Energy Efficiency of Mobile Edge Computing Systems

By:

Raisa Fariha

Md. Ziad Karim

Sheikh Faiyaz Mahamud

Supervised by:

Prof. Muhammad Mahbub Alam PhD
Professor
Department of Computer Science and Engineering
Islamic University of Technology

Department of Computer Science and Engineering
Islamic University of Technology (IUT)
04 March, 2021

Declaration of Authorship:

This is to certify that the work presented in this thesis is the outcome of the analysis and simulations carried out by Raisa Fariha, Md. Ziad Karim and Sheikh Faiyaz Mahamud under the supervision of Prof. Muhammad Mahbub Alam PhD, Professor of Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

Raisa Fariha

Raisa Fariha

Student ID – 160041049

Ziad Karim

Md. Ziad Karim

Student ID – 160041046

Sheikh Faiyaz Mahamud

Student ID – 160041009

Supervisor:

Prof. Muhammad Mahbub Alam PhD

Professor

Department of Computer Science and Engineering

Islamic University of Technology (IUT)

Abstract:

Quality of Service in case of IoT devices like Mobile Edge Computing Systems widely depends on the way the offloading of applications is done and how the devices are placed in the system. In case of Mobile Edge Computing Systems, the users are constantly moving so it is very important to find an optimal placement arrangement for them. But due to continuous movement, any optimal placement might even turn into an inefficient one within minutes. So, it is very important to design the placement of the applications keeping in mind the dynamics of the whole system. Again, the energy consumption done by the server in a MEC is an integral part while calculating the cost of services of the system.

That is why in our paper we will address the problem of the optimal placement of the devices in a MEC as a multi-stage stochastic program. Our main goal will be to improve the Quality of Services as much as possible. Each of the mobile edge servers has a specific energy budget which we will also take into account. We have designed an algorithm to solve this problem and we will perform an experimental analysis to evaluate the performance of our designed algorithm.

Contents:

Chapter 1:

Introduction:

1.1 Mobile Edge Computing System (MEC)

In different aspects of our day-to-day lives such as communication, connection, e-commerce, health we extensively use mobile devices. This has increased to such a great extent that it has one of the primary means for these aspects of communication. But this also arises the necessity of identifying the limitations that mobile devices have. These limitations include the capacity of the memory disk, battery, processing power, and so on [1]. To eradicate these issues and limitations a system called 'Mobile Cloud Computing' was developed which was successful in removing most of the issues of the mobile devices [2], [3]. MCC helped to remove the issues of mobile devices by enabling them to run their applications on servers from clouds. The performance measure of this Mobile Cloud Computing was adversely affected because of the inconsistency of the distance between the cloud servers and mobile users.

To remove this problem of inconsistency of the MCC, the Mobile Edge Computing system was introduced to reduce the long response time between mobile applications [4], [5], [6]. In case of MEC, the mobile users are provided with services from the mobile edge servers with lower latency. Thus, MEC has become a popular infrastructure for so many applications in our daily lives such as machine learning, Internet of Things (IoT) [7], [8].

Now MEC also has its fair share of challenges as well. This includes the efficient placement of the applications on the edge server. The inefficient placement of the applications ultimately results in poor Quality of Service (QoS).

In the present times with the rapid increase of population and increased energy requirements has led the researchers to learn more and become aware of the reduction of energy consumption. Especially this awareness has been more and more in practice in case of distributed systems. In case of MEC systems, while we are planning for the optimal placement of the applications in the edge servers, it is extremely important to keep in mind the energy consumption done by the servers. As the cost per operation in case of edge servers is dependent on the placement of the applications, it is expected to be operated with a higher operating cost. The energy consumption that occurred accounts for almost 25% of the operating cost of the data centres [9]. This way we can see that by optimizing the energy consumption we can also optimize the operating costs to a great extent.

It is very important to take care of the energy consumption in an edge-computing system. For that, we need to utilize the physical resources wisely and efficiently. This makes the efficient application placement in an edge-computing system an important factor in case of optimizing cost and energy consumption. The efficient placement of the applications problem can be shown as below:

If we have a n=set of edge servers and a set of user requests who wants to execute specific applications, we will assign the applications in such a way that the Quality of Service (QoS) for all the users is maximized. The energy budget of the servers, the availability of resources and the movement of the users are considered in this case.

