IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 21, 2020, accepted February 13, 2020, date of publication February 24, 2020, date of current version March 4, 2020.

*Digital Object Identifier 10.1109/ACCESS.2020.2975733*

# A Probability Preferred Priori Offloading Mechanism in Mobile Edge Computing

**JIN WANG[1,2], (Senior Member, IEEE), WENBING WU[1], ZHUOFAN LIAO[1], (Member, IEEE), R. SIMON SHERRATT[3], (Fellow, IEEE), GWANG-JUN KIM[4], OSAMA ALFARRAJ[5], AHMAD ALZUBI[5], AND AMR TOLBA[5,6]**

[1]Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410000, China
[2]School of Information Science and Engineering, Fujian University of Technology, Fuzhou 350118, China
[3]Department of Biomedical Engineering, The University of Reading, Reading RG6 6AY, U.K.
[4]Department of Computer Engineering, Chonnam National University, Gwangju 61186, South Korea
[5]Department of Computer Science, Community College, King Saud University, Riyadh 11437, Saudi Arabia
[6]Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Shebin Elkom 32511, Egypt

Corresponding author: Gwang-Jun Kim (kgj@chonnam.ac.kr)

This work was supported in part by the National Natural Science Foundation of China under Grant 61772454, Grant 61811530332, and Grant 61811540410, in part by the Degree & Postgraduate Education Reform Project of Hunan Province, and in part by the Deanship of Scientific Research at King Saud University through Research Group under Grant RG-1438-070.

**ABSTRACT** Mobile edge computing (MEC) can provide computation and storage capabilities via edge servers which are closer to user devices (UDs). The MEC offloading system can be viewed as a system where each UD is covered by single or multiple edge servers. Existing works prefer a posterior design when task offloads, which can lead to increased workloads. To investigate the task offloading of edge computing in multi-coverage scenario and to reduce the workload during task offloading, a probability preferred priori offloading mechanism with joint optimization of offloading proportion and transmission power is presented in this paper. We first set up an expectation value which is determined by the offloading probability of heterogeneous edge servers, and then we form a utility function to balance the delay performance and energy consumption. Next, a distributed PRiori Offloading Mechanism with joint Offloading proportion and Transmission (PROMOT) power algorithm based on Genetic Algorithm (GA) is proposed to maximize the utility of UD. Finally, simulation results verify the superiority of our proposed scheme as compared with other popular methods.

**INDEX TERMS** Probability preferred, workload, offloading, genetic algorithm, mobile edge computing.

## I. INTRODUCTION

With the rapid development of Internet of Things (IoTs), user devices (UDs) can provide more convenient services, which plays a more and more important role in our daily life [1]. However, high performance of low energy consumption and latency is unreachable due to the limitation of the UDs as per power battery, computation and storage capacity, and device size, etc.

Mobile edge computing (MEC), as a novel supporting mechanism of cloud computing, is developing very rapidly in recent years, which can provide an opportunity for UDs to overcome those limitations and reduce energy consumption or latency. MEC provides a cloud computing capacity at the edge of the network [2]. In this case, the demand of UDs for low energy consumption and latency can be achieved by offloading data or tasks to edge servers. The thought of bringing cloud capabilities closer to UDs has further improved the user's experience [3]–[5].

Since the tasks of UD can be executed locally or offloading to the edge server for processing, there is much literature focusing on partial offloading in MEC systems [6]–[9]. In partial offloading, one part of the task is executed locally and the rest part is executed remotely. Similarly, transmission power also becomes more important for the task that needs to be transferred to edge server in partial offloading. He *et al.* [6] optimized the energy minimization problem with the time constraint. Gu *et al.* [7] minimized energy consumption by using the matching game. He *et al.* [8] maximized the supported devices with communication and

The associate editor coordinating the review of this manuscript and approving it for publication was Tu Nguyen.

computation constraints. Mu *et al.* [9] reduced energy consumption by investigated joint partitioning decisions and subcarrier assignment. However, there is little information about the offloading proportion. Moreover, the transmission power is not jointly optimized during task offloading.

Besides, the base stations become denser and denser due to the development of 5th-Generation (5G) [10], and the workload of edge computing tasks offload decision is getting larger and larger which is caused by posterior mechanism. The posterior mechanism in edge computing offloading will cause every server connected to the UD to be evaluated once.

To deal with those challenges, in this work, a probability preferred priori offloading mechanism with joint optimized the offloading proportion and the transmission power method is presented. A tradeoff optimization problem with the expectation of time and energy consumption is formed, which maximize the utility of each UD. An M/M/1 queuing model is also adopted at the heterogeneous edge servers for reasonable task schedule. To solve the maximum utility problem and obtain the best offloading and transmission power, a distributed PRiori Offloading Mechanism with joint Offloading proportion and Transmission (PROMOT) power algorithm based on Genetic Algorithm (GA) is proposed. Simulations are conducted to evaluate the effectiveness of the presented schemes. The main contribution of this paper is summarized as follows:

1) A priori offloading model with probability preferred is proposed, where the expectation value is used. The utility maximum problem with jointly optimizing offloading proportion and transmission power is formed, which queue theory is also included.
2) A distributed PRiori Offloading Mechanism with joint Offloading proportion and Transmission (PROMOT) power algorithm based on Genetic Algorithm (GA) is proposed to tackle the utility problem to obtain the best offloading proportion and transmission power to achieve the maximum utility.
3) Simulation is carried out to evaluate the performance of the proposed solution which shows the superiority.

The rest of this paper is organized as follows. related work is introduced in Section II. The system model is introduced in Section III. Section IV presents the offloading scheme. In Section V, the illustrations of simulation results are discussed. And Section VI concludes the paper and made a view on future work.

## II. RELATED WORK

Computation offloading in MEC is a decentralized and distributed approach, which UD can offload its tasks to one edge server it connected with for execution. In that case, UD needs to choose the best edge server to offload tasks if it connected with multiple edge server. From the perspective of task offloading, there are two kinds of task offloading strategies, including fully offloading and partial offloading, the focus in this paper is partial task offloading.

