In future Random Forest can be merged with Ant colony or BAT multiheuristic algorithm for better performance in terms of makespan.

7. REFERENCES

[1] [1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Futur. Gener. Comput. Syst., vol. 25, no. 6, pp. 599–616, 2009.

[2] [2] Y. Jiang, “A Survey of Task Allocation and Load Balancing in Distributed Systems,” IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 2, pp. 585–599, Feb. 2016.

[3] [3] W. S. M. Systems, D. Zhou, S. Member, J. Guo, S. Member, and Y. Zhang, “Distributed Data Analytics Platform for,” pp. 1–9, 2016.

[4] [4] W. Jung, “Using Distributed Fault Tolerant Memory in Virtualized Data Centers Woon Jung PernixData Agenda Background Motivation for DFTM Survey of challenging problems with DFTM.”

[5] [5] Y. I. Abramovich, O. Besson, S. Member, B. A. Johnson, and S. Member, “Conditional expected likelihood technique for compound Gaussian and Gaussian distributed noise mixtures,” no. c, pp. 1–12, 2016.

[6] [6] S. Moradi, “A distributed method based on mobile agent to detect Sybil attacks in wireless sensor networks,” pp. 276–280, 2016.

[7] [7] K. Hwang, X. Bai, Y. Shi, M. Li, W.-G. Chen, and Y. Wu, “Cloud Performance Modeling with Benchmark Evaluation of Elastic Scaling Strategies,” IEEE Trans. Parallel Distrib. Syst., vol. 27, no. 1, pp. 130–143, Jan. 2016.

[8] [8] S. J. Sheather, “Logistic Regression,” A Mod. Approach to Regres. with R, pp. 263–303, 2009.

[9] [9] G. Vaishali and V. Kalaivani, “Big Data Analysis for Heart Disease Detection System Using Map Reduce Technique,” vol. V.

[10] [10] V. Ramesh, P. Parkavi, and K. Ramar, “Predicting Student Performance A Statistical and Data Mining Approach,” Int. J. Comput. Appl., vol. 63, no. 8, pp. 35–39, 2013.

[11] [11] M. Wang and W. Zeng, “A comparison of four popular heuristics for task scheduling problem in computational grid,” 2010 6th Int. Conf. Wirel. Commun. Netw. Mob. Comput. WiCOM 2010, pp. 3–6, 2010.

[12] [12] D. T. Pham, A. Ghanbarzadeh, E. Koç, S. Otri, S. Rahim, and M. Zaidi, “The Bees Algorithm - A Novel Tool for Complex Optimisation Problems,” Intell. Prod. Mach. Syst. - 2nd I*PROMS Virtual Int. Conf. 3-14 July 2006, pp. 454–459, 2006.

[13] [13] H. Duan, S. Member, and X. Wang, “Pigeon-Inspired Optimization for Image Restoration,” pp. 1–13, 2015.

[14] [14] Y. Zhou, N. Wang, and W. Xiang, “Clustering Hierarchy Protocol in Wireless Sensor Networks Using an Improved PSO Algorithm,” IEEE Access, vol. 5, no. c, pp. 2241–2253, 2017.

[15] [15] H. Lin, C. Gang, and Z. Guoqing, “Modeling and Characteristic Analysis of a Marine Condenser Based on Improved PSO-RBF Algorithm,” pp. 125–129, 2017.

[16] [16] H. D. Karatza and R. C. Hilzer, “Parallel Job Scheduling in Homogeneous Distributed Systems,” Simul. Trans. Soc. Model. Simul., vol. 79, no. 5, pp. 287–298, 2003.

[17] [17] M. Azharuddin and P. K. Jana, “A distributed algorithm for energy efficient and fault tolerant routing in wireless sensor networks,” Wirel. Networks, vol. 21, no. 1, pp. 251–267, 2015.

[18] [18] B. Yuce, M. S. Packianather, E. Mastrocinque, D. T. Pham, and A. Lambiase, “Honey bees inspired optimization method: The bees algorithm,” Insects, vol. 4, no. 4, pp. 646–662, 2013.

[19] [19] R. Stoleru, T. He, and J. A. Stankovic, “Range-free localization,” Secur. Localization Time Synchronization Wirel. Sens. Ad Hoc Networks, pp. 3–31, 2007.

[20] [20] Z. Yan and C. Xu, “Combining KNN algorithm and other classifiers,” Proc. 9th IEEE Int. Conf. Cogn. Informatics, ICCI 2010, no. 1, pp. 800–805, 2010.

[21] [21] R. Singh, “Task Scheduling in Parallel Systems using Genetic Algorithm,” vol. 108, no. 16, pp. 34–40, 2014.

[22] [22] A. Hossein and G. X. Yang, “Bat algorithm for constrained optimization tasks,” pp. 1239–1255, 2013.

[23] [23] K. Bousselmi, “Energy efficient partitioning and scheduling approach for Scientific Workflows in the Cloud,” 2016.

[24] [24] P. A. Sonawane and L. Ragha, “Hybrid Genetic Algorithm and TABU Search Algorithm to Solve Class Time Table Scheduling Problem,” Int. J. Res. Stud. Comput. Sci. Eng., vol. 1, no. 4, pp. 19–26, 2014.

[25] [25] J. H. Abawajy, S. Member, and A. Kelarev, “Large Iterative Multitier Ensemble Classifiers for Security of Big Data,” vol. 2, no. 3, 2014.

[26] [26] I. Guyon, “An Introduction to Feature Extraction.”

[27] [27] A. P. Yow et al., “Automated in vivo 3D High-Definition Optical Coherence Tomography Skin Analysis System,” pp. 3895–3898, 2016.

[28] [28] F. Xhafa and A. Abraham, "A compendium of heuristic methods for scheduling in computational grids," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 5788 LNCS, pp. 751–758, 2009.

[29] [29] W. Abdulal, A. Jabas, S. Ramachandram, and O. Al Jadaan, "Task Scheduling in Grid Environment Using Simulated Annealing and Genetic Algorithm," 2012.

[30] [30] J. Kołodziej and F. Xhafa, "A game-theoretic and hybrid genetic meta-heuristics model for security-assured scheduling of independent jobs in computational grids," CISIS 2010 - 4th Int. Conf. Complex, Intell. Softw. Intensive Syst., pp. 93–100, 2010.

[31] [31] R. Kumar and T. Prashar, "Performance Analysis of Load Balancing Algorithms in Cloud Computing," Int. J. Comput. Appl., vol. 120, no. 7, 2015.

[32] [32] Di Zhao2, Wei Wang3, and Haiwu He Yongfei Zhu1, "A Novel Load Balancing Algorithm Based on Improved Particle Swarm Optimization in Cloud Computing Environment," © Springer International Publishing Switzerland, pp. 634-645, 2016.