Now to develop an optimized application placement problem, considering the movement of the users in the system should be considered the most. But as the users are constantly moving across the network from time to time, any efficient placement application can turn into an inefficient one within minutes. On the other hand, frequent movement of the applications to various servers depending on the location of the users is not efficient. As it is hard to determine the future locations of the users, the efficient placement mechanism for the applications must consider this uncertainty.

We will consider a multi-stage stochastic programming method for the energy-aware application placement problem in MEC systems [10]. We will mainly focus on maximizing the Quality of Service (QoS) of the Mobile Edge Computing System and efficient placement of the applications by considering the limited energy budget of the energy edge servers and dynamics of the network. We will consider the Sample Average Approximation (SAA) to solve this problem [10].

1.2 Related Works:

Though there exist several ways of solving the application placement problem in case of cloud computing [11], [12], we cannot directly apply them in case of Mobile Edge Computing Systems (MEC). Because, in case of MEC the users are mobile all the time and they move from one place to another after the initial placement of the applications. So, it is very difficult to make an optimal arrangement of application placement because the user's mobility may make it last. So, any placement setting has to be adaptive to the dynamic setting.

In case of performance metrics, cost of services [13], [14], [15], energy consumption [16], [17], [18], [19] and Quality of Service (QoS) [20], [21] are the most important metrics to be considered. In order to consider the dynamic nature of the MEC systems, which stem mainly from the user's mobility many researchers came up with methods that do no require any details about the future location of the users. [22], [23].

In case of edge computing systems there are many nondeterministic parameters that has to be taken into account in order to identify the efficient application placement mechanism. As we know due to the dynamic nature of the users in the system, any optimal placement can turn into an inefficient one within minutes. Many researchers have used the Markov Decision Process (MDP) to model the uncertainties in case of application placement in Mobile Edge

Computing Systems [24], [25]. Most variants of the problems involving MDP are known to be P-complete, it is not possible to design highly efficient parallel algorithms to solve them unless all problems in P have such highly efficient parallel solutions [26].

1.3 Our Contributions:

This work is basically an extension of the work done on a stochastic task placement optimization in Mobile Edge Computing Systems [27]. There they developed a multi-stage stochastic programming process for optimized application placement in MEC systems. The main objective was to minimize the relocation cost, computation cost, communication cost, and overall total cost. Here, our main objective is to maximize the total Quality of Service of the MEC System. The Quality of Service depends on the latency, which means that the QoS is directly proportional to the distance between the request of the user and the server. The QoS is inversely proportional to the distance between the user and the server that processes the request. All these are done by keeping the limited energy budget of the servers in mind. Here we will consider an energy-aware application placement problem in edge computing systems as a multi-stage stochastic program where a time slot is shown and the location of users might change between time slots. The stochastic programming approach is considered because it becomes easy to consider the dynamicity of the users in the system. Because, in case of MEC Systems, the location of the users changes frequently which might turn an optimized placement of application into an inefficient one within few minutes.

So, our goal here is to consider the metrics of QoS in a system and keeping the problem of higher relocation cost in mind. For the relocation cost, we consider a recourse function, which will help both maximizing the QoS of a system and at the same time keeping the relocation cost minimum.

Here the main constraint is the limited energy budget in case of the servers because no matter how many requests are reaching the server, it cannot exceed its budgeted energy. As talked

before about the recourse function, its calculation might be a bit difficult because a very large number of scenarios has to be considered in this case [28].

For the multi-stage stochastic application placement problem, the use of the Sample Approximation Algorithm has been done here. Here a lot of scenarios with many subproblems are considered and in the case of multi-stage stochastic application placement problems, they can be solved faster with the SAA-based algorithm. A greedy approach is used to solve all the phases of the SAA-based algorithm. The experimental analysis is done with the help of real-world data.

1.3 Motivation:

With the continuous increase of IoT based and mobile applications, the energy requirement is increasing day by day. This is leading to an energy crisis all over the world. This makes it more important to save mankind by conservation of energy as much as possible. MEC reduces the response time of mobile applications removing the barriers of cloud computing. Aim to gain an optimum situation with minimum amount of wastage of energy. This will ultimately result in gaining an upper hand while operating the IoT based applications with a much lower energy consumption and cost.

