Computer Communications 118 (2018) 140–147

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A Continuous-Time Markov decision process-based resource allocation scheme in vehicular cloud for mobile video services

Lu Hou[a], Kan Zheng[*,a], Periklis Chatzimisios[b], Yi Feng[c]

[a] Intelligent Computing and Communication (IC^2) Lab, Wireless Signal Processing and Networks Lab (WSPN), Key Lab of Universal Wireless Communications, Ministry of Education, Beijing University of Posts and Telecommunications, Beijing, 100088, China
[b] CSSN Research Lab, Alexander Technological Educational Institute of Thessaloniki (ATEITHE), Sindos, Thessaloniki, 57400, Greece
[c] Wireless Technology Research Department, China Unicom Network Technology Research Institute, China

ARTICLE INFO

Keywords:
Mobile video services
Vehicular cloud
Social graphs
Continuous-time Markov decision process

ABSTRACT

The development of vehicular network technologies boosts the wide deployment of mobile video applications with high requirements of Quality of Experience (QoE) in the Fifth-Generation (5G) era. However, the limitation of computing capabilities of intelligent vehicles makes it difficult to meet the QoE demands. The offloading technique that is put forward in vehicular cloud can extend such limitations largely by offloading video processing tasks to cloud or other vehicles. On the other hand, the emerging mobile social networks create new patterns for mobile applications to serve people on the basis of social relations. The mobile video offloading services can also be influenced by social relations of users inside a cloudlet. Therefore, in this paper we study the impact of social graphs on mobile video offloading services and propose a Continuous-Time Markov Decision Process (CTMDP) based resource allocation scheme considering social graphs as constraints. By using relative value iteration algorithm, an optimal policy can be obtained, which aims at maximizing the average system rewards. Simulation results show that our CTMDP based scheme achieves an enhanced performance against Greedy benchmark under different metrics.

1. Introduction

Vehicular netowrks has become a significant part of Fifth-Generation (5G) era. With advances in these vehicles and 5G wireless communications, the demand to process real-time mobile videos is increasing, and the requirements of Quality of Experience (QoE) of users are also rising [1]. For example, city surveillance services require large capability of computing to process TeraBit video data [2]. Although utilizing both Central Processing Units (CPUs) and Graphics Processing Units (GPUs), vehicles still have difficulty handling the video processing tasks regarding delay and energy consumption [3,4]. Therefore, it is still challenging for intelligent vehicles to run mobile video applications due to the limitation of energy and computing capability.

The development of mobile cloud computing and wireless communication technologies enable intelligent vehicles to offload video processing tasks to remote cloud servers [5,6], mobile edge servers [7] or cloudlets [8,9]. By utilizing video offloading services, vehicles can obtain unlimited resources which help overcome the constraints on themselves [10]. Therefore, the compute-intensive video applications can be executed in shorter delay and energy consumption. However, for some real-time mobile video applications, offloading to remote servers is not proper since it takes a lot of time for data delivery. The upcoming Vehicle-to-Vehicle (V2V) communications, such as IEEE 1609/802.11p or 5G communications, enable vehicles to deliver data to nearby vehicles without employing any switches or base stations. As a result, large time can be saved [11]. A vehicular cloud can be established by cooperative communications of intelligent vehicles. In this cloud, each vehicle serves as either an offloading service provider or service consumer [12].

Mobile social networking and social cloud computing are expanding with unprecedented rates [13]. Social clouds create opportunities for new application scenarios in vehicular networks. On the other hand, social graphs which are formed by vehicle owners have impact on the resources management of offloading process [14]. Generally, people are more willing to share their communication and computing resources to families and friends [15]. For video offloading applications, the number of computing resources a provider can share is dependent on the social relation between provider and consumer. Similarly, the vehicles in social graph can also choose to relay or not to relay video processing tasks for others on the basis of social graphs [16,17]. Therefore, social relations act as key constraints in the video offloading services.

In this paper, we propose an offloading scheme for mobile video

* Corresponding author.
E-mail address: zkan@bupt.edu.cn (K. Zheng).

http://dx.doi.org/10.1016/j.comcom.2017.10.011
Received 30 April 2017; Received in revised form 14 September 2017; Accepted 11 October 2017
Available online 13 October 2017

applications on the basis of social graphs. Our scheme is applied in a vehicular cloud where vehicles can communicate with each other using V2V communications. In such scenario, video processing tasks can be offloaded to other vehicles directly or with the help of intermediate vehicles. Our main contributions include, 1) studying the impacts of social relations on the video application offloading services; 2) finding the optimal strategy for computational offloading under the constraints of social graphs. In this paper, we first study the influence of social graphs on the offloading process for a set of intelligent vehicles. We then build a system model where mobile offloading services are deployed and vehicles are constrained by social relations. Based on system model, a Continuous-Time Markov Decision Process (CTMDP) problem is formulated. The CTMDP problem is then solved by using Bellman equation and relative value iteration. Finally, numerical results are presented to demonstrate that our offloading policy is able to enhance the overall resources utilization resulting in improved performance for the mobile video applications.

The rest of paper is organized as follows. Section 2 provides detailed related works. We present our system model in Section 3. The CTMDP model is formulated in Section 4. Section 5 describes the optimal solution to the CTMDP model. Performance evaluation is conducted in Section 6. Section 7 concludes the paper.

2. Related works

The emerging mobile video applications require more and more energy consumption and computing capabilities on vehicles. Some of the literatures focus on saving the energy consumption of vehicles with some power-efficient algorithms [18]. Argyriou et al. [19] provided detailed descriptions over video delivery on both challenges and solutions in 5G. In [4], Xu et al. provide a slot allocation policy named “Delay-Aware Resource Allocation (DARA)” to ensure quality, while Zhang et al. [3] describe a cloud framework that can switch CPUs and GPUs during runtime. However, the resources are often limited in vehicles. Thus, computing offloading technology can remarkably mitigate the need of computing capabilities in individual vehicle to the cloud. It can be efficient for breaking the limitation, because vehicles are enabled to use resources in cloud, edge, cloudlet or even other vehicles. Zhang et al. [12] studied offloading of real-time video applications in mobile cloud. They proposed an energy-efficient solutions and gave detailed analysis.

