Evolutionary Intelligence (2021) 14:381–387
https://doi.org/10.1007/s12065-019-00233-6

SPECIAL ISSUE

Genetic algorithm for quality of service based resource allocation in cloud computing

Prasad Devarasetty[1] · **Satyananda Reddy**[1]

Received: 16 January 2019 / Revised: 9 March 2019 / Accepted: 3 April 2019 / Published online: 16 April 2019
© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract
In the recent years, cloud computing has emerged as one of the important fields in the information technology. Cloud offers different types of services to the web applications. The major issue faced by cloud customers are selecting the resources for their application deployment without compromising the quality of service (QoS) requirements. This paper proposed the improved optimization algorithm for resource allocation by considering the objectives of minimizing the deployment cost and improving the QoS performance. The proposed algorithm considers different customer QoS requirements and allocates the resources within the given budget. The experimental analysis is conducted on various workloads by deploying into the Amazon Web Services. The results shows the efficiency of the proposed algorithm.

Keywords Genetic algorithm · Quality parameters · Cloud · Cost · Execution time

1 Introduction

Cloud computing is a combination of resources such as computing servers, database servers and storage servers. Cloud provides these services to the clients through pay-as-you-use policy. The major organizations like Microsoft [1] and Google [2] utilizes the cloud services as a cost effective solution. On the other side, the development of web applications and deployment in the cloud causes many challenges such as increase in cost and execution time to the cloud service providers [3–5]. Resource allocation based on the customer application requirements is the major issue in the cloud. Different algorithms are proposed to provide the solution for resource allocation problem. Distinct properties of the resources like broad network access and on-demand services offered by cloud service providers' causes challenges to the customers for selecting the resources. Cloud service providers have the complex QoS policies and complex pricing strategy for their allocated resources. For instance, Amazon Web Services (AWS) [6] offers different type of services like, storage, computation, networking and database with different pricing models. Therefore, it is difficult to the customer to find the resources in their budget by considering QoS [3].

AWS OpsWorks [7], IBM Smart Cloud [8] and RightScale [9] are some of the tools used for the cloud management which can scale the services based on the customer requirements. These tools consider the application workload to scale the resources and also they are concentrated on reducing the deployment cost or workload demand but they are not concentrated on QoS performance [10–12]. The QoS performance is the major criteria in the perspective of customers. Misra et al. [12] elaborates on developing the algorithms to optimize the QoS performance along with the deployment cost using learning methods for cloud IaaS. In [13], the authors developed an optimization algorithm based on the objective function of minimizing the deployment cost and latency which is considered as QoS performance. In [14], the authors considered the factor of response time as the QoS to address the customer requirements.

In cloud data centers, the resource allocation is the major issue and commonly it is modelled as single objective function. Several models consider the QoS, deployment cost and energy consumption as the major objectives for problem formulation. In the proposed model, we considered the deployment cost and QoS performance as the major objectives to formulate the problem. The proposed algorithm uses branch and bound algorithm for finding the discrete solutions. The

✉ Prasad Devarasetty
pdevarasetty03@gmail.com

[1] Department of Computer Science and Systems Engineering, AU College of Engineering, Andhra University, Visakhapatnam, Andhra Pradesh, India

rest of the paper is organized as follows. Section 2 deals with the related work regarding the existing resource allocation methods. Section 3 explains about the problem formulation along with the resource allocation requirements. Section 4 deals with the optimization algorithm for resource allocation in cloud. Section 5 explains about the performance evaluation with respect to different workloads. Finally Sect. 6 concludes the research work.

2 Background

Cloud computing is the emerging platform that impacts the changes in the information technology. Resource allocation is the major concern that brings important research towards cloud computing. The enterprises are looking over the QoS requirements due to the increase in completion. The QoS is the major issue which should be considered at the time of allocating resources to the application. These papers address the two objectives such as deployment cost and QoS [15–21].

Many researchers addressed the issue of deployment cost in cloud. In [22], the author developed an algorithm for minimizing the deployment cost. In this algorithm, only one service provider is considered for deploying small scale and large scale applications. The results proved that the deployment cost is reduced up to 32% and the performance is reduced up to 5%. In [23], the authors considered the dynamic resource allocation and proposed the online resource allocation algorithm that reduced the deployment cost and resource rate. In [3], Huang et al. proposed the cloud model which considers the QoS requirements for service provisioning. The algorithm uses the directed acyclic graph for problem formulation. The algorithm showed the improvement in runtime and performance.

