Multi-objective Optimization for Computation Offloading in Mobile-edge Computing

Liqing Liu*, Zheng Chang†, Xijuan Guo*, Tapani Ristaniemi†
* College of Information Science and Engineering, Yanshan University, 066004 Qinhuangdao, China
E-mail: liuliqing_yanyan@163.com
† University of Jyvaskyla, Department of Mathematical Information Technology, P. O. Box 35, FI-40014 Jyvaskyla, Finland

***Abstract*—Mobile-edge cloud computing is a new cloud platform to provide pervasive and agile computation augmenting services for mobile devices (MDs) at anytime and anywhere by endowing ubiquitous radio access networks with computing capabilities. Although offloading computations to the cloud can reduce energy consumption at the MDs, it may also incur a larger execution delay. Usually the MDs have to pay cloud resource they used. In this paper, we utilize queuing theory to bring a thorough study on the energy consumption, execution delay and price cost of offloading process in a mobile-edge cloud system. Specifically, both wireless transmission and computing capabilities are explicitly and jointly considered when modelling the energy consumption and delay performance. Based on the theoretical analysis, the multi-objective optimization problem is formulated with the joint objectives to minimize the energy consumption, execution delay and price cost by finding the optimal offloading probability and optimal transmission power for each MD. The scalarization scheme and interior point method are applied to address the formulated problem. Through extensive simulations, the effectiveness of the proposed scheme can be demonstrated.**

I. INTRODUCTION

With the rapid development of ICT industry, mobile devices (MDs) have become an indispensable part of our daily life as they can provide convenient communication almost anytime and anywhere. The mobile application markets are also boosted by the advanced mobile technologies and high data rate wireless networks. However, due to the restrictions of the MDs on size, weight, battery life, ergonomics, and heat dissipation and so on. Many computational-intensive and latency-sensitive mobile applications have poor performance when they are performed on smart phones, such as image processing, chess gaming and so on [1].

Recent study shows that mobile cloud computing (MCC) technology provides a promising opportunity to overcome the limitation of hardware and save energy for MDs by offloading the computational-intensive tasks to the cloud for execution [2], [3], [4]. After execution is done at the cloud side, the final results are returned back to the MDs. The idea of bring cloud resource more closer to the MDs has been further pushed by the massive deployed MDs, which endows small-cell base stations with additional but limited cloud resources[5], [6], [7] and is refereed as mobile-edge cloud computing. As a novel MCC paradigm, mobile-edge cloud computing can provide cloud computing resource at the edge of radio access networks (RAN). In this case, the need for interactive response between edge computing and cloud center can be met by fiber transmission from the network edge to the cloud computing infrastructures, which is usually fast and low-latency. In this way, both computational and radio resource are brought closer to MDs, thus improving scalability in both computation and radio aspects [8]. Commonly, the cloud service can be provide by the cellular RAN, and the edge computing can be connected through WiFi networks [9]. However, the computing resource in mobile-edge cloud cannot be treated as sufficiently as the traditional central cloud, as it is usually targeted to serve a small portion of users.

With MCC, the mobile requests can be locally executed or offloaded to cloud for processing. From some previous literatures [2], [7], we can see that the transmission energy consumption is usually less than local execution energy consumption for the same size of requests for each MD. Meanwhile, we can also find that the offloading execution time is more than the local execution time for the same size of requests for each MD. Additionally, the MD has to pay the resource they used in edge cloud or the central cloud. Although some of the aforementioned literatures take the energy consumption, delay performance, or cost for utilizing the cloud resources individually into account when designing the offloading schemes [10]. However, to date, there hasn't been a paper considering these three optimization goals simultaneously in a MCC system. Moreover, most published articles take transmission power as a constant, which is too simplistic and are also inconsistent with reality. Also most of the works consider the cloud functionalities endowed on the base station as with infinite whereas the reality is against it.