Markov Decision Process (MDP) can be used to model the offloading problems. Zheng et al. have used MDP to solve their models, especially those offloading models [20]. Liang et al. [21] proposed an Semi-MDP (SMDP) model for inter-domain resource allocation in cloud networks. The computing request can be offloaded to a cloud domain or its neighbor domains in this paper. Another study in [9] uses SMDP to solve resource allocation problems in vehicular cloud, considering the mobility of vehicles. Besides, CTMDP model is used for totally different scenarios in [22]. Jia et al. provide detailed analysis for CTMDP model and evaluate the performance and complexity of it compared with normal MDP model. Therefore, in this paper, we use CTMDP to formulate our mobile video offloading problem.

On the other hand, the social networks starts to change the way people communicate. Social graphs can reflect trust between users and privacy of users. Therefore, Chen et al. [23] studied the trust-based service management in Internet of things system. Many other researches have focused on the routing problems with social graphs. For example, Gupta et al. [24] proposed a routing algorithm in mobile ad-hoc networks and made comparison with some existing algorithms. Social graphs are commonly restrictions on path routing, as described in [17]. Therefore, the authors exploit a Lagrangian relaxation solution with Dijkstra algorithm to find optimal paths for D2D communications. Moreover, social graphs can also have impact on the resource allocation in cloud computing. Social cloud computing is fully surveyed in [15]. Caton and coworkers [13,25,26] have studied resources allocations with social graphs under different scenarios. They all aimed at optimizing utilization of resources with social restrictions. So far as we can investigate, the study of social graphs on both path routing and computing offloading is still lacked in the state of the art.

In all, none of the literatures have studied the dynamic offloading considering the impact of social relations which is of great importance for the 5G age. Therefore, different from previous works, we combine social graph with communication graph, and study the impact that social graph has brought. We propose a CTMDP-based model to solve the computing offloading problems in mobile video applications. The social graph imposes restrictions mainly on two aspects, i.e., 1) the maximum shareable resources between vehicles and, 2) the routing path for packet delivery. Generally speaking, strong social relations lead to large number of shareable resources and low-cost communication paths. Our model aims at improving the QoS by reducing the block probability of offloading requests. A relative value iteration algorithm is introduced to solve the problem. An optimal policy can be obtained after several times of iterations.

3. System model

In this Section, the system model is described in details. Intelligent vehicles that are deployed V2V transmission modules form up a multi-hop vehicular cloud where each vehicle can request or allocate a number of resources for others. All the path routings and resource allocations are restricted by social graphs, as described in Fig. 1.

3.1. Communication and social graph

We consider a mobile vehicular cloud which consists of N vehicles and a main operator(can be acted typically by base station). All the vehicles are distributed uniformly inside the coverage of the operator. They can be linked physically with each other if they are in the range of V2V communications. We use $G_c = (\mathcal{N}, E_c)$ to represent the undirected communication graph where $\mathcal{N}$ is the set of all the vehicle nodes and E_c is formed up by all the available communication links between them. The adjacency matrix of G_c with N vertexes can be described by an $N \times N$ matrix, i.e.,

$$C_{N\times N} = \begin{array}{c c} & \begin{array}{cccccc} V_1 & V_2 & V_3 & V_4 & \cdots & V_N \end{array} \\ \begin{array}{c} V_1 \\ V_2 \\ V_3 \\ V_4 \\ \vdots \\ V_N \end{array} & \begin{pmatrix} - & c_{12} & c_{13} & c_{14} & \dots & c_{1N} \\ c_{21} & - & c_{23} & c_{24} & \dots & c_{2N} \\ c_{31} & c_{23} & - & c_{34} & \dots & c_{3N} \\ c_{41} & c_{24} & c_{43} & - & \dots & c_{4N} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{N1} & c_{N2} & c_{N3} & c_{N4} & \dots & - \end{pmatrix} \end{array}$$

where c_{ij} represents the transmission cost of link $(i, j) \in E_c$. For simplicity, we use Shannon's equation to calculate c_{ij}, i.e.,

$$c_{ij} = \frac{1}{r_{ij}} = \frac{1}{B\log_2(1 + f_{SNR}(d_{ij}))} \tag{1}$$

where r_{ij} is the transmission rate of the communication link between vehicle i and j. B is the bandwidth of wireless channel and f_{SNR} is a function that calculate signal to noise ratio (SNR) based on the distance d_{ij} between vehicle i and j.

It can be seen that under such assumption, we have $C_{N\times N} = C^T_{N\times N}$.

Similarly, all vehicle owners in this cloud form a social graph $G_s = (\mathcal{N}, E_s)$, where $\mathcal{N}$ is the same set as it in G_c and E_s represents the social relation in G_s. The adjacency matrix of G_s is expressed as

