DOI: http://dx.doi.org/10.26483/ijarcs.v8i7.4150

ISSN No. 0976-5697

Volume 8, No. 7, July – August 2017

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

INTEGRATED EXTREMAL OPTIMIZATION AND RANDOM FOREST BASED SCHEDULING FOR CLOUD COMPUTING ENVIRONMENT

Tanvi
Dept. Of Comp. Engg & Technology
Guru Nanak Dev University
Amritsar, India

Kiranbir Kaur
Dept. Of Comp.Engg & Technology
Guru Nanak Dev University
Amritsar, Indiaa

Abstract: Cloud computing is the most recent continuation of parallel computing, distributed computing and grid computing. Basically it is a parallel and distributed system containing various powerfully interconnected virtualized machines. Virtual machines are assigned over system to give administrations to users. The major problem nowadays occurs in cloud computing is Load Balancing. Basically it is a term used to spread jobs tasks over multiple networks. This paper defines the parallel processor scheduling technique in cloud computing environment. The proposed technique of Random Forest with integration of Extremal Optimization improves the efficiency and results of the tasks spreaded into multiple networks defined by the parameters like speedup, makespan time, migration cost etc. which indicates better outcomes of proposed technique than the existing one.

keywords: Scheduling, Random forest, Load balancing, distributed computing, Extremal Optimization, Cloud Computing, Processor scheduling.

I. INTRODUCTION

Nowadays, cloud computing is an emerging field in information technology, next generation of computing. It provides very extensive measure of computing and storage Service gave to users through the internet which follows pay-as-you-go model [1]. The emerging area in an IT environment is Cloud Computing. The term Cloud Computing has been termed with two parts in it i.e. one is cloud and other is computing [2]. With the improvement of parallel processing, circulated registering, framework figuring, another figuring model showed up called distributed computing. Focus on cloud computing is to provide a cost effective and user friendly platform for all resources to be utilized productively [2]. It becomes one of the exponentially growing technologies. It provides computing as utility to meet the needs of users. Cloud computing models broadly classified into four groups. Public, Private, Hybrid, Community. These services further broadly classified into three types: [3] [8] Platform as a Service, Software as a Service, and Infrastructure as a Service [8].

i. PaaS (Platform as a Service): It is given to user to show onto cloud framework buyer made or gotten applications made using programming languages. This model gives the client a stage which can be utilized to make, create, run and manage applications.
ii. SaaS (Software as a Service): It is given to client gave requests running on cloud framework. Requests are open from different user administrations. By utilizing this model, client can utilize an application online without the requirement for establishment.
iii. IaaS (Infrastructure as a Service): It is provided to consumers the arrangement of handling, storage networking and other critical figuring assets where user can quantify and run subjective programming which can incorporate working framework and programming. By utilizing this model client can lease infrastructure. [8].

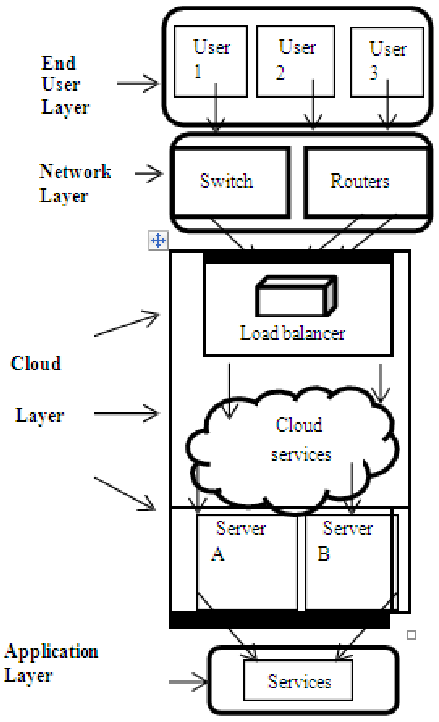


Figure 1: Load Balancing in Cloud Computing

In cloud computing environment, load balancing is an important aspect. The main objective behind load balancing is to allocate the whole load of work to entire cloud [1]. Load balancing algorithm helps the resource to have optimized utilization with proper destination of resources to the cloud user in pay-as-you-say manner [2]. It is a process of reallocating the total load of distributing system into individual nodes to ensure that no node is burdened and no nodes were loaded or idle. It is an optimization technique in which task scheduling is NP hard optimization problem.
Load balancing is used to divide the load of virtual machine over all nodes to enhance resources, service utilization and

give high fulfillment to clients. Because of load sharing each node can work effectively, information can be received and sent immediately [11] [22].

The circumstances of few servers extremely loaded rejects while servers that are thinly loaded as far as CPU loaded proportion in load balancing [24]. Let's take an example that shows load balancing in cloud computing. If we make one application on cloud and numerous users are required to get it at any one time. Response time to hundred persons will be moderate and hosts will end up plainly, active rapidly bringing about moderate reaction and unsuitable clients [9]. If we implement load balancing on our applications then task will be divided at different hubs and we can get huge improvement reaction and effective response [20].

Load balancing is a NP-complete issue technique as the extent of issue expands, the measure of the resolution extents too that implies more demands come to cloud, it gets hard to do adjusting among the several virtual machines [20] [21]. Genetic algorithm is much famous for resolving of NP- complete issues. This system has a place with the class of evolutionary algorithm that creates arrangements encouraged by natural solution [28]. By allocating the data centre using cloud sim software programme the performance of cloud computing is evaluated, having their purposes are: decrease total cost, recognize response time, classify data center hourly loading [6]. Distributed computing is a state in which a collecting of free and geographically dispersed system participate to take care of the complex issues each by clarifying the piece of organization and afterward interacting the outcomes from all computer systems [29].This paradigm allows before infeasible research. It also inspires public awareness of current scientific research. Access control is referred with supposing out in cloud computing environment that which user has which right for the organization or resources. This allots the whole load of the system to sub systems or servers. The aim of access control system is to keep the system resources against unauthorised or illegal access by users [37].

Access control is the field of dispersed applications, distributive supportive environments like cloud computing, where many clients access the resources and services are called distributed access control [4] [13]. Cloud computing exploits the information of virtual machines and collect material on virtual machines comparatively than physical machines. Load balancing is the main part of cloud computing environment [38]. It is a process of dividing the entire load amongst system, different node to make viable application of resource usage and to increase system execution.