In this work, we consider a joint E&D&P (**E**nergy consumption & **D**elay of execution & **P**rice cost)optimization for mobile-edge computing with explicit consideration of both the transmit power over wireless channel and computing capabilities. Accordingly, a multi-objective optimization problem is formulated, which involves minimizing the average energy consumption, the average execution time and average price cost by finding the optimal offloading probability and transmit power consumption. By using scalarization method, we are able to transform the multi-objective optimization problem into single-objective optimization problem and apply the Interior Point Method (IPM) to address transformed optimization

978-1-5386-1629-1/17/$31.00 ©2017 IEEE

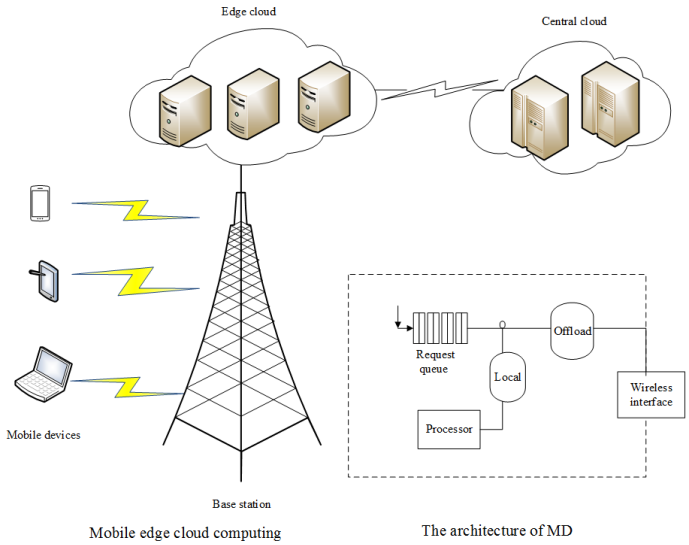

Figure 1. The model of mobile-edge cloud computing and the MD architecture

problem. Extensive simulations are conducted to evaluate the effectiveness of the presented schemes.

II. System Model and Problem Formulation

A. System Model

We consider a mobile-edge cloud computing model as in Fig. 1. We assume that the system consists of N MDs, a edge cloud located at the edge of the radio access networks, and a distant central cloud. Each MD executes an application and generates a series of homogeneous service requests. In this paper, we consider the queue model at MD as a $M/M/1$ queue, the edge cloud as a $M/M/c$ queue and the central cloud as a $M/M/\infty$ queue. For each MD, it can offload a portion or whole of its requests to edge cloud through an wireless channel, where the transmission suffers from interference generated by other MDs and noise power. If the total request rate is less than the maximum accepted rate of the edge cloud, then all the offloaded requests will be processed on the edge cloud. Otherwise, the edge cloud will further offload the overloaded requests to the central cloud for executing.

We assume that the requests generated by MD $i, i \in \{1, 2, \cdots, N\}$ follows a Poisson process with an average arrival rate of λ_i. Each request generating from the MD $i, \quad (i = 1, 2, \cdots, N)$ contains a data size of θ_i. The MD chooses to offload the service request with a probability $p_i^C \left(0 \le p_i^C \le 1\right)$. We refer to p_i^C as the offloading probability of the MD i hereinafter. According to the properties of the Poisson distribution, the service requests which are offloaded to cloud follow a Poisson process with an average rate of $p_i^C \lambda_i$, the service requests which are processed locally also follow a Poisson process with average rate of $\left(1 - p_i^C\right) \lambda_i$. In this paper, $p_i^C \ (i = 1, 2, \cdots, N)$ are optimization variable.

B. Local Execution Model and Transmission

Let u_i^M denotes the computing capability of MD i. Additionally we assume that l_i^M denotes the normalized workload on the MD i which represents the percentages of CPU that have been occupied. When considering a $M/M/1$ queue, the response time of a $M/M/1$ queue is $R = \frac{1/u}{1-\rho}$ [11], where $\rho = \frac{\lambda}{u}$ is the queue utilization, λ is the queue arrival rate, u is the service rate. Then the average response time T_i^M for locally processing requests at MD i is expressed as follows:

$$T_i^M \left(p_i^C\right) = \frac{1}{u_i^M \left(1 - l_i^M\right) - \left(1 - p_i^C\right) \lambda_i}. \tag{1}$$

The MD i transmits the data to the edge cloud through wireless base station. Considering the mutual interference caused by other MDs and the background interference, we can compute the uplink data rate for computation offloading of MD i as follows [7]:

$$R_i = W \log_2 \left(1 + \frac{P_i H_{i,s}}{\omega_i + \sum_{j \in N, j \ne i} P_j H_{j,s}}\right), \tag{2}$$

where W is the channel bandwidth and P_i is the transmission power of the MD i which are also the optimization variable in this paper. Additionally, $0 < P_i \le P_i^{th}$, where P_i^{th} is the maximum transmission power of MD i. $H_{i,s}$ is the channel gain between the MD i and the base station. ω_i denotes the background interference power.

