Energy-Efficient Joint Offloading and Resource Allocation Strategy in Vehicular Networks

Wei Wu(✉), Ning Wang, Xuanli Wu, and Lin Ma

Communication Research Center, Harbin Institute of Technology, Harbin 150080, China
{kevinking, xlwu2002, malin}@hit.edu.cn, 19S105158@stu.hit.edu.cn

Abstract. In the vehicular networks integrated with mobile edge computing (MEC), vehicle users are permitted to offload latency-sensitive and computation-intensive tasks to nearby MEC servers, which can extend battery life of the vehicle while improving the experience of users. In this paper, we consider a multi-user computation offloading scenario in vehicular networks with MEC server, in which tasks are executed at vehicle and MEC server parallelly through partial offloading. However, the finite communication and computation resource limit the flexibility of offloading. We propose a joint offloading and resource allocation algorithm based on improved hybrid particle swarm and simulated annealing to reduce the system energy consumption as much as possible. The simulation results demonstrate that our algorithm performs well in convergence and energy consumption under strict time constraint.

Keywords: Vehicular networks · Mobile edge computing · Partial offloading · Communication and computation resource allocation

1 Introduction

With the rapid development of emerging mobile applications, the communication and computation requirements of latency-sensitive and computation-intensive applications place higher demands on vehicle networks, which requires powerful computation capability to meet such applications with low latency requirements. In addition, with the introduction of Green 5G, energy consumption has also become the focus of many studies. MEC makes up for the limited computation capability and battery capacity of the vehicle by deploying computation resource and storage resource on the network edge. Moreover, compared with cloud computing, lower latency can be obtained through executing tasks at the MEC server closer to users, which is in line with the communication and computation requirements of the Internet of Vehicles [1].

In practice, the vehicle may need to run streaming data processing applications (such as video analysis). For such applications, the data partition can be used to divide the task into sub-tasks, which can be calculated parallelly at the vehicles and MEC servers [2].

There have been many studies on MEC, most of which focused on offloading decision and resource allocation strategies. On partial offloading, the interaction

Q. Guo et al. (Eds.): WiSATS 2021, LNICST 410, pp. 685–696, 2022.
https://doi.org/10.1007/978-3-030-93398-2_59

between the edge cloud and users is modeled with a Stackelberg game, and uniform and differentiated pricing schemes to find optimal offloading proportion for maximizing MEC server's computation revenue and minimizing users' cost are proposed in [3]. Some works researched the strategy of offloading decision and resource allocation jointly. [4] studied the offloading of VR applications via downlink in vehicular networks, and proposed a three-stage heuristic algorithm to minimize the maximal task completion time by jointly determining offloading proportion and resource allocation. In [5], the scenario where users offload tasks to macro base station through backhaul was considered, and the system computation overhead was minimized through joint optimization of task offloading, wireless backhaul bandwidth resource and computation resource allocation. Based on convex optimization and Hungarian algorithm, [6] proposed a bi-level programming algorithm to optimize binary offloading, power and subcarriers allocation to minimize the system energy consumption jointly. [7] proposed a collaborative scheme of optimizing binary offloading and computation resource allocation to maximize the utility fuction in Cloud-MEC based vehicular networks. [8] applied particle swarm optimization and Nash equilibrium to solve binary offloading and resource allocation respectively, and alternately iterated the two sub-algorithms to maximize the total utility of all vehicles under limited delay, wireless and computation resource. [9] studied the MEC system with physical layer security, and a scheme based on a convex algorithm was proposed to optimize the allocation of local computing tasks and CPU frequency, offloading power and timeslots allocation collaboratively.

The aforementioned works have rarely studied the energy-efficient multi-user MEC system with strict time constraint, limited communication and computation resource for streaming data processing applications. In this case, offloading proportion and resource allocation need to be considered jointly.

