2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems

Task Scheduling in Mobile Edge Computing with Stochastic Requests and M/M/1 Servers

Yuchong Luo, Jigang Wu, Yalan Wu, Long Chen
School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, P.R China, 510006

***Abstract*—A multi-user mobile edge computing system with stochastic requests and M/M/1 queuing based servers is proposed in this paper. The problem of minimizing the total response time of all tasks is formulated, which is proved to be NP-complete. A greedy algorithm is proposed to solve the mentioned optimization problem, which prefers to assign task to the server with minimum response time for the task. Simulation results show that, the average response time of tasks in the proposed greedy algorithm is saved by $20\% - 30\%$, in comparison to the proposed random algorithm. Meanwhile, the average response time of tasks in customized tabu search is decreased by 8.5%, compared to the greedy algorithm.**

***Index Terms*—mobile edge computing, task scheduling, M/M/1 queuing system.**

I. Introduction

Mobile edge computing (MEC) is a promising technology, in which the edge servers are deployed in base station (BS) to reduce the latency of offloading tasks and energy consumption of mobile devices [1]. Task offloading decided by mobile device is based on current network environment and resources (*i.e.*wireless channels, available resources of edge servers). However, due to mobile devices lack accurate information of MEC system, tasks may not be scheduled to appropriate edge servers which will affect quality of service (QoS) of mobile devices [2] [3]. Hence, in order to improve QoS of mobile devices, an effective task scheduling strategy is key point in MEC system.

Most of earlier works have focused on energy consumption [4] of mobile devices, latency of tasks [5] and resource allocation [6] in MEC system. Actually, energy consumption of edge servers is a major portion of MEC system. The traffic of BS will change periodically caused by the different time periods. BS can switch work state (ON) or sleep state (OFF) dynamically to save energy according to traffic changes [7]. However, it is a challenge to ensure tasks are executed in time and achieve load balancing of system by scheduling extensive tasks executed in edge servers.

To deal with this challenge, we investigate resource allocation in MEC system with stochastic tasks in this paper. The contributions are summarized as follows.

- We model stochastic user requests as an independent Poisson stream and model edge servers as a simple M/M/1 queuing system.
- We prove this optimization problem is a NP-complete problem and propose greedy, customized tabu search algorithm to find a feasible solution.
- Simulation results demonstrate that the average response time of obtained by greedy algorithm is saved about $20\% - 30\%$, in comparison to the random algorithm. The average response time of obtained by customized tabu search algorithm is decreased by 8.5%, compare to greedy algorithm.

II. System Model

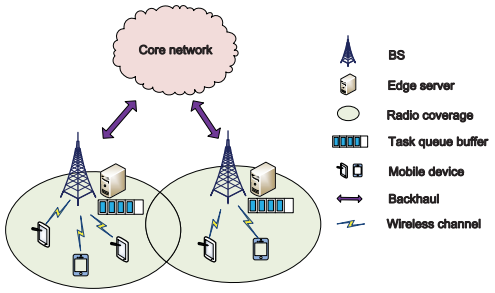

Fig. 1. Mobile edge computing system.

As illustrated in Fig. 1, we consider a multi-user MEC system consists of n mobile devices and m edge servers. Each server has a task queue buffer. Mobile devices migrate the computation tasks to edge servers through the wireless channels. We use random variables T^j_{ON}, T^j_{OFF} to denote the probability density functions of ON period and OFF period, respectively. Assumed that ON periods and OFF periods are independent and identically distributed (i.i.d.). We also assumed that ON and OFF periods are independent of each other [8]. Thus, F_j is the fraction of time when server s_j is ON state, which is defined as follows.

