



The Conjectural Framework for Finding Optimal DC using Modified PSO Algorithm in Mobile Cloud

M. Durairaj¹, P. Kannan²

¹Assistant Professor, ²Research Scholar, School of Computer Science, Engineering and Applications,
Bharathidasan University, Tiruchirappalli, Tamil Nadu, India

Abstract: Managing and scheduling the resources in the form of virtual resources which are allocated on-demand basis are the core works in cloud computing. Mobile and cloud computing have together evolved as a new technology called Mobile Cloud computing. Efficient resource allocation is one of the major challenging tasks in the mobile cloud environment as the virtual resources are always dynamic in mobile cloud. To find optimal cloud provider's datacenter for resource allocation is a serious challenging problem which is on-step. Hence, this paper proposes a conjectural framework for finding the nearest optimal datacenter by applying a modified PSO algorithm. The proposed framework is developed by using the basic of how PSO algorithm finds an optimal solution in the search space. This algorithm tends to find the optimal solution in minimum time.

Keywords: Mobile Cloud Computing, datacenter, Resource Management, PSO

I. Introduction

In this modern era, most of the business activity is processed with the help of mobile devices. A person or enterprise can provide services to the user, without considering mobile users movement they are able to continue their work seamlessly at anywhere anytime in 24×7 basis. Mobility [1] has many challenges such as limited energy, insufficiency of resource, connectivity maintenance. These kinds of identified problems have been solved by using new computing technology called cloud computing. According to National Institute of Standards and Technology [2] "Cloud Computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources such as networks, servers, storage, applications and services that can be rapidly provisioned and released with minimal management effort or service provider interaction".

Offloading the computation part to cloud providers is the background idea of cloud computing. The services [3] rendered by cloud providers are broadly classified into three major types as infrastructure as a service (IaaS), platform as a service (PaaS) and software as a service (SaaS). The combination of mobile computing and cloud computing has laid the instauration for a new computing approach called Mobile Cloud computing, which permits cloud users to access cloud resources like processing power, storage, application, etc., through mobile devices which act as a thin client. Dejan Kovachev defined [4] "Mobile cloud computing is a model for transparent elastic augmentation of mobile device capabilities via ubiquitous wireless access to cloud storage and computing resources, with context-aware dynamic adjusting of offloading in respect to change in operating conditions, while preserving available sensing and interactivity capabilities of mobile devices."

Resource [5] allocation is an important challenge to enhance resource utilization in cloud and mobile cloud computing environment. In mobile cloud, before allocating resources to find an optimal cloud provider's location or datacenter is very important challenge. In order to find an optimal datacenter, heuristics techniques are better than the traditional searching techniques. Normally, the mobile cloud architecture [6] should be proactive, self-adaptive, maintain mobility and location awareness.

In this paper, conjectural framework is proposed to find the nearest optimal datacenter by combining particle swarm optimization and genetic algorithm approach. PSO [7] [8] used for efficiently find an optimal DC in search space. Genetic approach is applied for resource allocation. This framework will be useful for normal cloud and mobile cloud. If the user access cloud resources such as virtual machines and applications from mobile devices then mobile cloud procedures will be applied. This procedure concentrates on network provider and cloud provider with single or multi backup systems.

However, existing research work concentrates on minimizing task processing time, load balancing in normal cloud computing environment which is in a static level. In mobile cloud, computing process happens in dynamic environment which will vary from time to time and based on network/bandwidth limit and availability of network coverage. In this paper, we designed a framework which combines normal cloud and mobile cloud workflow and apply a particle swarm optimization based algorithm for handling cloud resources efficiently to achieve a minimum response time to cloud users and try to provide better service.

The remaining part of this paper is organized as follows. Related work of this paper is briefly given in section II. Section III, discusses the existing heuristic techniques and their description. Proposed framework is presented in section IV. Section V, gives the conclusion and future direction of this paper.