In this paper, aim to minimize the system energy consumption while meeting the time constraint in the multi-user vehicular networks, we propose a joint optimization of the offloading proportion, communication and computation resource allocation. The main contributions can be summarized as:

1) We establish a joint optimization model for streaming data processing applications in the vehicular networks with limited resource, in which the system energy consumption is minimized while task completion deadline is met by jointly optimizing offloading proportion and allocation of communication and computation resource.
2) We propose a joint task offloading and resource allocation algorithm based on improved hybrid particle swarm and simulated annealing (IHPS-JORA) to solve the NP-hard problem. The simulation results show that the proposed algorithm can converge fast with strong global optimization capability, and can significantly reduce system energy consumption with time constraint by determining partial offloading decision and resource allocation reasonably.

The rest of this paper is organized as follows. In Sect. 2, system model is illustrated in detail. In Sect. 3, the problem of system energy consumption minimization is formulated. In Sect. 4, we propose a joint task offloading and resource allocation algorithm based on improved hybrid particle swarm and simulated annealing, and the execution steps are described in detail. The simulation results are presented in Sect. 5 to indicate the performance of our proposed algorithm. In Sect. 6, the conclusions are given.

2 System Model

In this part, we describe the task offloading scenario in vehicular networks in detail, including models of the network, communication and computation.

2.1 Network Model

As shown in Fig. 1, we considered a vehicular network covering a section of urban road, which consists of a micro base station (MBS) equipped with a MEC server and several vehicles. We assume that all vehicles are running streaming data processing applications. Since the MBS is equipped with a MEC server, it can provide communication resource and computation resource for vehicles in the coverage of MBS. Vehicles also have limited computation resource. In this network, tasks can be partitioned into two parts, which are called sub-tasks, at any proportion. One is transmitted to the MBS and executed by MEC server through partial offloading, the other can only be executed by vehicles locally. Here, we ignored the transmission time for return of the computation results the computation results from MBS [10, 11], since the output data is usually much less than input data. For simplicity, we assume that all sub-tasks offloaded to MEC server can be executed in parallel without queuing.

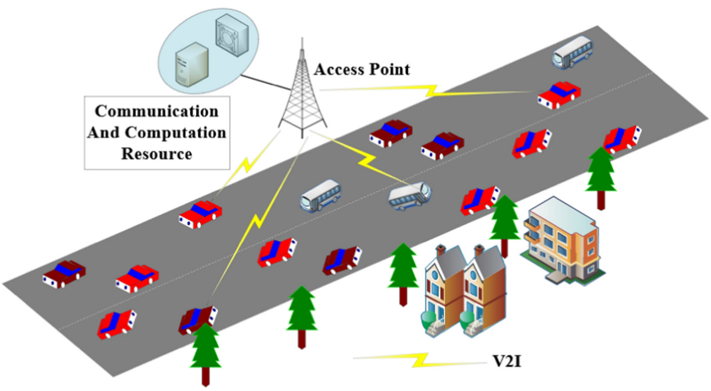


Fig. 1. Multi-user vehicular networks with MEC.

The set of vehicles is expressed as by $\mathbf{N} = \{1,2, \ldots, n, \ldots, N\}$. Task of vehicle n is described by a four-field variable $T_n = \{D_n, \beta_n, \tau_n, a_n\}$. This variable contains the size of the task input-data D_n (in Mb), the computation intensity β_n (in CPU cycles per bit), the requested completion deadline τ_n (in msec) and offloading proportion a_n, a_n [0,1]. Then sub-task $a_n D_n$ will be calculated by MEC server and sub-task $(1 - a_n)D_n$ will be executed by vehicle n. The offloading proportion vector is expressed as $\mathbf{A} = [a_1, a_2, \ldots, a_n, \ldots, a_n]$.

2.2 Communication Model

K available sub-channels can be allocated for wireless transmission of vehicles. The bandwidth of each sub-channel is W. The number of sub-channels allocated to vehicle n is denoted by b_n. Then the achievable data transmission rate for vehicle n is expressed as

$$R_n = b_n W \log_2\left(1 + \frac{p_n^t h_n \gamma_n d_n^{-\theta}}{N_0}\right) \tag{1}$$

Here, the path loss is modeled as $d_n^{-\theta}$, where d_n is the distance between vehicle n and the MEC server, θ denotes the path loss exponent. h_n and γ_n represent Rayleigh fading and shadow fading, respectively. p_n^t is the transmission power of vehicle n, and N_0 is the power of white noise.