Chapter 2:

System Model and Assumptions:

2.1 The Energy-Aware Application Placement Problem in MEC: Multi-Stage Stochastic Programming Formulation:

Here the Mobile Edge Computing Systems typically contains of a set of edge servers (M) denoted by $S = \{S_1, S_2, ..., S_M\}$, which is dedicated to providing service to a set of users (N), denoted by $U= \{U_1, U_2, ..., U_N\}$. The network model for the MEC system is shown below (Figure 1). The model is analogous to the network model but it doesn't include any cloud servers. In the network model, the edge servers are co-located with a base station and have a specific computational capacity. This computational capacity denoted by Q_j means the maximum number of unit-size containers that can be accommodated by a server. These unit-size containers have a fixed resource configuration that is set by the provider. When a user requests executing an application with the help of 4G/5G/Wi-Fi access networks on the servers. All the towers in the system are accessible to all the users. The user's request (U_i) has a size denoted by R_i which is expressed in terms of the unit size containers are required to execute the application completely. The time required to complete the execution is also specified by the user on any specific container with size R_i. One container will serve the request that came from one user.

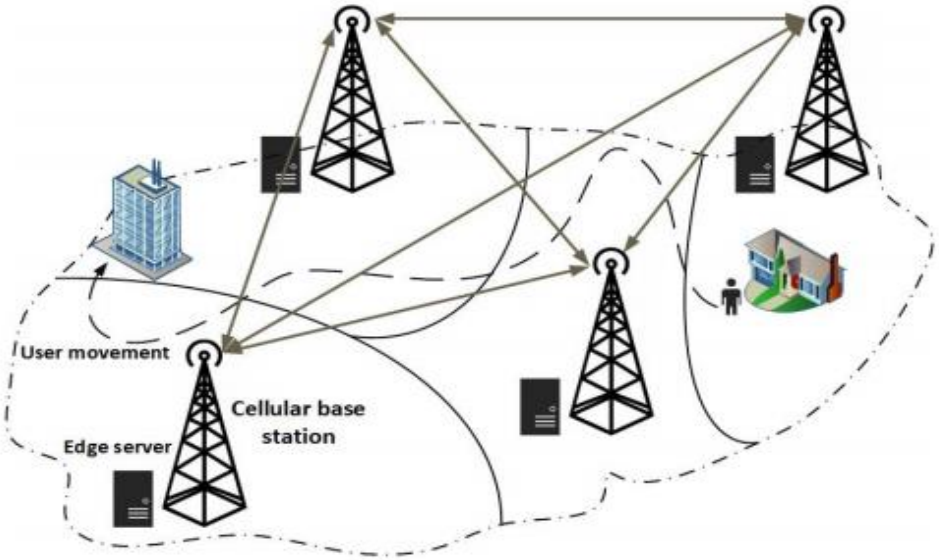


Fig. 1: MEC Network model.

The application placement decisions are made at periodic intervals known as time slots. The location of the users is usually specified with the help of a two-dimensional grid of cells. The users can usually from one cell to any other neighbouring cells within a time slot, which means the location of the users will change between two slots not instead of within a time slot. The main goal is to maximize the Quality of Service of the system keeping in mind the limited computational capacities of the servers and their energy budget. The problem here is represented as a multi-stage stochastic program.

The main objective of the multi-stage stochastic application placement problem is to maximize the QoS as much as possible. This QoS depends on the individual QoS of the users individually who receive services taking their relocation cost into account. The Quality of Service of the system here is determined with the help of two important factors which are used to calculate the latency. The term latency stands for denoting the distance between the user and the servers and the size of the user's request. If the distance between the user and the server is less, then the system is expected to have low latency and vice-versa. So, the QoS of a server that executes a user's request is:

$$QoS_{ij}^{t} = \frac{\gamma \cdot R_i}{d_{ij}^{t}}$$

(distance between Ui and Sj in time slot t)

(γ is a regulatory parameter)

Here, d^{t}_{ij} denotes the distance between the user and the server within a specific time slot along with a regulatory parameter. So, the QoS that a user receives from a server within a specific time slot is inversely proportional to the distance between the user and the servers executing the user's request.

The distance between the servers and users is the Manhattan distance. For example, if the user is located in a cell (x_1, x_2) and the server is located in the cell (x_3,x_4) then the distance between the user and the server is $|x_1-x_3| + |x_2-x_4|$.