$$F_j = E[T^j_{ON}]/(E[T^j_{ON}] + E[T^j_{OFF}]). \tag{1}$$

A. Network Model

In MEC system, let $D = \{d_1, d_2, \cdots, d_n\}$, where D is the set of mobile devices and d_i is the ith mobile device. Let $S = \{s_1, s_2, \cdots, s_m\}$, where S is the set of edge servers and s_j is the jth edge server. Each task in mobile device d_i can be denoted as t_i. Assume that the task t_i obeys an independent Poisson stream with parameter λ_i, where λ_i is the request rate of mobile device d_i. β_i denotes the data size of task t_i. w_i is the workload of task t_i. The edge servers

978-1-7281-2058-4/19/$31.00 ©2019 IEEE
DOI 10.1109/HPCC/SmartCity/DSS.2019.00331

are modeled as simple M/M/1 queuing system and the server time obey the exponential distribution with common service rate μ_j. C^j_{max} denotes the Maximum capacity of task queue buffer. Let L_{ij} denotes the transmission delay between d_i and s_j. To guarantee each task is executed in time, there is an upper bound W^{up} for the expected waiting time of tasks.

In MEC system, mobile device offloads the computational tasks to edge servers through wireless channel. Thus, we can compute the data rate of task t_i as follows.

$$R_i = Blog_2(1 + \frac{P_i H_i}{\sigma B}), \tag{2}$$

where B is channel bandwidth, P_i denotes the transmission power of task i, H_i denotes the channel power gain for task i, and σ denotes the noise power spectral density at the receiver.

B. Problem Formulation

We use a binary decision variable a_{ij} to represent which task t_i generated by d_i is assigned to server s_j. If t_i is scheduled to s_j, $a_{ij} = 1$. Otherwise, $a_{ij} = 0$.

Due to servers have two work state ON and OFF, we assume that the computation tasks from mobile devices only offloaded to the ON state server. Therefore, tasks arrived at server s_j in unit time can be formulated as,

$$A_j = \sum_{i,d_i \in D}^{n} a_{ij} \lambda_i F_j. \tag{3}$$

The delay between task t_i and server s_j can be calculated as $L_{ij} = \beta_{ij}/R_i$. For all tasks that arrive in the MEC system at the same time, the total transmission time of these tasks is given by,

$$T^{tr} = \sum_{i,d_i \in D}^{n} \sum_{j,s_j \in S}^{m} a_{ij} \lambda_i F_j L_{ij}. \tag{4}$$

The expected waiting time W_j of task at the queue buffer of server s_j is $1/(\mu_j - A_j)$. Accumulate the waiting time of tasks for each edge server in the MEC system. Then, the total waiting time of these tasks in unit time is given as,

$$W^{total} = \sum_{i,d_i \in D}^{n} \sum_{j,s_j \in S}^{m} a_{ij} \lambda_i F_j W_j. \tag{5}$$

Compared to cloud centers, edge servers have limited resources. The tasks executed by servers need some computation time. Let f_j denote the clock frequency of server j. The computational time of task t_i can be calculated as $T_c = w_i/f_j$. Hence, The computational time of tasks in edge servers is given as,

$$T^{com} = \sum_{i,d_i \in D}^{n} \sum_{j,s_j \in S}^{m} a_{ij} \frac{w_i}{f_j}. \tag{6}$$

Thus, The total response time of these tasks in unit time is given as,

$$T^{total} = T^{tr} + W^{total} + T^{com}. \tag{7}$$

In this paper, we minimize the total response time of these tasks in MEC system. We formulate the constrained minimization problem as,

$$\textbf{OPT} \quad \min \quad T^{total}, \tag{8}$$

$$\textbf{s.t.} \quad \sum_{j,s_j \in S}^{m} a_{ij} = 1, \tag{9}$$

$$\sum_{i,d_i \in D}^{n} a_{ij} \beta_i \leq C^j_{max}, \tag{10}$$

$$a_{ij} w_i \leq f_j t, \tag{11}$$

$$\sum_{i,d_i \in D}^{n} \sum_{j,s_j \in S}^{m} a_{ij} F_j (\mu_j - \lambda_i - \frac{1}{W^{up}}) \geq 0, \tag{12}$$

$$a_{ij} \in \{0, 1\}. \tag{13}$$

The objective is given in (8), which is to minimize the total response time of tasks. Constraint (9) ensures that each task only can be assigned to one server. Constraint (10) ensures for the data-size of tasks in each server do not exceeding the capacity of task queue buffer. Constraint (11) ensures that workload of task do not exceeding the workload of server in a time slot t. Constraint (12) ensures that the expected waiting time at any server does not exceed W^{up}. Constraint (13) ensures the binary decision variable a_{ij}.