In [17], the authors adopted genetic algorithm for resource allocation and QoS requirements are also considered. But this algorithm has high runtime due to the problem complexity and also it requires pre-specified application requirements. In [19], the authors developed the dynamic programming algorithm for optimizing the QoS requirements and deployment cost. In this algorithm, the service level agreements and the response time are considered as the requirements for QoS. In [24], Nan et al. developed an algorithm to reduce the total cost of the application development. The QoS requirements considered for the application are response time of the database instance and computing instance [11, 27]. Hao Zheng et al. [28] developed an approach for hybrid energy-aware resource allocation whereas the authors of [29] details about multi-objective particle swarm optimization algorithm for Service allocation in the cloud environments.

3 Preliminaries

Cloud computing is responsible for providing many type of resources with different prices for executing applications in the cloud computing environment. The major issue in the cloud computing is selecting an exact combination of resources based on the requirements of the user is a crucial task. This section defines the problem formulation by examining the existing system. Figure 1 Shows the general architecture of resource scheduling in cloud computing.

3.1 Problem formulation

The problem is formulated for optimal allocation of cloud resources and it takes the input data from the resources offered by cloud service providers. Although, the service providers provide similar type of resources to the customer but they are different in terms of price range, QoS performance and service type. The service providers display all the information about the resources required by the customer. The detailed information is given in Table 1.

The customers has to pay the price for the utilization of computing instance (X_c), database instance(X_{db}), IO requests (X_{IO}), storage space (X_s). In this research work, we are considered only the single cloud service provider. The rate of computing instance and the rate of the database also considered for the cloud based on the service level agreements. This problem statement also includes the requirements of the customers such

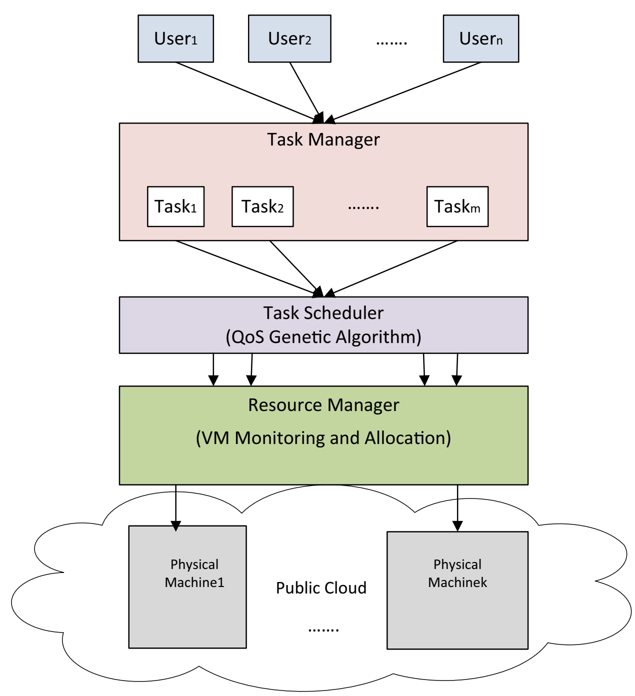

Fig. 1 Architecture of resource scheduling in cloud computing

Table 1 Information about the cloud resources

Notation	Description
X_c	Cost of computing instance
X_s	Cost of storage in database instance
X_{db}	Cost of database instance
X_{IO}	Cost of IO request capacity
R_c	Rate of computing instance (no. of request handling per hour)
R_{db}	Rate of database instance

Table 2 Customer requirements for application deployment in cloud

Notation	Description
T_D	Total time for deployment of application
E_C	Expected computing demand
E_{db}	Expected database demand
$ResT_{db}$	Response time of the database instance
$ResT_c$	Response time of the computing instance
ReqS	Required storage for application
B_{max}	Maximum budget for deployment of application

as QoS and budget to deploy the application. In Table 2, the customer requirements are presented.