From the communication model in (2), we can compute the transmission time of MD i for offloading the data from MD i to the base station as follows [7]:

$$T_i^t \left(p_i^C, P_i\right) = \frac{p_i^C \lambda_i \theta_i}{R_i} = \frac{p_i^C \lambda_i \theta_i}{W \log_2 \left(1 + \frac{P_i H_{i,s}}{\omega_i + \sum_{j \in N, j \ne i} P_j H_{j,s}}\right)}. \tag{3}$$

The energy consumption $E_i^M \left(p_i^C\right)$ of local executing the requests for MD i can be given as follows [9]:

$$E_i^M \left(p_i^C\right) = \kappa_i T_i^M \left(p_i^C\right) = \kappa_i \frac{1}{u_i^M \left(1 - l_i^M\right) - \left(1 - p_i^C\right) \lambda_i}, \tag{4}$$

where κ_i is the coefficient denoting the locally computation power of MD i.

We denote the energy consumption of transmitting the offloading requests from the MD to base station is $E_i^S \left(p_i^C\right)$, which can be given as follows [7]:

$$\begin{aligned} E_i^S \left(p_i^C, P_i\right) &= P_i T_i^t \left(p_i^C\right) = P_i \frac{p_i^C \lambda_i \theta_i}{R_i} \\ &= \frac{P_i p_i^C \lambda_i \theta_i}{W \log_2 \left(1 + \frac{P_i H_{i,s}}{\omega_i + \sum_{j \in N, j \ne i} P_j H_{j,s}}\right)}. \end{aligned} \tag{5}$$

C. Cloud Execution Model

Accordingly, we assume that there are c homogeneous servers deployed in the edge cloud, and the edge cloud is modeled as a $M/M/c$ queue. The service rate for each server is denoted as u^C. The maximum workload of the edge cloud is capped at a maximum request accepted rate denoted as $\lambda_{\max}^C$.

The requests from different MDs in the system are pooled together with a total rate λ_{Total}^{M} which can denoted as follows:

$$\lambda_{Total}^{M} = \sum_{i=1}^{N} \lambda_i p_i^C. \tag{6}$$

Then the fraction of the requests ψ^C that the edge cloud can process can be expressed:

$$\psi^C = \begin{cases} 1, & \lambda_{\max}^C \geq \lambda_{Total}^M; \\ \frac{\lambda_{\max}^C}{\lambda_{Total}^M}, & \lambda_{\max}^C < \lambda_{Total}^M; \end{cases} \tag{7}$$

Correspondingly, the actual execution rate at the edge cloud can be expressed as:

$$\lambda_p^C = \psi^C \lambda_{Total}^M = \begin{cases} \lambda_{Total}^M, & \lambda_{\max}^C \geq \lambda_{Total}^M; \\ \lambda_{\max}^C, & \lambda_{\max}^C < \lambda_{Total}^M. \end{cases} \tag{8}$$

To this end, based on the analysis of $M/M/c$ queue at the edge cloud and Erlang's Formula [12], we define

$$\rho^C = \frac{\lambda_p^C}{cu^C}. \tag{9}$$

Therefore, the average waiting time of each request at the edge cloud, which contains the queuing time and execution time, is denoted as follows

$$T_{wait}^C \left(\lambda_p^C\right) = \frac{C\left(c, \rho^C\right)}{cu^C - \lambda_p^C} + \frac{1}{u^C}. \tag{10}$$

where

$$C\left(c, \rho^C\right) = \frac{\left(\frac{\left(c\rho^C\right)}{c!}\right)\left(\frac{1}{1-\rho^C}\right)}{\sum_{k=0}^{c-1} \frac{\left(c\rho^C\right)^k}{k!} + \left(\frac{\left(c\rho^C\right)}{c!}\right)\left(\frac{1}{1-\rho^C}\right)}. \tag{11}$$

Assuming u_b^C is the transmission rate of the edge cloud, we can obtain the expected time T_b^C for the execution results waiting in the edge cloud before they are completely delivered out as follows:

$$T_b^C \left(\lambda_p^C\right) = \frac{1}{u_b^C - \lambda_p^C}. \tag{12}$$

The overloaded requests are transmitted to the central cloud through wired connection which incurs a fixed time delay T^O. As the central cloud is considered as $M/M/\infty$ with the service rate u^{CC}. Then, the waiting time T_{wait}^{CC} of overloaded request can be presented as follows:

$$T_{wait}^{CC} = T^O + \frac{1}{u^{CC}}. \tag{13}$$

The expected time T_b^{CC} for the results waiting in the cloud before they are completely sent out is denoted as

$$T_b^{CC}\left(p_i^C\right) = \frac{1}{u_b^{CC} - \left(\lambda_{Total}^M - \lambda_p^C\right)}. \tag{14}$$

We neglect the time and energy consumption for the MD to receive the processed outcome, which is similar to [7].