II. Related Works

Verbelen *et al.* [9] presents cloudlet based deployment optimization for handling offloading techniques. It also depends on WAN Latency, bandwidth limit and application nature. Considering data transmission cost and computation cost while scheduling applications in cloud resource management with the support of heuristic technique Particle Swarm Optimization (PSO) was proposed by S. Pandey *et al.* [10]. From the given set of grid resources PSO find a better optimal solution for coordinating all jobs in the workflow [11]. Satyanarayanan *et al.* [12] delivered the concept of cloudlet in which mobile devices act as a thin client and respect to a cloud providers service for managing cloud services depends on latency and bandwidth limit. The services are customized and accessed through virtual machine based on the user request in the wireless LAN.

W. Min *et al.* [13] proposed a particle swarm optimization algorithm to improve mobile agent scalability limitation based on a domain partition in hierarchical network management. This algorithm concentrates on domain partition, mobile agent migration, time efficiency and load balancing. Clustering strategy [14] was applied to perform multiple mobile agents for tasks simultaneously and the results show that the processing time was reduced. Particle Swarm Optimization based scheduling heuristic technique used to minimize execution cost and distribute workload in a better manner than Best Resource Selection algorithm in normal cloud not in mobile cloud environment [15].

Mingyue Feng *et al.* [16] designed a particle swarm optimization algorithm to resolve resource allocation problem by introducing Pareto-dominate theory which provides a good support for multi-objective optimization problem. It considers resource allocation, execution time of total task and quality of service for getting a better optimal solution. Wang *et al.* [17] pointed that mobile agent able to amend the time efficiency and scale down the network traffic based on the experimental results and mathematical analysis in network management. E.E. Marinelli [18] proposed a mobile-cloud infrastructure called Hyrax which allows smart phone application to effectively use data and perform computing jobs on heterogeneous and smart-phone networks. The concept was implemented by using hadoop framework and examined on android platform.

Anh-Dung Nguyen *et al.* [19] addressed how mobility impacts on mobile cloud computing environment from mobile ad-hoc network. They demonstrated the mobility can effectively and efficiently improve the distributed computation performance with mobile patterns which includes small-world network structure. This structure improves mobile cloud computing service resilience. J. Yang *et al.* [20] proposed a novel general provable data possession scheme which includes trusted third-party agent, merkle hash tree, trusted platform model chips and bilinear signature. This scheme improves offloading techniques and computational processing and transfer from mobile end user to cloud server side.

Yifei Yuan *et al.* [21] analyzed whether the bandwidth problem was solved incisively with acceptable overload and formulate alternate techniques for abstraction in datacenter topologies. Pelin Angin *et al.* [22] proposed a framework which support dynamic performance optimization and worked based on the mobile agent with application partition level in mobile cloud computing. B.G. Chun *et al.* [23] proposed a clonecloud framework which is used to partitioning the application and unites static code analysis with dynamic procedure profiling. This framework also concentrated energy consumption and try to optimize execution time on mobile devices with a support of optimization solver. Hence, this approach needs a full virtual machine copy at the remote execution site, it may lead to use more cloud resources and makes a chance for application code in vulnerable situation.

A mobile agent based middleware framework was proposed by Li *et al.* [24] for using mobile computation technique in cloud computing environment. This framework creates a simple model for mobile services for executing and migrating resources in a smooth manner. Brandoan Heller *et al.* [25] presented a network manager to manage datacenter traffic load an elastic manner called Elastic Tree. They equate multiple strategies for detecting low power resource level network subsets among the different traffic patterns. Moreover, they tested the trade-offs in e-commerce applications and save considerable network energy using heuristic techniques.

III. Particle Swarm Optimization

Particle Swarm Optimization [26] is a semi-robotic, conventional and optimization based swarm intelligence algorithm. Swarm intelligence is a part of artificial intelligence which takes the collective intelligent demeanor interaction between particles of the swarm. It considers intelligence as the amalgamation of the acquaintance gained by individuals through past experiences and other social interactions behavior. The inspiration for PSO is that the organism's social behavior patterns which populate and interact in a large search space. The PSO is better situated for operation execution than genetic algorithm. PSO algorithm does not have crossover or mutation process and its particle motion is affected by the function of velocity [27].