The customer application deployment in the cloud requires different type of resources at different levels. Therefore, the customer has to purchase the resources to satisfy the application requirements. The resources which will be charged for the customer are given below.

N_c: Number of computing instances used by the application
N_{db}: Number of database instances used by the application
N_s: Volume of storage required by each database instance
M_{Rc}: Minimum service rate offered by computing instance
M_{Rdb}: minimum service rate offered by database instance

The proposed algorithm is concentrated on developing the best combination of cloud resources. Therefore, the proposed algorithm optimizes the cloud resource variables.

Variable 1: Cost of the data base instance with respect to service time T_D is given as

$$V_1 = N_{db} \times (X_{db} + X_s \times N_s) \times T_D \tag{1}$$

Variable 2: Cost of the data base server with respect to service time T_D is given as

$$V_2 = N_c \times X_c \times T_D \tag{2}$$

Variable 3: Cost of the Storage server with respect to service time T_D is given as

$$V_3 = (M_{Rdb} + M_{Rc}) \times X_{IO} \times T_D \tag{3}$$

The deployment cost of the application into the cloud is given as follows and hence the waiting time of the resources allocation depends the cloud resource variables.

$$\delta = V_1 + V_2 + V_3 \tag{4}$$

3.2 Cost optimization

The resource allocation problem is formulated by considering the total deployment cost along with customer requirements in the Data Center (DC) as the problem constraints. The QoS based cost optimization is given below:

Constrain 1: $\underset{\{N_{db},N_c,N_s,N_{Rdb},N_{Rc}\}}{Min} DC$ such that $DC \leq B_{\max}$

Constrain 2: $\underset{\{N_{db},N_c,N_s,N_{Rdb},N_{Rc}\}}{Min} DC$ such that $T_{db}(M_{Rdb}) \leq \mathrm{Res}T_{db}$

Constrain 3: $\underset{\{N_{db},N_c,N_s,N_{Rdb},N_{Rc}\}}{Min} DC$ such that $T_c(M_{Rc}) \leq \mathrm{Res}T_c$;

Constrain 4: $\underset{\{N_{db},N_c,N_s,N_{Rdb},N_{Rc}\}}{Min} DC$ such that $N_c \times R_c \geq E_c$;

Constrain 5: $\underset{\{N_{db},N_c,N_s,N_{Rdb},N_{Rc}\}}{Min} DC$ such that $N_{db} \times R_{db} \geq E_{db}$;

Constrain 6: $\underset{\{N_{db},N_c,N_s,N_{Rdb},N_{Rc}\}}{Min} DC$ such that $1 \leq N_c$

Constrain 7: $\underset{\{N_{db},N_c,N_s,N_{Rdb},N_{Rc}\}}{Min} DC$ such that $1 \leq N_{db}$

Constrain 8: $\underset{\{N_{db},N_c,N_s,N_{Rdb},N_{Rc}\}}{Min} DC$ such that $N_{db} \times N_s \geq \mathrm{Reg}_s$

Constrain 9: $\underset{\{N_{db},N_c,N_s,N_{Rdb},N_{Rc}\}}{Min} DC$ such that $\min_s \leq N_s \leq \max_s$

The problem constraint 1 considered the customer budget for deploying the application. In problem constraints 2 and 3, represents the response time of the database servers and computing servers. In problem constraints 4, 5, 6 and 7 represents that the application has to utilize at least one database and one computing instance. In problem constraint 8 and 9 represents the storage requirements for the application.

4 QoS aware GA based resource allocation in cloud

In the proposed algorithm, many factors should be considered simultaneously to solve the problem. The main goal of the proposed algorithm is to balance the QoS performance and the deployment cost of the application at the time of resource allocation.