Yu *et al.* [11] proposed a dynamic mobility-aware partial offloading (DMPO) method to save energy consumption under delay constraint, which the short-term mobility prediction characteristic is used and combined with the mobile communication path decision. This algorithm predicts the time of handover and assigns a specific data size in each time slot to achieve the purpose of minimizing energy consumption. Kuang *et al.* [12] proposed a two-level alternation method framework based decomposition to solve the Partial Offloading Scheduling and Power allocation (POSP) problem. The framework can reduce time consumption. Wang *et al.* [13] proposed a Hungarian algorithm based MTMS algorithm to solve the energy minimization problem. In this algorithm, two phases are included, the first phase is that the task is divided into several subtasks and the second phase is that the subtasks are assigned to the neighbor edge servers which connected with the local edge server. In this case, the energy consumption can be efficiently reduced and the time consumption can also be reduced. Ning *et al.* [14] investigated the latency and offloading efficiency in MEC. Two situations are considered in this paper which is the single-user situation and multiple-user situation. In the former situation, the resources are not constrained and can be solved by the branch and bound algorithm. While the latter situation is resource competition and solved by Iterative Heuristic Resource Allocation (IHRA) algorithm. Tang *et al.* [15] investigated the time and energy consumption in MEC. A Mixed Overhead of Time and Energy (MOTE) minimization problem is proposed by joint optimized the time and energy consumption during task offloading. And the block coordinate descent method to solve each variable step by step is adopted. Ren *et al.* [16] investigated the latency minimization problem with joint optimize the communication and computation resource allocation. In this paper, three computation models consist of local compression, edge cloud compression, and partial compression is considered. By solving piecewise optimi-zation problem, the best communication and computation are obtained.

However, the literature [11], [13]–[15] have little information about the offloading proportion, they just consider a task can be divided into two or more part, one part can be executed locally, while the rest part can be executed remotely. The research [16] makes a specific offloading proportion during task offloading, while the transmission power is not considered. The literature [12] makes a specific offloading proportion and power allocation, while it just considers the situation in which a task is offloaded to single one edge server, the scenario that the user device is covered by multiple servers is not considered.

Tran *et al.* [17] investigated the scenario that a user device is covered by multiple edge servers, and optimized the offloading decision, transmission power, and computation resource. And a system utility function consist of the time and energy consumption is formed to balance time and energy. A strategy named hJTORA (heuristic Joint Task Offloading scheduling and Resource Allocation strategy) is proposed to solve the optimization problem for maximizing the

system utility. Zhang *et al.* [18] studied the resource scheduling problem under multi-server scenario for minimizing the latency. A delay based Lyapunov function and a resource scheduling algorithm are proposed, which can effectively reduce the latency. Those researches studied the scenario that user devices are covered by multiple edge servers; which user devices must choose a specific edge server to offload task. While those research are used a posterior mechanism, which the workload is large when the number of covered server is getting larger. Yang *et al.* [19] investigated a multi-access MEC server system to reduce energy consumption. Computational offloading, subcarrier allocation, and computing resource allocation are jointly optimized. The authors designed an improved branch and bound algorithm (BnB) to find the global optimal solution. A combination algorithm is proposed to obtain the practical application of the suboptimal solution.

Another challenge of task offloading in MEC is task scheduling at the edge server. Gao *et al.* [20] proposed an energy minimization task offloading strategy with a dynamic priority-based task scheduling scheme in the MEC system. The strategy can reduce latency and improve performance. Luo *et al.* [21] proposed a stochastic and M/M/1 queuing based edge computing system model. In that case, a response time minimization problem is given which can be proved as an NP-complete. And a greedy algorithm is used to solve the problem. Chen *et al.* [22] formulated a multi-user and multi-task offloading problem by considering the mobile edge cloud computing. An energy harvesting policy is proposed by using the Lyapunov approach. In that case, the centralized and distributed Greedy algorithm is proposed to solve the problem. Chen *et al.* [23] investigated the dynamic offloading problem in MEC. The problem is minimizing the energy consumption with ensuring the queue length. An Energy Efficient Dynamic Offloading Algorithm (EEDOA) is proposed further to solve this problem. Wang *et al.* [24] investigated the multi-task scheduling problem in hybrid mobile cloud computing (MCC). A Cooperative Multi-Task Scheduling based on Ant Colony Optimization algorithm (CMSACO) is proposed by jointly considering task profit, task deadline, task dependency, node heterogeneity and load balancing. And the network optimization algorithm in [25]–[28] provide an enlightening thought to the network construction in edge computing.

In this paper, we proposed a method called PRiori Offloading Mechanism with joint Offloading proportion and Transmission power (PROMOT). The method is probability preferred and can be adopted at the scenario that UDs are covered by one or multiple heterogeneous edge server. For reasonable task scheduling, an M/M/1 queuing is used at the edge server. By adopting the method, the utility of UDs can be efficiently improved.

## III. SYSTEM MODEL

MEC model is considered as in Figure 1, in which the system consists of $N$ UDs and $M$ edge servers, respectively.

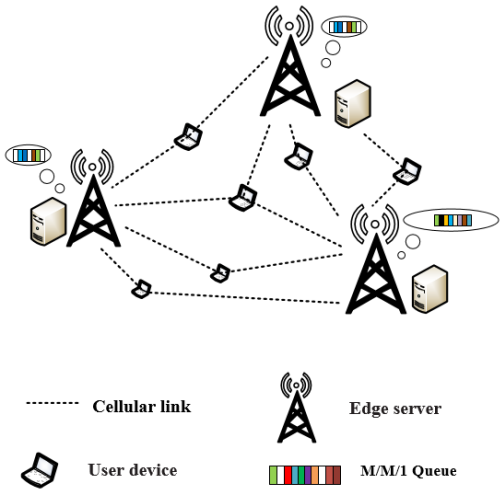


**FIGURE 1.** The system model of MEC.

Those edge servers located at the edge of the cellular network. In this paper, each server can communicate with UDs through cellular link.

