Original Article
ISSN (Online): 2582-7472

ShodhKosh: Journal of Visual and Performing Arts
January-June 2023 4(1), 1042–1050

FACTUAL-TIME ASSETS ALLOCATION CONFIGURATION FOR MANY-TASK UNLOADING IN MCC

S. Saranya [1], P. Jeevitha[2], M. Kavitha [2], K. Nithyashree [2], P. Pavithra [2]

[1] Assistant professor, Department of Computer Science and Engineering, Mahendra Engineering College
[2] UG Students, Department of Computer Science and Engineering, Mahendra Engineering College

DOI
10.29121/shodhkosh.v4.i1.2023.2851

Funding: This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

ABSTRACT

The purpose of using this paper cloudlet is remotely reduces the cloud communication delay and the system load. However, since the powerful system performance of the remote cloud is not in the cloudlet, as the no. of user's changes, the cloud provides resources for each user change. Moreover, as the cloudlet's service security is low, cloudlet's service for user, when of work activation is eliminated from coverage. In this case, users will not get the computational results, so the cloud compute fails. This paper provides a real-time resource allocation framework for users to allocate sufficient and necessary resources. This paper proposes the Movement Record-based Particle Swarm Optimization (MRPSO) method to solve the problem 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), MRPSO, Assets Allocation, Task Offloading

1. INTRODUCTION

Mobile devices include vehicles, smartphones, and laptops. As the number of mobile devices continues to rise, the need for services such as speech recognition software, online games and face recognition software is increasing. To solve problems such as resource limitation and mobility, research on MCC is very important. For a cloudlet of resources known as computational resources, via wireless access, by moving tasks, can improve the computing power of mobile devices.

To address the resource limit of mobile devices, 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 devices via high speed WLAN.

To diminish the power consumed by handheld devices, MCC is used. However, due to various challenges such as task calculation failure, long operational delay, and finite network bandwidth, cloud computing affects the actual usage.

In this paper, to solve the above problems, a real-time resource allocation framework is proposed. The system developed for this paper is, resources can be allocated based on information on resource loads and real-time tasks. And you can plan the task offloading system more efficiently and faster. This includes information such as the cloudlet and the user's location, the user's expected time and the amount of tasks.

In paper [1], that author have used the Decision making algorithm to solve the operational challenges in Mobile Cloud Coupling. In paper [2], that analyzer discuss how to optimize MCC performance, for that, they have proposed a code offloading framework. The purpose of paper [3] is to find the right mapping between cloud resources and mounted task. Heterogeneous and homogeneous configurations of virtual machines have been used. In paper [4], concerned about the resource savings of mobile devices, they have used the MCC technique.

In paper [5], they have used controller in cloud layer to achieve high response time and low latency challenges.The objective of this paper [6] is to obtain the location of users for security purposes. Longitude and Latitude information is collected using the longitude and latitude algorithm.

2. PROPOSED METHODOLOGY

In this section, the proposed resource allocation system is introduced through two assumptions. One is its function and the other is its structure.

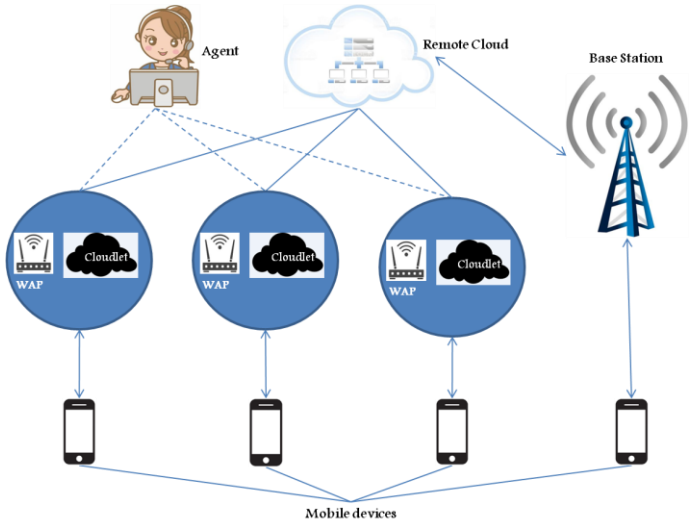

Figure1 Resource Allocation Structure

In figure 1, there are cloudlets, mobile devices, agencies and remote cloud. When mobile devices move from one location to another, they bring some tasks to the remote cloud or to the cloud in less running time, and save energy consumption. Mobile devices can join the cloudlet through the WAP (wireless access point) distributed through the cloudlets. With a remote cloud, the cloudlets can connect to a high-speed wire network. Mobile devices can also access remote cloud utilizing the base station. These two resource access points are more important. Because when a mobile device is running fast, in the Cloudlet's service coverage area, operating results do not result. In this situation, mobile device can offload with remote cloud, and function results can be easily obtained.