Theorem 1 : The proposed problem **OPT** is an NP-complete problem.

Proof : We consider this problem without constraints. the original problem can be reduce to that tasks are scheduled to be placed in the queue buffer of the edge server. With the limited capacity of queue buffer, place these tasks to minimize the total time for all offloading tasks. Obviously, This is equivalent to a special Knapsack problem. Knapsack problem has been proved to be NP-complete. The problem **OPT** has several constraints which is more complicated than Knapsack problem. Thus, the problem **OPT** is proved as NP-complete.

III. Algorithm Design

The problem **OPT** is an NP-complete problem, which is infeasible to find an optimal solution. Thus, we propose two algorithms, greedy (GD) algorithm and customized tabu search (TS) algorithm, to solve the problem.

A. Greedy Algorithm

Before the tasks assignment, We get the number of ON state servers by probability. We select the tasks to all ON state servers, and calculate the transmission delay between them. when the task i is assigned to the server j, we can calculate the computation time in server j. We can select the minimum total time of task in each ON state server j. If the minimum total time of task do not meet the constraint, we can select the second minimum until the traverse is over. In our constraint,

each task can only be assigned to one server, and the expected waiting time at any server does not exceed W^{up}.

B. Customized Tabu Search Algorithm

Once the offloading policy of tasks and solution are obtained by the greedy algorithm, the customized tabu search algorithm is implemented. In each iteration, we use domain function to swap the values in the offloading policy as much as possible to get the candidate solutions under the current policy. Then, it is judged that whether the candidate solution is better than the current optimal solution. If the candidate solution is better than the current optimal solution, replace the original current optimal solution with the candidate solution as the new current optimal solution. If not, select the offloading policy of task that is not in the tabu table as the current solution for the next iteration, and update the solution corresponding to the policy as the current optimal solution. Repeat the above steps until the conditions are met.

We introduce the following notations in Algorithm 1. Q is the offloading policy of tasks obtained by the greedy algorithm. L is the solution obtained by the greedy algorithm. Q^{bst} is to save the current optimal policy. Q^{tmp} is to save a better policy in iterations. Q_y is candidate policy not in tabu table. $temp$ is to save a better solution in iterations. Y is a tabu table. K is the threshold for algorithm termination. H is the threshold for termination during sub-iteration.

Algorithm 1 Customized Tabu Search (TS) Algorithm

Initialization:

Set Q, Q^{bst}, Q^{tmp}, L, Y, H, K, $temp$;

Iteration:

$temp \longleftarrow L$, $Q^{tmp} \longleftarrow Q$, $Q^{bst} \longleftarrow Q$;
while $k < K$ **do**
  while $h < H$ **do**
    Randomly swap the value in Q to get $Q^{'}$;
    if $Q^{'}$ not in Y **then**
      Compute the solution of current policy l;
      if $l < temp$ **then**
        $temp \longleftarrow l$, Save $Q^{'}$ in Y;
        $Q^{tmp} \longleftarrow Q^{'}$, update Y;
      else
        Select Q_y not in Y, $Q^{tmp} \longleftarrow Q_y$;
        update the solution of Q_y to $temp$;
      end if
    end if
    $k \longleftarrow k + 1$;
  end while
  if $temp < L$ **then**
    $L \longleftarrow temp$, $Q^{bst} \longleftarrow Q^{tmp}$;
  end if
  $h \longleftarrow h + 1$.
end while