We consider that each UD submits their tasks with arrival rate $\lambda_i$ according to the Poisson process [29] and one computation task needs to be offloaded at one time in a time slot. Each task $i$ generated by UD $i$ can be characterized by a quaternion of four parameters, $< s_i, c_i, T_i^{\max}, E_i^{\max} >$, where $s_i$ denotes the input size of the task $i$, $c_i$ denotes the necessary needed CPU cycle of executing task $i$, $T_i^{\max}$ and $E_i^{\max}$ denotes the maximum tolerate time and energy of task $i$, respectively. Each UD can save its energy and reduce execution time by offloading their tasks to edge server or servers, however excessive data will incur additional time and energy consumption during task transfer. The UD chooses to offload the task with an offloading proportion $p_i^{off}$ $(0 \leq p_i^{off} \leq 1)$, which $p_i^{off} = 1$ indicates that the task $i$ is offloaded entirely.

The queuing theory and heterogeneity is adopted in this system. In that case, each edge server maintains an M/M/1 queue and has different service rate. The service rate of server $j$ is denoted as $SR_j$. The used key notations in this paper are summarized in Table 1.

### A. LOCAL EXECUTION MODEL

Let's assume that $f_{i,loc} > 0$ denotes the local computing capability of UD $i$ according to the CPU cycles/s. Then the time consumption of task $i$ executed locally can be calculated as follows

$$t_{i,loc}^{comp} = \frac{(1 - p_i^{off})c_i}{f_{i,loc}} \tag{1}$$

The energy consumption of UD $i$ when executing its task locally can calculated by adopting the widely used model as $\varepsilon = \kappa f^2$ [30], where $\kappa$ is the energy coefficient which depends on the chip architecture and $f$ is the CPU frequency. Thus the energy consumption of task $i$ can be obtained as

**TABLE 1. Notations.**

| Notations | Descriptions |
|---|---|
| $N$ | The number of UDs in the system |
| $M$ | The number of edge servers in the system |
| $\lambda_i$ | The arrival rate of the UD $i$ |
| $p_i^{off}$ | The offloading rate of task $i$ |
| $s_i$ | The date size of task $i$ |
| $f_{i,loc}$ | The local computing capability of UD $i$ |
| $c_i$ | The necessary needed CPU cycle of executing task $i$ |
| $E_i^{loc}$ | Energy consumption of task $i$ when executing task locally |
| $t_{i,loc}^{comp}$ | Time consumption of task $i$ when executing task locally |
| $\kappa$ | Energy coefficient of user device |
| $k$ | Energy coefficient of edge computing |
| $B$ | The channel bandwidth |
| $N_0$ | The background noise interference |
| $Po_i$ | The transmission power of the UD $i$ |
| $Po_{\max}$ | The maximum transmission power of UD $i$ |
| $h_{i,j}$ | The channel gain between the UD $i$ and the edge server $j$ |
| $R_{i,j}$ | The uplink transmission rate |
| $SR_j$ | The service rate of edge server $j$ |
| $t_{i,off}^{trans}$ | The transmission time expectation of task $i$ |
| $E_i^{trans}$ | The transmission energy consumption expectation of task $i$ |
| $t_i^{exe}$ | The time consumption expectation of task $i$ when executing remotely |
| $E_i^{exe}$ | The energy consumption expectation of task $i$ when executing remotely |
| $T_i^{\max}$ | The maximum execution latency the UD $i$ |
| $E_i^{\max}$ | The maximum energy of the UD $i$ |

follow

$$E_i^{loc} = \kappa(f_{i,loc}^2)(1 - p_i^{off})c_i \tag{2}$$

### B. EDGE SERVER EXECUTION MODEL

The UD needs to choose an edge server to offload its task since the UD is covered by multiple edge servers. The mathematical expectation is used in this paper for solving the task offloading decision problem. For each edge server has its own service rate, we can calculate the offloading probability of server $j$ at UD $i$ as follows:

$$pro_{i,j} = \frac{SR_j}{\sum_k SR_k} \tag{3}$$

where the numerator is the service rate of server $j$, the denominator is the total service rate of all servers that connected with UD $i$ and $k$ is the total number of edge servers that UD $i$ connected with. According to the equation (1), the server with larger service rate has a higher probability to the offloading server, which is benefited to the UD and can improve the UD's performance.

When tasks processed in edge server, there are three parts of incurred delay: the time consumption $t_{i,off}^{trans}$ caused by transmitting the task to the edge server through cellular link; the time consumption $t_i^{off}$ caused by executing and waiting at the edge server; the time consumption caused by transmitting the result from edge server back to the UD. The time consumption of transferring the result can be ignored for the data size of the result is much smaller than that of the input [31].

In this work, the transmission rate is calculated by Shannon formula [32] with the consideration of the mutual interference caused by other UDs and the background interference, the transmission rate of UD $i$ transfer task to server $j$ is calculated as follows

$$R_{i,j} = B\log_2(1 + \frac{Po_i h_{i,j}}{N_0 + \Sigma_{i' \in k \setminus i} Po_{i'} h_{i',j}}) \tag{4}$$

where $B$ is the channel bandwidth and $N_0$ is the background noise interference. $Po_i$ $(0 < Po_i \leq Po_{\max})$ is the transmission power of the UD $i$, and $Po_{\max}$ is the maximum transmission power of UD $i$, $h_{i,j}$ is the channel gain between the UD $i$ and the edge server $j$, the rest of the denominator is interference between UDs.

Since the uplink transmission rate is obtained, we can calculate the transmission time of UD $i$ for offloading the data from UD $i$ to the edge server $j$ as follows:

$$t_{i,j}^{trans} = \frac{p_i^{off} s_i}{R_{i,j}} \tag{5}$$

Each edge server has a probability to be the server that UD offloads task to. To reduce the workload when UD evaluate every edge server, the expectation which is made by offloading probability is used. Thus, the expectation of transmission time of task $i$ can be calculated as follow

$$t_i^{trans} = \sum_k pro_k \frac{p_i^{off} s_i}{R_{i,k}} \tag{6}$$

Then the expectation of energy consumption of transmitting the data a from UD $i$ to the edge server is $E_i^{trans}$, which can be given as follows:

$$E_i^{trans} = \sum_k pro_k Po_i t_{i,k}^{trans} \tag{7}$$

The MEC server has the capability to provide computation service to multiple UDs, after receiving the task from a UD, the server will execute the task and return the result back to the UD. An M/M/1 queue is constructed at each edge server since the queuing theory is widely used in MEC [21], [33]–[35]. Then the execution time when task $i$ executed in edge server j can be obtained as follows:

$$t_{i,j}^{exe} = \frac{1}{SR_j - p_i^{off}\lambda_i} \tag{8}$$

Considering UD $i$ can connect several edge servers and UD can offload task with different specific probabilities, the expectation of execution time of task $i$ can be calculated as follow

$$t_i^{exe} = \sum_k pro_k t_{i,k}^{exe} = \sum_k pro_k \frac{1}{SR_k - p_i^{off}\lambda_i} \tag{9}$$