Particle swarm optimization in the orbit of computer science is an arithmetical computation method that produce a better solutions while optimizes the objectives by improving the candidate solution repeatedly. A problem has been evaluated in PSO with candidate solutions, which are recognized as particles. The particles movement in search space adopting to mathematical computation rules over the position and velocity of particles. The local best position affects the movement of each particle. It guides the particles to find the best known positions in the search space. The final best known positions are updated and known to the other particles.

Basically particle swarm comprises of 'n' particles. The potential solution of D-dimensional space, individuals, potential solutions and flow through hyper dimensional search space of each particle is indomitable by its position in the search space. The change of particle within the swarm is predisposed by the knowledge or familiarity from the neighbour. The algorithm involves three basic steps which comprises the fitness evaluation, updating of individual, global best functions and velocity with position updating of each particle. The best fit particle of the entire swarm ascertains the position of each particle. Each individual particle has current position in search space with current velocity v_i and a personal best position $P_{best,i}$, the objective function f determine the value of i . [28]. The $P_{best,i}$ is used to calculate the global best position G_{best} . The buck value is obtained by comparing all the $P_{best,i}$.

The $P_{best,i}$ is calculated by using the formula

$$P_{best,i} = \begin{cases} P_{best,i} & \text{if } f(x_i) > P_{best,i} \\ x_i & \text{if } f(x_i) \leq P_{best,i} \end{cases} \quad (1)$$

The formula used to calculate Global Best Position G_{best} is

$$G_{best} = \{ \min \{ P_{best,i} \}, \text{ where } i \in [1, \dots, n] \text{ where } n > 1 \} \quad \text{----} > (2)$$

Velocity can be updated by using the formula

$$V_i^{t+1} = wv_i(t) + c_1r_1[x_i(t) - x_i(t)] + c_2r_2[g(t) - x_i(t)] \quad \text{---} > (3)$$

where,

$v_i(t)$ -> velocity

w , c_1 and c_2 ->supplied co-efficient.

r_1 and r_2 -> random values

$x_i(t)$ ->individual best solution

$g(t)$ ->swarm's global best candidate solution.

$wv_i(t)$ -> inertia component.

$c1r1 [(x_i(t)-x_i(t))]$ ->cognitive component.

$c2r2 [(g(t)-x_i(t))]$ -> social component

The pseudo code of the Particle Swarm Optimization Algorithm is as given below.

Algorithm PSO

Input

m : the swarm size; c_1 , c_2 : positive acceleration constants; w : inertia weight

MaxV: maximum velocity of particles

MaxGen: maximum generation

MaxFit: maximum fitness value

Output:

P_{gbest} : Global best position

Begin

Swarms $\{x_{id}, v_{id}\} = \text{Generate}(m)$; /*Initialize a population of particles with random positions and velocities on S dimensions */

$P_{best}(i)=0$; $i = 1, \dots, m, d=1, \dots, S$

$G_{best} = 0$; Iter = 0;

While (Iter<MaxGen and $G_{best} < \text{MaxFit}$)

{For (every particle i)

{Fitness(i)=Evaluate(i);

IF(Fitness(i)> $P_{best}(i)$)

{ $P_{best}(i)=\text{Fitness}(i)$; $p_{id}=x_{id}$; $d=1, \dots, S$ }

IF(Fitness(i)> $G_{best}(i)$)

{ $G_{best}(i)=\text{Fitness}(i)$; $g_{best}=i$;

}

For (every particle i)

{For(every d)

$v_{id} = w*v_{id} + c_1*rand()*(p_{id}-x_{id}) + c_2*Rand()*(p_{gd}-x_{id})$

IF ($v_{id} > \text{MaxV}$) { $v_{id} = \text{MaxV}$;