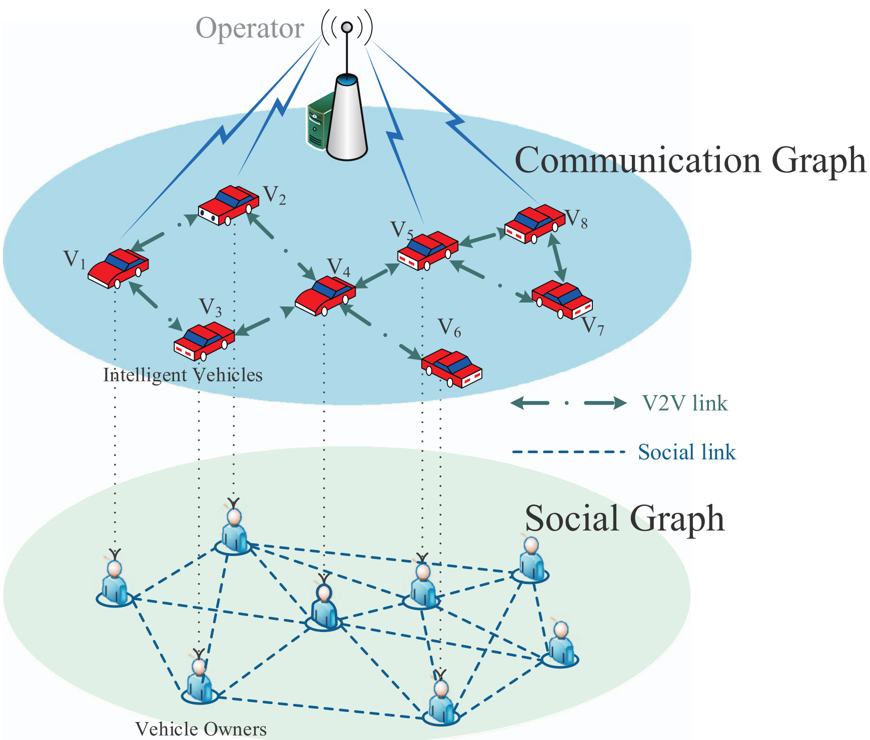

Fig. 1. Communication and Social Graphs.

$$S_{N\times N} = \begin{array}{c|c} & V_1 \quad V_2 \quad V_3 \quad V_4 \quad \dots \quad V_N \\ \begin{matrix} V_1 \\ V_2 \\ V_3 \\ V_4 \\ \vdots \\ V_N \end{matrix} & \begin{pmatrix} - & g_{12} & g_{13} & g_{14} & \dots & g_{1N} \\ g_{21} & - & g_{23} & g_{24} & \dots & g_{2N} \\ g_{31} & g_{23} & - & g_{34} & \dots & g_{3N} \\ g_{41} & g_{24} & g_{43} & - & \dots & g_{4N} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ g_{N1} & g_{N2} & g_{N3} & g_{N4} & \dots & - \end{pmatrix} \end{array}$$

where g_{ij} stands for the social relation in number between vehicle i and j. We use number between 0 and 1 ($0 < g_{ij} < 1$) to weigh the relation. We define that large number means strong relation. Social relations have two aspects of impacts in the system, i.e.,

1. **Data delivery path:** Vehicles are required to act as relays, thus they form up a multi-hop V2V communication network. However, this task raises concerns for vehicles as it induces cost due to energy and other resource consumption. The willingness for a vehicle to relay for others relies on both social graphs and resource consumptions. Inspired by Koutsopoulos et al. [17], we use a probability function $p_j = f(g_{sj}, g_{ij}, g_{jd})$ to quantify the possibility that vehicle j is willing to receive packets from vehicle i and deliver them further. These packets are originally generated by node s to node d.

$$f(g_{sj}, g_{ij}, g_{jd}) = \begin{cases} \dfrac{g_{sj} + g_{ij} + g_{jd}}{3} & s \neq i, j \neq d \\ \dfrac{g_{sj} + g_{jd}}{2} & s = i, j \neq d \\ 1 & j = d \end{cases} \tag{2}$$

An available path p_{sd} between node s and d is defined by the combination of edges and vertices that make s and d connectable. Note that "available" means this path satisfies the requirement of threshold of social constraint ξ, i.e.,

$$\sum_{(i,j)\in p_{sd}} f(g_{sj}, g_{ij}, g_{jd}) > \xi \tag{3}$$

Furthermore, we use $\mathbb{P}_{sd}$ to denote the set that includes all the available paths between two vehicles s and d. Evidently though, $\mathbb{P}_{sd} = \varnothing$ means it is socially unreachable between vehicle s and d.

2. **Maximum allocatable resources:** Each vehicle has $K_i (i \in \mathcal{N})$ units of resources in total. Note that the idle state of vehicle server systems also need to utilize some of the CPU resources, thus we use θ_i to denote the CPU utilization proportion of the idle state in vehicle i. Mobile video offloading services allows vehicles to share resources with others. We use o_{ij} to denote the number of resources that are allocated from vehicle i to vehicle j. Based on social graph, we can get an upper bound of o_{ij}, i.e.,

$$\mathcal{O}_i = \max_{\forall j\in\mathcal{N}} (o_{ij}) = \lfloor g_{ij}(1-\theta_i)K_i \rfloor \tag{4}$$

Similar with $C_{N\times N}$, $S_{N\times N} = S^T_{N\times N}$.

3.2. Mobile video offloading services

Generally, part of the mobile applications can be offloaded, such as CPU-involved computing, while other part of the mobile applications cannot be offloaded, such as Input/Output (I/O) processing. In this paper, $\eta (0 < \eta < 1)$ represents the proportion of components that are about to be offloaded in the whole application.

When a vehicle send an offloading request, the base station decides which vehicle should accept this request and how many computing resources can be allocated to it. Note that the number of resources is dependent on the social graph. Strong relation makes receiver willing to share more resources to the requester. On the other hand, offloading request cannot always be successful due to the limit of resources in other vehicles. Under such circumstances, the requester has to finish all the computation tasks locally. This causes a lot of energy loss, and the time for computing can be increased highly.

We define that the offloading request arrives as Poisson process with mean rate λ for each vehicle. Therefore, total arrival rate of requests is $N^*\lambda$. The service rate of each unit of resource is denoted as μ, and the service time follows an exponential distribution. Moreover, M_v denotes the average amount of computation for a single mobile video request.