Then, the expectation of execution energy consumption is shown as follows

$$E_i^{exe} = k t_i^{exe} \tag{10}$$

### C. PROBLEM FORMULATION

To this end, the total delay when UD $i$ offloads its task can be obtained. Due to both local and remote executing that can occur simultaneously, the total delay is the largest of the two-time consumption. Given the offloading proportion and transmission power, the total delay generated by UD $i$ when offloading its task in a distributed way is given by

$$T_i^{total} = \max\{t_{i,loc}^{comp}, t_i^{trans} + t_i^{exe}\} \tag{11}$$

The energy consumption of UD$i$ can also be obtained as follows

$$E_i^{total} = E_i^{loc} + E_i^{trans} + E_i^{exe} \tag{12}$$

In the mobile edge system, the UD utility usually consists of the time and energy which a task completed. Therefore, the utility of UD $i$ can be defined as

$$utility_i(p_i^{off}, Po_i) = \alpha \frac{T_i^{\max} - T_i^{total}}{T_i^{\max}} + (1-\alpha)\frac{E_i^{\max} - E_i^{total}}{E_i^{\max}} \tag{13}$$

where $0 < \alpha < 1$ is the weight between time and energy consumption. When $\alpha$ is large, it means that time consumption is more valuable than energy consumption, otherwise, the energy consumption is more important than time consumption. Each UD can set the value of $\alpha$ adjusting different practice situations, saving time or energy consumption.

Through the analysis above, the priori mechanism-based problem of joint task offloading proportion and transmission power is formulated here, which maximize the average utility of UDs. Then the problem is given as follows,

$$\max_{p_i^{off}, Po_i} \quad U = \sum_{i=1}^{M} utility_i(p_i^{off}, Po_i) \tag{14}$$

$$\text{subject to} 0 \le p_i^{off} \le 1 \tag{14a}$$

$$0 < Po_i \le Po_{\max} \tag{14b}$$

$$E_i^{total} < E_i^{\max} \tag{14c}$$

$$T_i^{total} < T_i^{\max} \tag{14d}$$

The constraints in the formulation above can be explained as given: constraint (14a) imply that the offloading proportion is no more lager than 1; constraint (14b) imply that the transmission power should be less than the maximum transmission power of UD; constraint (14c) imply that the energy consumption should be less than the maximum time consumption and constraint (14d) imply that the time consumption should be less than the maximum energy consumption.

## IV. ALGORITHM DESIGN

Under the scenario that UD covered by single or multiple edge servers, each UD should consider the offloading proportion and transmission power under every edge server that UD connected. In that way, which we called posterior offloading mechanism, the workload could be very large if the UD is covered by dense edge servers. Based on this, a priori offloading mechanism is proposed.

*Theorem 1:* The workload of the priori offloading mechanism is less than that of the posterior offloading mechanism.

*Proof:* Let assume that the workload of a UD to evaluate the utility and obtain the best offloading proportion and transmission power at the edge server is $\phi(\phi > 0)$. The total workload of a UD is $\beta\phi$ when the UD connects with $\beta(\beta \ge 1)$ edge server by using a posterior offloading mechanism. In that way, for each additional number of edge server that the UD connects with, the UD needs to evaluate the utility once more, and the workload will be increased once. With a priori offloading mechanism, the total workload of a UD is $\phi$ when the UD connects with $\beta$ edge server. As the increasing number of connected edge servers, the utility of UD changes only in the expectation value of time and energy consumption, but the algorithm only runs for one time. In that way, the number of edge server that connected with the UD has no effect on the number of times the algorithm is run, the algorithm runs only once, so the workload is $\beta$. Thus the workload of priori offloading mechanism is less than that of posterior offloading mechanism due to $\phi > 0$ and $\beta \ge 1$.

Through a detailed analysis, we propose an algorithm called a distributed PRiori Offloading Mechanism with joint Offloading proportion and Transmission (PROMOT) power algorithm to maximize the UD utility. The PROMOT algorithm is consisting of three phases: First, the UD gets the network information includes the number of connected edge servers, the position of those edge server, the capacity of those edge server etc. Based on those network information, the UD can calculate the offloading probability of each connected edge server. The expectation value is also calculated and UD forms a utility function with offloading proportion and transmission power. Then, the UD transmits the utility function to the nearest edge server. The Genetic Algorithm is running at edge server for searching the best offloading proportion and transmission power. Finally, the nearest edge server feedback the results to the UD. The UD generates a random number which is corresponding to the offloading probability among connected edge servers. The UD begins to offload task according to the random number with the best offloading proportion and transmission power. Specifically, the PROMOT algorithm is always running at the beginning of a time slot, which means the offloading proportion and transmission power keeps unchanged until the next time slot begins. The flowchart is described in Figure 2.

*Example* The UD connected with two edge servers, named edge server A and edge server B. The offloading probability of edge server A and B are calculated as $pro_A$ and $pro_B(pro_A + pro_B = 1)$, respectively. After the best offloading proportion and transmission power are obtained, the UD generates a random number $ran_{UD}(0 < ran_{UD} < 1)$. If $0 < ran_{UD} \le pro_A$, then the UD will offload task to edge server A, otherwise, $pro_A < ran_{UD} \le (pro_A + pro_B)$, then the UD will offload task to edge server B. If the number of edge servers is three or more than three, the interval from 0 to 1 can be divided into three or more probability intervals.