The fundamental issue in cloud computing system is related to task scheduling where scheduler finds an optimal solution in cost effective manner [39]. The process of scheduling in cloud is defined in three stages:

Resource Discovering and Filtering: Data centres broker learns the incomes present in network system and collect position material related to them [31].

Resource Selection: Based on definite parameters of job and resource target resource is designated. This is significant stage [31].

Task Submission: Task is submitted to the resource that is selected [31].

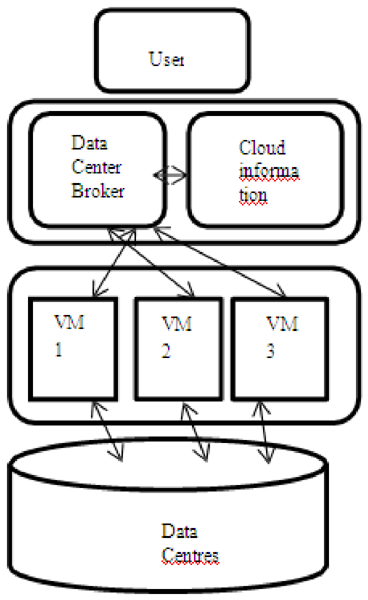


Figure 2: Scheduling in Cloud Computing

Task scheduling issue is mainly focus on to find the best or Optimal resource in order to minimize the total dispensation time of virtual machines [26]. Cloud task scheduling is a NP-hard optimization problem which is best resolvable using meta-heuristic algorithm [30]. Meta-heuristics are the strategies which are basically at high level that control on essential more problems to definite heuristic to raise their performance [31][27]. Different meta-heuristic methods useful like genetic algorithm (GA), ant colony optimization (ACO), simulated annealing (SA), particleswarm optimization (PSO), and many more to solve task scheduling issue [21]. In task scheduling, allocating a specific task at particular period of instance to definite resources are to be done. A task is only a little bit of work that is to be influenced within exact time frame [44]. Cloud information service manager gives the knowledge about the position of resources to the cloud task scheduler that is available to complete the allocated task. The major goal of task scheduling is to improve the resource utilization and waiting time of the task [6].

II. TECHNIQUES USED

Cloud Optimization is quick moving from a grand to need as cloud administrations come to build a growingly huge part of associations' general IT infrastructure [10]. Optimization plays a critical part in defining techniques for financial specialists and it is characteristically a discrete Multiobjective optimization issue whose choice criteria dispute with each other.

III. EXTREMAL OPTIMIZATION

Extremal Optimization is an stimulating nature- inspired optimization technique. It was suggested by Boettcher and Percuss influenced the Bak-Seneppen self- organized active critically [5].

It is well said that variety of everyday work optimization troubles can be expressed as controlled optimization problems[7].

The difficulties of supposing out the forced optimization issues get from different cut-off points on decision factors, the basics of included constraints, disturbance among restrictions and the interrelationship between the unbiased functions and desires [7].

Problems like multiplicity of combinatorial optimization extremal optimization have been effectively used [25][33]. Uses of EO to mandatory optimization issues are mostly rare. EO algorithm with adaptable control handling methods called EO-ACD for constraint optimization issues [15]. Basic idea behind EO-ACD is combining of actual coded EO and flexible dealing techniques of limits [23] [36]. It is even not a simple task to attain an executable result. As an outcome, various algorithms with constraints handling techniques have been proposed to achieve better execution. This optimization called Extremal Optimization (EO) so the constrained optimization has been resolved [36]. EO built load balancing algorithm offered in which determination of better constrained area of arrangement has being enhanced was completely arbitrary [27]. Load balancing with guided EO which has been discovered gives much of the time preferred outcomes over EO and hereditary calculation in view of completely irregular choice, EO for processor load balancing throughout execution of conveyed projects [18]. This approach is utilized to once in a while identify the best assignment as contender for relocated undertakings [34]. To diminish the complexity of determination for movement guided EO calculations accept two stage stochastic choices on arrangement flawlessness in light of two different appropriateness functions. [37]. The capacities depend on particular program models which figure the relations amongst projects and official equipment. Work shop is one of an outstanding NP-hard improvement issues. Extremal improvement is a transformative meta-heuristic strategy that more than once substitutes undesirable factors in current arrangement with an arbitrary esteem and advances itself toward ideal arrangement [38] [35].

A. RANDOM FOREST:

Random forest or random decision forests arejoint learning strategies for course of action, inversion, and various undertakings so that the work by making large count of trees chosen at the time of preparing and the class being afforded is the method of the classes or mean gauge of the discrete trees[14]. The general technique for irregular choice backwoods was first proposed by HO in 1995 who establishes that forestry of trees part. Irregular woodland has turned out to be basic place in numerous PC vision applications [49]. Their quality for the most part determined by their high computational adequacy amid both preparing and evaluation while as yet having the capacity to accomplish condition of craftsmanship exactness [40]. Arbitrary timberland is an order framework expects to help the demonstration of arrangement strategies. It depends on procedure of building various classifiers and afterward on the whole utilizes them all to distinguish unlabelled occurrences [41].

It is a machine learning strategy, based planned packing and irregular component choice numbers of choice trees are created [46]. Two broadly utilized methodologies could be recognized; to be specific boosting and bagging [2].

Boosting: It is an additive process of building a arrangement of classifiers, where work of each classifier on inaccurately categorized occurrences of the previous one in the sequence [43].

Bagging: It is the method where each classifier collectively built using randomly data drawn where each classifier gives an equal vote while making unmarked occurrences[43].

The more stronger than boosting against over fitting model is bagging. The main illustrative of bagging is Random Forest. Size of random forest is specific and differs from one data set to another. Due to randomization encouraged during creation, and its huge size random forest has at best been described as black-box [51]. Random forest fits into the family of classifier joint that use randomization to produce a diverse pool of individual classifiers [51]. Random forest has developed as one of the maximum normally used classification method. Several characteristics that make it ideal are:

It can be utilized when there is numerous bigger numbers of factors than illuminations

It has great prescient execution notwithstanding when most prescient factors are more

It doesn't over fit

It doesn't deal with mix of unmistakable and consistent indicators [51].

Random forest algorithm uses the concept bagging with different randomization technique called feature selection[42]. The two key of optimization problem must have in order for active programming to be applicable are:

Optimal Substructure: Optimal arrangement shown for problems, if problem of finest solution contains optimum solution to sub problems [42].