In the proposed framework, the part of the agent is very important. This is because it monitors mobile device information's, cloud resources, and work offloading requests. When users assign tasks to cloudlets, they have to send work offloading requests to nearby cloudlets. The agent then receives all requests from the cloud. Depending on the location 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 uses the resource allocation system.

This section introduces the proposed system. The proposed system consists of three volumes, which can be seen in Figure 2. They are, respectively, the Cloud Resource Manager (CRM), the User Information Manager (UIM), and the Task Offloading Agent (TOA). 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.

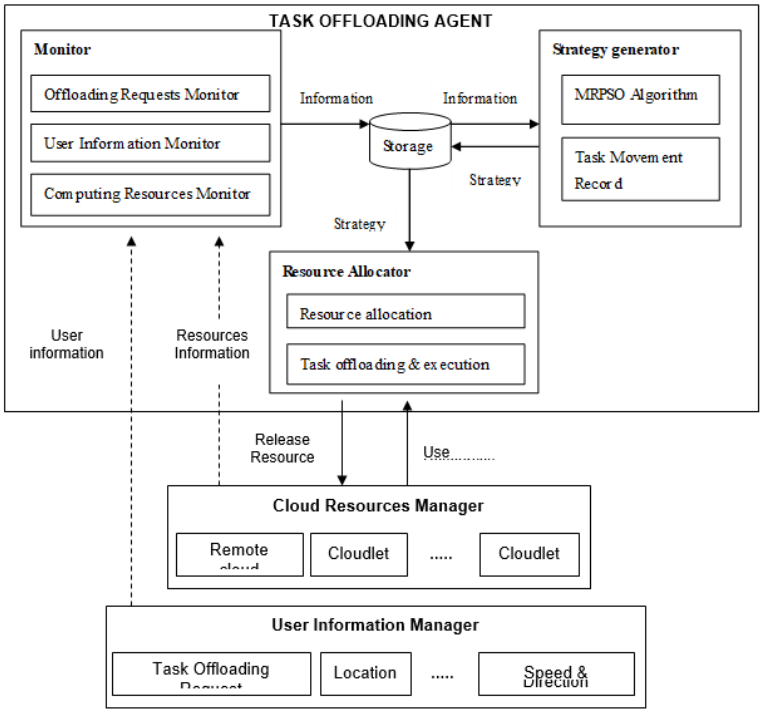

Figure2 Proposed System Architecture

1) Monitor

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

2) Data Storage

This unit is a database that is temporarily given to stored information. Due to the time distinguish between user cloud information and the collection of work offloading requests, the agent will continue to collect information and store them in storage. When retrieving data per block, it call information 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 task, to move to the most appropriate cloudlet the time of work activation and the location of the work movement are recorded. Then, based on the information of resources and information of task, SG calculates the tasks 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 section.

4) Resource Allocator

Cloudlet resources are assigning to task and obtaining the cloudlet resources from remote cloud and cloudlets, these element functions are based on offloading techniques. After stored the computing resources, tasks can be executed immediately. To use other tasks after completing tasks, resources are released immediately.

The first part of this section is devoted to the problem clarity of resource allocation in MCC. Then an improved formalized MRPSO method is presented.

- **Problem Definition**

Due to the dynamics of resources and the mobility of users, it is difficult to know precisely the status of future resources. The shared user's complex tasks need to be converted into sub-tasks. Since there is correlation between sub-

tasks, have to wait until some tasks are completed. Then they will be executed. If there is a large of time spent in the previous stage, when the Resources Plan was prepared at the time of the work's execution, it is not certain that the resources used are the same.

So based on currently known resources the work offloading program can be developed. Functions to utilize planned resources, Resource allocation should be done in a short period of time.

- **PSO Algorithm**

The multi-goal resource allocation problem is, because a NP-Complete Problem, to solve this problem in a polynomial time, PSO is used in this paper. PSO is a calculation method. It depends on a specific data; to improve a candidate solution trying will find a solution. However, to find an optimal solution in shorter time to the original PSO system, this paper was created. The optimal 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.