4. CTMDP based problem formulation

In order to solve the problem mentioned in Section 3, we proposed a CTMDP-based formulation. A CTMDP formulation can be described by the five following basic parts, i.e.,

- Decision epochs denoted as T;
- System state space marked as $\mathscr{S}$;
- Available action sets for each state, denoted as $\mathscr{A}_s$;
- State and action dependent rewards, defined as r;
- State and action dependent state transition probability matrix, denoted as $\mathscr{P}$;

$$\mathscr{M}_{CTMDP} \triangleq \{T, \mathscr{S}, \mathscr{A}_s(s \in \mathscr{S}), r, \mathscr{P}\} \tag{5}$$

When an event is arriving, the system state transfer from one to another. In the meantime, an action is made whenever the system state is changed, and the time spent in a particular state $t \in T$ follows exponential distribution. State transitions have probabilities which rely on the arriving of next events. Different states and event arriving rates conducts to different possibilities for transition. Furthermore, the system receives a reward on the basis of current state and current action, which makes the whole process Markovian. With the system running, the actions form up an action chain which is regarded as a policy or a strategy, and can be denoted as π. We denote that the set of all policies as Π. The optimal goal is to find a policy $\pi^* \in \Pi$, which enables the system to harvest maximum rewards.

In this section, specific description for CTMDP formulation is given for every basic parts, and some important notations are listed in Table 1.

4.1. Event

An event is part of the system state and can trigger state transition. The event set can be described as

$$e \in \mathscr{E} = \{A_1, A_2, \ldots, A_N, D_{1,1}, D_{1,2}, \ldots, D_{1,C}, D_{2,1}, \ldots, D_{N,C}\} \tag{6}$$

A_i means represents that an offloading request is issued by vehicle i with the intensity of λ. Naturally, arrival process is Poisson. When an offloading task in vehicle i is completed and the occupied o units of resources are released, the event is denoted as $D_{i,\ o}$. All the available events forms up an event set which is denoted as $\mathscr{E}$.

4.2. Decision epoch

The decision epoch set T can be characterized on the basis of event. A decision is made whenever an event happens and the system stays for an action-dependent random period of time until the arrival of next event. Therefore, T can be described by probability distribution of the emergence of events. The arrival process follows a Poisson distribution and task processing time follows an exponential distribution. Thus, T follows an exponential distribution, which gives a proof for that our system model is a CTMDP problem. Note that T is dependent on state and action, we denote the expectation of T as $\tau(s, a) = \frac{1}{\gamma(s,a)}$, which will be further discussed in the sequel.

Table 1
List of important notations.

Symbol	Description
N	Number of nodes
η	Proportion of component that can be offloaded
M_v	Average amount of computation
λ	Arrival rate of offloading request
μ	Average service rate
$\mathbb{P}_{sd}$	Set of available paths between node s and d
T	Set of decision epochs
$\mathscr{S}$	System state space
$\mathscr{A}_s$	Set of available actions on state s
r	Rewards
$\mathscr{P}$	Transition probability matrix
e	An event
$\mathscr{E}$	Event set

4.3. System state

A system state s can be described as

$$s = (s_1, s_2, \ldots, s_N, e) \tag{7}$$

where s_i represents the total number of resources that are occupied in node i, and $e \in \mathscr{E}$.

4.4. Action

When an event occurs, the base station has to make a decision. In summary, the available action set under certain system state can be written as,

$$a \in \mathscr{A}_s = \begin{cases} (V, o, \mathrm{p}_{iV}), & e \in \{A_1, \ldots, A_N\},\ V \in \mathscr{N},\ o \in (0, C] \\ (0) & e \in \{A_1, \ldots, A_N\} \\ 0, & e \in \{D_{1,1}, \ldots\} \end{cases} \tag{8}$$

When a request arrives from vehicle i, the base station can specify that it can be offloaded to vehicle V, and V can share o units of resources for the requester. Meanwhile, the transmission path p_{iV} is chosen for package delivery. On the other hand, the base station can reject the request due to the lack of resources, or the high cost of transmission. In this case, the system action is denoted as $a = (0)$. Another situation is that when system event is a departure, the base station needs to make no decision. Therefore, a is simply 0 when $e \in \{D_{1,1}, \ldots\}$.

4.5. Long-term system reward

For any state in this system, the base station chooses an action and the system gains a reward. By proper definition of system reward, the performance of policies can be evaluated. Firstly, the system instant income can be calculated as

$$w(s, a) = \begin{cases} E - \beta t_o - \zeta t_c, & e \in \{A_1, \ldots, A_N\},\ a = (V, o, \mathrm{p}) \\ -P, & e \in \{A_1, \ldots, A_N\},\ a = 0 \\ 0, & e \in \{D_{1,1}, \ldots\},\ a = 0 \end{cases} \tag{9}$$

When the operator receives an offloading request, it can accept or reject the request. If the request is accepted, the system can acquire an income E. However, the occupation of computing resources and communication resources require some cost of the system. Let t_c denotes the transmission time and t_o represents the service time. Also define β and ζ as the cost coefficients of time for t_c and t_o, thus, we have the instant income of state s with action a. The time spent for computing can be calculated as,

$$t_o = M_v/(\mu^* o)$$

and the time for delivering the task and results can be written as,

$$t_c = \rho^* \sum_{(i,j)\in \mathrm{p}} c_{ij}$$

If the base station reject an request, the request can never be finished, which causes bad QoS for vehicle users. Therefore, the system receives a penalty P for rejection. The last case is when a task is finished and the resources are released, the system needs to do nothing. As a result, no pay or gain is obtained during this process.