Overlapping sub-problems: When we get similar issues again and again with recursive algorithm then it is said that enhancement issue has covering sub-issues.

Random Forest method is existing as machine learning technique, which won't bring about over-fitting issue and parameters are not difficult to be tuned [42]. Grey Relational projection (GRP) is presented to select same ancient material to make random forest models. In random forest, number of trees formed having each tree made uses randomly drawn amounts from the informational collection [50]. Randomization is likewise linked while selecting the best node to split on for every one of the trees [43]. It is combination of tree analysts in a way that each tree is dependent upon the costs of a random vector tested discretely with the exact same distribution of all the trees in the forest [45].

The simplification error for forests joins i.e. as far as possible, for example, the quantity of trees in the forests turns out to be vast [46]. The simplification error of woods of tree classifiers relies on upon the quality of the individual trees in the forest and relationship between them [45]. Random Forest is a proficient information mining algorithm which can be utilized for system disruption acknowledgement [48] [47]. Network interference discovery system is one of the vital mechanisms in any network security infrastructure [48]. Machine learning classifiers such as random forest have showed unusual execution on several practical issues and are being developed for mapping and indicating natural resources and ranger service to a considerable degree.

I. RELATED WORK:

Neeta Patil et al. [2017] [32]evaluated an extremely effective processing condition is given by cloud computing where the clients or a few renters need various assets to be given as an administration over the internet. The use of assets is to be planned effectively with the goal that it helps in diminishing the ideal opportunity for task to be completed. Task scheduling is most basic and vital part in distributed computing condition. In task scheduling allotment of specific assignments to specific assets at a specific time case is finished. There are numerous methods which can be proposed to look after the difficulties of task scheduling. This characterized various resource scheduling algorithms in cloud computing environment.

Akash Dave et al. [2016] [17] suggested very extensive measure of computing and storage Service gave to users through the internet which follows pay-as-you-go model. Most of the issues noticed in cloud are resource uncovering fault tolerance, load balancing and security. Load balancing is one of the main challenges, important technique, and critical issue and plays an important role which is required to divide the whole load of work or task equally across the nodes or servers. It provides a detailed summary of the load balancing optimization techniques of evolutionary and swarm based algorithms which will help to overcome the optimization problems or resource utilization.

Jianguo Chen et al. [2017] [16] analysed a Parallel Random Forest (PRF) algorithm for vast information on the Apache Spark stage. The PRF algorithm is upgraded in view of a hybrid approach consolidating information parallel and task parallel optimization. From the point of view of data parallel optimization, a vertical information dividing strategy is performed to decrease the information correspondence cost viably, and an information multiplexing technique is performed to permit the preparation dataset to be reused and reduce the volume of data. From the point of view of task parallel optimization, a double parallel approach is completed in the preparation procedure of RF, and an errand Directed Acyclic Graph (DAG) is made by the parallel preparing procedure of PRF and the reliance of the Resilient Distributed Datasets (RDD) objects. At that point, diverse task schedulers are conjured for the undertakings in the DAG. In addition, to enhance the calculation's exactness for substantial, high-dimensional, and loud information, we play out a measurement diminishment approach in the preparation procedure and a weighted voting approach in the forecast procedure preceding parallelization. Broad test comes about demonstrate the prevalence and remarkable points of interest of the PRF calculation over the important calculations executed by Spark lib and different reviews as far as the grouping exactness, execution, and adaptability.

Hussain A Makasarwala et al. [2016][26] defined a genetic algorithm (GA) basedapproach for load balancing in cloud. For populace introduction, need of demand is viewed as in view of their time. The thought behind the considering the need is to get true perception. In Real World Scenario asks for have a few needs can utilize for our Algorithm. Recreation of the proposed technique is finished utilizing Cloud Analyst. Re-enactment Results demonstrates that proposed technique performs well at that point existing once with some real world picture.

Sidra Aslam et al. [2015] [19] detailed the load on the cloud is expanding hugely with the improvement of new applications. Load balancing is a critical piece of cloud computing condition which guarantees that all gadgets or processors perform same measure of work in equivalent measure of time. Distinctive models and algorithms for load balancing in cloud computing has been created with the intend to make cloud assets available to the end clients effortlessly and accommodation. It is organized and extensive structure of the exploration on load balancing algorithms in cloud computing. It reviews the best in class load balancing instruments and methods over the time of 2004-2015. We aggregate existing methodologies gone for giving burden adjusting in a reasonable way. we give a simple and brief perspective of the basic model embraced by each approach.

Xiaoyu WU et al. [2015] [42] formulateda new two-stage hybrid algorithm planned to take care of these two issues is proposed. Random Forest (RF) strategy is presented as the machine learning technique, which won't bring about over-fitting issue and parameters are anything but difficult to be tuned. Moreover, Gary Relational Projection (GRP) is acquainted with select comparative authentic information to prepare irregular woodland models. The last estimating outcomes in view of genuine load information demonstrate this new two-organize strategy performs superior to the next two normal strategies.

Ivanoe De Falco et al. [2014] [37]studied techniques for utilizing Extremal Optimization (EO) for processor load balancing throughout execution of appropriated tasks. A load balancing calculation for groups of multicore processors is introduced and talked about. In this algorithm the EO approach is utilized to intermittently identify the best jobs as possibility for movement and for a guided choice of the best processors to get the moved assignments. To diminish the unpredictability of choice for relocation, we propose a guided EO calculation which accept a two-stage stochastic determination amid the arrangement change in light of two separate wellness capacities. The capacities depend on particular program models which appraise relations between the projects and the official equipment. The proposed stack adjusting calculation is evaluated by analyses with mimicked stack adjusting of disseminated program diagrams. The algorithm is contrasted against an EO based algorithm and irregular arrangement of relocated undertakings and an exemplary hereditary algorithm.

Xu Wang et al. [2011] [3]populated that with the advancement of parallel processing, appropriated registering, matrix figuring, another registering model showed up, called distributed computing. It plans to share information, counts, and administrations straightforwardly among clients of a monstrous framework. It turned into a hot issue for its focal points, for example, "decrease costs", "increment business adaptability" as well as "give business progression". It depicted what is distributed computing and took Google's distributed computing methods for instance, summed up key procedures, for example, information stockpiling technology (Google File System), information administration innovation (Big Table), and in addition programming model and undertaking planning model (Map□□Reduce), utilized as a part of distributed computing, and afterward some case of distributed computing sellers were represented and looked at.