IF ($v_{id} < -\text{MaxV}$) { $v_{id} = -\text{MaxV}$;

$x_{id} = x_{id} + v_{id}$

}

}

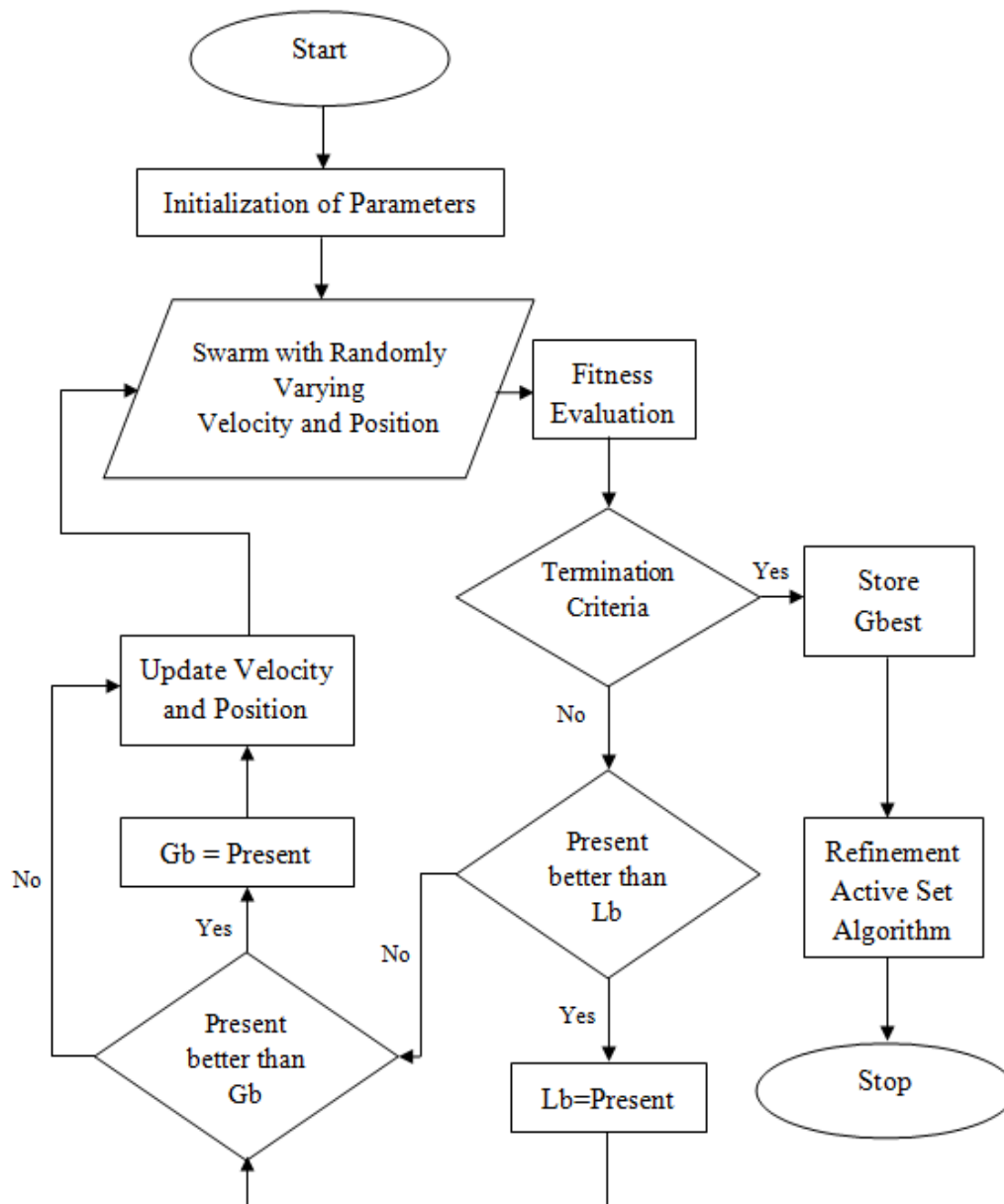
Iter=Iter+1;

```

}/* rand() and Rand() are two random functions in the range [0,1]*/
Return P_{gbest}
End

