Movement Record-Oriented Particle Swarm Optimization for Resource Allocation and Solving Job Failure Problems in Cloud.

K Shruthi
Department of Computer Science and Engineering
The Oxford College of Engineering Bengaluru, Karnataka, India
shruthireddy1551@gmail.com

R Ch A Naidu
Department of Computer Science and Engineering
The Oxford College of Engineering
Bengaluru, Karnataka, India
chanr789@gmail.com

Raghu Ramamoorthy
Department of Computer Science and Engineering
The Oxford College of Engineering
Bengaluru, Karnataka, India
raghuace85@gmail.com

ABSTRACT

The purpose of using Cloudlet is remotely reducing the cloud communication delay and the system load. Still, since the prevailing system performance of the remote cloud is not in the cloudlet, as the no. of users changes, the cloud provides resources for each user change. Moreover, as the cloudlet's service security is low, the cloudlet's service for users is eliminated from coverage when work activation. In this circumstance, users will not get the computational results, so cloud computing fails. This work offers a real-time resource allocation framework for users to allocate sufficient and necessary resources. This work proposes the Movement Record-Oriented Particle Swarm Optimization (MROPSO) method to unravel the issue of work failure and real-time resource allocation. The method proposed by this paper provides a better powerful result than the real PSO method.

Keywords: Mobile Cloud Computing (MCC), MROPSO, Assets allocation, Job Offloading.

A. INTRODUCTION

Mobile devices comprise vehicles, smartphones, and laptops. As the quantity of mobile entities continues to rise, the need for services such as speech recognition software, online games and face recognition software is increasing [1 and 2]. To solve problems such as resource limitation and mobility [3], research on MCC is very important. For a cloudlet of resources known as computational resources, via wireless access, by moving jobs, can improve the computing energy of mobile devices. To address the resource boundary of mobile entities, cloudlet is a great idea. Cloudlet's research advances nowadays. Cloudlet is, they are computers connected to a rich and reliable Internet. It is used by mobile entities via high speed WLAN. To diminish the power consumed by handheld devices, MCC is used. Though, due to various meets such as job calculation failure, long operational delay, and finite network bandwidth, cloud computing affects the actual usage.

In proposed system, to address the above problems, a real-time resource distribution structure is proposed. The system created for this work is, resources can be allocated based on information on resource loads and real-time jobs. And you can plan the job offloading system more efficiently and faster. This includes information such as the cloudlet and the user's place, the user's predictable time and the amount of jobs. In paper [4], that author have used the Decision making algorithm to resolve the operational challenges in Mobile Cloud Coupling. In paper [5], that analyzer discuss how to optimize MCC performance, for that, they have presented a code offloading framework. The purpose of paper [6] is to find the right mapping between cloud resources and mounted job. Heterogeneous and homogeneous configurations of virtual machines have been used. In paper [7], concerned about the resource savings of mobile devices, they have used the MCC technique. In paper [8], they have used controller in cloud layer to achieve high response time and low latency challenges. The objective of this work [9] is to obtain the location of users for security purposes. Longitude and Latitude information is collected using the longitude and latitude algorithm.

B. PROPOSED METHODOLOGY

In this segmnet, the introduced resource allocation scheme is introduced through two assumptions. One is its function and the other is its structure.