Apart from the instant income the system can gain, the occupation of resources in current state s needs some costs from system. We define such cost by the utilization of resources in s as $z(s, a)$ and because the state periods τ follows exponential distribution, $z(s, a)$ can be expressed as

$$\begin{aligned} z(s,a) &= o(s)\int_0^{\infty}(1-e^{-\gamma(s,a)t})dt \\ &= o(s)\tau(s,a) \end{aligned}$$

where

$$o(s)=\sum_{i\in\mathcal{N}} s_i$$

Since this system runs all the time, total rewards become infinite after a long time. The proposed reward model becomes insignificant because the rewards cannot be quantified. Therefore, a discount reward model is introduced according to [27]. It only considers the *discount expected* rewards as the criterion. In this way, the system reward can be numerically maximized. According to [27], we have the discount expected rewards as

$$r(s,a)=w(s,a)-\frac{o(s)}{\alpha+\gamma(s,a)} \tag{10}$$

5. Optimal solution to CTMDP formulation

Based on the CTMDP formulation, an optimal solution in the aim of maximizing the system rewards is introduced. The solution uses value function of state as the iteration object in the form of Bellman equation, which needs to be normalized in the first place. Besides, a relative value iteration algorithm is proposed to solve the Bellman equation.

5.1. Mean rate of events

When the operator takes an action, the system transits from one state to another. The next state is dependent with the coming event. Therefore, it is important to obtain the occurrence rate of every event. As assumed above, all the time intervals of events in this system follows exponential distribution, thus we use mean rate of events $\gamma(s, a)$ to represents the event rate at state s with action a. Evidently, the time interval between two continuous decision epochs $\tau(s,a)=\frac{1}{\gamma(s,a)}$.

$$\gamma(s,a)=\begin{cases}\gamma_0(s), & e\in\{D_{1,1},\ldots\} \\ & e\in\{A_1,\ldots\},\ a=0 \\ \gamma_0(s)+o\mu, & e\in\{A_1,\ldots\},\ a=(V,o,\text{p})\end{cases} \tag{11}$$

where

$$\gamma_0(s)=N^*\lambda+\mu\sum_{1\leqslant i\leqslant N}s_i$$

5.2. Probability of state transition

The probability of state transition depends on the rate of events. If the system transit from state s to state j with an action a, then the state transition probability can be calculated as

$$p(j|s,a)=\frac{rate(e_j)}{\gamma(s,a)}$$

where e_j is the event of j state, and $rate(e_j)$ is the occurrence rate for e_j. The $p(j|s, a)$ can be specified in three different cases, i.e.,

5.2.1. $e_j = A_i$

The system transits to a state where an offloading request arrives, thus,

$$p(j|s,a)=\frac{\lambda}{\gamma(s,a)} \tag{12}$$

5.2.2. $e_j = D_{m,n}$, $a = (k, o)$, $k \neq m$

In this case, vehicle m has received a request and allocated o units of resources. In the meantime, n units of resources in vehicle k are leaving at j state. If $k \neq m$, the transition probability is calculated as,

$$p(j|s,a)=\frac{n\mu}{\gamma(s,a)} \tag{13}$$

5.2.3. $e_j = D_{m,n}$, $a = (m, o)$

If $k = m$, we have

$$p(j|s,a)=\frac{(n+o)\mu}{\gamma(s,a)} \tag{14}$$

Denote impossible transition between state i and state j as $p(j|s, a) = 0$, a state transition probability matrix can be derived, which contains all the transition probabilities from every state to every other state. We denote this matrix as $\mathscr{P}$, which satisfies

$$\forall\ i\in[1,N],\ \sum_{1\le j\le N}\mathscr{P}_{ij}=1$$

and

$$\forall\ j\in[1,N],\ \sum_{1\le i\le N}\mathscr{P}_{ij}=1$$

5.3. Bellman equation and normalization

In order to solve the CTMDP formulation, a Bellman equation is involved in this solution. Let $\nu(s)$ represents a numeric value of a certain state s, and all the $\nu(s)$ can be written in the form of vector $\boldsymbol{\nu}$. According to discounted reward model, we derive that,

$$\nu(s)=\max_{a\in\mathscr{A}_s}\left\{r(s,a)+\iota\sum_{j\in\mathscr{S}}p(j|s,a)\nu(j)\right\} \tag{15}$$

which can also be written in form of policy π, i.e.,

$$\boldsymbol{\nu}=\max_{\pi\in\Pi}\{\mathbf{r}_\pi+\iota\mathscr{P}_\pi\boldsymbol{\nu}\} \tag{16}$$

where $\iota=\gamma(s,a)/(\alpha+\gamma(s,a))$.

To solve the Bellman equation for CTMDP formulation, the Bellman Equation needs to be normalized at first in order to convert CTMDP to discrete time Markov decision process model [27]. Parameter κ is introduced and defined as

$$\kappa=N^*\lambda+N^*K^*\mu$$

Thus, the normalized reward function and the normalized transition probability can be expressed as

$$\tilde{r}(s,a)=r(s,a)\frac{\gamma(s,a)+\alpha}{\kappa+\alpha} \tag{17}$$

$$\tilde{q}(j|s,a)=\begin{cases}1-\frac{[1-q(s\mid s,a)]\gamma(s,a)}{\kappa}, & j=s\\ \frac{q(j\mid s,a)\gamma(s,a)}{\kappa}, & j\neq s\end{cases} \tag{18}$$

And the normalized Bellman equation can be written as

$$\tilde{\nu}(s)=\max_{a\in\mathscr{A}_s}\left\{\tilde{r}(s,a)+\tilde{\iota}\sum_{j\in\mathscr{S}}\tilde{q}(j|s,a)\tilde{\nu}(j)\right\} \tag{19}$$

where $\tilde{\iota}=\kappa/(\kappa+\alpha)$.

5.4. Relative value iteration algorithm

In order to solve the CTMDP model, a relative value iteration algorithm is applied as shown in Algorithm 1. Denote the span of vector $\boldsymbol{\nu}$ as

$$\Psi(\boldsymbol{\nu})=\max_{s\in\mathscr{S}}\nu(s)-\min_{s\in\mathscr{S}}\nu(s),$$

which is apparently a seminorm of vector $\boldsymbol{\nu}$ on $(\mathscr{V})$. Thus, the optimal

Step 1. Set $\boldsymbol{\phi}_0 = \tilde{\boldsymbol{v}}^0 - \tilde{v}^0(s')\boldsymbol{e}$, where $\tilde{\boldsymbol{v}}^0 \in \boldsymbol{N}$, s' is an arbitrary state and $\boldsymbol{e} = (1, 1, \ldots, 1)$. Also set $n = 0$ and $\varepsilon > 0$.