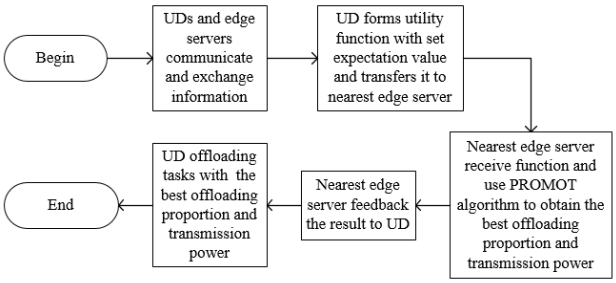


**FIGURE 2.** The flowchart of the PROMOT algorithm.

The pseudo-code of the PROMOT algorithm can be described in algorithm 1.

**Algorithm 1** Proposed PROMOT Algorithm

**Input:** network information, task's information, UD's information, edge servers' information.

**Output:** offloading proportion, transmission power

**Initialization** with task set, server set, information broadcast
UD $i$ calculates the offloading probability, expectation value of time and energy consumption, forms a utility function and transfers it to nearest edge servers
//using GA algorithm to search the optimal offloading proportion and transmission power
**Initialization** with precision $\varepsilon$, Maximum Generation (MG), crossover probability (CP), mutation probability (MP), generate first generation population (Pop),Population Size (PS).
**while** evolution generation $<$ MG:
  decode the population to obtain the representation
  get utility value U($i$) and cumulative probability value
  select a new population
  cross new population according to CP
  mutate operation according to MP
  merge parents and children into new population
  final decode
  utility value U($i$) assessment
  select the greatest utility value U($i$) between populations
  find the maximum utility value in the round
**end while**

The pseudo-code for solve the problem is summarized in algorithm 1, where $\varepsilon = 0.001$, maximum generation is set as 50, crossover probability is set as 0.8, mutation probability is set as 0.01, and the number of newly formed population is set as 50. According to Figure 3, we can see that the algorithm is converged at 20-30 generations. In Figure 3, NR means Number time of algorithm Runs, we run the PROMOT algorithm for 5, 10, 20 times, and the algorithm

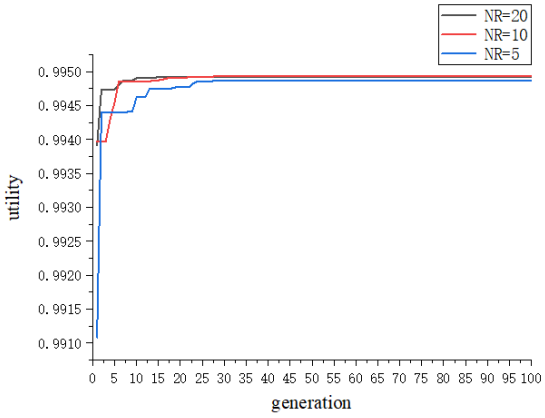


**FIGURE 3.** Convergence of the PROMOT algorithm.

converges in the twenty to thirty generations. Thus we set the maximum generation as 50, and our algorithm has great convergence according to Figure 3. From the proposed PROMOT Algorithm, we can catch the optimal offloading proportion and transmission power for each UD to maximize the utility function.

The major advantage of the Genetic Algorithm is the ability to search quickly, randomly, and independently of the problem domain [36]. The search starts from the population, which has the potential parallelism. The Genetic Algorithm can carry on many individuals simultaneously the comparison and the process is simple. It has the randomness by using the probability mechanism to carry on the iteration. It is extensible and easy to be combined with other algorithms.

## V. EXPERIMENTAL RESULTS

In this section, we investigate the performance of our system for validating the efficiency of the offered algorithm for the joint time and energy optimization problem. We simulated the experiment on a windows operating system that contains CPU with dual-core, eight gigabytes (GB) of memory and one terabyte of second storage memory. The python language is the programming language we use. Here, ten edge servers are install in the range of 50m $\times$ 50m. The user devices and edge servers use a single antenna for communicating. We also assume that the maximum latency and energy consumption in our system are 2s and 15J respectively [35]. The rest main parameters can be found in Table 2.

The utility performance of our scheme strategy is compared with the following method.

1) Optimizing the offloading Proportion and transmission Power withOut Queue theory method (PPOQ) [12]: the queue theory is not adopted. The optimization variables are the offloading proportion and transmission power. In that case, the edge server only maintains a normal queue

2) Optimizing the Offloading Proportion with Queue theory and Random Transmission Power (OPQR) method: queuing theory is adopted and the offloading proportion is optimized. The transmission power is randomly allocated.

**TABLE 2. Simulation parameters.**

| Parameters | value |
|---|---|
| $\lambda_i$ | 1.5 MIPS |
| $s_i$ | $[2.3,4.2]\times10^6$ bits |
| $f_{loc}$ | 1GHz |
| $c_i$ | 2000 Megacycles |
| $\kappa$ | $5\times10^{-27}$ |
| $k$ | 16 [35] |
| $B$ | 20MHz |
| $N_0$ | -110dBm |
| $Po_{\max}$ | 20dBm |
| $h_{i,j}$ | $140.7+36.7\log_{10} d$ [17] |
| $SR$ | [2.5, 4.5]MIPS |

3) Optimizing the Offloading Proportion withOut Queuing theory and Random transmission power method (OPOQR) [16]: the queuing theory is not adopted and the transmission power is randomly allocated. The optimization variable is only the offloading proportion.

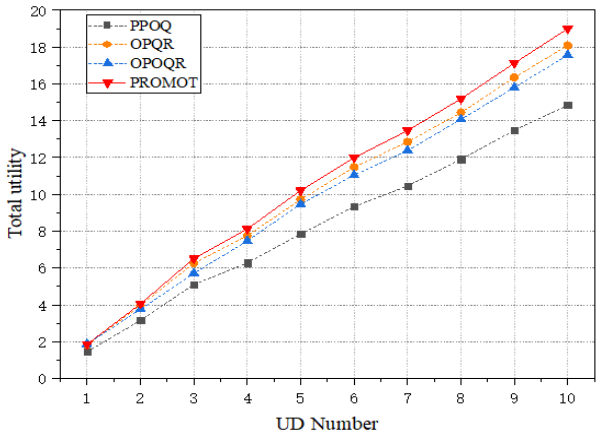


**FIGURE 4. Comparison of total system against different UD number.**

First, we investigate the impact of increasing number on system utility performance. From Figure 4, we can see that, with the increasing number of UD, the system utility is always higher than the other three methods. When the number of UD is one or two, the utility is very close to OPQR method, however, the utility gap between PROMOT and OPQR is getting larger and larger. In that case, the scalability is good.