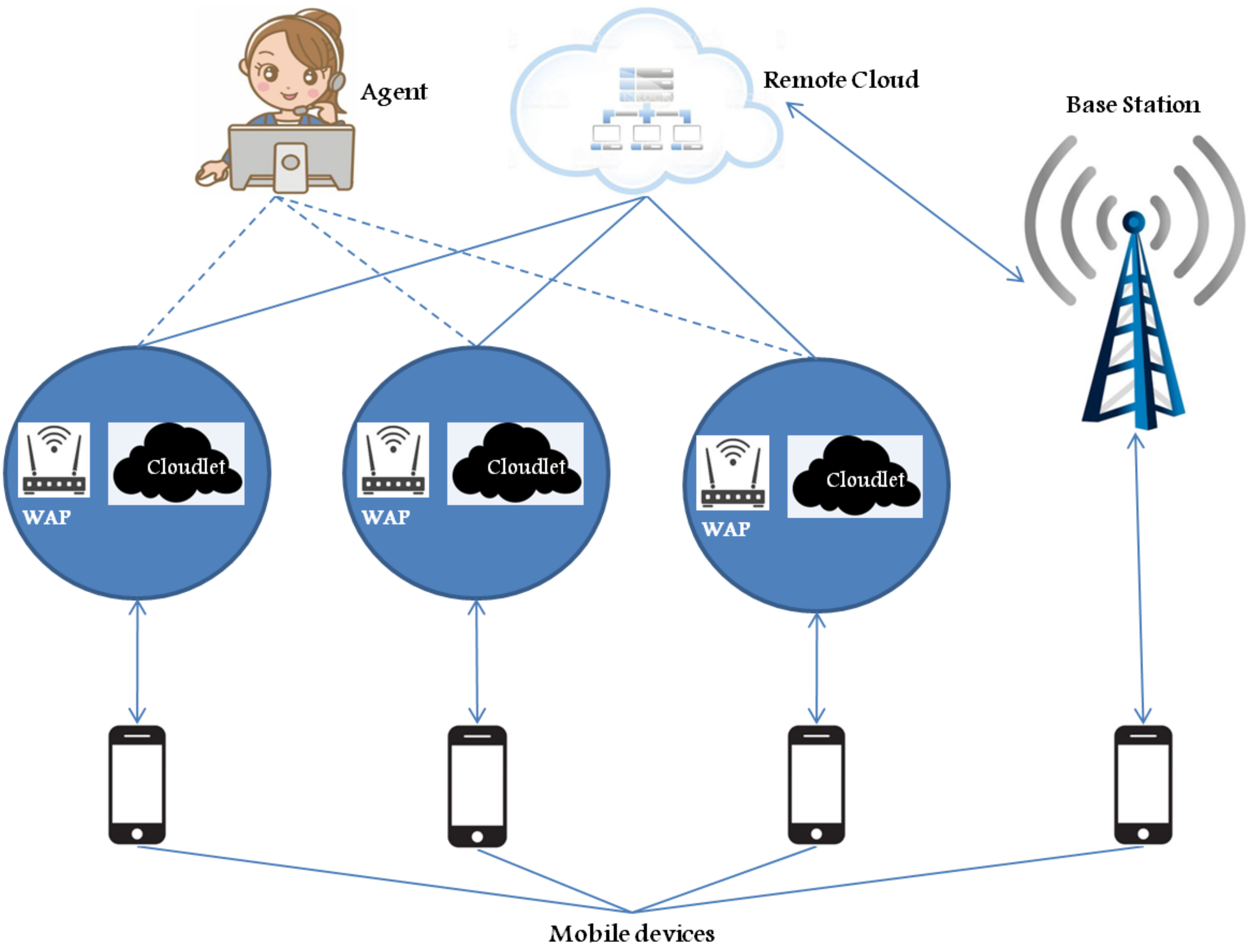

Fig.1: Resource Allocation Structure

In figure 1, there are cloudlets, mobile devices, agencies and remote cloud. When mobile devices travel from one location to another, they bring some jobs to the remote cloud or to the cloud in less running time, and save energy consumption. Mobile devices be able to join the cloudlet over the WAP (wireless access point) distributed through the cloudlets. With a remote cloud, the cloudlets can attach to a high-speed wire network. Mobile devices be able to access remote cloud utilizing the base station. These two source access sockets are more important. Because while a mobile device is running fast, in the Cloudlet's service coverage area, operating results do not result.

