Dynamic Programming Based Cooperative Mobility Management in Ultra Dense Networks

Ziyue Zhang[1,2], Jie Gong[3], and Xiang Chen[1,2](✉)

[1] School of Electronics and Information Technology, Sun Yat-sen University, Guangzhou 510006, China
chenxiang@mail.sysu.edu.cn

[2] Key Lab of EDA, Research Institute of Tsinghua University in Shenzhen (RITS), Shenzhen 518075, China

[3] School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China

Abstract. In ultra dense networks (UDNs), base stations (BSs) with mobile edge computing (MEC) function can provide low latency and powerful computation to energy and computation constrained mobile users. Meanwhile, existing wireless access-oriented mobility management (MM) schemes are not suitable for high mobility scenarios in UDNs. In this paper, a novel dynamic programming based MM (DPMM) scheme is proposed to optimize delay performance considering both wireless transmission and task computation under an energy consumption constraint. Based on markov decision process (MDP) and dynamic programming (DP), DPMM utilizes statistic system information to get a stationary optimal policy and can work in an offline mode. Cooperative transmission is further considered to enhance uplink data transmission rate. Simulations show that the proposed DPMM scheme can achieve close-to-optimal delay performance while consume less energy. Moreover, the handover times are effectively reduced so that quality of service (QoS) is improved.

Keywords: Mobile edge computing · Mobility management · Cooperative transmission · Markov decision process · Dynamic programming

1 Introduction

With the increasing requirements of massive access and multimedia service in future radio access networks, ultra dense networks (UDNs) [5] and mobile edge

Supported in part by Science, Technology and Innovation Commission of Shenzhen Municipality (No. JCYJ20170816151823313), NSFC (No. 61771495), States Key Project of Research and Development Plan (No. 2017YFE0121300-6), Guangdong Provincial Special Fund For Modern Agriculture Industry Technology Innovation Teams (No. 2019KJ122), the Science and Technology Program of Guangzhou under Grant (No. 201707010166).

H. Gao et al. (Eds.): ChinaCom 2019, LNICST 313, pp. 252–263, 2020.
https://doi.org/10.1007/978-3-030-41117-6_21

computing (MEC) [6] are urgently required in 5G mobile communication systems. UDN aims at satisfying high data rate requirements to enhance the overall network capacity. In MEC, cloud computing and storage resources are placed at the network edge, so that mobile users experience lower latency, higher reliability and reduced computing power. However, due to the intensive deployment of base stations (BSs) in the UDN network and the high mobility of users, handover among BSs may frequently occur during data transmission or task computing. Frequent handover among BSs causes additional time delay and reduces the quality of service. Therefore, mobility management (MM) needs to be taken care of to guarantee low-latency and reliable service.

Mobile edge computing has drawn increasing attentions recently. The existing works mainly focus on task offloading policies and resource management schemes, i.e., how and when to offload a computation task from a mobile device to the edge or cloud systems. Ref. [8] proposed a partial computation offloading policy by comprehensively considering cloud computing and MEC in Internet of Things (IoTs), but the resources of MEC are not limited, which is unrealistic. Moreover, the user's mobility was not considered. Some other works consider service migration due to user's mobility, which is a key component of MM in MEC. With the consideration of the complex dynamics of UDN and the low mobility of users, a non-stochastic online-learning approach was proposed in [10] to reduce unnecessary handover. However, the high mobility of users makes the mobility management policy ineffective.

Due to the mobility of users and the change of wireless environment, the information obtained will be inaccurate, which brings challenges to wireless transmission and computation offloading. A Q-learning based MM was proposed in [13] to handle the information uncertainties, considering the wireless transmission and computation offloading. An energy efficient mobility scheme was designed in [15] to improve computation delay performance while satisfying communication energy constraint. Ref. [15] proposed a method to solve the optimization task using Lyapunov optimization and Multi-armed Bandits. Different from [15], a user-oriented energy-aware MM policy was proposed in [11] considering handover cost and the BSs' random ON/OFF. However, all these works are online-learning approachs which do not take advantage of user mobility statistics. In addition, inter-BS cooperation is not considered.

There are a few works considering cooperation to improve communication and computation performance. A joint computation and communication cooperation approach was proposed in [1] by using the helper user as a relay. In practice, using another BS as a relay would cause more access problems. Ref. [4] proposed a scalable BS switching strategy, which applied cooperative communications and power control, to extend the network coverage to the service areas of the switched-off BSs. However, the cooperative data transmission among BSs has not been considered in those works.