As shown in Figure 5, the trend of average utility with increasing task arrival rate can be investigated after the number of UD is fixed. As the task arrival rate increases, the overall trend is downward, which means the larger task arrival rate, the lower average utility. In that case, the execution time and energy will increase with the task arrival rate increases due to the M/M/1 queuing model adopted at the edge server. From the Figure, the utility of NQNE and NEBE methods is lower than QBNE and PROMOT method, which means the M/M/1 queuing model can effectively improve the system utility. And the system utility of the PROMOT method

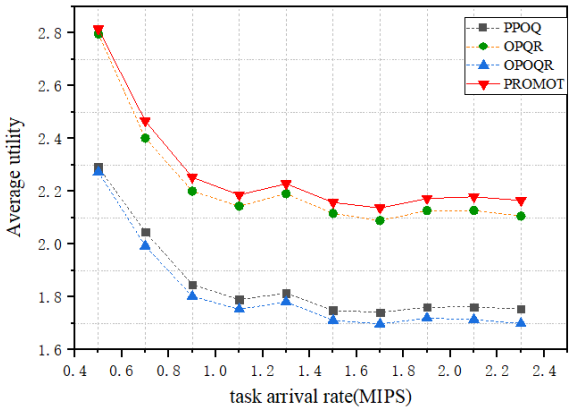


**FIGURE 5. The impact of task arrival rate, UD = 10.**

is higher than that of QBNE method because the transmission power is optimized.

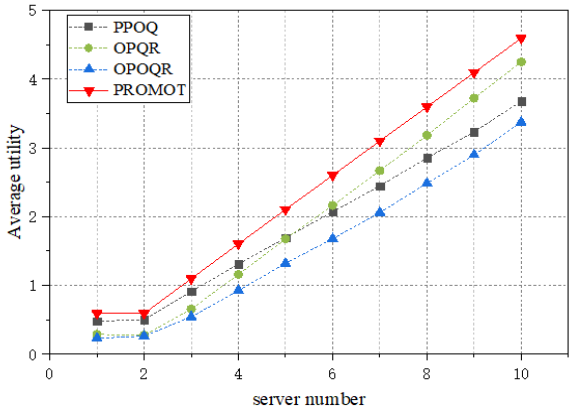


**FIGURE 6. The impact of server number on average utility, UD = 10.**

As shown in Figure 6, the average utility of four methods is shown as an uptrend with the increasing of the number of edge server, which means the average increases with the increasing of the number of edge server. In that case, UDs have more choice to offload tasks for UDs are covered by multiple edge server. And the average utility of PROMOT method is always higher than that of the other three methods. As the number of the edge server is six, the utility of the QBNE method is higher than that of the NQBE method, which means queue theory can improve the utility of UDs.

Then, we examine the data size impact on system utility. As shown in Figure 7, the average utility is a general trend of decline with the increase of data size, and the average utility of PROMOT method is always higher than that of the other three methods. This implies that the task with a small size is benefit more from offloading than those with large data size do.

From Figure 8(a) and 8(b), the time consumption and energy consumption with different weight is investigated, which the weight $\alpha$ is changed from 0.1 to 0.9. It can be seen that the energy consumption decrease when the weight $\alpha$ increases and the time consumption increases with the

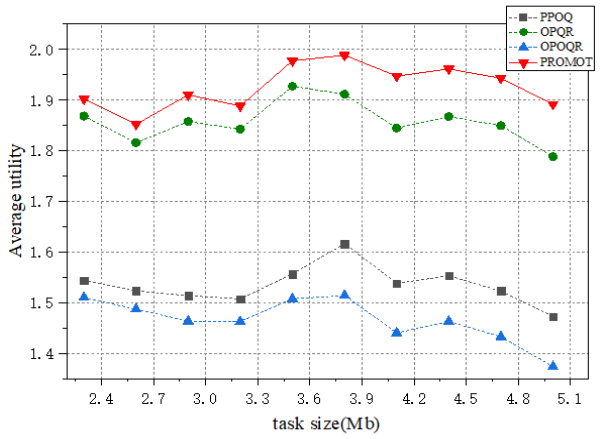


**FIGURE 7.** The impact of data size on average utility.

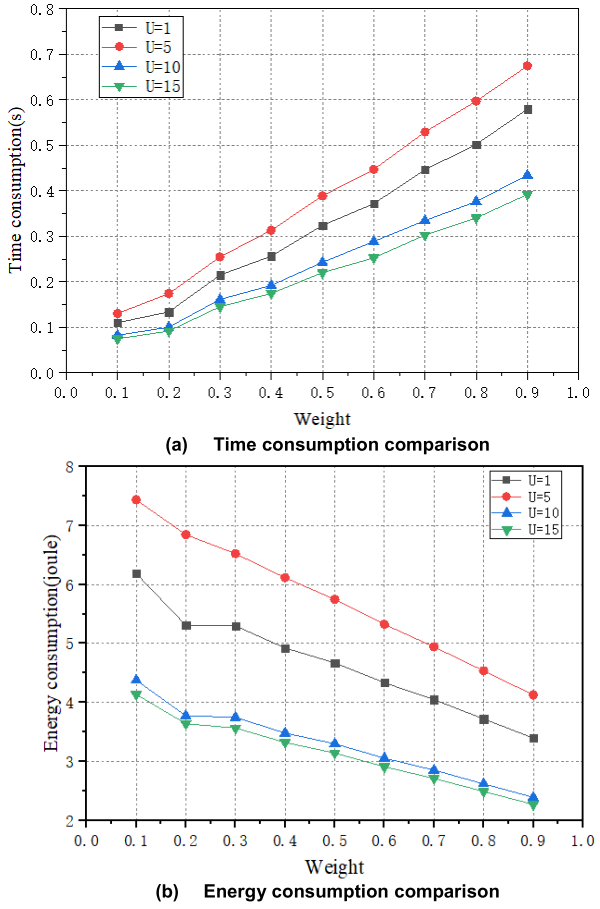


**FIGURE 8.** The impact of weight on time and energy consumption.

weight $\alpha$ increases. This is because UD takes more attention to energy consumption when the weight $\alpha$ increases, which means energy consumption is more important than time consumption.