4.1 Requirements for QoS performance

To deploy the customer satisfied application into the cloud, several factors need to be considered. Those factors are response time of the database instance and response time of the computing instance. The response time of the database majorly depends on the expected demand (E_{db}). The database demand is modelled using passion distribution process. Equation 5 shows the response time of the database instance of an application with a service rate $M_{Rdb.}$

$$T_{db}(M_{Rdb}) = \frac{(M_{Rdb})^{-1}}{1 - \frac{E_{db}}{M_{Rdb}}} \tag{5}$$

The response time of the database instance should not be less than T_{db}. The T_{db} is the maximum value specified by the customer for response time of the database instance.

$$T_{db}(M_{Rdb}) \leq \mathrm{Res}T_{db} \tag{6}$$

Another major issue to deploy the application into the cloud is the response time of the computing instance. Assume that M_{Rc} is the minimum response time of the computing instances and the expected arrival rate of the application request to the computing instance are given as E_c. Equation 7 shows the model for response time of the computing instance.

$$T_c(M_{Rc}) = \frac{(M_{Rc})^{-1}}{1 - \frac{E_c}{M_{Rc}}} \tag{7}$$

The response time of the computing instance should not be less than the T_c. The T_c is the maximum response time set by the customer.

$$T_c(M_{Rc}) \leq \mathrm{Res}T_c \tag{8}$$

4.2 Resource allocation model

The two main objectives considered for the algorithm is reducing the deployment cost and improving the QoS performance. The mathematical model for resource allocation is as follows:

$$\text{Deployment Cost}: \quad \text{Minimize}\ \delta$$
$$\text{QoS performance}: \quad \text{Maximize}\ T_{db}(M_{Rdb}),\ T_c(M_{Rc}) \tag{9}$$

4.3 Algorithm for cloud resource allocation

The output of the proposed algorithm is guaranteed to be best optimal solution. However, the optimal solutions are continuous. In general, the cloud service providers offer discrete number of computing instances and database instances. Algorithm 1 is developed to find the suitable cloud resources based on the customer application requirements. Here, we considered the Genetic algorithm [26] for finding the optimal solutions. Genetic algorithm is considered as one of the suitable approach for solving the optimization problems which includes resource allocation.

4.3.1 Chromosome representation

In the chromosome representation, we considered the customer QoS requirements along with the available Resources. Equation 10 shows the chromosome representation in genetic algorithm.

$$\lambda(i,j) = Q_i(R_j) \quad \text{Where i} = 1 \ldots \text{n and j} = 1 \ldots \text{m} \tag{10}$$

Where λ(i,j) is the chromosome representation, 'i' is the QoS requirements and j is the available resources.

4.3.2 Initialization of the population

In genetic algorithm [26], population initialization is the major factor involved and the suitable population results in the fine-tuned outcome. Algorithm 1 shows the population initialization in genetic algorithm.

```
Algorithm 1: Population initialization
Input: Customer QoS requirements Q
Output: Suitable Resources R
 Begin
Step 1: for i in 1 to n do
Step 2:     for j in 1 to n do
Step  3:       if (i<=j) do
 Step 4:            Compute Deployment cost of the application by using Eq. 4
                  End if
 Step 5:              if ( δ> threshold) then
 Step 6:                 R = R- Rj
 Step 7:               Otherwise
Step 8:                 Compute  Eq. 5 and Eq. 7
Step 9:                  end if
Step 10:      End for
Step 11: End for
End
```

4.3.3 Fitness function

The fitness function is calculated for each chromosome which was initialized. This fitness function enhances the quality of the solution. The objective function of the resource allocation mechanism is to maximize the response time of the resources and reduces the deployment cost of the application. The fitness function of the two objectives is shown in Eq. 11.

$$f = w_1(\min(\delta)) + w_2(\max((T_{db}(M_{Rdb}), (T_c(M_{Rc})))) \tag{11}$$

Where w_1 and w_2 denotes the priority of the objectives.

4.3.4 Cross over and mutation

Initially, two chromosomes are selected based on the random value G ϵ {0, 1}allocated for each chromosome. Cross over mechanism is applied for the chromosomes which are selected with the probability P_c random number G ∈ {0,1} assigned to each chromosome. If $G < P_c$ then, the chromosomes follows the one point crossover. The low mutation level is maintained for the optimal results. Here, one single bit is position is changed in the chromosomes for performing the mutation.

5 Performance comparison

The proposed algorithm is developed with C++ that is run on Windows 10 OS with Intel Core i7, 1.7 GHz and 16 GB of RAM. The two workload benchmarks RUBiS [25] and SPEC [25] are considered for the evaluation. Table 3 shows the detailed characteristics of the two workloads.