In this paper, an MM scheme considering both user's high mobility and BS cooperation is proposed. A high mobility user is guided to connect to delay-optimal BSs and performs handover if necessary. By utilizing the statistic information of user mobility and edge servers' status, the MM problem is formulated as a Markov decision process (MDP) and solved by the dynamic programming (DP) algorithm. Simulations show that delay performance can be greatly enhanced by using cooperative data transmission. Moreover, the proposed algorithm can effectively reduce handover times compared with the benchmark algorithm to guarantee UE's quality of service (QoS).

2 System Model and Problem Formulation

In this section, we will introduce the system model. The user mobility model, task generation and computation are first introduced. Then, the cooperative data transmission, the time delay and energy consumption model, the handover cost are described in detail. Finally, the mobility management problem is formulated.

2.1 System Overview

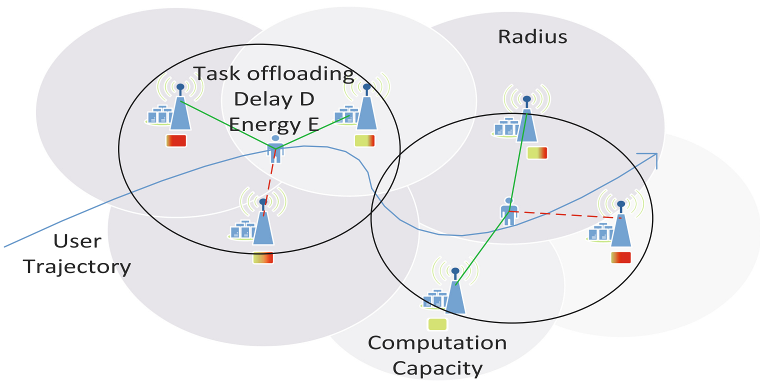


Fig. 1. Illustration of the considered mobility management in UDN with MEC-enabled BSs. A mobile UE with random walk trajectory offloads tasks to two of the candidate BSs via cooperative data transmission. The blue curve is the walk trajectory. The green lines mean connections between the UE and its serving BSs while the red lines mean possible connections between the UE and the candicate BSs. Moreover, the red/green colored bar below each BS refers to BS's computation status (red: heavy loaded, green: idle or lightly loaded). (Color figure online)

As shown in Fig. 1, the system, which is considered as a UDN environment, is running a set of MEC-enabled BSs which denoted as $\mathcal{S}_N = \{1, 2, \ldots, N\}$. A high mobility user equipment (UE) moves randomly in the network, which is modeled as a two-dimensional Brown motion, generating totally M tasks to be offloaded to the BSs for computing. Let $L_m \in \mathcal{S}_L$ denoting the UE's location where task m is generated. Due to the UDN environment, there are always several candidate service BSs $\mathcal{S}_A(L_m) \subseteq \mathcal{S}_N$, where $\mathcal{S}_A(L_m)$ is the set of candicate BSs in location L_m.

2.2 Task Computation

The computation task m is characterized using a two-parameter model [7]: input data of size λ_m bits that needs to be offloaded and computation intensity μ indicating how many CPU cycles are required to compute one bit input data. Without loss of generality, we assume that tasks are all of equal size λ. The following analysis can be extended to the cases where the data sizes are not equal. Moreover, the CPU frequency distribution of BSs is uniform.

Each BS $n \in \mathcal{S}_N$ is equipped with an MEC server of CPU frequency $f \in \mathcal{S}_F$ and supports cooperative data reception with the other BS. A UE who needs task offloading can choose one BS or two BSs according to the candicate BSs and time delay. Once the task is offloaded to two BSs via cooperation, it can be computed jointly by the two BSs. The equivalent CPU frequency is modeled as the sum of the two serving BSs' CPU frequency for simplicity. If BS n_1 and BS n_2 are selected to task computation, the computation delay is

$$d^c_{m,n_1,n_2} = \frac{\mu\lambda}{f_{m,n_1} + f_{m,n_2}}, \tag{1}$$

where f_{m,n_1} and f_{m,n_2} are the CPU frequencies that BS n_1 and BS n_2 can provide to task m respectively. If the UE selects one BS to compute tasks, the computation delay is

$$d^c_{m,n_1} = \frac{\mu\lambda}{f_{m,n_1}}, \tag{2}$$

where f_{m,n_1} is the CPU frequency of BS n_1.