Qingyang Meng et al. [2009] [12]formulated that in parallel cluster registering, an un-adaptable load adjusting calculation can strongly influence the execution of figuring. To go for this case, advances a straight powerful load adjusting model and examinations the steadiness of this direct model on the state of existing time delay. Base on dissecting comes about, this paper utilizes a heap adjusting addition to control this model with the expanding framework scale. At last, a more helpful nonlinear model is proposed and the re-enactment results are given to contrast and examining comes about and other load adjusting techniques.

II. GAPS:

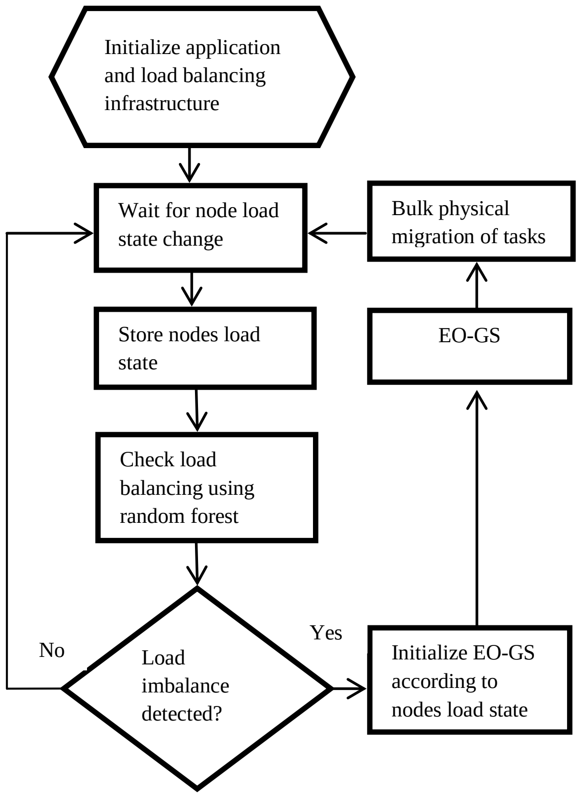
i. Variables do not clearly define general definition of fitness that may prove ambiguous.
ii. Slows down significantly by the fitness re-evaluating process in the case of highly connected systems.
iii. The parallel extremal optimization based load balancing has not considered the tunning parameters.
iv. The multi-objective tunning is also ignored in the existing literature.
v. The uses of machine learning strategies for efficient load balancing have also been ignored.

III. PROBLEM DEFINITION

Extremal Optimization is implemented in processor load balancing in accomplishment of distributed programming. It is a nature-inspired meta-heuristic technique. This algorithm finds an optimized strategy of task migration. This helps in minimizing the program execution time. Experimental results supports applications of parallel EO for load balancing successfully an evolutionary based on working of top solution in optimization process by improvement of choice of poorest mechanisms. EO may reduce count of job migration needed to equalize processors loads. EO simply based on betterment of only resolution representation with low memory and operational complexities focused with requirements of load balancing problems. The improved algorithm of extremal optimization is formed that is guided-state extremal optimization is produced to get the improved results of the solutions of worst components. This approach evaluates the several versions of EO based on parallelism. This dissertation will propose machine learning based EO technique for efficient load balancing. The proposed technique will utilize random forest algorithm to construct optimal schedulers. Also multi-objective criteria will be used to enhance the result further.

IV. PROPOSED METHODOLGY:

A. This section contains the graphical representation of proposed technique, consisting of various steps which are required to successfully accomplish the suggested algorithm.


Initialize application and load balancing infrastructure
Wait for node load state change
Bulk physical migration of tasks
Store nodes load state
EO-GS
Check load balancing using random forest
No
Yes
Load imbalance detected?
Initialize EO-GS according to nodes load state

B. The proposed technique is designed using MATLAB R2010a.

Following are the steps which explain the extremal optimization algorithm with random forest in the form of pseudo code and defined in step by step:

Algorithm: Extremal Optimization with Random Forest (EORF)

```
Initialize configuration Y at will
Ybest ← Y
While total number of iterations Niter not reached do
Evaluate ¥i for each variable yi of the current solution
Rank the variables yi based on their local fitness ¥i
Choose the rank z according to z
                    − α so that the variable ym with
 m = Ω (m) is selected
Evaluate βy for each neighbour Yv
∈ Neighbor(y, yj), generated by changing yj in the
current solution Y
Rank neighbours Yv
∈ Neigh(Y, yj) based on the random forest trees( ¥i)
Choose Y' ∈ Neighbor(Y, yj)according to the
exponential distribution
Accept Y ← Y' unconditionally
If ¥(Y) < ¥ (Ybest) then
Ybest ← Y
End if
End while
Return Ybest and ¥ (Ybest)
```

• In definite sequential EO algorithms iterative updates of single solution Y formed with number of elements $y_i$that are variables of problems [52].

- Value of $¥_i$ that is local fitness function of ¥ measures the worst component of $¥_v$ in the solution.
- At every iteration Y is modified by arbitrarily updating the worst element. Solution Y' that is global fitness function formed in place of Y because Y is improved at every repetition step by arbitrarily informing poorest adaptable.
- Y' replaces Y if global fitness function is better than that of Y.
- By using the probabilistic function α-EO version, local optimum EO avoids, where α is user defined parameter.
- Minimum problem solved by α-EO solution elements assigned ranks Z, 1<=Z=>n, where n is number of elements consistently increases with their local fitness value.
- Worst component Y_i is of rank 1 while finest one is of rank n.
- Element assortment possibility $p_z$over rank Z is defined as proportional to $Z^{-\alpha}$ for value of parameter α.
- At every iteration component rank z nominated in modern result Y affording to Y_i using roulette wheel.
- It is supposed that load balancing algorithms enthusiastically control assignments of program tasks T_z , z€ {1……[T]} to processors n, n€ {0,1….[N]-1}, where T is set of tasks and N are computing nodes.
- For the evaluation of load of the system two indicators are used
- First one is that job refunding calculating power of nodes take place at different nodes
- Function standardised in range of[0,1]
- Homogeneous communication is a link of whole value of total exterior announcement capacity and total communication volume of all communication.
- in heterogeneous decision-making systems analogous measures of the communication time are used:

$$\text{attrexttotal}(Y) = \frac{\text{totalext}(Y)}{\text{CT}}$$
$$\text{where CT} = \sum v\, y, d \in \text{Tcom}(y, d) \text{and totalext}(y) = \sum y, d \in \text{T}: \mu y = !\ \mu d \text{ com}(y, d)$$