D. Problem Formulation

From (4) and (5), we can obtain the average energy consumption for the MD i, which is denoted as follows:

$$E_i\left(p_i^C, P_i\right) = \left(1 - p_i^C\right) E_i^M\left(p_i^C, P_i\right) + p_i^C E_i^S\left(p_i^C, P_i\right). \tag{15}$$

From (1), (3), (10), (12), (13) and (14), we can acquire the average execution time for the MD i, which is denoted as follows

$$\begin{aligned} T_i\left(p_i^C, P_i\right) = &\left(1 - p_i^C\right) T_i^M\left(p_i^C\right) + p_i^C T_i^t\left(p_i^C, P_i\right) \\ &+ p_i^C \psi^C \left(T_{wait}^C + T_b^C\right) \\ &+ p_i^C \left(1 - \psi^C\right)\left(T_{wait}^{CC} + T_b^{CC}\right) \end{aligned}. \tag{16}$$

So the average energy consumption and average execution time of all MDs in the system are denoted in (17) and (18).

$$E\left(p_i^C, P_i\right) = \frac{1}{N} \sum_{i=1}^{N} E_i\left(p_i^C, P_i\right). \tag{17}$$

$$T\left(p_i^C, P_i\right) = \frac{1}{N}\left[\sum_{i=1}^{N} T_i\left(p_i^C, P_i\right)\right]. \tag{18}$$

Additionally, the MD has to pay the resource they used in edge cloud or the central cloud. We assume that the per price for the edge cloud is r^C and the central cloud is r^{CC}. We can compute the average price cost of the MDs as follows:

$$M\left(p_i^C\right) = \frac{1}{N}\left\{r^C \lambda_p^C\left(p_i^C\right) + r^{CC}\left[\lambda_{Total}^M\left(p_i^C\right) - \lambda_p^C\left(p_i^C\right)\right]\right\}. \tag{19}$$

To this end, with above analytic results on the expected energy consumption, execution delay and cost performance, we are able to formulated E&D&P minimization problem. The problem can be considered as a multi-objective optimization which involves minimizing energy consumption, execution delay and price cost, which can be denoted as follows:

$$\min_{\left\{p_i^C, P_i\right\}} \left\{E\left(p_i^C, P_i\right), T\left(p_i^C, P_i\right), M\left(p_i^C\right)\right\}, \tag{20}$$

subject to

$$\left(1 - p_i^C\right)\lambda_i < u_i^M\left(1 - l_i^M\right), \tag{21}$$

$$\lambda_p^C < cu^C, \tag{22}$$

$$\lambda_p^C < u_b^C, \tag{23}$$

$$\lambda_{Total}^M - \lambda_p^C < u_b^{CC}, \tag{24}$$

$$0 < P_i \leq P^{th} \qquad (i = 1, 2, \cdots, N), \tag{25}$$

$$0 \leq p_i^C \leq 1 \qquad (i = 1, 2, \cdots, N). \tag{26}$$

It can be noticed that Problem (20) is a multi-objective nonlinear optimization problem with various constraints. Moreover, we assume that the MDs in the system have an expected energy consumption, execution delay performance and price cost, which is denoted as $\tilde{E}$, $\tilde{T}$, $\tilde{M}$ respectively, which are all constants. To address such a kind of problem, scalarization method can be applied. To qualify the tradeoff, we incorporate

a set of weight factors:$\{\alpha_1,\ \alpha_2, \alpha_3\}$, where $\alpha_1+\alpha_2+\alpha_3=1$, to reflect the relative importance of the energy costs, execution time and price cost, respectively. Then the multi-objective optimization system could be transformed to a single objective optimization problem, which is

$$\min_{\{p_i^C, P_i\}} \quad \alpha_1 \frac{E\left(p_i^C, P_i\right)}{\tilde{E}} + \alpha_2 \frac{T\left(p_i^C, P_i\right)}{\tilde{T}} + \alpha_3 \frac{M\left(p_i^C\right)}{\tilde{M}} \quad (27)$$

subject to: (21), (22), (23), (24), (25), (26).