2.3 Cooperative Data Transmission

Maximal ratio combining (MRC) is one of diversity merging technologies [14]. Compared with selection combining (SC) and equal gain combining (EGC), MRC can get the best performance by getting a higher signal noise ratio (SNR). According to Ref. [12], the SNR can be calculated as follows:

$$\mathrm{SNR}_{m,n_1,n_2} = \frac{P_t(H_{m,n_1} + H_{m,n_2})}{\sigma^2}, \tag{3}$$

where H_{m,n_1} and H_{m,n_2} represent the channel power gain at task m between the UE and the serving BSs n_1 and n_2 respectively. σ^2 denotes the noise power and P_t denotes the transmitted power.

Compared to the non-cooperative transmission model's SNR:

$$\mathrm{SNR}_{m,n_1} = \frac{P_t H_{m,n_1}}{\sigma^2}. \tag{4}$$

The transmission diversity of MRC can greatly improve data transmission rate by increasing SNR.

2.4 Time Delay and Energy Consumption

The task is offloaded to the serving BSs through the wireless uplink channel. Assume that the SNR of uplink channel is constant during task transmission. The uplink transmission rate is represented as follows:

$$r_{m,n_1,n_2} = W\log_2\left(1+\mathrm{SNR}_{m,n_1,n_2}\right), \tag{5}$$

where SNR_{m,n_1,n_2} is the equivalent SNR using MRC while connecting to BS n_1 and BS n_2. W is the channel bandwidth. The transmission delay sending the task data of size λ to BS n_1 and BS n_2 is

$$d^t_{m,n_1,n_2} = \frac{\lambda}{r_{m,n_1,n_2}}. \tag{6}$$

The energy consumption of uplink transmission is

$$e^t_{m,n_1,n_2} = \frac{P_t\lambda}{r_{m,n_1,n_2}}. \tag{7}$$

Meanwhile, cooperative data transmission also brings extra energy consumption

$$e^{cop}_{m,n_1,n_2} = \frac{P_{cop}\lambda}{r_{m,n_1,n_2}}, \tag{8}$$

where P_{cop} is the energy consumption per second.

Obviously, if the UE only connects to one BS, the uplink transmission rate between task m and serving BS n_1 is

$$r_{m,n_1} = W\log_2\left(1+\mathrm{SNR}_{m,n_1}\right). \tag{9}$$

The transmission delay sending the task data of size λ to BS n_1 is

$$d^t_{m,n_1} = \frac{\lambda}{r_{m,n_1}}. \tag{10}$$

And the energy consumption of uplink transmission is

$$e^t_{m,n_1} = \frac{P_t\lambda}{r_{m,n_1}}. \tag{11}$$

2.5 Handover Cost

For each task m, it must be computed in the same one BS or two BSs according to the communication conditions. Due to the high mobility of the UE, different tasks may be offloaded to different BSs for a lower time delay. When handover is executed, there is an additional delay cost.

Let c_m be one-time handover cost for task m and $a_m \in \mathcal{S}_A(L_m)$ be the set of the index of serving BSs for task m, the overall handover cost is

$$D_h = c_m \sum_{m=1}^{M-1} d^h_m = c_m \sum_{m=1}^{M-1} \mathbb{I}\left\{a_m \neq a_{m+1}\right\}, \tag{12}$$

where d^h_m is the handover delay of task m. $\mathbb{I}\{x\}$ is an indicator function with $\mathbb{I}\{x\}=1$ if x is true and $\mathbb{I}\{x\}=0$ otherwise.