- function migration(Y) is a migration costs metrics , value of this function is in the range [0, 1], i.e., it is equal to 0 when there is no migration, when all tasks have to be migrated , migration(Y) = 1, otherwise 0 ≤ migration(Y) ≤ 1 [3]:

$$\text{migration}(S) = |\{t \in T : \mu Yt! = \mu Y * t\}|/|T|$$

Where Y is the currently considered solution and Y* is the previous solution

- function random forest tress(Y) represents the numerical computational load random forest tress metrics in the solution Y, equal to 1 when in Y there exists at least one unpacked calculating node, else equal to the standardized average absolute computational load deviation of tasks in Y, determined in the definition below:

$$\text{random forest tress}(Y) = \begin{cases} 1 \quad \text{exists at least} & \text{one unloaded node} \\ \dfrac{\text{deviation}(Y)}{2 * |N| * \text{WT}} & \text{otherwise} \end{cases}$$

- global fitness function ¥(Y) is defined as follows:

$$¥(S) = \text{attrexttotal}(Y) * §1 + \text{migration}(Y) * §2 + \text{random forest tress}(Y) * [1 - (\&§1 + \&§2)]$$
$$\text{where } 1 > §1 \geq 0, 1 > §2 \geq 0 \text{ and } §1 + §2 < 1$$

- local fitness function of the component t is computed assuming that the task t is assigned to the processor μt, ¥parameter (0 < ¥ < 1) allows modification of the weight of the computational load metrics in contradiction of the communication metrics anxious with a component t (task t assigned to processor _t)

$$¥(t) = ¥ * \text{load}(\mu t) + (1 - ¥) * \text{rank}(t)$$

V. RESULTS AND DISCUSSIONS:

i. PERFORMANCE ANALYSIS:

Performance analysis comprises the efficient interpretations to improve the performance and expand the decision making, mostly provided through the establishment of objective statistically. We have used the initially loaded workload that contains 100 jobs that need to be balanced. Each job contains its own processor. More is the speed of processors, lesser is the execution time of the jobs. Number of parameters of the problem to be optimized is defined as by D is 100. Total 5 parameters used for the evaluation of the problem and to derive the better results and efficiency. Destination number that is the number of optimization sources equal to half of the colony size is defined by NP/2. Maximum numbers of iterations are maxcycle is 100.The results are taken after the loop of 100 iterations, out of which, maximum execution time is marked as makespan of the first schedule. In the same way, all the parameters are calculated.

ii. RESULTS:

This section defines the graph methodology with the help of parameters like execution time, make span time, mean turnaround time, speedup, migration cost to describe the effective results of scheduling in cloud computing environment. In the tables given below, computed 15 values for each parameter and compared in three ways that is denoted as without optimization, extremal optimization(denoted as existing technique), extremal with random forest(denoted as proposed technique). The graphical results for all the metrices are shown in figures given below:

EXECUTION TIME: The execution time for a job is characterized as how much time a system takes to execute a given assignment which incorporates the time taken, executing run time or system benefits for its sake. The mechanism or formula used to calculate execution time for a particular task is defined as follows:

Execution Time = Stop Time – Start Time

While comparing the readings in Table 1, we can say that proposed method has been proved better than Without Optimization and Extremal Optimization. Figure 3 shows that the proposed algorithm optimizes the execution time in an efficient way, in comparison to both the techniques.

Table 1: Execution Time Values

Execution Time	Without optimization	Existing technique	Proposed technique
1.	15.1840	15.0259	14.8238
2.	3.7853	3.6336	3.4647
3.	3.7617	3.6197	3.4454
4.	3.7505	3.549	3.4198
5.	3.7830	3.6374	3.4654
6.	3.7183	3.5774	3.4072
7.	3.7370	3.5955	3.4229
8.	3.7408	3.5980	3.4114
9.	3.7324	3.5941	3.4180
10.	3.7289	3.5904	3.4162
11.	3.7350	3.5945	3.4258
12.	3.7252	3.5850	3.4105
13.	3.7097	3.5759	3.4072
14.	3.8279	3.6787	3.4827
15.	3.7208	3.5807	3.4111

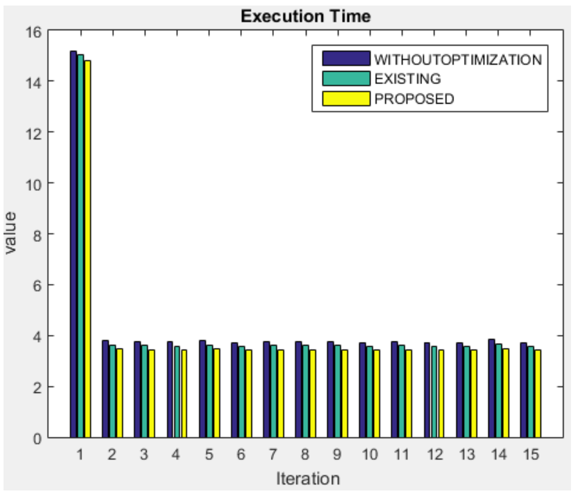


Figure 1: Execution Time Comparison

MEAN TURNAROUND TIME: Turnaround time is characterized as the aggregate time between the job submission and the yield given to the client. It might shift for different programming languages relying upon the designer of the product or the program.

$\sum_{i=1}^{N}(Execution\ time + waiting\ time)/N$ where N is no.of jobs.

Table 2 shows the results of comparison in the terms of mean turnaround values. Figure 4 describes that the computed the values of turnaround time while using the proposed method is lesser than the values evaluated using existing techniques.