```

Figure 1: Work Flow of PSO

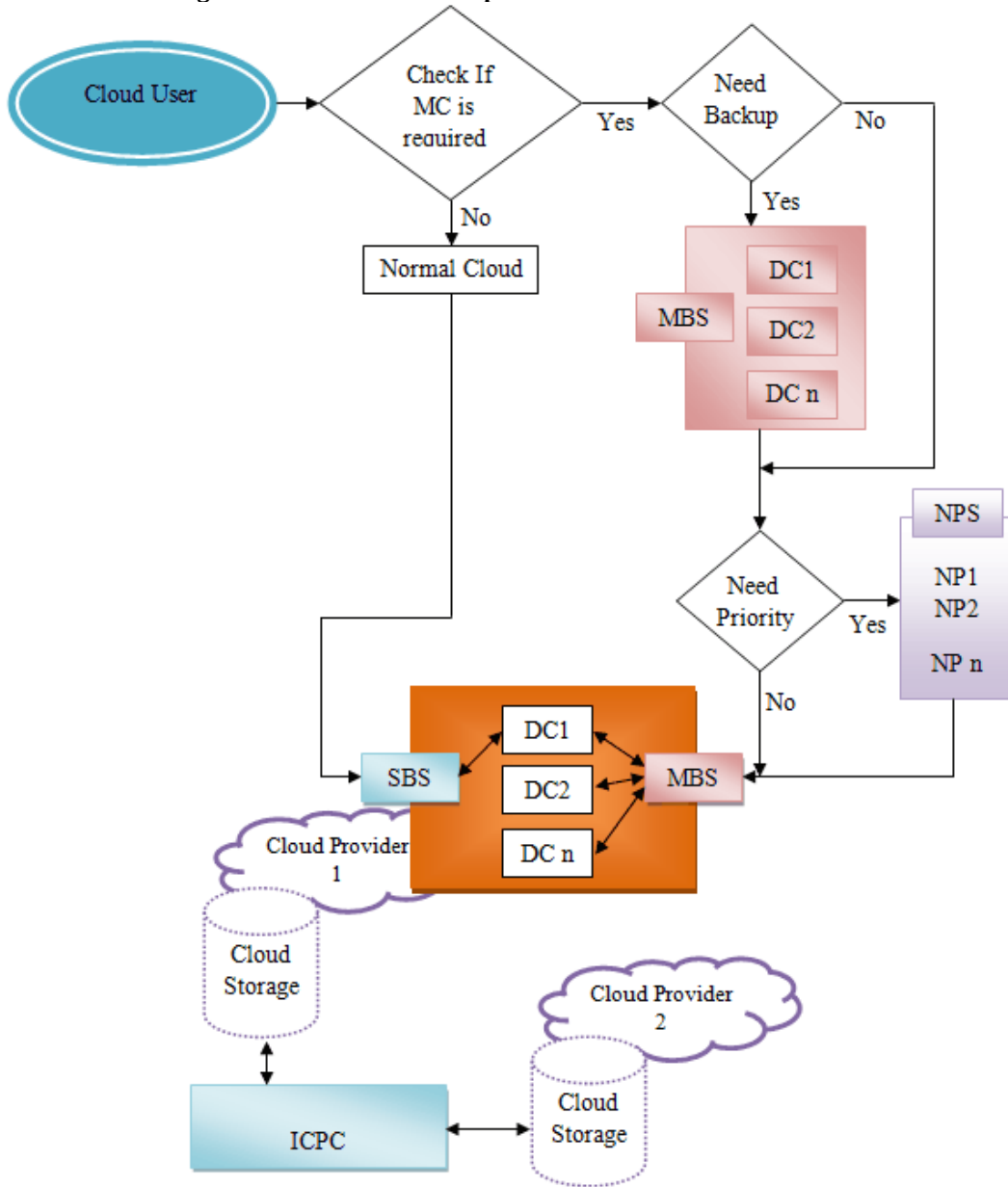


IV. Proposed Methodology

The framework in the Fig. 2 finds the optimal datacenter as follows: As the initial state, user requests the MCP. The provider checks whether the requested resource needs normal/mobile cloud environment for processing. If the resource needs static cloud, the request will be directed to the single backup system and associated with corresponding datacenter. If the resource requires dynamic cloud environment with mobile devices, then the request is directed to mobile cloud