2.6 Problem Formulation

In this paper, we consider the problem of minimizing the average delay under the constraint of average energy consumption, to determine which BSs serve the user and calculate the task. For task m, the overall delay is

$$D_{m,a_m} = d^c_{m,a_m} + d^t_{m,a_m} + d^h_{m,a_m}, \tag{13}$$

consisting of computing delay d^c_{m,a_m}, transmission delay d^t_{m,a_m} and handover delay d^h_{m,a_m}. As we focus on the user's energy, the overall energy consumption for task m is

$$E_{m,a_m} = e^t_{m,a_m} + e^{cop}_{m,a_m}. \tag{14}$$

We formulate the problem as an infinite horizon problem and an average cost problem. Therefore, the problem is formulated as follows:

$$\begin{aligned} \min_{a_1,a_2,\ldots} \quad & \lim_{M\to+\infty} \frac{1}{M}\sum_{m=1}^{M} D_{m,a_m} \\ s.t. \quad & \lim_{M\to+\infty} \tfrac{1}{M}\sum_{m=1}^{M} E_{m,a_m} \leq \alpha B \\ & a_m \in \mathcal{S}_A(L_m), \quad \forall m. \end{aligned} \tag{15}$$

where B is the battery capacity and $\alpha \in (0,1]$ indicates the desired energy consumption for all tasks.

3 Proposed Mobility Management Scheme

In this section, a dynamic programming algorithm is proposed to make decision based on statistic information. Handover and cooperative data transmission are considered. Moreover, a benchmark algorithm with full information is given for comparison.

3.1 Cooperative Mobility Management with DP

Dynamic programming is used to solve the MM problem. Firstly, the problem (15) can be formulated as an MDP consisting of state set $\mathcal{S}$, action set $\mathcal{A}$, state transition probability P and reward function R. The parameters of the DP-based cooperative mobility management (DPMM) scheme are defined as follows:

(1) **Agent:** The agent is a UE who decides to select BSs to ensure task computation with the shortest delay.
(2) **State:** The state is defined as $s = ((n_i, n_j), l, (f_i, f_j))$, where n_i and n_j represent the service BSs of the previous task. l represents the location of UE. The available computing power of service BSs is denoted by f_i and f_j respectively. If the UE only connects to a single BS, set n_2 and f_2 equal to 0. Then, the state space can be expressed as

$$\begin{aligned} \mathcal{S} = \{ & s = ((n_i, n_j), l, (f_i, f_j)) \mid n_i \in \mathcal{S}_N, n_j \in \mathcal{S}_N \cup \{0\}, \\ & l \in \mathcal{S}_L, f_i \in \mathcal{S}_F, f_j \in \mathcal{S}_F \cup \{0\}\}. \end{aligned} \tag{16}$$

(3) **Action:** The action is a decision to select one or two BSs as the service BSs for the current task. The set of action per state is defined as

$$\mathcal{A} = \{a = (n_i, n_j) \mid n_i \in \mathcal{S}_N, n_j \in \mathcal{S}_N \cup \{0\}\}. \tag{17}$$

By selecting a set that consists of two elements at each state, the UE can use cooperative data transmission method which is MRC as introduced in Sect. 2.3 to get the improvement of transmission rate. If only one BS is selected, we have $n_j = 0$.

(4) **Reward:** The reward for executing an action a at state s is the negative execution time of each task, which is defined as $R_{s,a} = -D_m$.

(5) **State Transition Probability:** The state transition probability $P \in \mathbb{R}^{s\times s\times a}$ is generated according to the statistic information of the random walk trajectory model and BSs' CPU frequency distribution.

To generate the state transition probability P, we need to know the state set, the random walk trajectory model and the BSs' CPU frequency distribution. As the two-dimensional Brown motion is continuous, the probability of UE's each location is zero. To directly calculate the state transition probability, network areas need to be discretized, resulting in discretization errors. Therefore, we propose a simulation based method to calculate P.

In this method, all states are initialized firstly and then the next state of each state is counted. Each experiment needs to be done until termination, or until it has been done a relatively large number of times. After these experiments, the state transition probability P can be concluded using following formula:

$$P_{ss'}^a = \frac{T_{ss'}^a}{T_s^a}, \tag{18}$$

where $P_{ss'}^a$ is the state transition probability of turning to state s' after the user takes action a at state s. $T_{ss'}^a$ is the number of times when the user takes action a at state s and then turns to state s'. T_s^a is the number of times that the user takes action a at state s.