The Amazon Web Services (AWS) [6] is considered as cloud service provider for deploying the workloads and the pricing is given in Table 4. The price of the resource is charged based on the usage.

The performance of the proposed model is compared with the QCost model [10]. The results are given in Table 5. From the table, it is observed that the proposed algorithm outperforms the existing algorithm. The deployment cost of the proposed model is less when compared to the QCost in both workloads.

The performance of the QoS genetic algorithm (QGA) is compared with the QCost algorithm [10] and Conventional algorithm [10]. The proposed algorithm considered the response time of the database, response time of the computation instance and deployment cost are the performance metrics considered for evaluation. Figure 2 shows the response time of the database instance. By observing the figure, it is proved that the proposed algorithm is efficient in reducing the response time of the database instance compared to the QCost and conventional algorithms. Figure 3 shows the response time of the computational instance. Though, proposed algorithm response time is less compared to other algorithms, it is slightly expensive compared to the QCost. Figure 4 shows the deployment cost of the proposed and existing algorithms. It is prove that, the proposed algorithm is efficiently balanced the trade-off between the QoS performance and the deployment cost.

Table 3 Input parameters of the proposed algorithm

Workload	Storage (GB)	E_{db} (1/h)	E_c (1/h)	$ResT_{db}$ (s)	$ResT_c$ (s)
SPEC 10	760	3121	1854	8.92	10.25
SPEC 20	760	3245	1945	15.41	12.46
SPEC 40	760	3286	1989	18.54	15.91
RUBiS 800	320	1452	398	9.86	4.52
RUBiS 1600	320	1682	989	11.23	7.31
RUBiS 2400	320	1935	1254	15.26	9.88

Table 4 AWS parameters

Parameter	Value
X_c ($)	0.02
X_s ($/h*GB)	0.0003
X_{db} ($)	0.15
X_{IO} ($)	0.1000
R_c (1/h)	4434.1
R_{db} (1/h)	5285.5

6 Conclusion

The major challenge faced by the cloud environment is resource allocation by satisfying the customer QoS requirements with in the budget. The customers can go through the cloud service providers to choose the optimal resources for their application deployment. Selecting the optimal resource is a big task to the customers with in the complex pricing structure. This paper proposed an optimal resource allocation method by considering the deployment cost and QoS requirements as the major objectives. The proposed

Table 5 Results obtained for the proposed model and QCost [10]

Workload	QCost [10]			QGA		
	DC ($)	T_{db} (s)	T_c (s)	DC ($)	T_{db} (s)	T_c (s)
SPEC 10	0.92	9.8	11.8	0.81	1.23	2.84
SPEC 20	1.84	17.2	14.2	1.24	2.54	3.15
SPEC 40	2.93	21.3	18.9	2.38	3.26	4.11
RUBiS 800	0.81	9.6	4.2	0.39	1.48	0.98
RUBiS 1600	0.84	12.4	7.6	0.78	2.10	1.12
RUBiS 2400	1.24	13.2	9.5	1.10	2.25	1.32

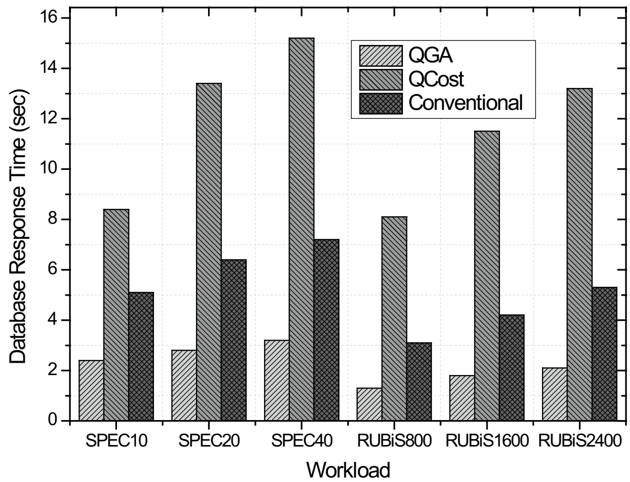

Fig. 2 Response time of the database instances with respect to QGA, Qcost and Conventional method