When the user needs backup, then the request is directed to Multi Backup System. In MBS, the nearest optimal datacenter to user is assigned with data backup using modified PSO algorithm. The algorithm will initialize datacenter as particle and $c1$ & $c2$, which are known as social/cognitive accelerants. The fitness value of every particle is calculated by using the Rosenbrock function. The fitness function for rosenbrock is given in equation 4.

Fig. 2 Framework to Find Optimal DC in MCC Environment



The fitness value is compared with the particle best (Pbest). If the CFV (Current Fitness Value) is better than Pbest, then the same is assigned to Pbest as fitness value. Then the Gbest and Pbest is compared by comparison of the results obtained, the nearest optimal DC is found. Later the data backup is provided to optimal DC and to the nearest DC. After assigning the backup, if the user/requests need any priority then the request is redirected to the NPS. If there is no need for priority mean, then the request is processed with the MBS in ICPC environment.

$$f(x) = \sum_{x=1}^i 100x (x_{i+1} - x_i^2) + (1 - x_i)^2 \text{ ----> (4)}$$

V. Conclusion

In this paper, the conjectural framework for finding nearest datacenter is explained which manages the elastic resource allocation in normal cloud and mobile cloud computing environment by using heuristic techniques. This framework handles resources of cloud user, cloud provider and network service provider in effective and efficient manner. In future, the resource allocation strategies will be implemented using combined heuristics techniques with multiple backup support and priority based network service provider in mobile cloud environment.