III. Algorithm Design

In the above section, we formulate the optimization problem in order to jointly optimize the E&D&P performance in the mobile-edge cloud computing scenario. In this section, we utilize interior point method (IPM) [13] to design the corresponding algorithm and address the formulated problem. By comparing the values of $\lambda_{\max}^C$ and λ_{Total}^M, we can further divided the case into two sub-cases.

1) In the first sub-case, we assume $\lambda_{\max}^C \geq \lambda_{Total}^M$. In other words, all the MDs' requests in the system can be processed at the edge cloud. Substituting (1), (3), (4), (5), (10), (11), (12) and (19) into (27), we can obtain the E&D&P optimization problem in a specific analytical expression as $V_1\left(p_i^C, P_i\right)$.
2) In the second subcase, $\lambda_{\max}^C < \lambda_{Total}^M$. In other words, the edge cloud can only process as much as $\lambda_{\max}^C$ workload, and the overloaded requests will be further offloaded to the central cloud to execute. Substituting (1), (3), (4), (5), (10), (11) , (12), (13), (14) and (19) into (27), we can obtain the E&D&P optimization problem in a specific analytical expression as $V_2\left(p_i^C, P_i\right)$.

In order to solve these nonlinear programming problems, we may consider using one special "punishment" approach as presented in [13]. Correspondingly, the penalty functions for the first subcase and second subcase can be denoted as (28) and (29).

In (28) and (29), $\xi_j^{(k)} > 0\,(j=1,2; k=0,1,2\cdots)$ are the penalty coefficients, and $\xi_j^{(k)}$ satisfies the following iterative rules:

$$\xi_j^{(k+1)} = \beta_j \xi_j^{(k)} \quad (j=1,2; k=0,1,2\cdots), \quad (30)$$

where $\beta_j\,(j=1,2)$ are the reduction factors, which are usually in the range $(0.1, 0.7)$, $\xi_j^{(0)}$ is usually taken 1 as the initial value, too large or too small value of $\xi_j^{(0)}$ may significantly affect the convergence rate of IPM. In addition, initial points $\left(\left(p_i^C\right)^0, \left(P_i\right)^0\right)_{i=1}^N$ are needed which are in the feasible region.

We can obtain the extreme points $\left(p_i^C\left(\xi_j^{(k)}\right), P_i\left(\xi_j^{(k)}\right)\right)_{i=1}^N \quad (j=1,2)$ of these two penalty functions through unconstrained optimization method. Through iteration, we can obtain the optimal $\left(\left(p_i^C\right)^*, \left(P_i\right)^*\right)_{i=1}^N$.

The detailed procedure of the proposed algorithm is depicted in Algorithm 1. With Algorithm 1, we can find the optimal offloading probability and the optimal transmission power for each MD in order to minimizing the E&D&P in the system under different cases. Additionally, in terms of complexity theory, the algorithm converges in $\mathrm{O}\,(k)$ iterations [13].

Table I
Simulation parameters of single-user scenario

Parameters	u^C (MIPS)	u_b^C (MIPS)	u^M (MIPS)	λ (MIPS)
Value	5.6	10	1.5	1.1
Parameters	κ (J/S)	θ (bits)	H_s	l^M
Value	10	1.8e+6	1.0	0.2

Algorithm 1 Proposed IPM Algorithm

1: Initialization:
initial feasible point $\left(\left(p_i^C\right)^0, \left(P_i\right)^0\right)_{i=1}^N$; initial value of penalty coefficients $\xi_j^{(0)}$; the reduction factor β_j, $k=0$.
2: Define ε_j as a sufficiently small positive real number.
3: Solving the extreme points of the penalty functions as $\left(p_i^C\left(\xi_j^{(k)}\right), P_i\left(\xi_j^{(k)}\right)\right)_{i=1}^N \quad (j=1,2)$.
4: **while** $\left(\left\|\left(\left(p_i^C\left(\xi_j^{(k)}\right), P_i\left(\xi_j^{(k)}\right)\right)_{i=1}^N\right) - \left(\left(\left(p_i^C\right)^0, \left(P_i\right)^0\right)_{i=1}^N\right)\right\| > \varepsilon_j\right)$ **do**
5: Iteration: $\xi_j^{(k+1)} = \beta_j \xi_j^{(k)} \quad (j=1,2; k=0,1,2\cdots)$, $\left(\left(p_i^C\right)^0, \left(P_i\right)^0\right)_{i=1}^N = \left(p_i^C\left(\xi_j^{(k)}\right), P_i\left(\xi_j^{(k)}\right)\right)_{i=1}^N \quad (j=1,2), k=k+1$.
6: **end while**
7: **return** $\left(p_i^C\left(\xi_j^{(k)}\right), P_i\left(\xi_j^{(k)}\right)\right)_{i=1}^N \quad (j=1,2)$