Step 2. Calculate

$$\tilde{\boldsymbol{v}}^{n+1} = \max_{\pi\in\Pi}\{\tilde{\boldsymbol{r}}_\pi + \tilde{\iota}\tilde{\mathcal{P}}_\pi\boldsymbol{\phi}^n\}$$

$$\boldsymbol{\phi}^{n+1} = \tilde{\boldsymbol{v}}^{n+1} - \tilde{v}^{n+1}(s')\boldsymbol{e}$$

Step 3. If

$$\Psi(\tilde{\boldsymbol{v}}^{n+1} - \tilde{\boldsymbol{v}}^n) < \varepsilon,$$

go to **Step 4**. Otherwise, let $n = n + 1$ and return to **Step 2**.

Step 4. Choose $\pi^* = \arg\max\{\tilde{\boldsymbol{r}} + \tilde{\iota}\tilde{\mathcal{P}}\tilde{\boldsymbol{v}}\}$

Algorithm 1. Relative value iteration algorithm.

Table 2
Simulation parameters.

Parameter	Value	Parameter	Value
N	4	ξ	0.2, 0.7
K_i	4	θ_i	0.5
P	100	η	0.5
α	0.9	β	1
ζ	1	E	100

policy π^* can be derived when Ψ converges after a few times of iterations.

6. Performance evaluation and analysis

To evaluate the performance of our CTMDP scheme, a MATLAB simulator is developed in this section. The simulation environment and parameter configurations are first described. We also proposed the Greedy scheme as the benchmark. The simulation results show that our CTMDP-based scheme transcends the benchmark under different metrics.

6.1. Simulation configuration

In this section, we first introduce the simulation environment and then analyze the numerical results. We consider totally $N = 4$ vehicles in our system, the parameters of which as well as some other key arguments are all listed in Table 2. We use the arriving rate of offloading requests λ and the processing rate of video tasks μ to study the performance of our CTMDP scheme. For comparison, we choose greedy policy as the benchmark. In greedy policy, the operator always chooses the action which makes system get highest instant reward, without considering the following states.

In order to evaluate the performance of different schemes, the steady state probabilities are required to be calculated by solving the equation

$$\begin{cases} \mathbf{x}\mathscr{P} = \mathbf{x} \\ \sum \mathbf{x} = 1 \end{cases} \tag{20}$$

where $\mathbf{x}$ represents the steady probability of each state. The evaluation relies on steady probability since it can reflect the long-term rewards and block probabilities.

6.2. Performance results and analysis

The proof of convergence, which is represented by Ψ, is shown in Fig. 2. It testifies the results is credible. As illustrated in Fig. 3, the system reward rises to the highest value then starts to decline with the increasing of arriving rate of offloading request. This is because when the offloading requests arrive at low rate, the resources are sufficient to process the tasks. Therefore, the more requests that come, the more reward system gains. However, when the arriving rate exceeds 0.41, the system goes beyond the maximum processing capacity and the reward starts to decrease because more and more requests are blocked. The same trend can be seen in Fig. 4, where the block probability of new requests under the increasing of arriving rate is illustrated. The fast growing number of new requests result in the rising of block rate, ulteriorly reducing the system reward.

As shown in Fig. 5, the system reward rises up and reduces with the increasing of average computational quantity μ. This is because when $\mu < 1.4$, the resources can handle the video tasks with little hindrance. Thus, the system can process more tasks in unit time. This makes the influence of the steady probability of state with new requests become larger than that produced by blocking. In the end, the system reward grows up. However, the system becomes unable to handle those

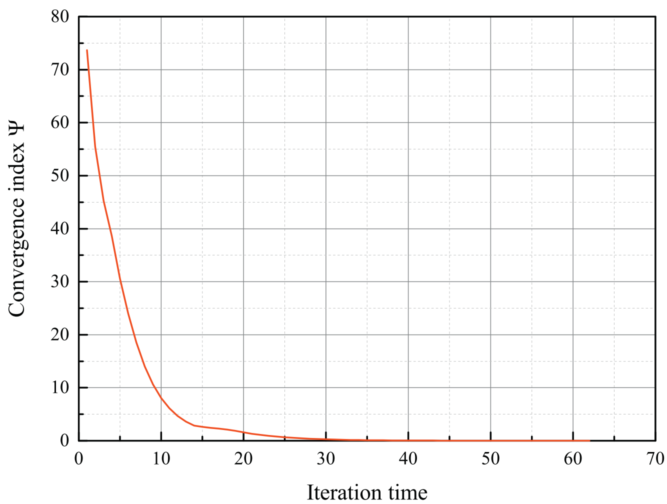

Fig. 2. Proof of convergence.

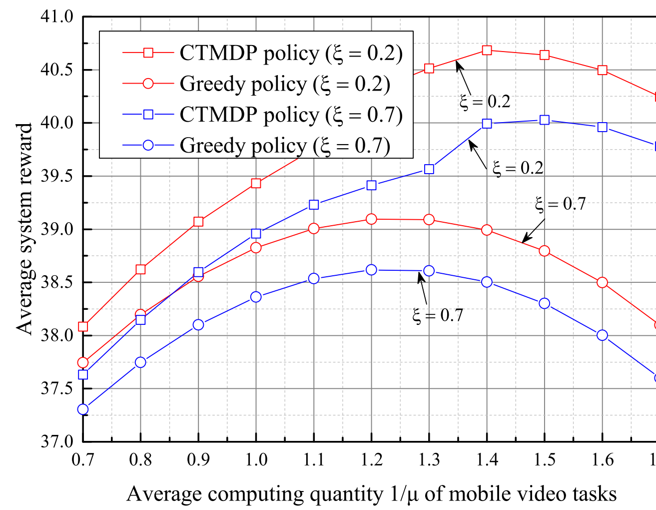

Fig. 5. Average system reward of average computing quantity.