Since the number of sub-channels of MBS is limited, the allocation of sub-channels is limited by

$$\sum_{n=1}^{N} b_n \leq K \tag{2}$$

The communication resource allocation vector is defined as $\mathbf{B} = [b_1, b_2, \ldots, b_n, \ldots, b_N]$. For vehicle n, the uplink transmission time is given by

$$t_n^{trans} = \frac{a_n D_n}{R_n} \tag{3}$$

Then the corresponding transmission energy consumption can be given by

$$e_n^{trans} = p_n^t t_n^{trans} \tag{4}$$

2.3 Computation Model

In this paper, the task n can be divided into 2 sub-tasks. a_n is the proportion of subtask with is computed remotely. Therefore, $a_n D_n$ and $(1 - a_n)D_n$, is the sub-task size of remote execution and local execution respectively. Vehicle uses these two execution modes parallelly to complete the execution of task.

With respect to different execution modes, time and energy consumption are presented separately as below.

Local Execution. For vehicle n, the frequency of local CPU is $f_{n,loc}$, thus the local execution time of sub-task at vehicle is expressed by

$$t_n^{loc} = \frac{(1 - a_n) D_n \beta_n}{f_{n,loc}} \tag{5}$$

The energy consumption for each CPU cycle of the vehicle n is $k_u f_{n,loc}^2$ according to [12], where k_u is a constant. Then, the energy consumption of sub-task at vehicle for local execution can be given by

$$e_n^{loc} = k_u(1 - a_n)D_n\beta_n f_{n,loc}^2 \tag{6}$$

Remote Execution. We assume that MEC server's CPU frequency is F_{MEC}, which is much higher than that of any vehicle. The computation resource allocated by the MEC server to vehicle n is denoted by f_n, thus the remote execution time is given by

$$t_n^{exe} = \frac{a_n D_n \beta_n}{f_n} \tag{7}$$

The computation resource allocation vector is denoted as $\mathbf{F} = [f_1, f_2, \ldots, f_n, \ldots, f_N]$. Since the the MEC server's computation capability of is limited, the allocation of computation resource is limited by

$$\sum_{n=1}^{N} f_n \le F_{MEC} \tag{8}$$

The remote execution time includes transmission time and actual computation time, thus it is given by

$$t_n^{off} = t_n^{trans} + t_n^{exe} \tag{9}$$

Similarly, the energy consumption of sub-task at MEC server for remote execution can be given by

$$e_n^{exe} = k_s a_n D_n \beta_n f_n^2 \tag{10}$$

Where k_s is a constant, as a rule, $k_s < k_u$. It can be seen that computation time is shorter, and the corresponding energy consumption is higher, when the computation resource MEC server allocates to vehicle is more. Then, the total energy consumption for vehicle n is given by

$$e_n = e_n^{loc} + e_n^{trans} + e_n^{exe} \tag{11}$$

3 Problem Formulation

In this paper, it is considered that time and energy consumption for streaming data processing applications, aiming to reduce the system energy consumption as much as possible while the task completion deadline is met. The system energy consumption is the total energy consumption of all vehicles. The optimal problem can be formulated as

$$\min_{\mathbf{A},\mathbf{F},\mathbf{B}} E(\mathbf{A},\mathbf{B},\mathbf{F}) = \sum_{n=1}^{N} e_n \tag{12}$$

$$s.t. \quad \text{C1: } \max\left\{t_n^{off}, t_n^{loc}\right\} \le \tau_n, \forall n \in \mathrm{N}$$

$$\text{C2: } \sum_{n=1}^{N} b_n \le K$$

$$\text{C3: } \sum_{n=1}^{N} f_n \le F_{MEC}$$

$$\text{C4: } 0 \le a_n \le 1, \forall n \in \mathrm{N}$$

$$\text{C5: } b_n > 0, \forall n \in \mathrm{N}$$

$$\text{C6: } f_n > 0, \forall n \in \mathrm{N}$$

Where $\mathbf{A} = [a_1, a_2, \ldots, a_n, \ldots, a_N]$, $\mathbf{B} = [b_1, b_2, \ldots, b_n, \ldots, b_N]$, $\mathbf{F} = [f_1, f_2, \ldots, f_n, \ldots, f_N]$. Constraint C1 limits the completion time of all tasks. C2 is the constraint on the number of available sub-channels of MBS. C3 constrains the allocation of computation capability of MEC server. C4 indicates the offloading proportion's range. C5 and C6 represents that the resource allocation of communication and computation is always positive.