- The jth particle's location: aj= (aj1, aj2, …,ajX).
- The jth particle's speed: sj= (sj1, sj2, …,sjX).
- The best location of jth particle: bj= (bj1, bj2, …,bjX).
- The best location of all particles: bi= (bi1, bi2, …,biX).
- The k1 and k2 are represent the rate of learning from the particle's own optimal solution and acceleration constants and the optimal 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 taken 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 + k1ξ (b_jX^C- a_jX^C) + k2η (b_iX^C- a_jX^C) (1)

a_jX^(C+1) = a_jX^C + s_jX^(C+1) (2)

- **MRPSO Algorithm**

When the unique problem solving, such as resource allocation, in original PSO mode, there is no specific 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 problem, this paper proposes the PSO method for work operating logging.

To record task movement information, used a two-dimensional array called TD[C] [T]. Where, C is the no. of cloudlets. T is the no. of tasks. The horizontal axis of the two-dimensional line represents the task ID, and the vertical axis represents the cloudlet ID. Thus each cloudlet will overlap with each task.

Let td^i_{tCT} be the particle's t^{th} task movement record at i^{th} iteration

$$td^i_{tCT} = \begin{bmatrix} td^i_{t11} & \dots & td^i_{t1T} \\ \vdots & \ddots & \vdots \\ td^i_{tC1} & \dots & td^i_{tCT} \end{bmatrix} \quad (3)$$

Here, T is the task number and C is the cloudlet number.

$$td^i_{tCT} = \frac{b_e t_T}{n_e t^i_{CT}} \quad (4)$$

Let n_e t_CT^ibe the fresh result time of Tth task in Cth cloudlet at ith iteration. Let b_e t_T= {b_e t_1,…,b_e t_2,…,b_e t_T} be the top result time of task. The value is calculated by assuming that each task is loaded separately for the most 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 tasks, it means that the task has to move much higher. The target location of the task 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 task may take a short time to load to the cloud. When all tasks are executed in legitimate clouds, all tasks will have reached very little execution time. Therefore, the resource allocation plan is appropriate.

- **User's Mobility Problem**

There is concern about users' mobility in MRPSO. Since cloudlets have a certain 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 task, functional results of the task are not obtained. This will result in failures in loading tasks. Therefore, after computing the operating time of each task, the MRPSO 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 selected. If the activation time exceeds the dependent time, the task 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.

- Pseudo Code

Let $Cur_e^i = [Cur_{e0}^i, \dots, Cur_{et}^i, \dots, Cur_{eT}^i]$ be the particles eth execution location ID of task t at ithiteration.

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

Let fitness () $=\sum_0^T et_t$= the execution time of task t, Imx= the maximum no. of iterations. Tpar= swarm size.

Pseudo code of MRPSO algorithm

Initialize Cur^1 randomly, set td[1] with 1;
Calculate the $b_{et_T;}$
For all particles in the swarm, td_{tCT}^i(e), Cur_t^i (e), e∈ $[1, T_{par}]$
    Calculate the fitness value of particle e;
    Let $O_l^1 = Cur^1;$
End for;
Copy O_g^1 to be the one with the best fitness in Cur[1] (k), 1 ≤ k≤ T_{par};
t = 1;
While t ≤ I_{mx};
For all particles in the swarm, td_{tCT}^i(e),Cur_t^i (e), e∈ $[1, T_{par}]$
    Based on Cur_t^i (e), find the task t of the smallest value in the td_{tCT}^i(e);
    Based on td_{tCT}^i(e), find the cloudlet c with the biggest value of task t;
    Change the execution location Cur_t^i (e) into cloudlet c;
    Calculate the fitness value of particle e;
    Update the td_{tCT}^i(e) with the fitness value;
        If fitness (Cur_t^i (e)) is better than fitness (O_l^1)
            $O_l^1 = Cur_t^i$ (e);

        End if
End for
    Set O_g^1 to be the one with the best fitness in Cur_t^i (e), 1 ≤ k≤ T_{par};
    i + +;
End while

3. RESULTS AND DISCUSSION

In this part, the effectiveness of the proposed resource allocation mechanism is evaluated. First, the simulation scheme is introduced, 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 tasks. Many of these tasks have different computation sizes and data sizes.