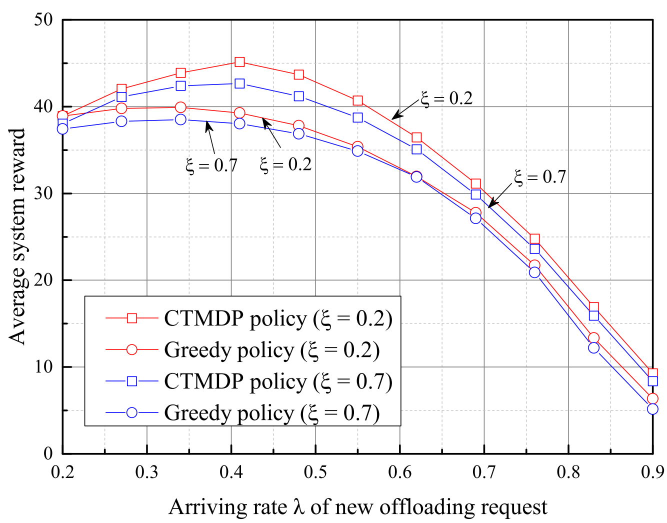

Fig. 3. Average system reward of different arriving rate.

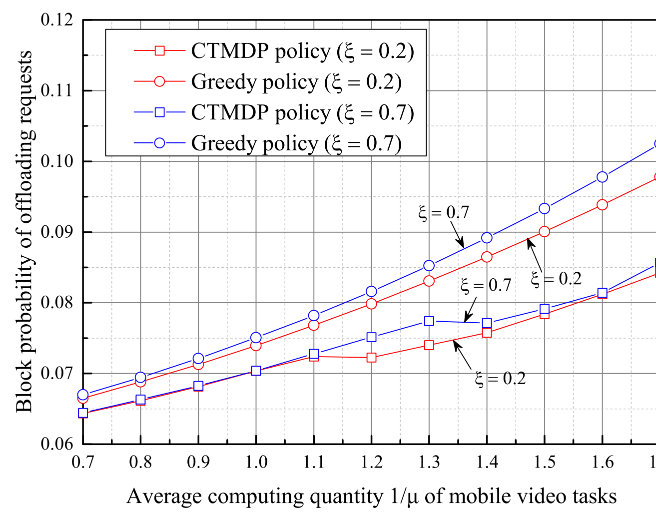

Fig. 6. Block probability of average computing quantity.

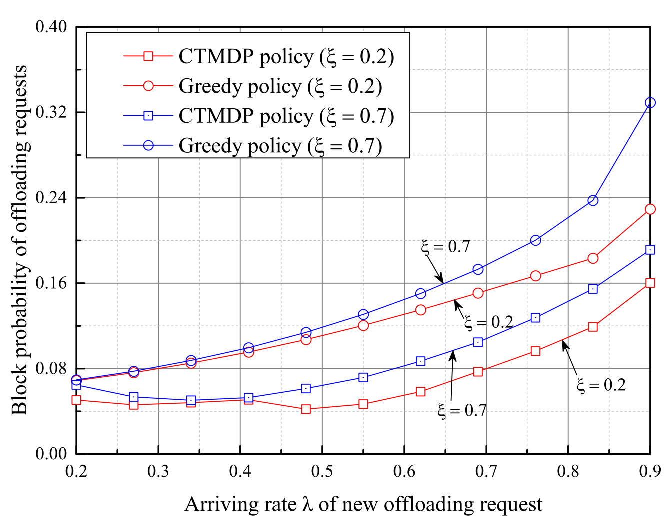

Fig. 4. Block probability of different arriving rate.

requests with larger average computing quantity than $\mu = 1.4$. In this situation, the block rate becomes high enough to reduce the system reward, and with the increasing of μ, the system reward become worse and worse. Fig. 6 shows that the more the computing quantity raises, the more block probability becomes, which has the same trends in Fig. 3.

Compared with greedy policy, our CTMDP policy performs better. The key to this result is that CTMDP policy considers the long-term rewards instead of the current ones. CTMDP policy tries not to allocate as many resources as possible, but leaves some resources for new requests in the future. In this way, the block rate is situated at the optimal position, while the long-term system reward keeps at optimal value. The greedy policy however, only focuses on maximizing the reward of current state. This kind of strategy gets maximum instant reward on the cost of high block probability, causing lower long-term system reward than CTMDP policy. Another interesting thing is that, a higher threshold ξ of social constraint conducts to a higher block probability and lower average system rewards, as shown in all the figures above. This is simply because ξ directly influences the transmission paths. More paths are to be unavailable with a large threshold. This clearly shows that the social graph has a definite effect on the whole system.

7. Conclusion

In this paper, we proposed a CTMDP-based model aiming at maximizing the system average rewards for mobile video offloading services under the constraints of social graphs. The problem was formulated as a CTMDP and solved by relative value iteration algorithm. The optimal policy indicates the resources allocation strategy on certain system state. The performance evaluation shows that our CTMDP model is superior to the reference benchmark.

Future work includes the further study of social graphs in the mobile cloud computing or mobile edge computing context. We decided to dig deeply for the potential value of social graphs.

Acknowledgment

This work is funded in part by Key Laboratory of Universal Wireless Communications (Beijing University of Posts and Telecommunications), Ministry of Education, P.R. China, No. KFKT-2016101, National Science Foundation of China (No. 61331009), Fundamental Research Funds for the Central Universities (No. 2014ZD03-02) and China Unicom Network Technology Research Institute.