Aim to minimize the system energy consumption while all the constraints are satisfied, the solution of the optimal offloading proportion solution $\mathbf{A}^*$, the solution communication resource allocation $\mathbf{B}^*$ and the solution of computation resource allocation $\mathbf{F}^*$ for all vehicles need to be find.

4 Joint Offloading and Resource Allocation Algorithm

This optimization problem is a mixed integer nonlinear programming (MINLP) problem. It can be proved to be a NP-hard problem. Next, the intelligent optimization algorithms will be applied to design a joint task offloading and resource allocation algorithm.

To solve the problem, global optimization algorithm is needed. Particle Swarm Optimization (PSO) is one of them. It is a swarm intelligence algorithm and can converges fast. PSO is usually used to solve unconstrained optimization problems, then we introduce the exterior penalty function method to enable it to handle problems with constraints. PSO tends to be trapped in the local optimal solution, while simulated annealing (SA) algorithm has a strong ability in breaking away from the local optimal solution. In view of this, we consider combining PSO with SA.

The proposed joint task offloading and resource allocation algorithm based on improved hybrid particle swarm and simulated annealing (IHPS-JORA) is illustrated in **Algorithm 1**. The corresponding specific steps are detailed as:

Step 1: Initialization. Inertia weights $\omega_{\max}$ and $\omega_{\min}$, acceleration factors c_1 and c_2, number of particles M, maximum iterations S of PSO, initial temperature T_0, freezing rate R_T, maximum iterations C of SA. Then, the position matrix P_m and velocity matrix V_m of each particle are generated randomly under constraints, that is, the initial offloading proportion and resource allocation of all vehicles are specified. And calculate fitness $f(P_m)$, i.e. the system energy consumption.

Step 2: SA Iterations. Each particle is based on SA independently. For the particle m, its position of the s-th iteration is given as

$$P_m(s) = \left(P_m^1, P_m^2, \ldots, P_m^n, \ldots, P_m^N\right) \tag{13}$$

Where $P_m^n = (a_n, b_n, f_n)$ represents the offloading proportion, communication and computation resource allocation for vehicle n.

At present temperature $T(s)$ of the s-th iteration, a new local optimal position $P_{best}^{m,trial}$ of particle m by adding perturbation on the previous P_{best}^{m} is generated, then boundary processing and exterior penalty. Exterior penalty can force particles that do not meet the constraints of (12) to return to feasible region.

If $f(P_{best}^{m,trial}) \leq f(P_{best}^{m})$ then $P_{best}^{m} = P_{best}^{m,trial}$. Otherwise, whether to accept $P_{best}^{m,trial}$ according to Metropolis Criterion. This step is repeated until iterations C of SA is reached.

Step 3: Update the position of particles and velocity. For the particle m, its velocity is given as

$$V_m(s) = \left(V_m^1, V_m^2, \ldots, V_m^n, \ldots, V_m^N\right) \tag{14}$$

For each iteration of PSO, updating of velocity and position in the d-dimension of particle m is given by

$$\omega = \omega_{\max} - \left(\frac{\omega_{\max} - \omega_{\min}}{S}\right)s \tag{15}$$

$$V_{md}(s+1) = \omega V_{md}(s) + c_1 \mathrm{rand}_1\left(P_{best}^{m}(s) - P_{md}(s)\right) + c_2 \mathrm{rand}_2\left(G_{best}(s) - P_{md}(s)\right) \tag{16}$$

$$P_{md}(s+1) = P_{md}(s) + V_{md}(s+1) \tag{17}$$

$$P_{md}(s+1) = round\left(P_{md}(s) + V_{md}(s+1)\right) \tag{18}$$

Where rand_1 and rand_2 are random variables in [0,1], P_{best}^{m} is the local optimal position of particle m, and G_{best} represents the global optimal position of swarm. Note that the offloading proportion and computation resource allocation are both continuous variables, and positions are updated according to (17); the wireless resource allocation is a discrete variable, which needs to be updated according to (18).