Table 2: Mean Turnaround Time:

Turnaround Time	Without optimization	Existing technique	Proposed technique
1.	6.0800	5.6863	3.9983
2.	6.0800	6.1184	3.9748
3.	6.0800	4.7098	3.7309
4.	6.0800	5.1941	3.5737
5.	6.0800	5.2813	4.0110
6.	6.0800	4.7354	3.7529
7.	6.0800	4.9628	4.1118
8.	6.0800	5.1180	3.7974
9.	6.0800	4.6445	4.0516
10.	6.0800	6.1809	4.4230
11.	6.0800	4.6613	3.8132
12.	6.0800	4.7965	4.0472
13.	6.0800	5.2615	3.8721
14.	6.0800	3.6787	3.8645
15.	6.0800	5.1761	3.777

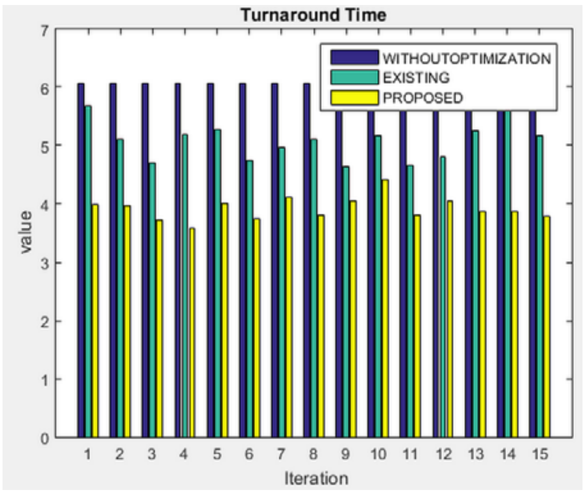


Figure 2: Turnaround Time Comparison

MAKESPAN TIME: The makespan of a undertaking is the aggregate time that omissions by from the earliest starting point to the end. The term usually shows up with regards to planning. There is a confusing scheme that is made out of a few sub-undertakings

$Makespan = Max(SL_i)$, where i=1, 2….P and P represents the total number of processors.

Table 3 shows the results of comparison in the terms of makespan values. Figure 5 describes that the computed makespan while using the proposed method is lesser than the values evaluated using existing techniques.

Table 3: Makespan Time

Makespan	Without optimization	Existing technique	Proposed technique
1.	43	8	5
2.	42	8	5

3.	44	6	5
4.	44	7	5
5.	44	7	5
6.	44	6	5
7.	44	7	5
8.	43	7	5
9.	43	6	5
10.	44	8	6
11.	44	6	5
12.	44	6	5
13.	44	7	5
14.	44	7	5
15.	44	7	5

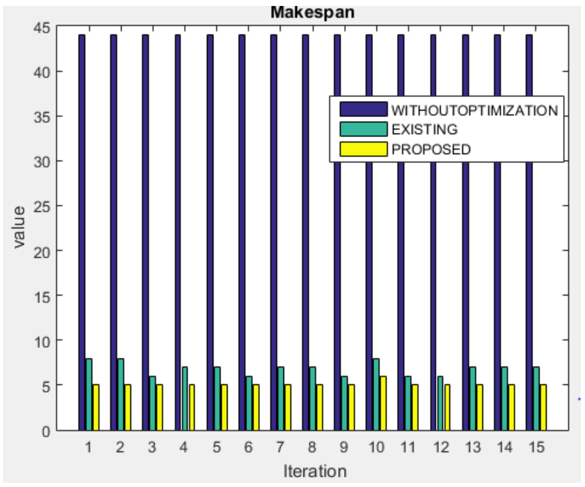


Figure 3: Makespan Comparison

SPEED UP: Speedup is defined as a process that increases the performance between two systems handling the same problem. It is the development in speed of execution of a task executed on two parallel architectures with different resources.

The experimental results of existing and proposed techniques are listed in Table 4. The graph in Figure 6 is showing comparison of speedup between the traditional algorithm and proposed algorithm. As it can be seen that the proposed method is more efficient than other methods

Table 4: Speedup Values

Speed Up	Without optimization	Existing technique	Proposed technique
1.	0.5212	0.5256	0.5308
2.	1.2844	1.3099	1.3395
3.	1.2883	1.3123	1.3430
4.	1.2902	1.3166	1.3476
5.	1.2848	1.3093	1.3394
6.	1.2955	1.3196	1.3500
7.	1.2924	1.3165	1.3471
8.	1.2918	1.3161	1.3492
9.	1.2932	1.3167	1.3480
10.	1.2938	1.3174	1.3483
11.	1.2928	1.3167	1.3466
12.	1.2944	1.3183	1.3493
13.	1.2970	1.3199	1.3500
14.	1.2774	1.3022	1.3363
15.	1.2951	1.3191	1.3492

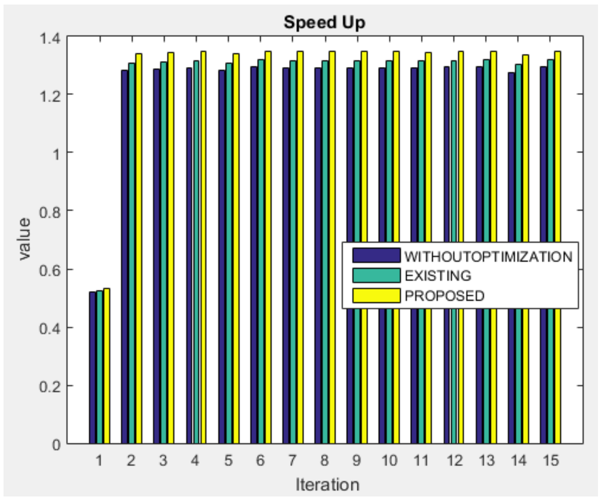


Figure 4: Speedup Values Comparison

MIGRATION COST: Migration cost is defined as the ratio of migration to be done while calculating the values of the existing and the proposed technique to number of processors allocating to each job.

$\sum_{j=1}^{NP} Total\ migration\ /NP$; where NP is number of processors

Table 5 shows the results of migration cost as obtained by implementing existing and proposed techniques. The Figure 7 depicts better performance of suggested method than the existing algorithms in terms of migration cost.