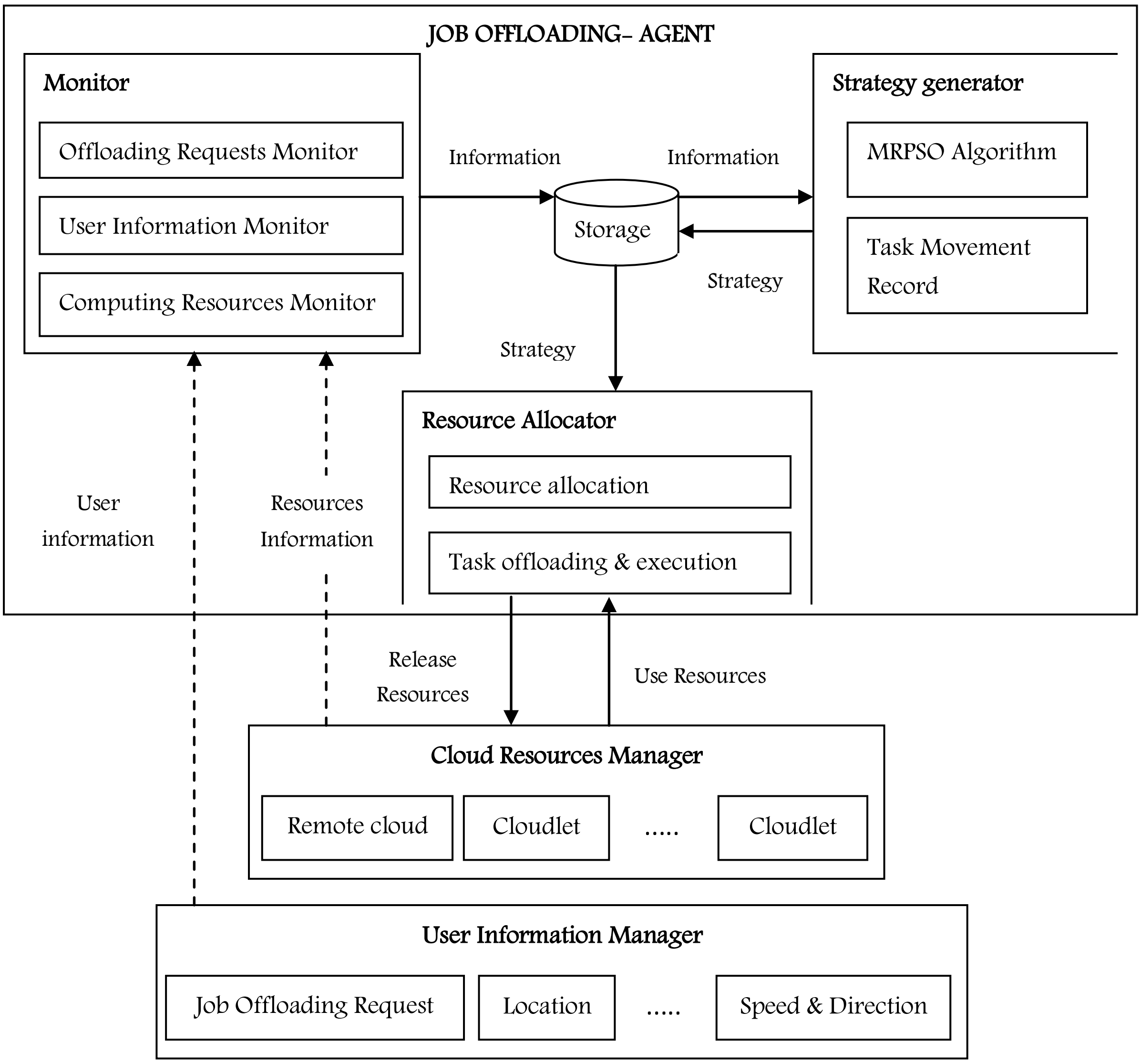

Fig.2: Proposed Scheme Architecture

In this state, mobile device be able to offload with remote cloud, and function results can be easily obtained. In the introduced framework, the portion of the agent is very important. This is because it monitors mobile device information's, cloud resources, and work offloading requests. When users assign jobs to cloudlets, they have to send work offloading needs to nearby cloudlets. The agent then receives all demands from the cloud. Depending on the position of the customer, the speed of the customer, the size of the works, the direction of the consumers and the load of the cloud assets, to get the work allotment system, the agent practices the resource allocation system. This segment presents the proposed system. The proposed scheme consists of three volumes, which can be shown in Figure 2. They are, respectively, the Cloud Resource Manager (CRM), the User Information Manager (UIM), and the Job Offloading Agent (TOA) [15]. The CRM receives remote cloud computing resources, cloudlets and bandwidth resources. The work offloading agent has four volumes. They are resource allocation, data storage, monitor and strategic generator. Each module is introduced individually.

1. Monitor

This fragment is installed on the address and the status of assets, the position of mobile apparatus and the request for job allotment can be tracked in real time. Resource sustainability brings the accessible cloud resources in real time. When the monitor passes enough work offloading demands, it takes about five seconds, and sends the collected data to the monitor.

2. Data Storage

This component is a database that is temporarily given to stored information. Due to the time distinguish between user cloud information and the group of work offloading requests, the agent will continue to collect information and store them in storage. When retrieving data per block, it call data from the storage. After using this information, it will be released from storage.

3. Strategy Generator (SG)

This area is important in the proposed system. In SG, includes work Movement Record and PSO. The latter a unique PSO is by moving from one cloudlet to another, helps detect short running time location. When selecting the right work and performing the next job, to move to the most appropriate cloudlet the time of work activation and the position of the work movement are recorded. Then, based on the data of resources and information of job, SG analyzes the jobs offloading strategies. Cash allocation strategies are moved to storage for use in the resource allocation area. Information about the algorithm is provided in the next division.

4. Resource Allocator

Cloudlet resources are assigning to job and obtaining the cloudlet resources from remote cloud and cloudlets, these element functions are based on offloading techniques. After stored the computing resources, jobs can be accomplished instantaneously. To use other jobs after completing jobs, resources are free immediately. The first portion of this segment is devoted to the problem clarity of resource allocation in MCC. Then an improved formalized MROPSO method is discussed.

a. Problem Definition

Due to the dynamic nature of resources and the movement of users, it is hard to know precisely the status of future resources [10,11,12]. The shared user's composite jobs essential to be converted into sub-jobs. Meanwhile there is correlation between sub-jobs, have to wait until some jobs are completed. Then they will be accomplished. If there is a large of time consumed in the preceding stage, when the Resources Plan was prepared at the time of the work's execution, it is not certain that the resources utilized are the same.

So based on presently recognized resources the work offloading program can be developed. Functions to utilize planned resources, Resource allocation should be done in a tiny period of time [13, 14,15].

b. PSO Algorithm

The multi-goal resource allocation issue is, because a NP-Complete Problem, to unravel this issue in a polynomial time, PSO is used in this paper. PSO is a calculation method. It depends on a specific data; to increase a candidate solution trying will find a solution. However, to discover a best solution in shorter time to the unique PSO system, this paper was created. The best solution is, with very little operating time is the work allotment scheme. Next, the PSO algorithm is introduced. The definition of each of the parameters is given below.

1. The j^{th} particle's location: $a_j = (a_{j1}, a_{j2}, \ldots, a_{jX})$.
2. The j^{th} particle's speed: $s_j = (s_{j1}, s_{j2}, \ldots, s_{jX})$.
3. The best location of j^{th} particle: $b_j = (b_{j1}, b_{j2}, \ldots, b_{jX})$.
4. The best position of all type of particles: $b_i = (b_{i1}, b_{i2}, \ldots, b_{iX})$.

5. The *k1* and *k2* are represent the rate of learning from the particle's own optimal solution and acceleration constants and the best result of the all particles, respectively. The η and ξ are random values that vary between zero and one.

The pace and status of the particles are engaged in a continuing and actual space. The equation of speed change and the particle swarm state are as follows:

$$s_{jX}^{C+1} = s_{jX}^{C} + k_1\xi(b_{jX}^{C} - a_{jX}^{C}) + k_2\eta(b_{iX}^{C} - a_{jX}^{C}) \quad (1)$$

$$a_{jX}^{C+1} = a_{jX}^{C} + s_{jX}^{C+1} \quad (2)$$

c. MROPSO Algorithm

When the unique problem solving, such as resource allocation, in original PSO mode, there is no exact meaning as to the direction of the particle and the speed of moving. So the original PSO method takes a lot of time to find a perfect solution. To solve this issue, this effort suggests the PSO method for work operating logging.

To record job movement information, used a two-dimensional array called *TD[C] [T]*. Where, C is the no. of cloudlets. 'T' is the no. of jobs. The horizontal axis of the two-dimensional line signifies the job ID, and the vertical axis denotes the cloudlet ID. Thus each cloudlet will overlap with every job.