In order to simulate this as realistic as possible, in this paper, the no. of works is 500, 1000 and the no. of cloudlets is 10, 50 quantified. This paper, PSO, MRPSO, COD and Greedy algorithms is executed fifty times respectively. The population of PSO and MRPSO 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 tasks offloading, so the proposed method of this paper has proven to be superior.

2) Simulation Results

Depending on the results of the task 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 MRPSO methods have a better work offloading system than other methods. This means that the total time of tasks is very limited.When there are more cloudlet resources, although the COD algorithm is better in one location than the MRPSO method and but their results are not consistent. Therefore, the results of the task offloading mechanism of the PSO and MRPSO methods are consistent.

Task Prediction Execution Time (sec)

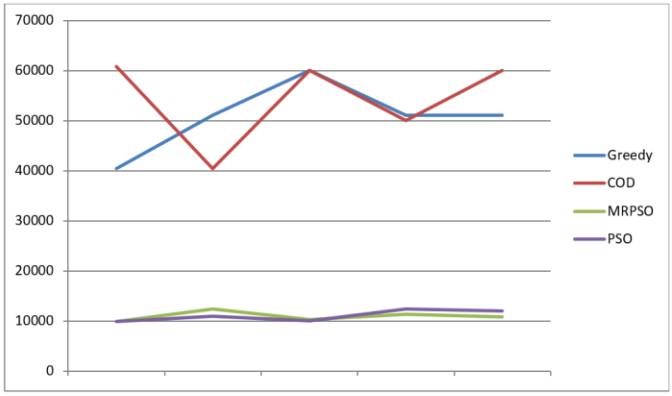

Figure3 Task number = 500, Cloudlet Number = 10

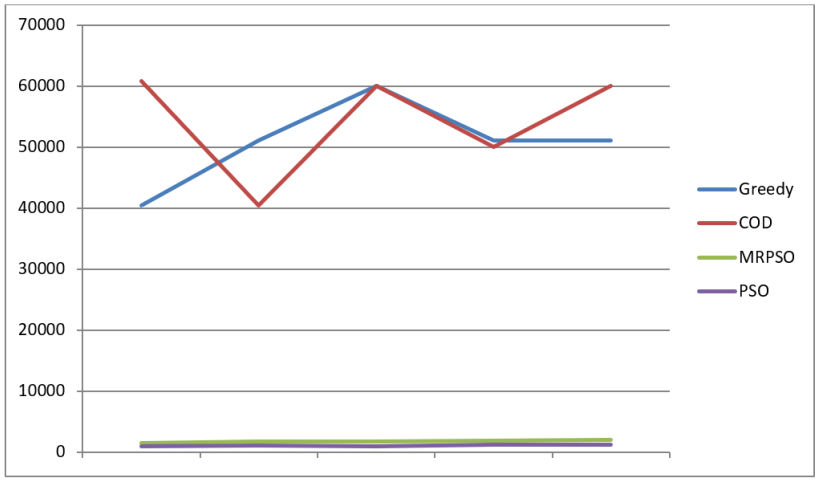

Figure 4 Task number = 1000, Cloudlet Number = 10

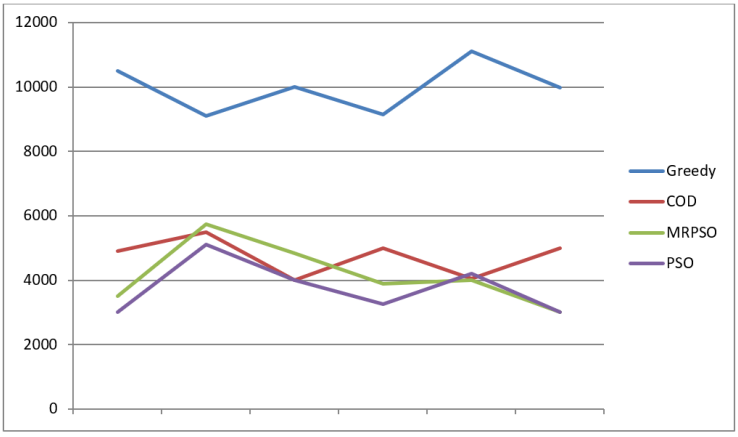

Figure 5 Task number = 500, Cloudlet Number = 50

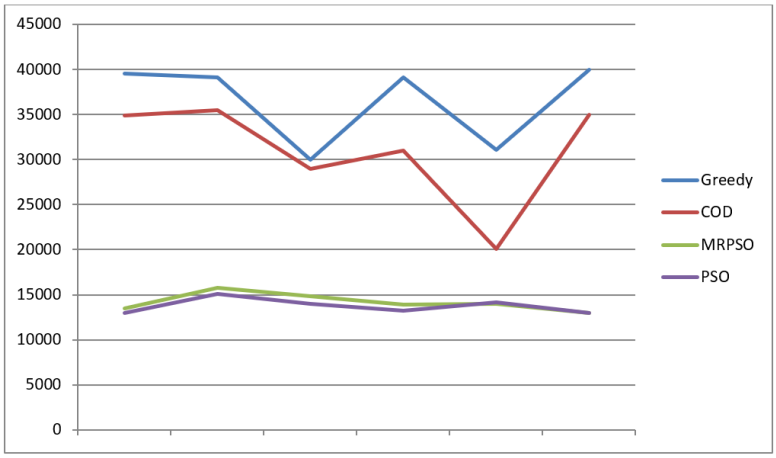

Figure 6 Task number = 1000, Cloudlet Number = 50

Algorithm Running Time (sec)