## VI. CONCLUSION AND FUTURE WORK

In this work, we investigated the mobile edge computing (MEC) model which UDs are covered by single or multiple edge servers. In that case, we proposed a probability preferred priori offloading mechanism with joint optimized offloading proportion and transmission power. A utility function with UD's time and energy consumption is formed, then the GA based PROMOT algorithm is used to solve the problem and the best offloading proportion and transmission power can be obtained for maximizing the UD utility.

We only consider offloading task to one single edge server under the scenario that UDs are covered by single or multiple edge server(s) in this paper, and if UD is covered by multiple edge servers, we can split a task into multiple subtasks, and offloading those subtasks in a distributed way. In that case, we need to calculate the best proportion for each subtask and find a unified transmission power. A device-to-device (D2D) offloading is not considered in this paper. Thus, we can add a D2D offloading in the future work for some UDs have more enough computation capacities.

## REFERENCES

[1] M. Z. Ge, H. Bangui, and B. Buhnova, "Big data for Internet of Things: A survey," *Future Gener. Comput. Syst.*, vol. 87, pp. 601–614, Oct. 2018.
[2] E. Ahmed and M. H. Rehmani, "Mobile edge computing: Opportunities, solutions, and challenges," *Future Gener. Comput. Syst.*, vol. 70, pp. 59–63, May 2017.
[3] W. Yu, F. Liang, X. F. He, W. G. Hatcher, C. Lu, J. Lin, and X. Y. Yang, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, Nov. 2017.
[4] W. Li, Z. Chen, X. Gao, W. Liu, and J. Wang, "Multimodel framework for indoor localization under mobile edge computing environment," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4844–4853, Jun. 2019.
[5] D. Cao, Y. Liu, X. Ma, J. Wang, B. Ji, C. Feng, and J. Si, "A relay-node selection on curve road in vehicular networks," *IEEE Access*, vol. 7, pp. 12714–12728, 2019.
[6] X. He, H. Xing, Y. Chen, and A. Nallanathan, "Energy-efficient mobile-edge computation offloading for applications with shared data," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.
[7] B. Gu, Y. Chen, H. Liao, Z. Zhou, and D. Zhang, "A distributed and context-aware task assignment mechanism for collaborative mobile edge computing," *Sensors*, vol. 18, no. 8, p. 2423, Jul. 2018.
[8] Y. He, J. Ren, G. Yu, and Y. Cai, "D2D communications meet mobile edge computing for enhanced computation capacity in cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1750–1763, Mar. 2019.
[9] S. Mu, Z. Zhong, D. Zhao, and M. Ni, "Latency constrained partial offloading and subcarrier allocations in small cell networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–7.
[10] J. Xu, L. Chen, and P. Zhou, "Joint service caching and task offloading for mobile edge computing in dense networks," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Honolulu, HI, USA, Apr. 2018, pp. 207–215.
[11] F. Yu, H. Chen, and J. Xu, "DMPO: Dynamic mobility-aware partial offloading in mobile edge computing," *Future Gener. Comput. Syst.*, vol. 89, pp. 722–735, Dec. 2018.
[12] Z. Kuang, L. Li, J. Gao, L. Zhao, and A. Liu, "Partial offloading scheduling and power allocation for mobile edge computing systems," *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6774–6785, Aug. 2019.
[13] J. Wang, W. Wu, Z. Liao, A. K. Sangaiah, and R. S. Sherratt, "An energy-efficient off-loading scheme for low latency in collaborative edge computing," *IEEE Access*, vol. 7, pp. 149182–149190, Oct. 2019.
[14] Z. Ning, P. Dong, X. Kong, and F. Xia, "A cooperative partial computation offloading scheme for mobile edge computing enabled Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4804–4814, Jun. 2019.
[15] Q. Tang, H. Lyu, G. Han, J. Wang, and K. Wang, "Partial offloading strategy for mobile edge computing considering mixed overhead of time and energy," *Neural Comput. Appl.*, Aug. 2019.
[16] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5506–5519, Aug. 2018.
[17] T. X. Tran and D. Pompili, "Joint task offloading and resource allocation for multi-server mobile-edge computing networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 1, pp. 856–868, Jan. 2019.

[18] Y. Zhang, P. Du, J. Wang, T. Ba, R. Ding, and N. Xin, "Resource scheduling for delay minimization in multi-server cellular edge computing systems," *IEEE Access*, vol. 7, pp. 86265–86273, 2019.
[19] X. Yang, X. Yu, H. Huang, and H. Zhu, "Energy efficiency based joint computation offloading and resource allocation in multi-access MEC systems," *IEEE Access*, vol. 7, pp. 117054–117062, Aug. 2019.
[20] L. F. Gao and M. Moh, "Joint computation offloading and prioritized scheduling in mobile edge computing," in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Orléans, France, Oct. 2018, pp. 1000–1007.
[21] L. Yuchong, W. Jigang, W. Yalan, and C. Long, "Task scheduling in mobile edge computing with stochastic requests and M/M/1 servers," in *Proc. IEEE 21st Int. Conf. High Perform. Comput. Commun.; IEEE 17th Int. Conf. Smart City; IEEE 5th Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Zhangjiajie, China, Aug. 2019, pp. 2379–2382.
[22] W. Chen, D. Wang, and K. Li, "Multi-user multi-task computation offloading in green mobile edge cloud computing," *IEEE Trans. Services Comput.*, vol. 12, no. 5, pp. 726–738, Sep. 2019.
[23] Y. Chen, N. Zhang, Y. Zhang, X. Chen, W. Wu, and X. S. Shen, "Energy efficient dynamic offloading in mobile edge computing for Internet of Things," *IEEE Trans. Cloud Comput.*, to be published.
[24] T. Wang, X. Wei, C. Tang, and J. Fan, "Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints," *Peer-Peer Netw. Appl.*, vol. 11, no. 4, pp. 793–807, May 2017.
[25] J. Wang, Y. Gao, X. Yin, F. Li, and H.-J. Kim, "An enhanced PEGASIS algorithm with mobile sink support for wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–9, Dec. 2018.
[26] J. Wang, X. J. Gu, W. Liu, A. K. Sangaiah, H. J. Kim, "An empower hamilton loop based data collection algorithm with mobile agent for WSNs," *Hum.-Centric Comput. Inf. Sci.*, vol. 9, pp. 1–9, May 2019.
[27] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S. J. Lim, "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2579, Jun. 2019.
[28] J. Wang, Y. Gao, C. Zhou, R. Simon Sherratt, and L. Wang, "Optimal coverage multi-path scheduling scheme with multiple mobile sinks for WSNs," *Comput., Mater. Continua*, vol. 61, no. 3, pp. 695–711, 2019.
[29] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Trans. Cloud Comput.*, vol. 5, no. 4, pp. 725–737, Oct. 2017.
[30] X. Chen, "Decentralized computation offloading game for mobile cloud computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 4, pp. 974–983, Apr. 2015.
[31] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
[32] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Upper Saddle River, NJ, USA: Prentice-Hall, 1996.
[33] D. Li, B. Dong, E. Wang, and M. Zhu, "A study on flat and hierarchical system deployment for edge computing," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Las Vegas, NV, USA, Jan. 2019, pp. 163–169.
[34] X. Li, C. Zhang, B. Gu, K. Yamori, and Y. Tanaka, "Optimal pricing and service selection in the mobile cloud architectures," *IEEE Access*, vol. 7, pp. 43564–43572, 2019.
[35] L. Liu, Z. Chang, X. Guo, S. Mao, and T. Ristaniemi, "Multiobjective optimization for computation offloading in fog computing," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 283–294, Feb. 2018.
[36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