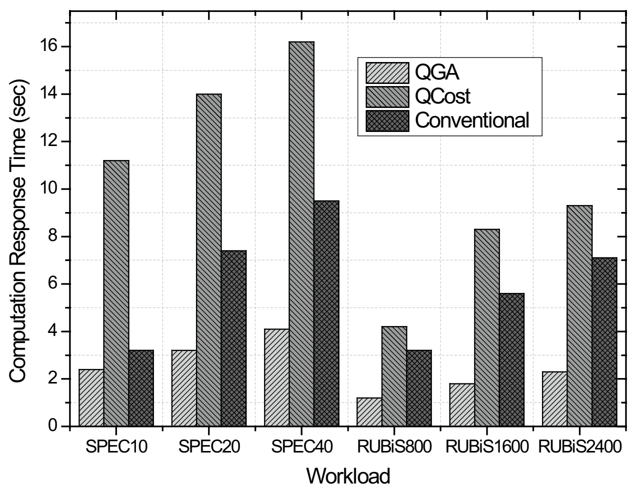

Fig. 3 Response time of the computing instance QGA, Qcost and Conventional method

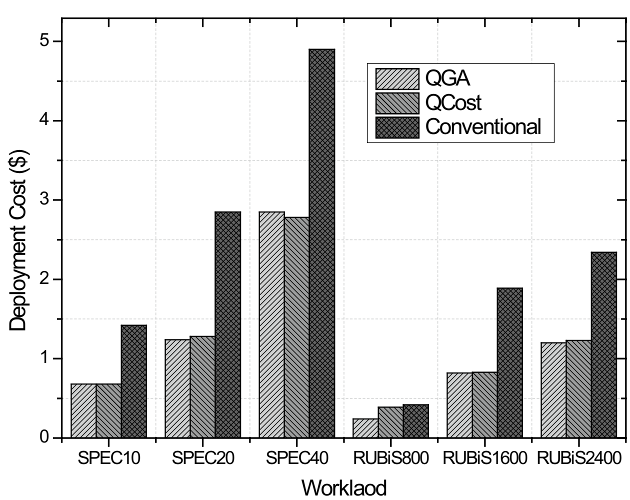

Fig. 4 Deployment cost of the QGA, QCost and Conventional model

algorithm uses the genetic algorithm for finding the best discrete solution to the problem. The experimental results are conducted over different workloads using the real time cloud service provider. The results proved that the proposed algorithm is efficient in balancing the trade-off between the deployment cost and QoS performance.

References

1. https://azure.microsoft.com, Microsoft. Accessed 11 Jan 2018
2. https://cloud.google.com, Google. Accessed 11 Jan 2018
3. Hu X, Ludwig A, Richa A, Schmid S (2015) Competitive strategies for online cloud resource allocation with discounts: the 2-dimensional parking permit problem. In: Proceedings of IEEE 35th international conference on distributed computing systems (ICDCS), June 2015, pp 93–102
4. Serrano D, Bouchenak S, Kouki Y, Oliveira FA Jr, Ledoux T, Lejeune J, Sopena J, Arantes L, Sens P (2016) SLA guarantees for cloud services. Fut Gener Comput Syst 54:233–246
5. Fan G, Yu H, Chen L (2016) A formal aspect-oriented method for modeling and analyzing adaptive resource scheduling in cloud computing. Proc IEEE Trans Netw Serv Manag (TNSM) 13(2):281–294
6. https://aws.amazon.com, Amazon. Accessed 12 Dec 2017
7. https://aws.amazon.com/opsworks, amazon. Accessed 12 Dec 2017
8. http://www-03.ibm.com/software/products/en/category/it-servicemanagement, IBM IT service management. Accessed 12 Dec 2017
9. http://www.rightscale.com, rightscale. Accessed 12 Dec 2017
10. Mireslami S, Rakai L, Wang M, Far BH (2015) Minimizing deployment cost of cloud-based web application with guaranteed QoS. In: Proceedings of the 2015 IEEE global communications conference (GLOBECOM), Dec 2015, pp 1–6
11. Nagaraju D, Saritha V (2017) An evolutionary multi-objective approach for resource scheduling in mobile cloud computing. Int J Intell Eng Syst 10(1):12–21
12. Misra S, Krishna PV, Kalaiselvan K, Saritha V, Obaidat MS (2014) Learning automata-based QoS framework for cloud IaaS. IEEE Trans Netw Serv Manag 11(1):15–24
13. Dastjerdi A, Garg S, Buyya R (2011) QoS-aware deployment of network of virtual appliances across multiple clouds. In: Proceedings of the third IEEE international conference on cloud computing technology and science (CloudCom), Athens, Greece, 29 Nov–1 Dec 2011, pp 415–423
14. Rajeshwari BS, Dakshayini M (2015) Optimized service level agreement based workload balancing strategy for cloud environment. In: Proceedings of IEEE international advance computing conference (IACC), June 2015, pp 160–165
15. Shi H, Zhan Z (2009) An optimal infrastructure design method of cloud computing services from the BDIM perspective. In: Proceedings of the second Asia-Pacific conference on computational intelligence and industrial applications (PACIIA), vol 1, Wuhan, China, 28–29 Nov 2009, pp 393–396
16. Yang Z, Liu L, Qiao C, Das S, Ramesh R, Du AY (2015) Availability aware energy-efficient virtual machine placement. In: Proceedings of IEEE international conference on communications (ICC), June 2015, pp 5853–5858
17. Huang J, Liu Y, Duan Q (2012) Service provisioning in virtualization based cloud computing: modeling and optimization.