Let td_{tCT}^{i}be the particle's t^{th}job movement record at i^{th}iteration.

$$td_{tCT}^{i} = \begin{bmatrix} td_{t11}^{i} & \cdots & td_{t1T}^{i} \\ \vdots & \ddots & \vdots \\ td_{tC1}^{i} & \cdots & td_{tCT}^{i} \end{bmatrix} \quad (3)$$

Here, 'T' is the job number and C is the cloudlet number.

$$td_{tCT}^{i} = \frac{b_e t_T}{n_e t_{CT}^{i}} \quad (4)$$

Let $n_e t_{CT}^{i}$be the fresh result time of T^{th} job in C^{th} cloudlet at i^{th} iteration. Let $b_e t_T$= $\{b_e t_1,\ldots,b_e t_2,\ldots,b_e t_T\}$ be the top result time of job. The value is computed by considering that each job is loaded separately for the best powerful and efficient cloud. Thus, the activating time of each work is lower.

Value rescued by *TD[i] [j]* is the main running time of the work. In Cloudlet *'j'*, separated by an *'i'* ongoing time. If *TD[i] [j]* is too big, the working time of the work is restricted. If the value of a work is too little for all jobs, it means that the job has to move much higher. The target location of the job is, this is the place with the greatest value in all the cloud related work. If the value is not the 1, but a great value, specifies that the job may take a short time to load to the cloud. When all jobs are executed in legitimate clouds, all jobs will have reached very little execution time. Therefore, the resource allocation strategy is appropriate.

d. User's Mobility Problem

There is concern about users' mobility in MROPSO. Since cloudlets have a definite level of service security, every user has time to stay in the Cloudlet service security area. If the user's carrying time is less than the ongoing time of the job, functional results of the job are not obtained. This will result in failures in loading jobs. Therefore, after computing the operating time of each job, the MROPSO mechanism differentiates the out coming time of the user with the Cloudlet's service coverage. If the activation time is less than the stay time, the cloudlet can be designated. If the activation time exceeds the dependent time, the job will be activated for the remote cloud. Even if users leave the cloudlet coverage service, the execution results can be obtained using the base station.

e. Pseudo Code

Let $Cur_e^i = \left[Cur_{e0}^i, \ldots, Cur_{et}^i, \ldots, Cur_{eT}^i\right]$ be the particles e^{th} execution location ID of job t at i^{th}iteration.

Let $b_j^i = [O_{l1}^i, \ldots, O_{li}^i, \ldots, O_{lT}^i]andO_i^i = [O_{g1}^i, \ldots, O_{gt}^i, \ldots, O_{gT}^i]$ be the local & global optimum position.

Let fitness () $=\sum_0^T et_t$. Where, et_t= the execution time of job t, I_{mx}= the maximum no. of iterations. T_{par}= swarm size. Table 1 shows the algorithm of MROPSO.

Table 1. Pseudo code of MROPSO algorithm

```
Initialize Cur^1 randomly, set td^1 with 1;
Calculate the b_e t_T;
For all types particles in the swarm, td^i_tCT(e), Cur^i_t (e), e∈ [1, T_par]
        Compute the fitness price of particle e;
        Let O^1_l = Cur^1;
End for;
Copy O^1_g to be the one with the optimal fitness in Cur^1 (k), 1 ≤ k≤ T_par;
t = 1;
While t ≤ I_mx;
For all types particles in the swarm, td^i_tCT(e),Cur^i_t (e), e∈ [1, T_par]
        Based on Cur^i_t (e), discover the job t of the lowest value in the td^i_tCT(e);
        Based on td^i_tCT(e), discover the cloudlet c with the largest value of job t;
        Alter the execution position Cur^i_t (e) into cloudlet c;
        Calculate the fitness price of particle e;
        Update the td^i_tCT(e) with the fitness cost;
                If fitness (Cur^i_t (e)) is better than fitness (O^1_l)
                        O^1_l = Cur^i_t (e);

                End if
End for
        Set O^1_g to be the one with the best fitness in Cur^i_t (e), 1 ≤ k≤ T_par;
        i + +;
End while
```

C. RESULTS AND DISCUSSION

In this portion, the effectiveness of the projected resource allocation mechanism is evaluated. First, the simulation scheme is presented, and then the outcomes of the simulation are given.

1. Simulation Scheme

In the simulation Schemes, this paper simulates some users with different locations, remote cloud resources and cloudlet resources jobs. Many of these jobs have different computation sizes and data sizes. In order to simulate this as realistic as possible, in this work, the no. of works is 500, 1000 and the no. of cloudlets is 10, 50 quantified. This paper, PSO, MROPSO, COD and Greedy algorithms is executed fifty times correspondingly. The population of PSO and MROPSO is ten. In each situation, the iteration times of the two algorithms are both hundred. Algorithms running time is compare by with function time of jobs offloading, so the proposed method of this paper has proven to be superior.