VI. References

- [1] M. Satyanarayanan, "Fundamental challenges in mobile computing", in: Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing, PODC'96, ACM, New York, NY, USA, 1996, pp. 1–7.
- [2] P. Mell, T. Grance, "The NIST Definition of Cloud Computing", National Institute of Standards and Technology, Information Technology Laboratory, Technical Report Version 15, 2009.
- [3] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, Above the clouds: a Berkeley view of cloud computing, Technical Report UCB/EECS-2009-28, 2009.
- [4] Dejan Kovachev, Yiwei Cao and Ralf Klamma, "Mobile Cloud Computing: A Comparison of Application Models", Information Systems & Database Technologies, RWTH Aachen University, Germany.
- [5] M. Durairaj, P. Kannan, "A Novel Approach for Elastic Application Partitioning in Mobile Cloud", IEEE - ICAET-4th International Conference on Advances In Engineering & Technology, 2014.
- [6] Niroshinie Fernando, Seng W. Loke, Wenny Rahayu, "Mobile cloud computing: A survey", Future Generation Computer Systems, 29, 2013, pp. 84–106.
- [7] Gang Zhao, "Cost-Aware Scheduling Algorithm Based on PSO in Cloud Computing Environment", International Journal of Grid and Distributed Computing, Vol.7, No.1 2014, pp.33-42.
- [8] Durairaj. M, Sivagowry. S, "Feature diminution by using Particle Swarm Optimization for envisaging the Heart syndrome", International Journal of Information Technology and Computer Science(IJITCS), MECS Publisher, Vol. 7, No. 2, 2015.
- [9] T. Verbelen, P. Simoens, F. De Turck, B. Dhoedt, "Cloudlets: bringing the cloud to the mobile user", in: Proc. of the 3rd ACM Workshop on Mobile Cloud Computing & Services, MCS'12, 2012.
- [10] S. Pandey et al., "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in Proc. 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010.
- [11] L. Zhang, Y. Chen, R. Sun, S. Jing, and B. Yang, "A task scheduling algorithm based on pso for grid computing", International Journal of Computational Intelligence Research, Vol. 4, No.1, 2008.
- [12] Mahadev Satyanarayanan, Victor Bahl, Ramon Caceres, and Nigel Davies, "The Case for VM-based Cloudlets in Mobile Computing", IEEE Pervasive Computing, 1, 2011.
- [13] Weidong MIN, Yonghui LIU, Yongzhen KE, Xuemei SUN, "Using Particle Swarm Optimization Algorithm to Improve Multi-agents Network Management", Journal of Computational Information Systems, Vol. 10, No. 2, 2014, pp. 739-746.
- [14] C. Y. Huang, K. Cheng, A. Holt, "An integrated manufacturing network management framework by using mobile agent", The International Journal of Advanced Manufacturing Technology, Vol. 32, 2007, pp. 822-833.
- [15] Suraj Pandey, Linlin Wu, Siddeswara Mayura Guru, Rajkumar Buyya, "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments", technical reports, University of Melbourne, 2009.
- [16] Mingyue Feng, Xiao Wang, Yongjin Zhang, Jianshi Li, "Multi-Objective Particle Swarm Optimization For Resource Allocation In Cloud Computing", IEEE, 2012.
- [17] Y. Wang, W. Qu, "A Framework for the Modeling and Evaluation of the Mobile Agent in Network Fault Management", in: China grid Conference'11, 2011, pp. 220-226.
- [18] E.E. Marinelli, "Hyrex: cloud computing on mobile devices using MapReduce", Technical Report September, DTIC Document, 2009.
- [19] Anh-Dung Nguyen, Patrick Senac and Victor Ramiro, "How Mobility Increases Mobile Cloud Computing Processing Capacity", IEEE, 2011, pp.50-55.
- [20] J. Yang, H. Wang, J. Wang, C. Tan and D. Yu1, Provable data possession of resource constrained mobile devices in cloud computing, Journal of Networks, Vol. 6, No. 7, 2011, pp.1033–1040.
- [21] Yifei Yuan, Anduo Wang, Rajeev Alur, and Boon Thau Loo, "On the Feasibility of Automation for Bandwidth Allocation Problems in Data Centers", IEEEExplore, 2013, pp.42-45.
- [22] Pelin Angin and Bharat Bhargava, "An Agent-based Optimization Framework for Mobile-Cloud Computing", Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, Vol. 4, No. 2, pp. 1-17.
- [23] B.G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud", in Proc. of the 6th European Conference on Computer Systems (EuroSys'11), Salzburg, ACM, 2011, pp. 301-314.
- [24] X. Li, H. Zhang, and Y. Zhang, "Deploying mobile computation in cloud service", in Proc. of the 1st International Conference on Cloud Computing (CloudCom'09), Beijing, China. Springer-Verlag, 2009, pp. 301–311.
- [25] Brandon Heller, Srini Seetharaman, Priya Mahadevan, Yiannis Yiakoumis, Puneet Sharma, Sujata Banerjee, Nick McKeown, "ElasticTree: Saving Energy in Data Center Networks", NSDI'10 Proceedings of the 7th USENIX conference on Networked systems design and implementation, ACM Digital Library, USENIX Association Berkeley, CA, USA, 2010.
- [26] J. Kennedy and Russell C Eberhart, "Swarm intelligence", Vol. 78, Springer, 2006.
- [27] Chen Hua-sha, Shen jiajie, "Based on PSO cloud computing server points location searching", International Workshop on Cloud Computing and Information Security (CCIS 2013), Atlantis Press, 2013.
- [28] Durairaj M, Sivagowry S, "An Intelligent Hybrid Quick Reduct Particle Swarm Optimization Algorithm for Feature Reduction in Cardiac Disease Prediction", International Journal of Emerging Technologies in Computational and Applied Sciences, 12(2), March-May 2015, pp. 163-173.