Step 4: Check for improvement in the local and global optimal position. Boundary processing and exterior penalty on the new position of each particle, then update P_{best}^{m} and G_{best}.

Step 5: Convergence. If iterations S of PSO is reached, output G_{best} as the best solution, i.e.$G_{best} = [\mathrm{A}^*, \mathbf{B}^*, \mathbf{F}^*]$. Otherwise, update temperature as (19) and go to step 2.

$$T(s) = R_T T(s-1) \tag{19}$$

Algorithm 1: IHPS-JORA Algorithm

```
Input: ω_max, ω_min, c_1, c_2, M, S, T_0, R_T, C
Output: G_best = [A*, B*, F*] : optimal offloading proportion policy and resource allocation
Initialization
while s ≤ S do
   s = s + 1;
   for m = 1 : M do
      while c ≤ C do
       c = c + 1;
      Generate new P_best^{m,trial} by adding perturbation, then boundary processing, exterior penalty
and calculate f(P_best^{m,trial});
       if f(P_best^{m,trial}) ≤ f(P_best^m), P_best^m = P_best^{m,trial} then
       else Whether to accept P_best^{m,trial} according to Metropolis Criterion;
       end if
       Update G_best;
       end while
       Update velocity and position of each particle according to (15), (16) and (17), (18)
respectively, then boundary processing;
       Exterior penalty and update P_best^m and G_best;
     end for
     Update temperature according to (19);
end while
return G_best
```

5 Performance Evaluations

We verify our proposed algorithm's performance through simulations.

5.1 Parameters

In the simulations, a vehicular network with multiple vehicles and one base station with MEC server is taken into account. Note that in practice, the computation capability, transmission power, and computation intensity of each vehicle are different, but the heterogeneity of users has no effect on the performance of our algorithm. The basic simulation parameters are shown in the Table 1.

Table 1. Simulation parameters.

System parameters	Values
Shadow fading standard deviation	8 dB
Sub-channel bandwidth W	0.4 MHz
Number of sub-channels K	150
Transmission power of vehicle $p_n^t, n \in \mathbf{N}$	30 dBm
Computation capacity of MEC server F_{MEC}	50 GHz
Computation capacity of vehicle $f_{n,loc}, n \in \mathbf{N}$	0.8 GHz
Capacitance coefficient of MEC server k_s	10^{-25}
Capacitance coefficient of vehicle k_u	10^{-24}
Input data size $D_n, n \in \mathbf{N}$	1.8–2 Mb
Computation intensity $\beta_n, n \in \mathbf{N}$	60 cycles/bit
Task completion deadline $\tau_n, n \in \mathbf{N}$	80 ms

5.2 Simulations

Since the proposed algorithm is an iterative algorithm, we first verify its convergence. As shown in Fig. 2, as the iterative number increasing, the system energy consumption gradually reduces, and eventually tends to stabilize and converge. In addition, as the number of vehicles increases, the convergence of the our algorithm will not deteriorate.