IV. Performance Evaluations

First, we investigate the impact of offloading probability p_i^C and transmission power P_i on the energy consumption and delay performance. The simulation parameters can be found in Table 1, which are refer to [10]. Comparing Fig.2 and Fig.3, we can clearly observe the necessity for investigating the tradeoff between the energy consumption and execution delay with offloading probability and transmission power.

With IPM algorithm, we can easily obtain the optimal offloading probability and optimal transmission power for each MD at any arrival rate at a certain weight set. For example, we determine $(\alpha_1, \alpha_2, \alpha_3) = (0.4, 0.5, 0.1)$, when the arrival rates are $(1.4, 1.8, 1.6)$, the optimal transmission power and optimal offloading probability is $(3.0573, 0.8021)$, $(0.8395, 0.7699)$, $(1.0461, 0.8357)$ for MD 1, MD 2, and MD 3 respectively. Additionally, we investigate the impact of the number of MDs on E&D&P, which is displayed in Fig. 4. From Fig. 4, we can find that with the number of MDs increasing, the energy consumption and execution delay also increase, but the price cost decreases. There is no doubt that resources contention and sharing can cause delay and performance degradation that might result in higher and higher response time. With execution delay increasing, some MDs prefer to execute some requests by themselves, so the energy consumption also increases. Because of using less cloud resource, the price cost decreases with the number of MDs increasing.

$$
\begin{aligned}
&\Phi_1\left(p_i^C, P_i, \xi_1^{(k)}\right) \\
&= V_1\left(p_i^C, P_i\right) - \xi_1^{(k)} \ln\left[\prod_{i=1}^{N}\left|\left(1-p_i^C\right)\lambda_i - u_i^M\left(1-l_i^M\right)\right|\right] - \xi_1^{(k)} \ln\left|\sum_{i=1}^{N}\lambda_i p_i^C - cu^C\right| \\
&- \xi_1^{(k)} \ln\left|\sum_{i=1}^{N}\lambda_i p_i^C - \lambda_{\max}^C\right| - \xi_1^{(k)} \ln\left|\sum_{i=1}^{N}\lambda_i p_i^C - u_b^C\right| - \xi_1^{(k)} \ln\left(\prod_{i=1}^{N}\left|p_i^C\right|\right) \\
&- \xi_1^{(k)} \ln\left(\prod_{i=1}^{N}\left|p_i^C - 1\right|\right) - \xi_1^{(k)} \ln\left(\prod_{i=1}^{N}\left|P_i\right|\right) - \xi_1^{(k)} \ln\left(\prod_{i=1}^{N}\left|P_i - P_i^{th}\right|\right)
\end{aligned} \tag{28}
$$

$$
\begin{aligned}
&\Phi_2\left(p_i^C, P_i, \xi_2^{(k)}\right) \\
&= V_2\left(p_i^C, P_i\right) - \xi_2^{(k)} \ln\left[\prod_{i=1}^{N}\left|\left(1-p_i^C\right)\lambda_i - u_i^M\left(1-l_i^M\right)\right|\right] - \xi_2^{(k)} \ln\left|\lambda_{\max}^C - \sum_{i=1}^{N}\lambda_i p_i^C\right| \\
&- \xi_2^{(k)} \ln\left|\sum_{i=1}^{N}\lambda_i p_i^C - \lambda_{\max}^C - u_b^{CC}\right| - \xi_2^{(k)} \ln\left(\prod_{i=1}^{N}\left|p_i^C\right|\right) - \xi_2^{(k)} \ln\left(\prod_{i=1}^{N}\left|p_i^C - 1\right|\right) \\
&- \xi_2^{(k)} \ln\left(\prod_{i=1}^{N}\left|P_i\right|\right) - \xi_2^{(k)} \ln\left(\prod_{i=1}^{N}\left|P_i - P_i^{th}\right|\right)
\end{aligned} \tag{29}
$$