**JIN WANG** (Senior Member, IEEE) received the B.S. and M.S. degrees from the Nanjing University of Posts and Telecommunications, China, in 2002 and 2005, respectively, and the Ph.D. degree from Kyung Hee University, South Korea, in 2010. He is currently a Professor with the Changsha University of Science and Technology. He has published more than 300 international journal and conference papers. His research interests mainly include wireless sensor networks, and network performance analysis and optimization. He is a member of ACM.

**WENBING WU** received the B.S. degree from Hunan Agricultural University, China, in 2018. He is currently pursuing the master's degree with the Changsha University of Science and Technology, Changsha, China. His research interests include mobile edge computing, fog computing, and machine learning. He is good at Python and C, and familiar with Linux OS.

**ZHUOFAN LIAO** (Member, IEEE) received the Ph.D. degree in computer science from Central South University, China, in 2012. Supported by China Scholarship Council, she was a Visiting Scholar with the University of Victoria, from 2017 to 2018. She is currently an Assistant Professor with the School of Computer and Communication Engineering, Changsha University of Science and Technology, China. Her research interests include wireless networks optimization, big data, and edge computing. She has published articles in leading transaction and top conferences as the first author in the above areas. She has served as the reviewer of top transaction, such as TPDS, TVC, and so on. She is a member of the China Computer Federation.

**R. SIMON SHERRATT** (Fellow, IEEE) received the B.Eng. degree in electronic systems and control engineering from Sheffield City Polytechnic, U.K., in 1992, and the M.Sc. degree in data telecommunications and the Ph.D. degree in video signal processing from the University of Salford, in 1994 and 1996, respectively. In 1996, he has appointed as a Lecturer in electronic engineering at The University of Reading, where he is currently a Professor of consumer electronics and the Head of the wireless and computing research. He is also a Guest Professor with the Nanjing University of Information Science and Technology, China. His research topic is on signal processing in consumer electronic devices concentrating on equalization and DSP architectures, specifically for personal area networks, USB, and Wireless USB. He has served the IEEE Consumer Electronics Society as the Vice President (Conferences), in 2008 and 2009, an AdCom Member, from 2003 to 2008 and since 2010, and the Awards Chair, in 2006 and 2007. He received the IEEE Chester Sall First Place Best Transactions on Consumer Electronics Paper Award, in 2004, and the Best Paper Award in the IEEE International Symposium on Consumer Electronics, in 2006. He served as the IEEE International Conference on Consumer Electronics General Chair, in 2009, and the IEEE International Symposium on Consumer Electronics General Chair, in 2004. He has been a member of the IEEE Transactions on Consumer Electronics Editorial Board, since 2004, and the Editor-in-Chief, since 2011.

**GWANG-JUN KIM** received the B.E., M.E., and Ph.D. degrees in computer engineering from Chosun University, in 1993, 1995, and 2000, respectively. From 2000 to 2001, he was a Researcher with the Department of Electrical and Computer Engineering, University of California Irvine. He joined the Department of Computer Engineering, Chonnam National University, in 2003, and became an Associate Professor, in 2009. Since 2015, he has been a Professor in computer engineering with Chonnam National University. His current research interests include the area sensor networks, the IoT, real-time communication, and various kinds of communication systems.

**OSAMA ALFARRAJ** received the master's and Ph.D. degrees in information and communication technology (ICT) from Griffith University, in 2008 and 2013, respectively. His doctoral dissertation investigates the factors influencing the development of e-Government in Saudi Arabia, and it is a qualitative investigation of the developers' perspectives. He is currently an Associate Professor with the ICT, King Saud University, Riyadh, Saudi Arabia. His research interests include electronic commerce, M-government, the Internet of Things, cloud computing, and big data analytics.

**AHMAD ALZUBI** received the Ph.D. degree in computer networks engineering from the National Technical University of Ukraine (Ukraine), in 1999. He is currently a Professor with King Saud University (KSU). His current research interests include computer networks, grid computing, cloud computing, big data, and data extracting. He also served for three years as a consultant and a member of the Saudi National Team for measuring e-Government in Saudi Arabia.

**AMR TOLBA** received the M.Sc. and Ph.D. degrees from the Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Egypt, in 2002 and 2006, respectively. He is currently an Associate Professor with the Faculty of Science, Menoufia University. He is on leave from Menoufia University to the Department of Computer Science, Community College, King Saud University (KSU), Saudi Arabia. He has authored or coauthored over 50 scientific articles in top ranked (ISI) international journals and conference proceedings. His main research interests include socially aware networks, vehicular ad hoc networks, the Internet of Things, intelligent systems, and cloud computing.