IV. Numerical Results

In this section, we conduct simulation experiments over a topology of MEC system, and evaluate the efficiency of algorithm GD and TS. We simulate the experiment using Matlab R 2014a. In our simulation, the parameters we used refer to [9] [10]. In order to facilitate calculation, the unit of time used in this experiment is milliseconds(ms). Thus, We consider a MEC system which has 8 servers. The number of mobile device is between 10 and 30. The common service rate μ_j is between 250 and 330. F_j is between is 0.5 and 0.9. The capacity of task queue buffer is between 5 and 10. λ_i is between 25 and 45. The data rate of task i is 50 kB/s and 100 kB/s. The data size of task is between 0.5 KB and 4 KB. The workload of task is between 30 cycles and 50 cycles. The CPU frequency of the server is between 1800 MHz and 3000 MHz. W^{up} is between 30 ms and 50 ms.

In this section, we evaluate the performance of three algorithms. The results are shown in Fig. 2, as the server rate increases, the total delay of tasks with three algorithms have a linear downward trend. This is because that as the server rate increases, the service capacity of servers are increasing in unit time, then the total time of tasks is decreasing. Since the service rate is 290, the decreasing rate of algorithm GD is faster than the random algorithm, while the decrease rate of algorithm TS is slower than that of algorithm GD. The total response time obtained by algorithm GD is saved by 18%, compare to random algorithm, and the total response time obtained by algorithm TS is decreased by 9.3%, compare to algorithm GD.

The results are shown in Fig. 3, the total time of tasks increases following the increasement of queue length. When the length of queue is 7, the increasing rate of time is gradually slowed down. After the length of queue is 7, the total response time of tasks obtained by the algorithm GD is basically unchanged. The total response time obtained by algorithm GD is saved about 20%, compare to the random algorithm. The length of queue between 5 and 6, the increasing rate of algorithm is less improved than algorithm GD. Starting from the length of queue is 7, the total response time obtained by TS algorithm is decreased by 8.5%, compare to algorithm GD. It can be learned from formula (5) that, when the service rate is constant, as the length of queue increases, the number of tasks in the queue would increase, resulting in increasement of the waiting time for tasks.

We investigate the relationship between the number of tasks and the performance of three algorithms. The results are shown in Fig. 4. As the number of tasks increases, the time-saving performance of algorithm GD is gradually increased compared with the random algorithm. The time-saving performance of the algorithm TS is gradually increased compared with algorithm GD. As the number of tasks increases, the time-saving performance of algorithm TS becomes more efficient. Algorithm TS is suitable for scenarios with large number of tasks.

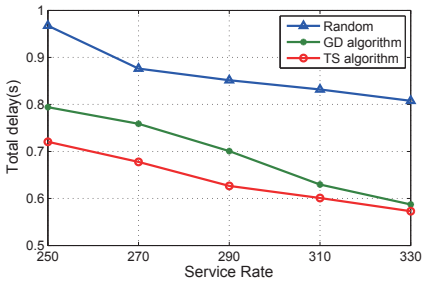

Fig. 2. Total delay in different service rates.

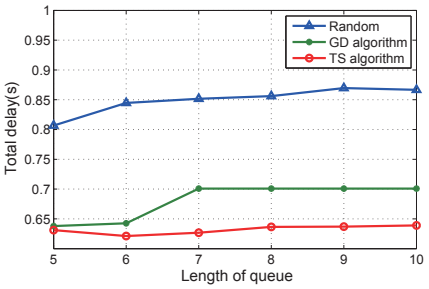

Fig. 3. Total delay in different length of queues.

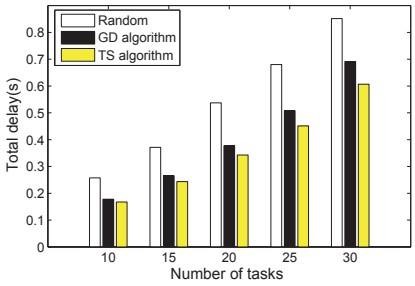

Fig. 4. Total delay in different number of tasks.