The cost that is incurred when the assignment of the user's request is changed from one server to another during execution is called the relocation cost. This means if the execution of any request is completed in one server, then it is not required to shift or change the assignment of the request, therefore, no relocation cost is incurred. This cost is directly proportional to the distance between the servers and the users. The location of the users at the beginning of a time slot can be known. The future location of the users in future time slots cannot be determined.

If the limited energy budget of a server is denoted by E, then we can say that the total energy consumption done by a server while processing a request is:

$$\epsilon_{ij} = \frac{\sigma \cdot R_i}{Q_j}$$

(σ is a regulatory constant coefficient)

From the above equation, we can see that the total energy consumption that occurred is directly proportional to the utilization of computational resources. Each server differs from the other with respect to energy budget and computation capacity.

If we consider the given two edge computing systems, we can see that the application placements are different for both cases. Numerous servers are represented by a rectangle, the radius of the circles representing the size of the user requests. The line that connects the user with the server shows the placement of the user-requested application on the edge server. Here the servers are assumed to be identical with the respect to their energy budget and computation capacities. Let's assume that each server has an energy budget of 10 units

and a capacity of 30 containers. The requests of users range from 2 to 10 containers. We also set γ = 10 and σ = 20. The Average Utilization Energy of the system is:

$$\mathrm{AEU} = \frac{\sum_{i \in \mathcal{U}} \sum_{j \in \mathcal{S}} \frac{\epsilon_{ij}}{E_j} \cdot x_{ij}}{M}$$

X_{ij} is a binary variable which is 1 if the request of user Ui is assigned to server S_j, and 0, otherwise.

The energy-efficient application placement problem in the MEC system is represented as a multi-stage stochastic programming model here. The variables that denote the assignments of the user's request are called the first stage variable because they are connected with the current stage. All these first-stage variables are decided before the uncertain parameters are known. The objective is to maximize the sum of Quality of Service in the first stage and the expected objective value function.

The objective function is to maximize the total QoS in the first stage and minimize the recourse cost. The recourse cost is the difference between the expected value of the total QoS and the relocation cost in the upcoming stages. Some constraints are required to be considered. Like:

(1) The energy budget of the servers cannot be exceeded. This constraint is maintained by the limited energy budget that a server has.
(2) The user request is assigned to at most one server during each time slot.
(3) It has to be guaranteed that the processing of a user's request is not interrupted in the system.
(4) With every relocation, the decision variables are set accordingly.
(5) All the decision variables have to be in binary.

Our main goal is to solve the multi-stage stochastic program and obtain the expected value of the recourse costs. With the help of a multi-stage stochastic problem, a large number of scenarios are possible to be solved in a

very reasonable amount of time. So here we have used the Sample Average Approximation (SAA) method which is a Monte Carlo simulation-based approach, to find out a reliable estimation of the expected value of the recourse cost function.

Chapter 3:

Mechanism:

3.1 Proposed Mechanism:

The network model for the efficient energy-aware application placement is a multi-stage stochastic program with an integer recourse function. The SAA algorithm has been proved to be an efficient method for solving multistage stochastic programs. The basic idea is to approximate the recourse function by the sample average function. The samples are mutually exclusive and identically distributed containing a constant number of scenarios. This is applied in each time slot/stage.

Algorithm 1 SAA algorithm

{Executed every time slot t}
1: Generate H independent scenario samples, each of size L, ($k \in \{1, \ldots, H\}$)
2: **for** $k = 1$ **to** H **do**
3: Solve the deterministic equivalent problem corresponding to each sample.
4: **end for**
5: Generate a sufficiently large sample of size L', $L' \gg L$.
6: **for** $k = 1$ **to** H **do**
7: Evaluate the candidate solutions obtained in line (3) by solving the deterministic equivalent problem corresponding to the sample of size L', while the current stage variables are fixed to their values obtained in line (3).
8: **end for**
9: Out of H candidate solutions, choose the one that has the largest estimated objective value.