2. Simulation Results

Depending on the outcomes of the job forecasting time, when in the figure 3 to 6 shows that the *available* cloudlet resources are small and the number of works is large. It can be seen that both PSO and MROPSO methods have a better work offloading system than other methods. This means that the total time of jobs is very limited. When there are additional cloudlet resources, although the COD algorithm is better in one location than the MROPSO method and but their results are not consistent. Therefore, the results of the job offloading mechanism of the PSO and MROPSO methods are consistent. The job prediction execution based on time is discusses in section A. The figures from 3 to 6 depicts the same. The proposed MROPSO outperforms PSO, Cod and Greedy in efficient manner. The job predictions consider varies from 10000 to 70000 with the implementations of 1 to 5. Jobs varies from 500 to 1000 with the cloud sets of 10. For various jobs the proposed MROPSO performance is superior in all aspects. Using PSO and considering mobility problem for consideration improves the performance of MROPSO for all job sets. The PSO algorithm search ability will allocates resources to services in optimal manner without complexity. The resource allocation with overhead will produce optimal solutions. Optimally, the PSO makes use of fitness function for evaluation of cloud sets for job allocation. The highest fitness function contains resource with optimal connectivity and degree, which enables jobs to find resources.

A. Job Prediction Execution Time (sec)

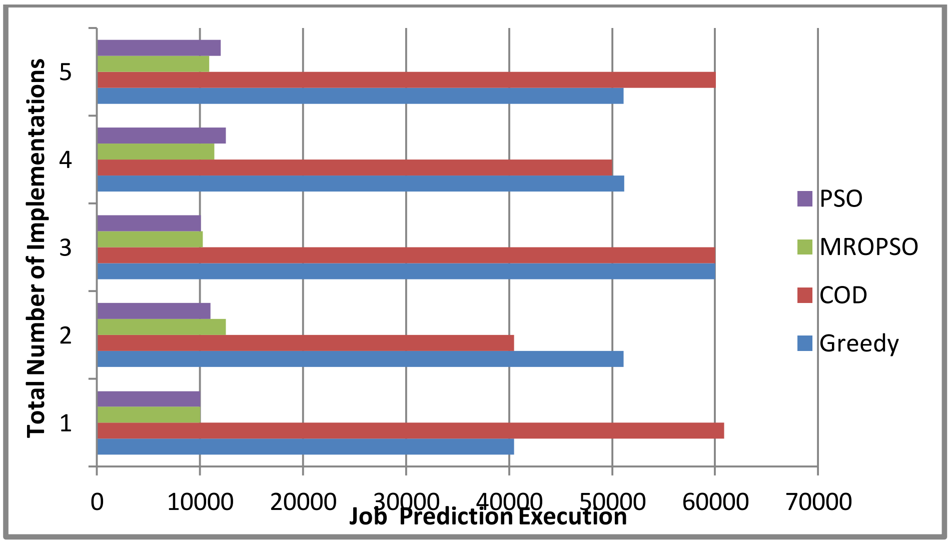

Fig.3: Job number = 500, Cloudlet Number = 10

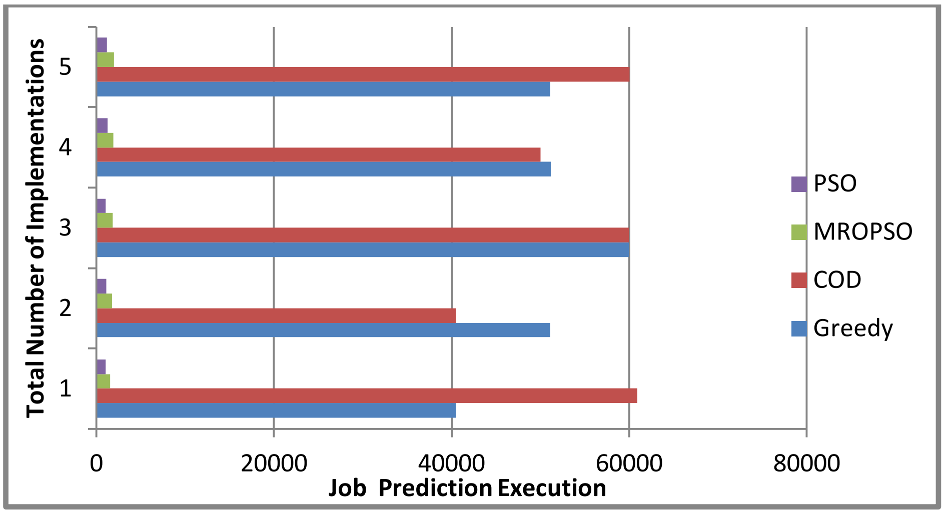

Fig.4: Job number = 1000, Cloudlet Number = 10

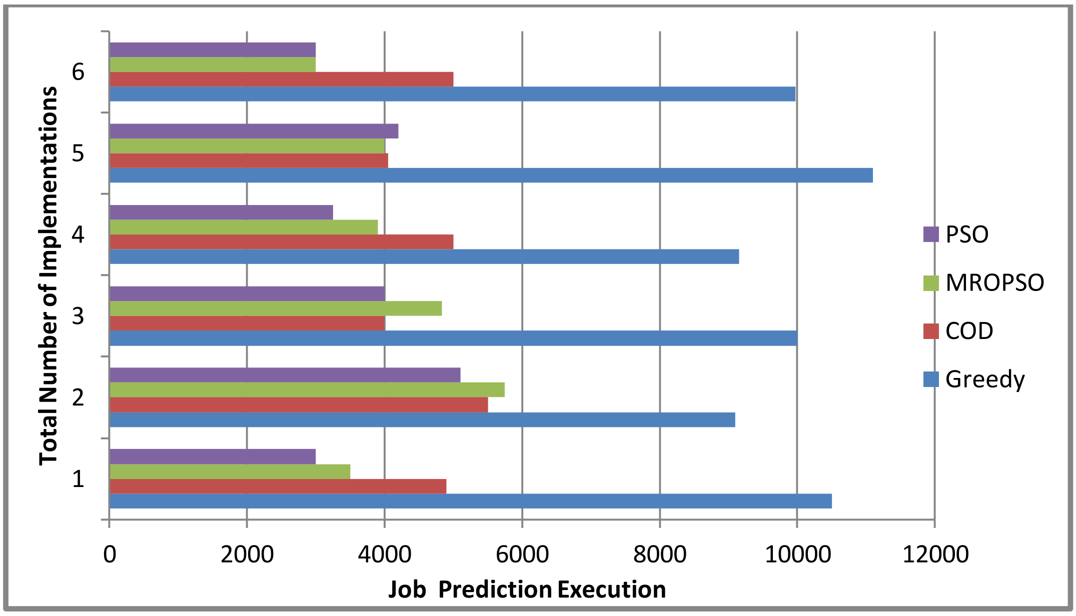