By using DP algorithm, the optimal value function is calculated by value iteration algorithm [2]. Each state-action pair (s, a) corresponds to an action-value function, the action-value function can be defined as follows:

$$q_\pi(s,a) = R_s^a + \sum_{s'\in\mathcal{S}} P_{ss'}^a \nu_\pi(s'), \tag{19}$$

where π is the action selection policy. $\nu(s)$ is the value function of state s defined as follows:

$$\nu_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) q_\pi(s,a). \tag{20}$$

The i-stage value function is defined as

$$\nu_{i+1}(s) = \max_a R_s^a + \sum_{s'\in\mathcal{S}} P_{ss'}^a \nu_i(s'), \tag{21}$$

and the differential utility is defined as $v_i(s) = \nu_i(s) - \nu_i(s_0)$, where s_0 is some fixed state. Then Bellman equation [2] can be given as follows:

$$\gamma^* + v^*(s) = \max_a \left[R_s^a + \sum_{s' \in \mathcal{S}} P_{ss'}^a v^*(s') \right], \tag{22}$$

where γ^* is the optimal average utility. The differential utility $v_i(s)$ represents the maximum difference between the expected utility from state s to a given state s_0 and the utility if the utility of each stage is γ^* [3]. In value iteration algorithm, we first calculate

$$\gamma_{i+1}(s_0) = \max_a \left[R_{s_0}^a + \tau \sum_{s' \in \mathcal{S}} P_{s_0 s'}^a v_i(s') \right]. \tag{23}$$

Then we calculate the differential utility as

$$v_{i+1}(s) = (1-\tau) v_i(s) + \max_a \left[R_s^a + \tau \sum_{s' \in \mathcal{S}} P_{ss'}^a v_i(s') \right] - \gamma_{i+1}(s_0). \tag{24}$$

Then the optimal policy $\pi^*(s)$ can be found by calculating the optimal $v^*(s)$.

Algorithm 1. DP-based Mobility Management

1: Generate the set of state $\mathcal{S}$, the set of action $\mathcal{A}$, the reward of state-action pair $R(s, a)$, and state transition probability P accord ing to the known models.
2: Initialize the differential utility $v_0(s) = 0$, the initial policy π_0 and parameter τ.
3: Choose a fixed state s_0.
4: **repeat**
5:    **for** every s **do**
6:       $v_{i+1}(s) = (1-\tau) v_i(s) + \max_a \left[R_s^a + \tau \sum_{s' \in \mathcal{S}} P_{ss'}^a v_i(s') \right] - \gamma_{i+1}(s_0).$
7:    **end for**
8: **until** $v_{i+1} = v_i$
9: **output** $\pi^*(s)$

Note that the parameter $\tau \in (0, 1)$ is used to guarantee the convergence of relative value iteration and does not change the optimal value [2]. Since the optimal average utility is irrelative with the initial state, $\gamma_{i+1}(s_0)$ converges to γ^*.

The proposed DPMM scheme with statistic information is summarized in **Algorithm** 1. The MDP parameters are generated at first. Then the differential utility, the policy and a fixed state are initialized. Line 4 to line 9 is the value iteration algorithm to get the optimal policy according to the known models. The iteration will stop when the maximal differential utility does not increase.

3.2 Mobility Management with Greedy Strategy

For comparison, a delay optimal greedy strategy (DOGS) is proposed as a baseline. In this strategy, the UE simply selects its serving BSs with minimum delay, without considering energy consumption and handover. The algorithm is summarized in **Algorithm** 2.

Algorithm 2. Delay Optimal Greedy Strategy (DOGS)

1: **if** $\exists$ task m **then**
2: **input** L_m, $\mathcal{S}_A(L_m)$, λ, μ and $\forall n \in \mathcal{S}_A(L_m)$, $f_{m,n}$, $H_{m,n}$ at the beginning of task m.
3: Calculate the expected delay for each BS or each two BSs as in (13);
4: Select one BS or two BSs with the shortest expected delay.
5: **end if**

4 Simulations

In this section, we evaluate the average delay performance and the average energy consumption of the proposed DPMM scheme and the DOGS scheme. Moreover, the times of handover is an important indicator to ensure the quality of service. Simulations are run to show the performance of the proposed scheme.