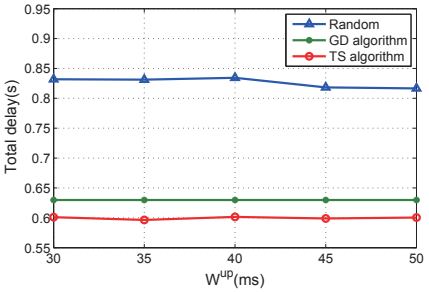

Fig. 5. Total delay in different W^{up}.

We study the relationship between the upper bound of task's waiting time and the total time of tasks. The results are shown in Fig. 5. As the waiting time upper bound of tasks in queue buffer increases, the total response time of tasks remains essentially the same. The total response time obtained by algorithm GD is saved about 23%, compare to random algorithm. We find that the upper bound of task's waiting time in queue buffer are not a factor affecting the total time of tasks.

V. Conclusion

In this paper, we study a multi-user MEC system with stochastic tasks to optimize the total response time of tasks. In order to simulate a system service scenario, we model the edge servers as a simple M/M/1 system. We prove this optimization problem is a NP-complete problem. Two heuristic algorithms are proposed to find a feasible solution. Simulation results show that the average response time of tasks in the proposed greedy algorithm is saved by $20\% - 30\%$, in comparison to the proposed random algorithm, and the average response time of tasks in customized tabu Search is decreased by 8.5%, compared to the greedy algorithm.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Grant No. 61672171, No.61702115, Guangxi Natural Science Foundation of China under Grant No. 2018GXNSFAA138082, Guangxi Key Laboratory of Cryptography and Information Security under Grant No. GCIS201816. The corresponding author is Jigang Wu (asjgwucn@outlook.com).

References

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[2] X. Lyu, W. Ni, H. Tian, R. P. Liu, X. Wang, G. B. Giannakis, and A. Paulraj, "Optimal schedule of mobile edge computing for internet of things using partial information," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2606–2615, 2017.

[3] Y. Sun, S. Zhou, and J. Xu, "Emm: Energy-aware mobility management for mobile edge computing in ultra dense networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2637–2646, 2017.

[4] X. Lyu, H. Tian, W. Ni, Y. Zhang, P. Zhang, and R. P. Liu, "Energy-efficient admission of delay-sensitive tasks for mobile edge computing," *IEEE Transactions on Communications*, vol. 66, no. 6, pp. 2603–2616, 2018.

[5] J. Ren, G. Yu, Y. Cai, and Y. He, "Latency optimization for resource allocation in mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. 17, no. 8, pp. 5506–5519, 2018.

[6] Y. Mao, J. Zhang, S. H. Song, and K. B. Letaief, "Stochastic joint radio and computational resource management for multi-user mobile-edge computing systems," *IEEE Transactions on Wireless Communications*, vol. 16, no. 9, pp. 5994–6009, Sep. 2017.

[7] A. Bousia, A. Antonopoulos, L. Alonso, and C. Verikoukis, ""green" distance-aware base station sleeping algorithm in lte-advanced," in *2012 IEEE International Conference on Communications*, June 2012, pp. 1347–1351.

[8] H. Kim and K. G. Shin, "Efficient discovery of spectrum opportunities with mac-layer sensing in cognitive radio networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 533–545, May 2008.

[9] W. Zhang and Y. Wen, "Energy-efficient task execution for application as a general topology in mobile cloud computing," *IEEE Transactions on Cloud Computing*, vol. 6, no. 03, pp. 708–719, jul 2018.

[10] C. Yang, L. Huang, B. Leng, H. Xu, and X. Wang, "Replica placement in content delivery networks with stochastic demands and m/m/1 servers," in *2014 IEEE 33rd International Performance Computing and Communications Conference*, Dec 2014, pp. 1–8.