Fig.5: Job number = 500, Cloudlet Number = 50

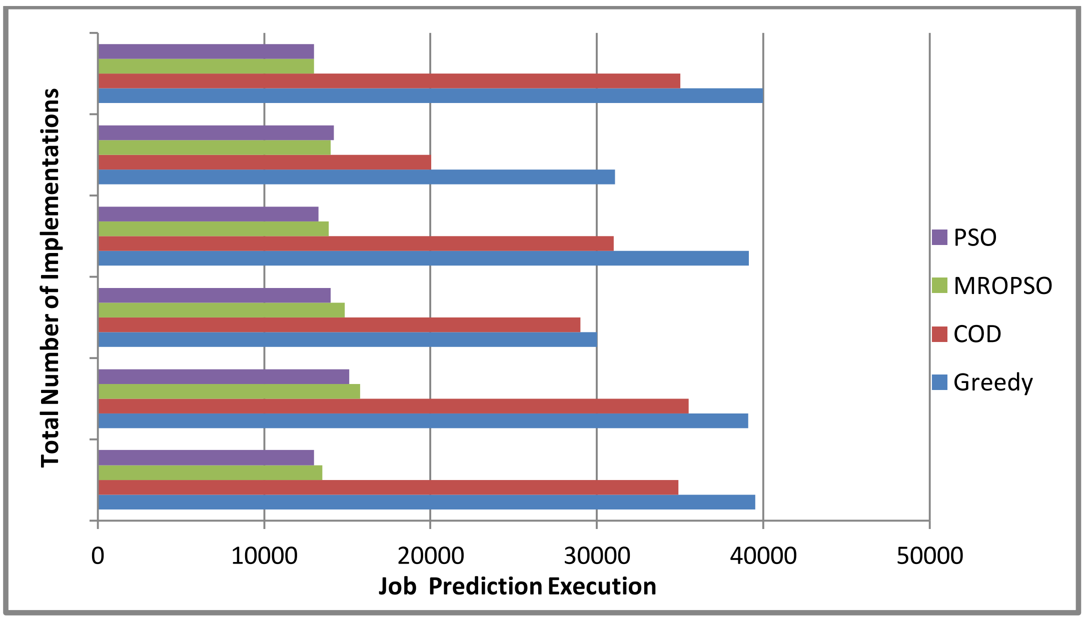

Fig.6: Job number = 1000, Cloudlet Number = 50

B. Algorithm Running Time (sec)

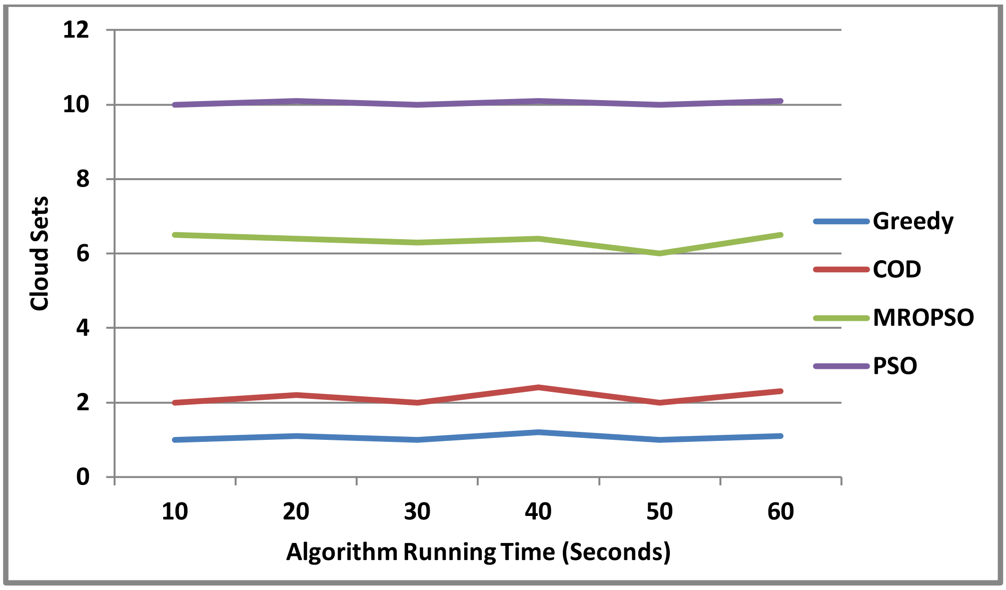

Fig.7: Job number = 500, Cloudlet Number = 10

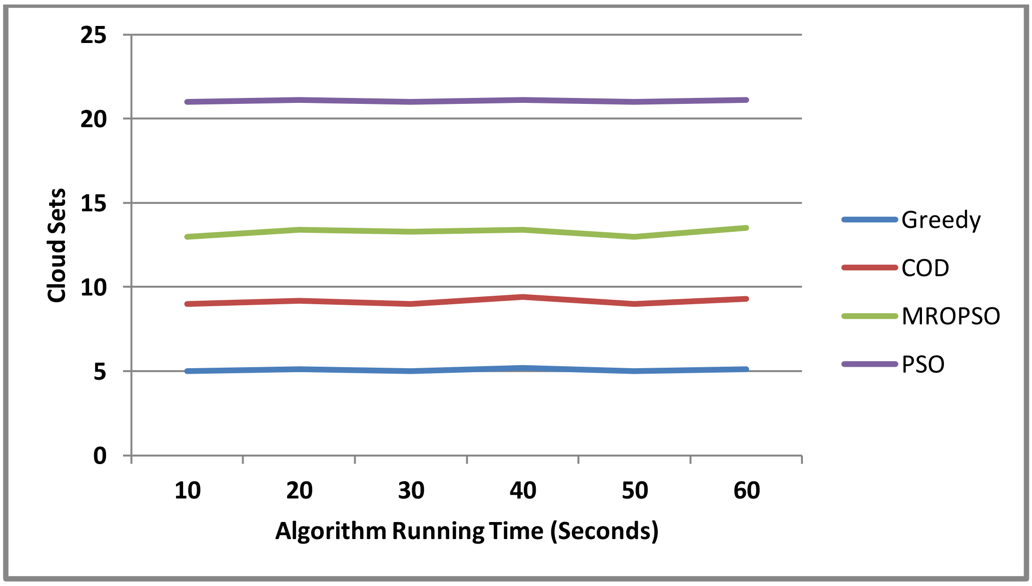

Fig.8: Job number = 1000, Cloudlet Number = 25

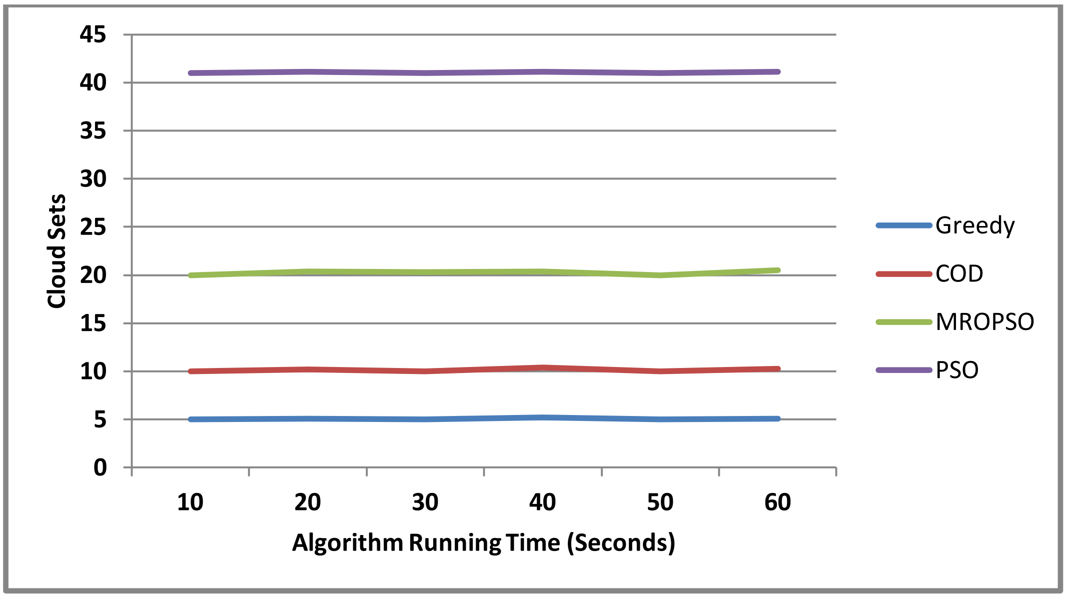

Fig.9: Job number = 500, Cloudlet Number = 50

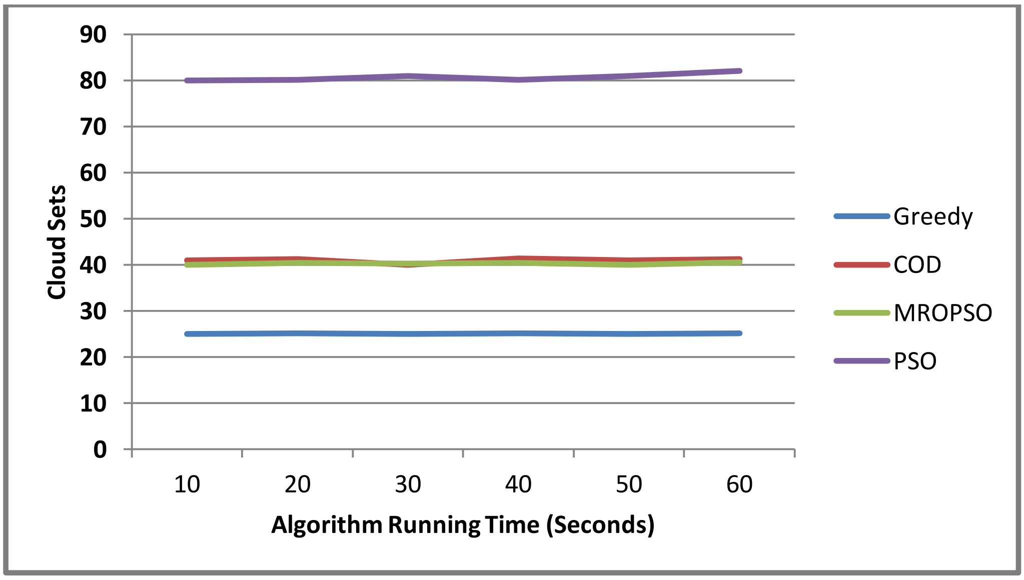

Fig.10: Job number = 1000, Cloudlet Number = 50

Based on the results of the activating time of the systems, when the cloudlet assets and works in fig. 7 to 10 are little, the ongoing time of the COD and greedy procedures is little than the other systems. On the other hand, when there are numerous work and cloudlet assets, the operating time of the COD and the MROPSO is almost matching. Although the predecessor works of the PSO and MROPSO operations are only matching time activating. The operation time of the PSO algorithm is longer than that of the MROPSO method.

D. CONCLUSION

The main purpose of this paper is solving the problem of resource allocation for mobile devices jobs. A real-time job offloading system has been proposed to offer precise proof's for users. Compared with the PSO technique with the MROPSO technique, it is feasible to discover a system that has time to perform almost the same jobs. However it's running time is small especially when there are more jobs and clouds. Compared to the COD and greedy mechanisms, although the MROPSO process usually takes longer, in less work, finding the job offloading system. In future, the MROPSO can be enhanced to improve security based on efficient protocols.[16,17 and 18].

REFERENCES