Table 5: Migration Cost:

Migration Cost	Without optimization	Existing technique	Proposed technique
1.	26	19	16
2.	27	15	10
3.	21	11	10
4.	26	19	13
5.	26	18	19
6.	25	17	15
7.	24	20	18
8.	23	12	17
9.	24	20	13
10.	28	19	19
11.	26	12	14
12.	22	13	13
13.	27	20	17
14.	25	20	15
15.	27	20	13

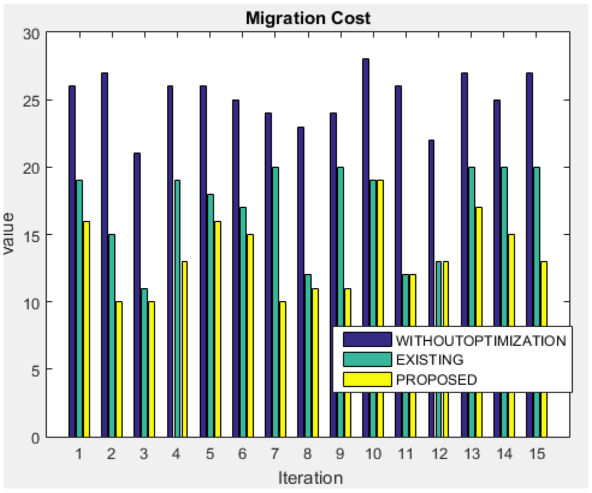


Figure 5: Migration Cost Comparison

VI.CONCLUSION:

Much attention has been paid to parallel processor scheduling synthesis and optimization used Extremal Optimization and Random Forest algorithms. In general, Extremal Optimization (EO) is able to provide good solutions, but with large computational efforts. In this novel study, the integration technique for parallel cloud scheduling using mutation and crossover operators based on EO and Random Forest is introduced. The experimental results demonstrate that the suggested method outperforms the available techniques with respect to different quality metrics. The further enhancement of this study involves the evaluation of algorithm by Random Forest, its actual scientific workload in a real-time cloud computing environment. In this work we have neglected the effect of failures. The better and efficient results are produced with optimized techniques used in cloud computing environment.

VII.REFERENCES

[1]. Dave, Akash, Bhargesh Patel, and Gopi Bhatt. "Load balancing in cloud computing using optimization techniques: A study." Communication and Electronics Systems (ICCES), International Conference on. IEEE, 2016.

[2].Thingom, Chintureena, Ganesh Kumar, and GuydeukYeon. "An analysis of load balancing algorithms in the cloud environment." Communication and Electronics Systems (ICCES), International Conference on. IEEE, 2016.

[3]. Wang, Xu, Beizhan Wang, and Jing Huang. "Cloud computing and its key techniques." Computer Science and Automation Engineering (CSAE), 2011 IEEE International Conference on. Vol. 2. IEEE, 2011.

[4]. Ribeiro, Anderson Souza, and David Bianchini. "The deployment of Systems in Cloud Computing environment: A Methodology to Select and Prioritize projects." IEEE Latin America Transactions 15.3 (2017): 557-562.

[5]. Alkayal, Entisar S., Nicholas R. Jennings, and Maysoon F. Abulkhair. "Efficient Task Scheduling Multi-Objective Particle Swarm Optimization in Cloud Computing." Local Computer Networks Workshops (LCN Workshops), 2016 IEEE 41st Conference on. IEEE, 2016.

[6]. Alnazir, MnahilKherAlseed Mohammed, et al. "Performance analysis of Cloud Computing for distributed data center using cloud-sim." Communication, Control, Computing and Electronics Engineering (ICCCCEE), 2017 International Conference on. IEEE, 2017.

[7]. Grandison, Tyrone, et al. "Towards a formal definition of a computing cloud." Services (services-1), 2010 6th World Congress on. IEEE, 2010.

[8]. Harikrishna, Pillutla, and A. Amuthan. "A survey of testing as a service in cloud computing." Computer Communication and Informatics (ICCCI), 2016 International Conference on. IEEE, 2016.

[9]. Toland, Tyrone S. "C2Cloud: A cloud computing framework." Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual. IEEE, 2017.

[10]. Zarra, Taoufiq, et al. "Cloud computing and sentiment analysis in E-learning systems." Cloud Computing Technologies and Applications (CloudTech), 2016 2nd International Conference on. IEEE, 2016.

[11]. Li, Ying, Kai Ma, and Jiong Zhang. "An Efficient Multicore Based Parallel Computing Approach for TSP Problems." Semantics, Knowledge and Grids (SKG), 2013 Ninth International Conference on. IEEE, 2013.

[12]. Meng, Qingyang, et al. "Research on the Stability of Load Balancing Algorithm for Scalable Parallel Computing." Communication Software and Networks, 2009. ICCSN'09. International Conference on. IEEE, 2009.

[13]. Wang, Weina, and Lei Ying. "Resource allocation for data-parallel computing in networks with data locality." Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on. IEEE, 2016.

[14]. Yang, Kang, Xiao Song, and Xiang Li. "Parallel Web server load balancing technology of cloud computing environments." Guidance, Navigation and Control Conference (CGNCC), 2014 IEEE Chinese. IEEE, 2014.

[15]. Zamanifar, Kamran, NaserNematbakhsh, and Razieh Sadat Sadjady. "A new load balancing algorithm in parallel computing." Communication Software and Networks, 2010. ICCSN'10. Second International Conference on. IEEE, 2010.

[16]. Chen, Jianguo, et al. "A parallel random forest algorithm for big data in a Spark cloud computing environment." IEEE Transactions on Parallel and Distributed Systems 28.4 (2017): 919-933.

[17]. Dave, Akash, Bhargesh Patel, and Gopi Bhatt. "Load balancing in cloud computing using optimization techniques: A study." Communication and Electronics Systems (ICCES), International Conference on. IEEE, 2016.

[18]. Kang, Lu, and Xing Ting. "Application of adaptive load balancing algorithm based on minimum traffic in cloud computing architecture." Logistics, Informatics and Service Sciences (LISS), 2015 International Conference on. IEEE, 2015.

[19]. Aslam, Sidra, and Munam Ali Shah. "Load balancing algorithms in cloud computing: A survey of modern techniques." Software Engineering Conference (NSEC), 2015 National. IEEE, 2015.

[20]. Ding, Zhixiong, Xingjun Wang, and Wenming Yang. "A Dynamic Load Balancing Algorithm in Heterogeneous Network." Intelligent Systems, Modelling and Simulation (ISMS), 2016 7th International Conference on. IEEE, 2016.

[21]. Ding, Zhixiong, Xingjun Wang, and Wenming Yang. "A Dynamic Load Balancing Algorithm in Heterogeneous Network." Intelligent Systems, Modelling and Simulation (ISMS), 2016 7th International Conference on. IEEE, 2016.

[22]. Hammoudi, Sarra, et al. "Load balancing in the cloud using specialization." Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), IEEE Annual. IEEE, 2016.