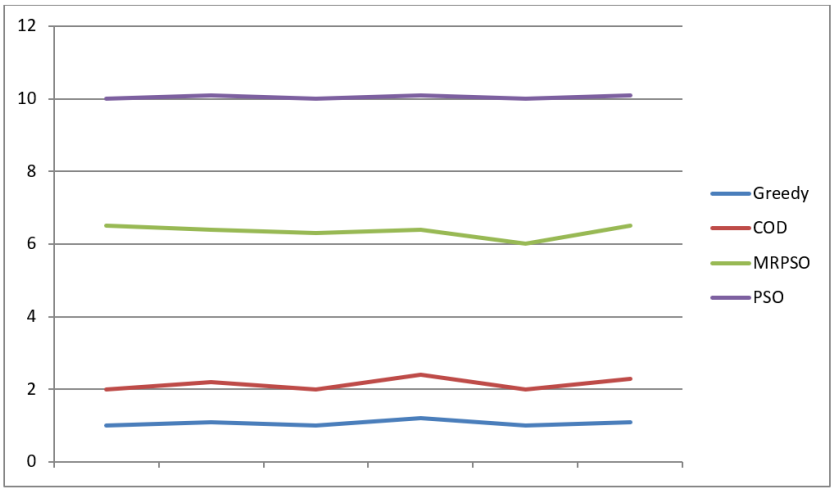

Figure 7 Task number = 500, Cloudlet Number = 10

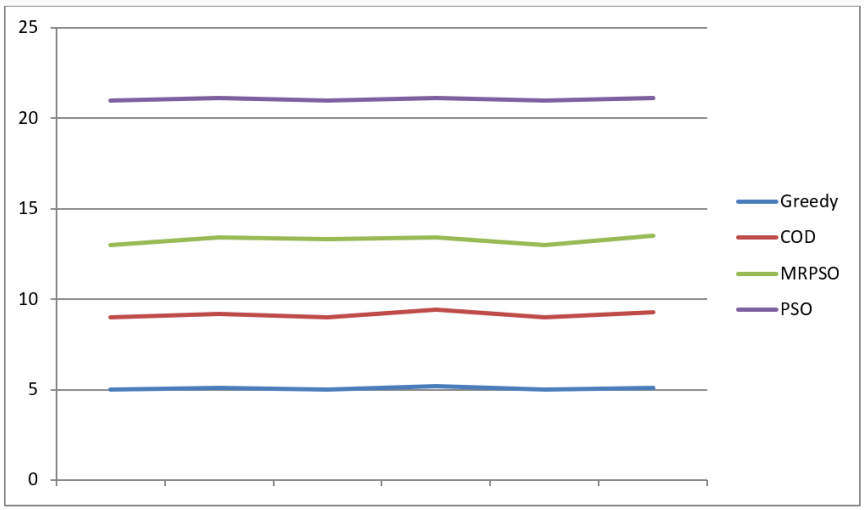

Figure 8 Task number = 1000, Cloudlet Number = 10

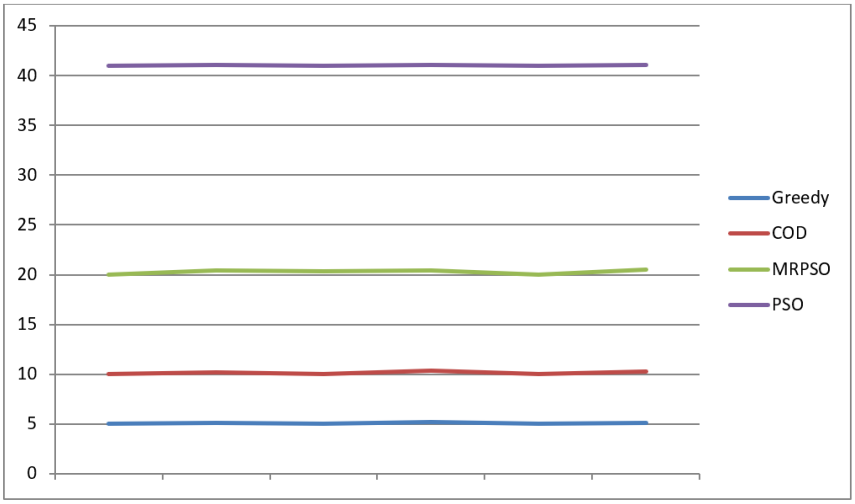

Figure9 Task number = 500, Cloudlet Number = 50

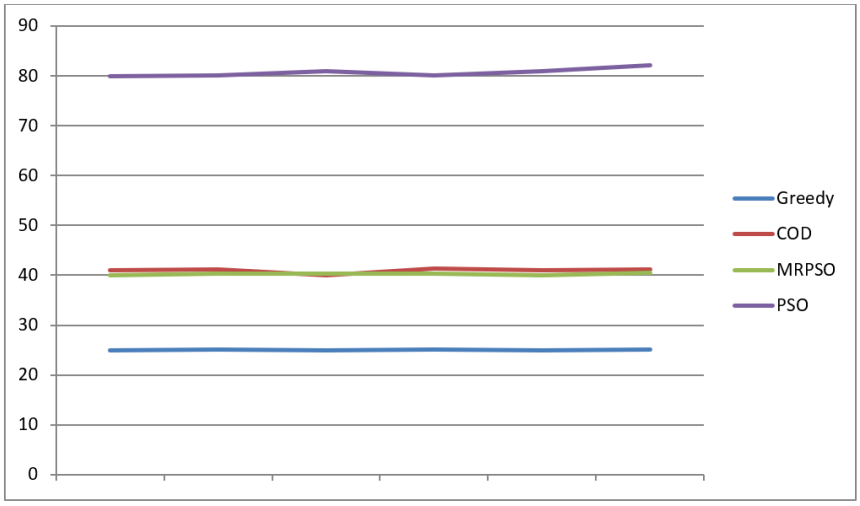

Figure10 Task number = 1000, Cloudlet Number = 50

According to the results of the activating time of the systems, when the cloudlet assets and works in fig. 8 to 10 are little, the ongoing time of the COD and greedy procedures is little than the other systems. However, when there are many work and cloudlet assets, the operating time of the COD and the MRPSO is almost matching. Although the predecessor works of the PSO and MRPSO operations are only matching time activating. The operation time of the PSO algorithm is longer than that of the MRPSO method.

4. CONCLUSION

The main purpose of this paper is solving the problem of resource allocation for mobile devices tasks. A real-time task offloading system has been proposed to provide accurate proof's for users. Compared with the PSO method with the MRPSO technique, it is feasible to find a system that has time to perform almost the same tasks. However it's running time is small especially when there are more tasks and clouds. Compared to the COD and greedy mechanisms, although the MRPSO process usually takes longer, in less work, finding the task offloading system.

CONFLICT OF INTERESTS

None.

ACKNOWLEDGMENTS

None.

REFERENCES

SajeebSaha, Mohammad S. Hasan, "Effective task migration to reduce execution time in mobile cloud computing", 2017 23rd International Conference on Automation and Computing (ICAC), 26 October 2017.

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.

V. Meena, M. HariPrasath, "Optimal resource reservation for offloaded tasks in mobile cloud computing", 2017 2nd International Conference on Communication and Electronics Systems (ICCES), 22 March 2018.

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.

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.

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.

IbrarYaqoob, Ejaz Ahmed, "Heterogeneity-Aware Task Allocation in Mobile Ad Hoc Cloud", IEEE Access (Volume: 5), 14 February 2017.

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.

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.

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).