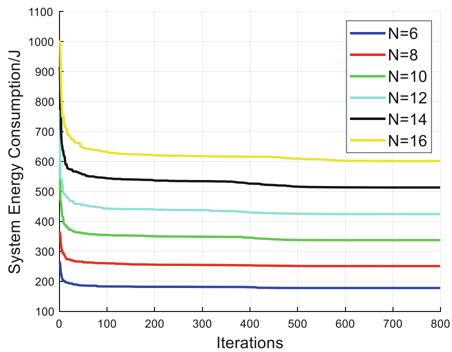
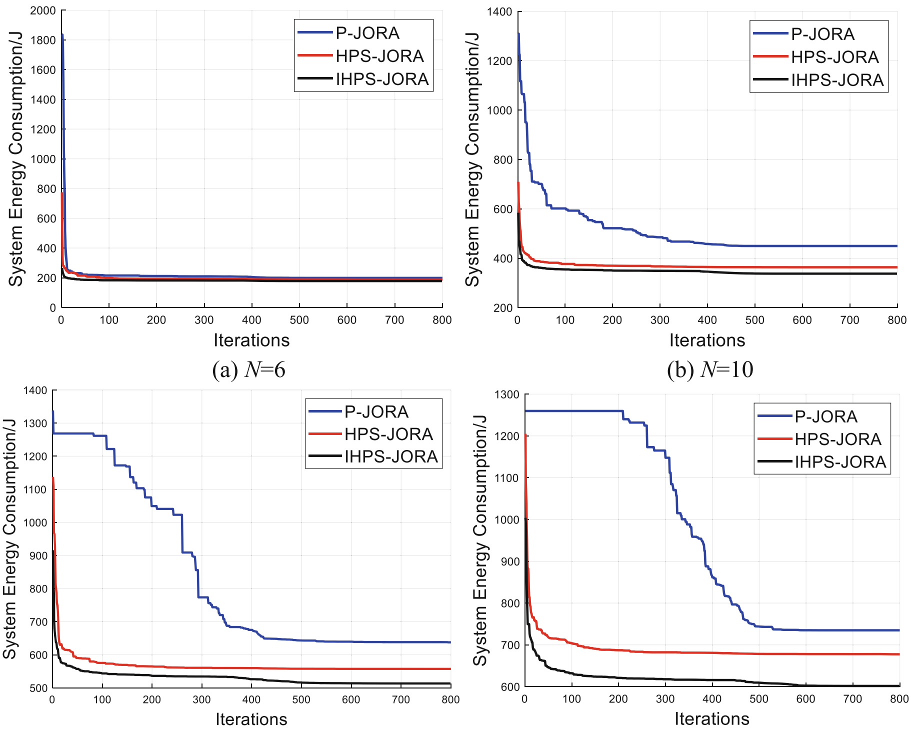


Fig. 2. Convergence for different numbers of vehicles.

Furthermore, we compared proposed algorithm with other joint offloading and resource allocation algorithms, such as particle swarm algorithm, another hybrid particle swarm algorithm and simulated annealing algorithm [13], named as P-JORA and HPS-JORA respectively, as shown in Fig. 3. It shows that, for the same number of vehicles, the proposed IHPS-JORA converges faster and can obtain a better approximate global optimal solution. Besides, when there are fewer vehicles, the convergence of the three algorithms is very close; when there are more vehicles, the convergence of the three algorithms differs greatly. It is because that with the number of vehicles



(a) N=6 (b) N=10 (c) N=14 (d) N=16

Fig. 3. Convergence comparison of algorithms.

increases, resource gradually becomes insufficient, which further tests the algorithm's global optimization capability.

We introduce two baseline schemes as comparisons. In the All-Local, all tasks are computed at vehicles; in the All-Edge, all tasks are offloaded to the MEC server for compution.

In Fig. 4, as the number of vehicles increases, the system energy consumption continues to increase. Then, when there are fewer vehicles, the performance of All-Edge, P-JORA, HPS-JORA and IHPS-JORA is similar. This is because the resource is sufficient and can easily meet the requirements of all vehicles. With the number of vehicles increases, resource is gradually becoming insufficient. Compared with other algorithms, the proposed IHPS-JORA algorithm always achieves lower system energy consumption. Note that for the simulation parameters of this paper, the execution time of All-Local scheme is at least 135 ms, which cannot meet the limit of task completion time.

Figure 5 presents the relationship between the usage of computation resource of edge server and the number of vehicles. When the number of vehicles increases, the computation resource usage of MEC server for all algorithms continues to increase. Then, for the same number of vehicles, the computation resource usage of server of IHPS-JORA is the least. The less the MEC server's computation resource is used, the

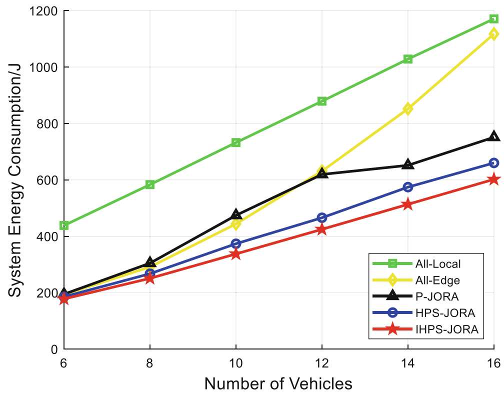


Fig. 4. System energy consumption vs. the number of vehicles.