Algorithm 2 PG-SAA: Parallel Greedy SAA-based application placement algorithm

{Executed every time slot t}
Generate H independent scenario samples ($k \in \{1, \ldots, H\}$), each of size L.
for $k = 1$ **to** H **do in parallel**
  for $i = 1$ **to** N **do**
    for $\ell = 1$ **to** L **do**
      Create graph $G_i(V, E)$ for user U_i under scenario ℓ.
      Use Dijkstra's algorithm to find the shortest paths between vertices S_j^t and vertex D in graph $G_i(V, E)$ for scenario ℓ.
      $w_{ij}^{k,\ell} \leftarrow$ cost of the shortest paths.
    end for
    for $j = 1$ **to** M **do**
      $\bar{w}_{ij}^{k} = \frac{1}{L} \sum_{\ell=1}^{L} w_{ij}^{k,\ell}$
    end for
  end for
  Call G-MAP ($\bar{W}^k$) to obtain the placement.
  Record the optimal solution as ($\bar{X}^{k,t}, \bar{Y}^{k,t}$).
end for
Generate a sufficiently large sample of size L', $L' \gg L$.
for $k = 1$ **to** H **do in parallel**
  for $i = 1$ **to** N **do**
    for $\ell' = 1$ **to** L' **do**
      Create graph $G_i'(V, E)$ for user U_i under scenario ℓ'.
      Use Dijkstra's algorithm to find the shortest paths between vertex $S_{j_i^*}$ and dummy vertex D in graph $G_i'(V, E)$ for scenario ℓ'.
      $w_{ij}^{k,\ell'} \leftarrow$ cost of the shortest paths.
    end for
    for $j = 1$ **to** M **do**
      $\bar{w}_{ij}^{k} = \frac{1}{L'} \sum_{\ell'=1}^{L'} w_{ij}^{k,\ell'}$
    end for
  end for
  $Z^k \leftarrow$ objective value corresponding to sample k.
end for
$Z^* \leftarrow \max_{k \in \{1,\ldots,H\}} Z^k$
Allocate users' requests to servers according to $\bar{X}^{k,t}$ corresponding to Z^*.

3.2 Our Improved Mechanism:

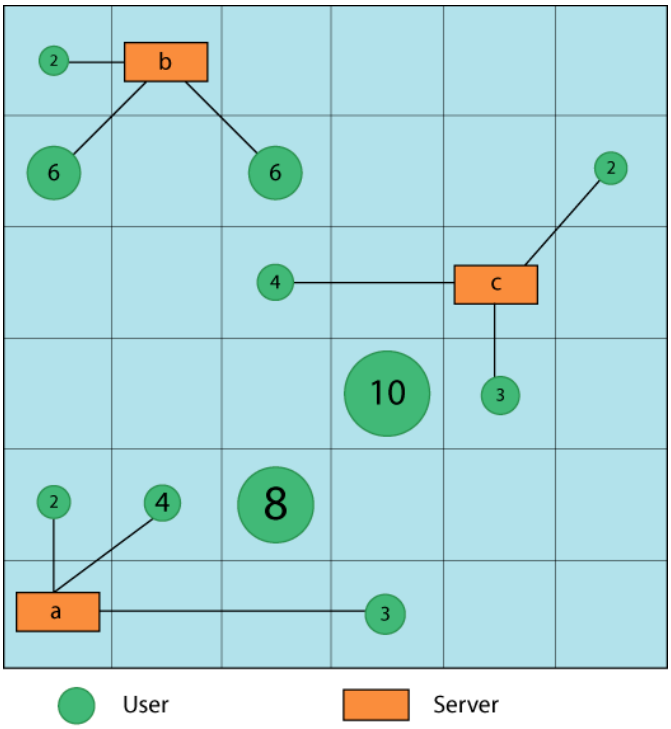
In case of the PG-SAA algorithm mentioned before, they treated all the servers equally. They didn't consider any server with a higher or lower number of hits, that means the servers which has the highest or comparatively lower number of user request assigned. That is why we can have a higher optimization by considering this case. In case of the PG-SAA algorithm mentioned before, we saw that they didn't consider any server differently with respect to their number of hits and energy consumption. So, by finding the weighted average in place of the normal average, the server that has the most hits, i.e., with the greatest number of queries, can be assigned with higher value of weight, so that the energy can be optimized more by reducing energy consumption done by the most hit server along with the other ones.

3.3 Experimental Setup:

In our target paper they have incorporated a real-world data set of cab mobility traces and they used Dell PowerEdge R740 Rock Server as the edge servers. Our experimental environment is fully C++, where we stored the outputs in different arrays and vectors and printed them in our console for further analysis. We randomly generated users and servers and used the same set of data for both the existing and our improved algorithm to perform comparative analysis.