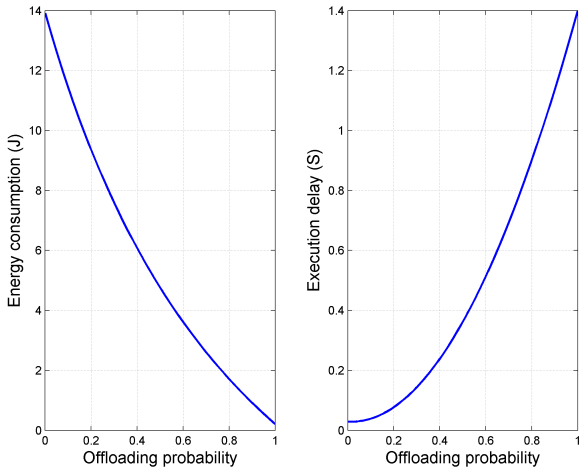

Figure 2. The impact of offloading probability on E&D

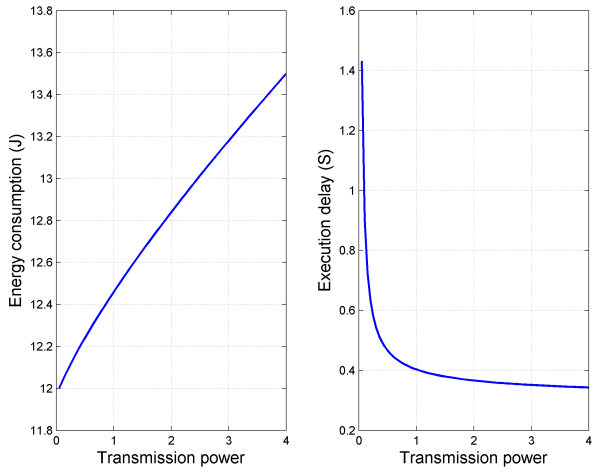

Figure 3. The impact of transmission power on E&D

In Fig. 5, we investigate the impact of transmission power on the total weighted E&D&P at different offloading probabilities. At first, with transmission power increasing, the total weighted E&D&P decrease. The E&D&P reaches the minimum, at a certain transmission power value, which is the optimal transmission power. Then the total weighted E&D&P increase with the transmission power increasing. This rule can be found from any curve in Fig.5, which denotes different offloading probabilities. Moreover, the larger offloading probability, the less E&D&P consumption, which can be found by comparing the four curves in Fig. 5.

At last, we compare our proposed scheme with other schemes proposed in [7], [9], which is denoted in Fig. 6. In our scheme, we optimize both offloading probability and transmission power, which obtain lower weighted E&D&P consumption. The method in [7] only optimize the offloading probability and the method in [9] only optimize the transmission power. We can see that our method can achieve better performance in E&D&P, which demonstrate the comprehensiveness and validity of the study.