1. Ashwin, M., Kumar, E. S., Naidu, R. C. A., & Ramamoorthy, R. (2023, March). IoT based Innovative Teaching Learning using Smart Class Rooms. In 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS) (pp. 1143-1148). IEEE.
2. Ramamoorthy, R., Kumar, S., Naidu, R. C. A., & Sathya, M. (2022, December). Hybrid MultiHop Routing Mechanism with Intelligent Transportation System architecture for Efficient Routing in VANETs. In 2022 International Conference on Disruptive Technologies for Multi-Disciplinary Research and Applications (CENTCON) (Vol. 2, pp. 69-74). IEEE.
3. Smitha, D., Ramamoorthy, R. A. G. H. U., & Naidu, D. R. C. A. (2021). OHAR: optimized human action recognition paradigm using optimized type-2 neuro-fuzzy classifier. Period Mineral, 91(4), 684-699.
4. SajeebSaha, Mohammad S. Hasan, "Effective job migration to reduce execution time in mobile cloud computing", 2017 23rd International Conference on Automation and Computing (ICAC), 26 October 2017.
5. Bowen Zhou, Amir VahidDastjerdi, "mCloud: A Context-Aware Offloading Framework for Heterogeneous Mobile Cloud", IEEE Transactions on Services Computing, Volume: 10, Issue: 5 , Sept.-Oct. 1 2017.
6. V. Meena, M. HariPrasath, "Optimal resource reservation for offloaded jobs in mobile cloud computing", 2017 2nd International Conference on Communication and Electronics Systems (ICCES), 22 March 2018.
7. R. G. Alakbarov, O. R. Alakbarov, "Mobile clouds computing: Current state, architecture and problems", 2017 Second International Conference on Electrical, Computer and Communication Technologies (ICECCT), 23 November 2017.
8. M ShanthiThangam, M Vijayalakshmi, "Data-intensive Computation Offloading using Fog and Cloud Computing for Mobile Devices Applications", 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), 01 July 2019.