In: Proceedings of IEEE global communications conference (GLOBECOM), Dec 2012, pp 1710–1715
18. Chaisiri S, Lee B, Niyato D (2012) Optimization of resource provisioning cost in cloud computing. IEEE Trans Serv Comput 5(2):164–177
19. Goudarzi H, Ghasemazar M, Pedram M (2012) SLA-based optimization of power and migration cost in cloud computing. In: Proceedings of the 12th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGrid), Ottawa, ON, 13–16 May 2012, pp 172–179
20. Feng M, Wang X, Zhang Y, Li J (2012) Multi-objective particle swarm optimization for resource allocation in cloud computing. In: Proceedings of IEEE 2nd international conference on cloud computing and intelligent systems (CCIS), Oct 2012, vol 03, pp 1161–1165
21. Moorthy RS (2015) An efficient resource allocation (era) mechanism in IAAS cloud. In: Proceedings of international conference on advances in computing, communications and informatics (ICACCI), Aug 2015, pp 412–417
22. Nir M, Matrawy A, St-Hilaire M (2014) An energy optimizing scheduler for mobile cloud computing environments. In: Proceedings of IEEE conference on computer communications workshops (INFOCOM WKSHPS), April 2014, pp 404–409
23. Aniceto IS, Moreno-Vozmediano R, Montero R, Llorente I (2011) Cloud capacity reservation for optimal service deployment. In: Proceedings of the second international conference on cloud computing, GRIDs, and virtualization (CLOUD COMPUTING), Rome, Italy, 25–30 Sept 2011, pp 52–59
24. Nan X, He Y, Guan L (2011) Optimal resource allocation for multimedia cloud based on queuing model. In: Proceedings of the 13th IEEE international workshop on multimedia signal processing (MMSP), Hangzhou, China, 17–19 Oct 2011, pp 1–6
25. Ersoz D, Yousif M, Das C (2007) Characterizing network traffic in a cluster-based, multi-tier data center. In: Proceedings of international conference on distributed computing systems (ICDCS), 2007, p 59
26. Ye Z, Zhou X, Bouguettaya A (2011) Genetic algorithm based QoS-aware service compositions in cloud computing. In: International conference on database systems for advanced applications. Springer, Berlin, pp 321–334
27. Vankadara S, Dasari N (2019) Energy-aware dynamic task offloading and collective task execution in mobile cloud computing. Int J Commun Syst. https://doi.org/10.1002/dac.3914
28. Zheng H, Feng Y, Tan J (2017) A hybrid energy-aware resource allocation approach in cloud manufacturing environment. IEEE Access 5:12648–12656
29. Sheikholeslami F, Navimipour NJ (2017) Service allocation in the cloud environments using multi-objective particle swarm optimization algorithm based on crowding distance. Swarm Evolut Comput 35:53–64

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.