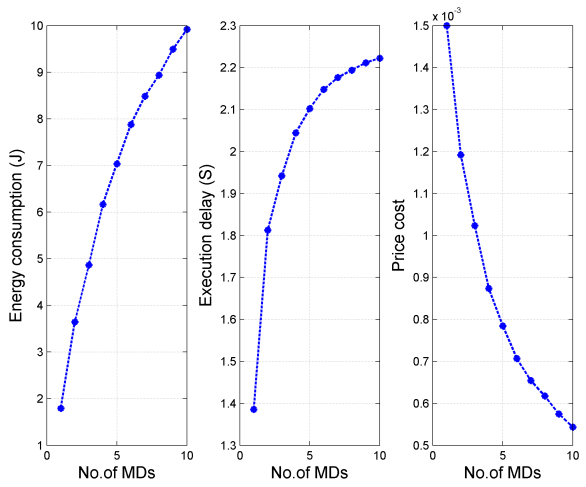

Figure 4. The impact of number of MDs on E&D&P

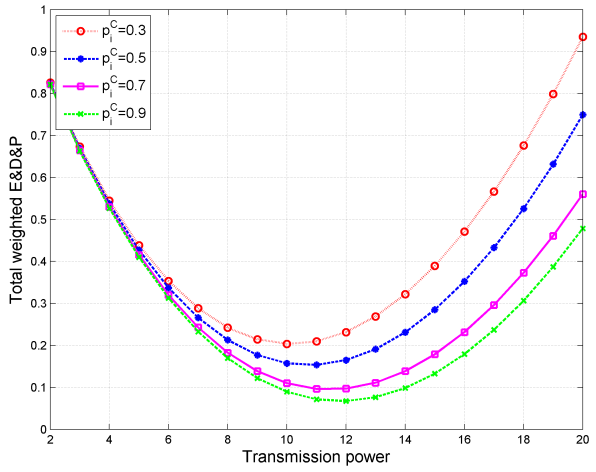

Figure 5. The impact of transmission power on E&D&P

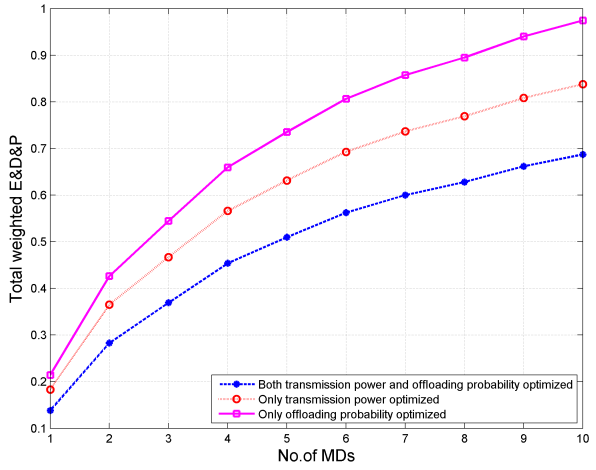

Figure 6. Comparing among different schemes

V. CONCLUSION

In this paper, we investigated the tradeoff among energy consumption, delay performance and price cost in a mobile-edge cloud system. Specifically, we optimized the offloading probability and transmission power for each MD to minimize the E&D&P optimization issue. We derived the analytic results of energy consumption, delay performance and price cost with assumption of three different queue models at mobile devices, edge cloud and central cloud. By leveraging the obtained results, we then formulated the multi-objective problem with various constraints and addressed it by using IPM-based algorithm. The extensive performance evaluations presented to illustrate the effectiveness of the proposed scheme.

ACKNOWLEDGEMENT

This work is partly supported by the Academy of Finland (Decision number 284748) and Hebei NSF (F2016203383).

REFERENCES

[1] 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, Jun. 2013.

[2] S. Deng, L. Huang, J. Taheri, and A. Y. Zomaya, "Computation offloading for service workflow in mobile cloud computing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 12, pp. 3317-3329, Dec. 2015.

[3] J. Vazifehdan, R. V. Prasad, M. Jacobsson, and I. Niemegeers, "An analytical energy consumption model for packet transfer over wireless links," *IEEE Communications Letters*, vol. 16, no. 1, pp. 30-33, Jan. 2012.

[4] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open Challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 369-392, Feb. 2014.

[5] B.G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: elastic execution between mobile device and cloud," *Conference on Computer Systems*, pp. 301-314, 2011.

[6] N. Fernando, S. W. Loke, and W. Rahayu, "Computing with nearby mobile devices: a work sharing algorithm for mobile edge-clouds," *IEEE Transactions on Cloud Computing*, in press, Apr. 2016, DOI: 10.1109/TCC.2016.2560163.

[7] X. Chen, L. Jiao, W. Z. Li, and X. M. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computin," *IEEE/ACM Transactions on Networking*, in press, Oct. 2015, DOI: 10.1109/TNET.2015.2487344.

[8] M. Villari, M. Fazio, S. Dustdar, O. Rana, and R. Ranjan, "Osmotic computing: a new paradigm for edge/cloud integration," *IEEE Cloud Computing*, vol. 3, no. 6, pp. 76-83, Dec. 2016.

[9] S. Barbarossa, S. Sardellitti, and P. D. Lorenzo, "Communicating while computing: distributed mobile cloud computing over 5G heterogeneous networks," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 45-55, Oct. 2014.

[10] Z. Jiang, and S. Mao, "Energy delay tradeoff in cloud offloading for multi-core mobile devices," *IEEE Access*, vol. 3, pp. 2306-2316, 2015.

[11] A. Lazar, "The throughput time delay function of an M/M/1 queue (Corresp.)," *IEEE Transactions on Information Theory*, vol. 29, no. 6, pp. 914-918, Jan. 2003.

[12] B. Ngo, and H. Lee, "Analysis of a pre-emptive priority M/M/c model with two types of customers and restriction," *Electronics Letters*, vol. 26, no. 15, pp. 1190-1192, July 1990.

[13] J. Gondzio, "Interior point methods 25 years later," *European Journal of Operational Research*, vol. 218, no. 3, pp. 587-601, May 2012.