9. QassimBani Hani, Julius P. Ditcher, "Mobile-Based Location Tracking without Internet Connectivity Using Cloud Computing Environment", 2017 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), 12 June 2017.
10. IbrarYaqoob, Ejaz Ahmed, "Heterogeneity-Aware Job Allocation in Mobile Ad Hoc Cloud", IEEE Access (Volume: 5), 14 February 2017.
11. V. Kalpana, S. Swathikha, "A profile guided, analysis for energy-efficient computational offloading for mobile cloud computing environment", 2017 2nd International Conference on Communication and Electronics Systems (ICCES), 22 March 2018.
12. InduSahu, U.S. Pandey, "Mobile Cloud Computing: Issues and Challenges", 2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), 01 July 2019.
13. Dezhong Yao, Chen Yu, "Using Crowdsourcing to Provide QoS for Mobile Cloud Computing", IEEE Transactions on Cloud Computing (Volume: 7, Issue: 2 , April-June 1 2019).
14. Saravana Kumar, E., Ramkumar, P., Naveen, H. S., Ramamoorthy, R., & Naidu, R. C. A. (2023). Classification of COVID-19 with Belief Functions and Deep Neural Network. SN Computer Science, 4(2), 178.
15. Ramamoorthy, R., & Thangavelu, M. (2022). An enhanced distance and residual energy- based congestion aware ant colony optimization routing for vehicular ad hoc networks. International Journal of Communication Systems, 35(11), e5179.
16. Ramamoorthy, R., & Thangavelu, M. (2021). Group based dual mode key management scheme for secure communication in vehicular ad hoc networks. Wireless Personal Communications, 120, 949-973.

17. Raghu, R., R. Prabhushankar, J. Rajaram, and M. Vaiyapuri. "Efficient dead reckoning approach for localization prediction in VANETs." J Appl Sci Comput 6, no. 3 (2019): 2093-2099.
18. Raghu, R., & Menakadevi, T. (2016). A survey on anonymous secure on-demand routing protocols in MANETs. Middle East J. Sci. Res, 24, 3869-80.