As shown in Fig. 2, a 300 m $\times$ 300 m square area with four densely deployed BSs is simulated. The UE can associate with BSs within a distance of 100 m. The trajectory of UE is generated by the classic random walk model with speed $v \in [5, 10]$ m/s. The wireless channel gain is modeled as $H_{m,n} = 127 + 30 \times \log_{10} d$ as suggested in [9]. The cooperative power $P_{cop} = 0.2W$. Other simulation parameters are based on [11,15], including channel bandwidth $W = 20$ Mhz, noise power $\sigma^2 = 2 \times 10^{-13}W$, transmit power $P_t = 0.5W$. We consider a video stream analysis with some tasks generated during the whole moving process. The size of every task is $\lambda = 100$ Mbits and the computation intensity is $\mu = 20$ cycles/bit referring to [11]. The available CPU computation frequency is $f_{m,n} \in [\frac{1}{2}F, F]$, where $F = 25$ GHz.

Figure 3(a) and (b) compare the average delay performance and the average energy consumption of DPMM and DOGS over M tasks, respectively. It can be seen that although the DPMM scheme sacrifices about 10% of the delay performance, the energy consumption can be reduced. More importantly, the DPMM scheme can greatly reduce handover times compared with DOGS as shown in Fig. 4, which significantly reduces signalling cost for handover. This is because in order to effectively control the handover times, the DPMM algorithm will select the service BSs according to the average reward of the long-term state. As a result, the single selection of BSs in a certain state is not optimal.

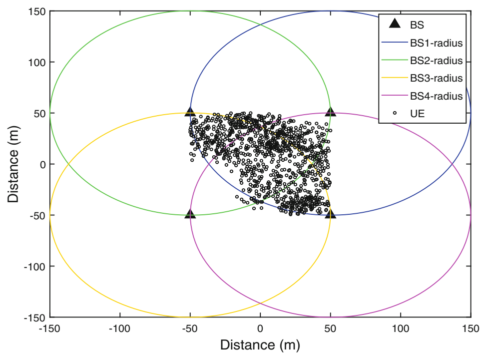
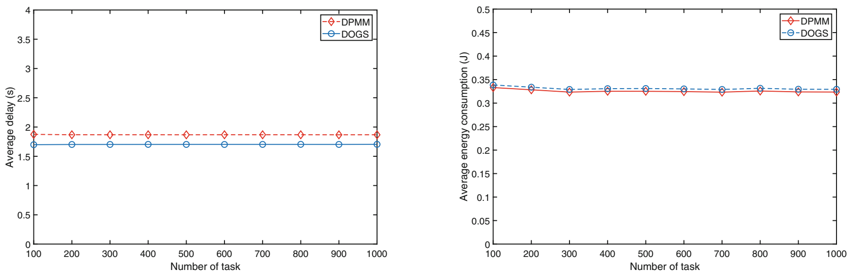


Fig. 2. Network topology



(a) Average delay (b) Average energy consumption

Fig. 3. Performance of DPMM compared with DOGS.

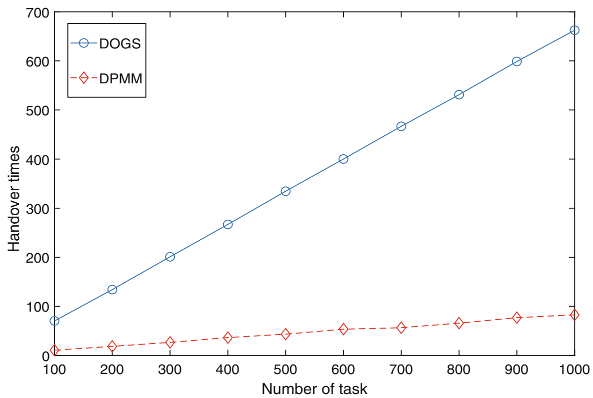
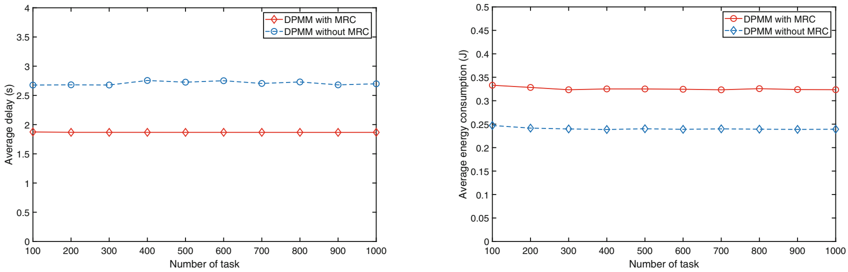


Fig. 4. Handover times

To show the benefits of cooperation, we also compare the DP algorithms with and without MRC. The result is shown in Fig. 5(a). It can be seen that by using MRC, the average delay is reduced by 50% due to the increased data transmission



(a) Average delay (b) Average energy consumption

Fig. 5. DPMM with MRC and without MRC

rate. Figure 5(b) shows that the energy consumption of DPMM with MRC is 20% higher than the energy consumption of DPMM without MRC. This is because the transmission diversity of MRC can greatly improve data transmission rate by increasing SNR. However, it will consume extra energy. Therefore, by sacrificing a certain amount of energy, the delay performance can be greatly improved.