[23]. Ragmani, Awatif, et al. "A performed load balancing

algorithm for public Cloud computing using ant colony optimization." Cloud Computing Technologies and Applications (CloudTech), 2016 2nd International Conference on. IEEE, 2016.

[24]. Sharma, Agraj, and Sateesh K. Peddoju. "Response time based load balancing in cloud computing." Control, Instrumentation, Communication and Computational Technologies (ICCICCT), 2014 International Conference on. IEEE, 2014.

[25]. Kaur, Rajwinder, and NavtejGhumman. "Hybrid Improved Max Min Ant Algorithm for Load Balancing in Cloud." International Conference on Communication, Computing & Systems (ICCCS–2014).

[26]. Makasarwala, Hussain A., and PrasunHazari. "Using genetic algorithm for load balancing in cloud computing." Electronics, Computers and Artificial Intelligence (ECAI), 2016 8th International Conference on. IEEE, 2016.

[27]. Pilavare, Mayur S., and Amish Desai. "A novel approach towards improving performance of load balancing using Genetic Algorithm in cloud computing." Innovations in Information, Embedded and Communication Systems (ICIIECS), 2015 International Conference on. IE, 2015.

[28]. Arani, AtefehHajijamali, et al. "Distributed Load Balancing User Association and Self-Organizing Resource Allocation in HetNets." Vehicular Technology Conference (VTC-Fall), 2016 IEEE 84th. IEEE, 2016.

[29]. Dharmapala, Prashan, et al. "Peer-to-peer distributed computing framework." Technology and Management (NCTM), National Conference on. IEEE, 2017.

[30]. Ibrahim, Elhossiny, Nirmeen A. El-Bahnasawy, and Fatma A. Omara. "Task Scheduling Algorithm in Cloud Computing Environment Based on Cloud Pricing Models." Computer Applications & Research (WSCAR), 2016 World Symposium on. IEEE, 2016.

[31]. Mandal, Tripti, and SriyankarAcharyya. "Optimal task scheduling in cloud computing environment: Meta heuristic approaches." Electrical Information and Communication Technology (EICT), 2015 2nd International Conference on. IEEE, 2015.

[32]. Patil, Neeta, and Deepak Aeloor. "A review-different scheduling algorithms in cloud computing environment." Intelligent Systems and Control (ISCO), 2017 11th International Conference on. IEEE, 2017.

[33]. Kai, Sun, Genke Yang, and Changchun Pan. "Solving hot rolling scheduling problem by a new population-based extremal optimization algorithm." Bio-Inspired Computing: Theories and Applications (BIC-TA), 2010 IEEE Fifth International Conference on. IEEE, 2010.

[34]. Azadehgan, Vahid, et al. "A New Hybrid Algorithm for Multiobjective Optimization." Tools with Artificial Intelligence (ICTAI), 2011 23rd IEEE International Conference on. IEEE, 2011.

[35]. Chen, Min-Rong, JianWeng, and Xia Li. "Multiobjective extremal optimization for portfolio optimization problem." Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on. Vol. 1. IEEE, 2009.

[36]. Chen, Jie, et al. "Extremal optimization algorithm with adaptive constants dealing techniques for constrained optimization problems." Industrial Electronics and Applications (ICIEA), 2014 IEEE 9th Conference on. IEEE, 2014.

[37]. De Falco, Ivanoe, et al. "Extremal optimization with guided state changes in load balancing of distributed programs." Parallel, Distributed and Network-Based Processing (PDP), 2014 22nd Euromicro International Conference on. IEEE, 2014.

[38]. Gharehjanloo, Masoud, et al. "Extremal optimization for solving job shop scheduling problem." Computer and Knowledge Engineering (ICCKE), 2011 1st International eConference on. IEEE, 2011.

[39]. Nakada, Akihiro, Keiichi Tamura, and Hajime Kitakami. "Optimal protein structure alignment using modified extremal optimization." Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on. IEEE, 2012.

[40]. Qi, Jie, and Liping He. "Extremal optimization for a dye vat scheduling problem." Natural Computation (ICNC), 2011 Seventh International Conference on. Vol. 4. IEEE, 2011.

[41]. Tamura, Keiichi, et al. "A new distributed modified extremal optimization for optimizing protein structure alignment." Computational Intelligence and Applications (IWCIA), 2015 IEEE 8th International Workshop on. IEEE, 2015.

[42]. Wu, Xiaoyu, et al. "A two-stage random forest method for short-term load forecasting." PowerTech, 2015 IEEE Eindhoven. IEEE, 2015.

[43]. Bader-El-Den, Mohamed. "Self-adaptive heterogeneous random forest." Computer Systems and Applications (AICCSA), 2014 IEEE/ACS 11th International Conference on. IEEE, 2014.

[44]. Benedict, Shajulin, et al. "Energy prediction of OpenMP applications using random forest modeling approach." Parallel and Distributed Processing Symposium Workshop (IPDPSW), 2015 IEEE International. IEEE, 2015.

[45]. Breiman, Leo. "Random forests." Machine learning 45.1 (2001): 5-32.

[46]. Leistner, Christian, et al. "Semi-supervised random forests." Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009.

[47]. Ma, Kai, and Jezekiel Ben-Arie. "Compound exemplar based object detection by incremental random forest." Pattern Recognition (ICPR), 2014 22nd International Conference on. IEEE, 2014.

[48]. Prashanth, G., et al. "Using random forests for network-based anomaly detection at active routers." Signal Processing, Communications and Networking, 2008. ICSCN'08. International Conference on. IEEE, 2008.

[49]. Shahana, K., SubhajitGhosh, and C. Jeganathan. "A survey of particle swarm optimization and random forest based land cover classification." Computing, Communication and Automation (ICCCA), 2016 International Conference on. IEEE, 2016.

[50]. Uriarte, Rafael Brundo, SotiriosTsaftaris, and Francesco Tiezzi. "Service clustering for autonomic clouds using random forest." Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on. IEEE, 2015.

[51]. Kulkarni, Vrushali Y., Aashu Singh, and Pradeep K. Sinha. "An Approach towards Optimizing Random Forest using Dynamic Programming Algorithm." International Journal of Computer Applications 75.16 (2013)

[52]. De Falco, Ivanoe, et al. "Parallel extremal optimization in processor load balancing for distributed applications." Applied Soft Computing 46 (2016): 187-203