References

[1] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, Y. Zhou, Heterogeneous vehicular networking: a survey on architecture, challenges, and solutions, IEEE Commun. Surv. Tut. 17 (4) (2015) 2377–2396.
[2] N.H. Motlagh, M. Bagaa, T. Taleb, UAV-Based iot platform: a crowd surveillance use case, IEEE Commun. Mag. 55 (2) (2017) 128–134.
[3] W. Zhang, P. Duan, W. Gong, Q. Lu, S. Yang, A load-aware pluggable cloud framework for real-time video processing, IEEE Trans. Ind. Inf. 12 (6) (2016) 2166–2176.
[4] J. Xu, Y. Andrepoulos, Y. Xiao, M. Schaar, Non-stationary resource allocation policies for delay-constrained video streaming: application to video over internet-of-things-enabled networks, IEEE J. Sel. Areas Commun. 32 (4) (2014) 782–794.
[5] S. Guo, B. Xiao, Y. Yang, Y. Yang, Energy-efficient dynamic offloading and resource scheduling in mobile cloud computing, IEEE International Conference on Computer Communications (INFOCOM), (2016), pp. 1–9, http://dx.doi.org/10.1109/INFOCOM.2016.7524497.
[6] D. Huang, P. Wang, D. Niyato, A dynamic offloading algorithm for mobile computing, IEEE Trans. Wirel. Commun. 11 (6) (2012) 1991–1995.
[7] Y. Mao, J. Zhang, K.B. Letaief, Dynamic computation offloading for mobile-edge computing with energy harvesting devices, IEEE J. Sel. Areas Commun. 34 (12) (2016) 3590–3605.
[8] M. Chen, Y. Hao, Y. Li, C. Lai, D. Wu, On the computation offloading at ad hoc cloudlet: architecture and service modes, IEEE Commun. Mag. 53 (6) (2015) 18–24.
[9] K. Zheng, H. Meng, P. Chatzimisios, L. Lei, X. Shen, An SMDP-based resource allocation in vehicular cloud computing systems, IEEE Trans. Ind. Electron. 62 (12) (2015) 7920–7928.
[10] T. Taleb, Towards carrier cloud: potential, challenges, and solutions, IEEE Commun. Mag. 21 (3) (2014) 80–91.
[11] X. Xiong, L. Hou, K. Zheng, W. Xiang, M. Hossain, S. Rahman, SMDP-based radio resource allocation scheme in software-defined internet of things networks, IEEE Sens J 16 (20) (2016) 7304–7314.
[12] L. Zhang, D. Fu, J. Liu, E. Ngai, W. Zhu, On energy-efficient offloading in mobile cloud for real-time video applications, IEEE Trans. Circuits Syst. Video Technol. 27 (1) (2017) 170–181.
[13] S. Caton, C. Haas, K. Chard, K. Bubendorfer, O. Rana, A social compute cloud: allocating and sharing infrastructure resources via social networks, IEEE Trans. Serv. Comput. 7 (3) (2014) 359–372.
[14] Z. Su, Q. Xu, H. Zhu, Y. Wang, A novel design for content delivery over software defined mobile social networks, IEEE Netw. 29 (4) (2015) 62–67.
[15] K. Chard, S. Caton, O. Rana, K. Bubendorfer, Social clouds: a retrospective, IEEE Cloud Comput. 2 (6) (2015) 30–40.
[16] X. Chen, B. Proulx, X. Gong, J. Zhang, Exploiting social ties for cooperative d2d communications: a mobile social networking case, IEEE/ACM Trans. Netw. 23 (5) (2015) 1471–1484.
[17] I. Koutsopoulos, E. Noutsi, G. Iosifidis, Dijkstra goes social: social-graph-assisted routing in next generation wireless networks, 20th Proceedings of European Wireless, (2014), pp. 200–706.
[18] F. Liu, K. Zheng, W. Xiang, H. Zhao, Design and performance analysis of an energy-efficient uplink carrier aggregation scheme, IEEE J. Sel. Areas Commun. 32 (2) (2014) 197–207.
[19] A. Argyriou, K. Poularakis, G. Iosifidis, L. Tassiulas, Video delivery in dense 5g cellular networks, IEEE Netw. 31 (4) (2017) 28–34.
[20] K. Zheng, F. Liu, L. Lei, C. Lin, Y. Jiang, Stochastic performance analysis of a wireless finite-state markov channel, IEEE Trans. Wirel. Commun. 12 (2) (2013) 782–793.
[21] H. Liang, L.X. Cai, D. Huang, X. Shen, D. Peng, An SMDP-based service model for interdomain resource allocation in mobile cloud networks, IEEE Trans. Veh. Technol. 61 (5) (2012) 2222–2232.
[22] S. Jia, X. Wang, L. Shen, A continuous-time markov decision process-based method with application in a pursuit-evasion example, IEEE Trans. Syst. Man Cybern. 46 (90) (2016) 1215–1225.
[23] I. Chen, F. Bao, J. Guo, Trust-based service management for social internet of things systems, IEEE Trans. Dependable Secure Comput. 13 (6) (2016) 684–696.
[24] R. Gupta, N. Krishnamurthi, U. Wang, T. Tamminedi, M. Gerla, Routing in mobile ad-hoc networks using social tie strengths and mobility plans, IEEE Wireless Communications and Networking Conference, (2017), pp. 1–6.
[25] Z. Feng, Z. Feng, T. Gulliver, Effective small social community aware d2d resource allocation underlaying cellular networks,. IEEE Wirel. Commun. Lett., Vol. PP, no. 99.
[26] P. Zhao, L. Feng, P. Yu, W. Li, X. Qiu, A social-aware resource allocation for 5g device-to-device multicast communication, IEEE Access 5 (2017) 15717–15730.
[27] M. Puterman, Markov Decision Processes: Discrete Stochastic Dynamic Programming, Wiley, New York, 2005.