Chapter 4:

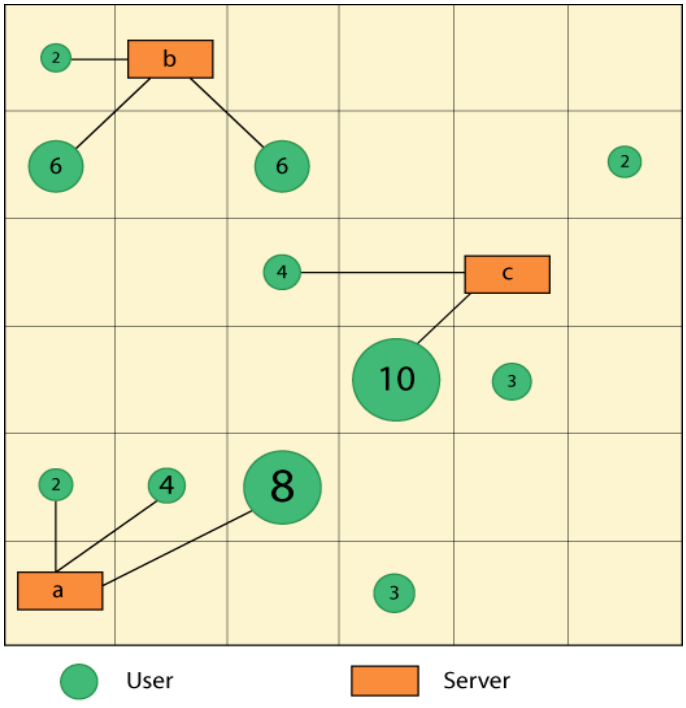
Performance Evaluation:



PG-SAA Algorithm:

AEU= 0.71

QoS= 190



PG-SAA Algorithm (Improved):
AEU= 0.93

QoS= 206.67

Chapter 5:

Conclusion:

We addressed the energy-aware application placement problem in mobile edge computing systems.

To take into account the mobility of users, a challenging issue in the resource management of MEC systems, we formulated the problem as a multi-stage stochastic program. The results of the experimental analysis showed that the proposed algorithm is able to obtain very good quality solutions for the problem. The proposed algorithm requires a very small execution time when executed on large multi-core systems, making it very suitable for deployment on current and future mobile edge computing systems.

Chapter 6:

Future Works:

Employ the proposed approach on the design of algorithms for other challenging stochastic problems in the management of mobile edge computing systems. Problems include the mobility of users, which is an important factor affecting the offloading and provisioning decisions. Employ a risk-based optimization approach to address the QoS in MEC systems considering the movement of users.

References:

1. M. Satyanarayanan, “Fundamental challenges in mobile computing,” in Proc. 15th ACM Symp. on Principles of Distrib. Comp., 1996, pp. 1–7

2. H. T. Dinh, C. Lee, D. Niyato, and P. Wang, “A survey of mobile cloud computing: architecture, applications, and approaches,” Wireless Comm. and Mobile Comp., vol. 13, no. 18, pp. 1587–1611, 2013.

3. F. Liu, P. Shu, H. Jin, L. Ding, J. Yu, D. Niu, and B. Li, “Gearing resource-poor mobile devices with powerful clouds: architectures, challenges, and applications,” IEEE Wireless Communications, vol. 20, no. 3, pp. 14–22, 2013

4. M. T. Beck, M. Werner, S. Feld, and S. Schimper, “Mobile edge computing: A taxonomy,” in Proc. 6th International Conference on Advances in Future Internet, 2014, pp. 48 – 54

5. B. Liang, “Mobile edge computing,” Key Technologies for 5G Wireless Systems, p. 76, 2017.] [S. Davy, J. Famaey, J. Serrat-Fernandez, J. L. Gorricho, A. Miron, M. Dramitinos, P. M. Neves, S. Latre, and E. Goshen, “Challenges ´ to support edge-as-a-service,” IEEE Communications Magazine, vol. 52, no. 1, pp. 132–139, 2014

6. Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “Mobile edge computing: Survey and research outlook,” arXiv preprint arXiv, vol. 1701, 2017

7. F. Cicirelli, A. Guerrieri, A. Mercuri, G. Spezzano, and A. Vinci, “Itema: A methodological approach for cognitive edge computing iot ecosystems,” Future Generation Computer Systems, vol. 92, pp. 189–197, 2019

8. Y. Liu, C. Yang, L. Jiang, S. Xie, and Y. Zhang, “Intelligent edge computing for iot-based energy management in smart cities,” IEEE Network, vol. 33, no. 2, pp. 111–117, 2019.