Finally, DPMM algorithm is an offline algorithm, which means that it only needs to use the current statistics information to calculate once. Then it can get the optimal mobility management strategy. Considering that the statistical information of the BSs side and the user side will not change frequently, the computational complexity of the algorithm will not affect the user's long-term experience.

5 Conclusions

In this paper, we investigated the problem of MM based on BS cooperation. The DPMM scheme was proposed to make MM decision based on statistic information, taking UDN environment and high mobility of user into consideration. Simulations show that the proposed algorithm has greatly reduced handover times. At the same time, compared with greedy algorithm, it has similar delay performance. Furthermore, a cooperative data transmission scheme was proposed, which used the cooperation of BSs to improve data transmission rate. Through cooperative data transmission, the delay performance can be significantly improved. Future researches will include designing MM schemes for multiple users scenarios where the users may compete for transmission and computing resources.

References

1. Cao, X., Wang, F., Xu, J., Zhang, R., Cui, S.: Joint computation and communication cooperation for energy-efficient mobile edge computing. IEEE Internet Things J. **6**(3), 4188–4200 (2019)

2. Dimitri, P.B.: Dynamic Programming and Optimal Control, vol. 2, 3rd edn. Athena Scientific, Belmont (2005)
3. Gong, J., Zhou, S., Zhou, Z.: Networked MIMO with fractional joint transmission in energy harvesting systems. IEEE Trans. Commun. **64**(8), 3323–3336 (2016)
4. Han, F., Safar, Z., Lin, W.S., Chen, Y., Liu, K.J.R.: Energy-efficient cellular network operation via base station cooperation. In: 2012 IEEE International Conference on Communications (ICC), pp. 4374–4378 (2012)
5. Kamel, M., Hamouda, W., Youssef, A.: Ultra-dense networks: a survey. IEEE Commun. Surv. Tutor. **18**(4), 2522–2545 (2016)
6. Mach, P., Becvar, Z.: Mobile edge computing: a survey on architecture and computation offloading. IEEE Commun. Surv. Tutor. **19**(3), 1628–1656 (2017)
7. Mao, Y., You, C., Zhang, J., Huang, K., Letaief, K.B.: A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutor. **19**(4), 2322–2358 (2017)
8. Ning, Z., Dong, P., Kong, X., Xia, F.: A cooperative partial computation offloading scheme for mobile edge computing enabled internet of things. IEEE Internet Things J. **6**(3), 4804–4814 (2019)
9. Niu, C., Li, Y., Hu, R.Q., Ye, F.: Fast and efficient radio resource allocation in dynamic ultra-dense heterogeneous networks. IEEE Access **5**, 1911–1924 (2017)
10. Shen, C., Tekin, C., van der Schaar, M.: A non-stochastic learning approach to energy efficient mobility management. IEEE J. Sel. Areas Commun. **34**(12), 3854–3868 (2016)
11. Sun, Y., Zhou, S., Xu, J.: EMM: energy-aware mobility management for mobile edge computing in ultra dense networks. IEEE J. Select. Areas Commun. **35**(11), 2637–2646 (2017)
12. Tse, D., Viswanath, P.: Fundamentals of Wireless Communication. Cambridge University Press, Cambridge (2005)
13. Wang, J., Liu, K., Ni, M., Pan, J.: Learning based mobility management under uncertainties for mobile edge computing. In: 2018 IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2018)
14. Zhang, X., Beaulieu, N.C.: SER of threshold-based hybrid selection/maximal-ratio combining in correlated nakagami fading. IEEE Trans. Commun. **53**(9), 1423–1426 (2005)
15. Xu, J., Sun, Y., Chen, L., Zhou, S.: E2m2: energy efficient mobility management in dense small cells with mobile edge computing. In: 2017 IEEE International Conference on Communications (ICC), pp. 1–6 (2017)