9. A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, “The cost of a cloud: research problems in data center networks,” ACM SIGCOMM Computer Comm. Rev., vol. 39, no. 1, pp. 68–73, 2008

10. Energy-Aware Application Placement in Mobile Edge Computing: A Stochastic Optimization Approach
11. M. Chowdhury, M. R. Rahman, and R. Boutaba, “Vineyard: Virtual network embedding algorithms with coordinated node and link mapping,” IEEE/ACM Transactions on Networking, vol. 20, no. 1, pp. 206–219, 2012.

12. D. Dutta, M. Kapralov, I. Post, and R. Shinde, “Embedding paths into trees: Vm placement to minimize congestion,” in European Symposium on Algorithms. Springer, 2012, pp. 431–442

13. T. Bahreini and D. Grosu, “Efficient placement of multi-component applications in edge computing systems,” in Proc. of the Second ACM/IEEE Symposium on Edge Computing, 2017, pp. 5:1–5:11

14. S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. K. Leung, “Dynamic service placement for mobile micro-clouds with predicted future costs,” IEEE Transactions on Parallel and Distributed Systems, vol. 28, no. 4, pp. 1002–1016, 2017.

15. S. Wang, M. Zafer, and K. K. Leung, “Online placement of multicomponent applications in edge computing environments,” IEEE Access, vol. 5, pp. 2514–2533, 2017

16. A. Al-Shuwaili and O. Simeone, “Energy-efficient resource allocation for mobile edge computing-based augmented reality applications,” IEEE Wireless Comm. Letters, vol. 6, no. 3, pp. 398–401, 2017

17. C. You, K. Huang, H. Chae, and B.-H. Kim, “Energy-efficient resource allocation for mobile-edge computation offloading,” IEEE Trans. Wireless Comm., vol. 16, no. 3, pp. 1397–1411, 2017

18. J. Guo, Z. Song, Y. Cui, Z. Liu, and Y. Ji, “Energy-efficient resource allocation for multi-user mobile edge computing,” in IEEE Global Communications Conference, 2017, pp. 1–7

19. J. Zhang, X. Hu, Z. Ning, E. C.-H. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, “Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks,” IEEE Internet of Things Journal, vol. 5, no. 4, pp. 2633–2645, 2018

20. S. Wang, Y. Zhao, L. Huang, J. Xu, and C.-H. Hsu, “Qos prediction for service recommendations in mobile edge computing,” Journal of Parallel and Distributed Computing, vol. 127, pp. 134–144, 2019

21. A. H. Sodhro, Z. Luo, A. K. Sangaiah, and S. W. Baik, “Mobile edge computing based qos optimization in medical healthcare applications,” Intl. J. of Information Management, vol. 45, no. 1, pp. 308–318, 2019

22. B. Gao, Z. Zhou, F. Liu, and F. Xu, “Winning at the starting line: Joint network selection and service placement for mobile edge computing,” in Proc. IEEE INFOCOM, 2019, pp. 1459–1467

23. Y. Sun, S. Zhou, and J. Xu, “Emm: Energy-aware mobility management for mobile edge computing in ultra dense networks,” IEEE J. on Selected Areas in Comm., vol. 35, no. 11, pp. 2637–2646, 2017

24. S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, “Dynamic service migration in mobile edge-clouds,” in IFIP Networking Conference, 2015, pp. 1–9

25. R. Urgaonkar, S. Wang, T. He, M. Zafer, K. Chan, and K. K. Leung, “Dynamic service migration and workload scheduling in edgeclouds,” Performance Evaluation, vol. 91, pp. 205 – 228, 2015

26. C. H. Papadimitriou and J. N. Tsitsiklis, “The complexity of markov decision processes,” Mathematics of Operations Research, vol. 12, no. 3, pp. 441–450, 1987

27. H. Badri, T. Bahreini, D. Grosu, and K. Yang, “A sample average approximation-based parallel algorithm for application placement in edge computing systems,” in Proc. of IEEE International Conference on Cloud Engineering, 2018, pp. 198–203

28. H. Badri, T. Bahreini, D. Grosu, and K. Yang, “A sample average approximation-based parallel algorithm for application placement in edge computing systems,” in Proc. of IEEE International Conference on Cloud Engineering, 2018, pp. 198–203