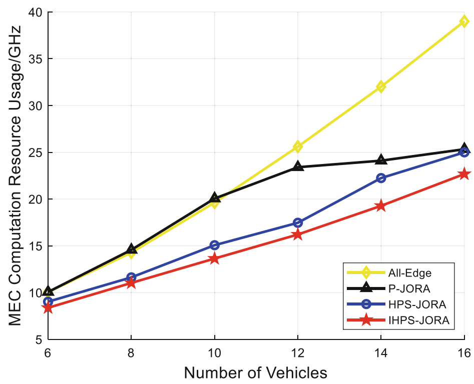


Fig. 5. The computation resource usage of MEC server for different number of vehicles.

more computation resource can be reserved to carry more new offloading tasks. Therefore, under the premise of meeting the predetermined goals, it is hoped that the usage of MEC server's computation resource is as small as possible.

6 Conclusion

In this paper, we studied the multi-user vehicular networks integrated with MEC. The aim is to minimize the system energy consumption for task under strict time constraint, and we formulated a joint optimization model for offloading proportion and resource allocation. We proposed an IHPS-JORA algorithm to achieve the approximate optimal solution to solve this problem. The simulation results reveal that our algorithm converges fast while having strong global optimization capability. In addition, our algorithm can reduce system energy consumption significantly while taking up less computation resource of MEC server, so it has better task offloading and resource allocation performance.

References

1. Abbas, N., Zhang, Y., Taherkordi, A., Skeie, T.: Mobile edge computing: a survey. IEEE Internet Things J. **5**(1), 450–465 (2018)
2. Jiang, C., Cheng, X., Gao, H., Zhou, X., Wan, J.: Toward computation offloading in edge computing: a survey. IEEE Access **7**, 131543–131558 (2019)
3. Liu, M., Liu, Y.: Price-based distributed offloading for mobile-edge computing with computation capacity constraints. IEEE Wirel. Commun. Lett. **7**(3), 420–423 (2018)
4. Zhou, J., Wu, F., Zhang, K., Mao, Y., Leng, S.: Joint optimization of offloading and resource allocation in vehicular networks with mobile edge computing. In: 2018 10th International Conference on Wireless Communications and Signal Processing (WCSP), Hangzhou, China (2018)
5. Pham, Q., Le, L.B., Chung, S., Hwang, W.: Mobile edge computing with wireless backhaul: joint task offloading and resource allocation. IEEE Access **7**, 16444–16459 (2019)
6. Cheng, K., Teng, Y., Sun, W., Liu, A., Wang, X.: Energy-efficient joint offloading and wireless resource allocation strategy in Multi-MEC server systems. In: 2018 IEEE International Conference on Communications (ICC), Kansas City, pp. 1–6 (2018)
7. Zhao, J., Li, Q., Gong, Y., Zhang, K.: Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks. IEEE Trans. Veh. Technol. **68**(8), 7944–7956 (2019)
8. Wu, W., Wang, Q., Wu, X., Zhang, N.: Joint offloading and resource allocation for scalable vehicular edge computing. In: VTC2020-Fall, pp. 1–5 (2020)
9. Wang, J.-B., Yang, H., Cheng, M., Wang, J.-Y., Lin, M., Wang, J.: Joint optimization of offloading and resources allocation in secure mobile edge computing systems. IEEE Trans. Veh. Technol. **69**(8), 8843–8854 (2020)
10. You, C., Huang, K., Chae, H., Kim, B.: Energy-efficient resource allocation for mobile-edge computation offloading. IEEE Trans. Wirel. Commun. **16**(3), 1397–1411 (2017)
11. Mao, S., Leng, S., Yang, K., Huang, X., Zhao, Q.: Fair energy-efficient scheduling in wireless powered full-duplex mobile-edge computing systems. In: GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, pp. 1–6 (2017)
12. 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)
13. Javidrad, F., Nazari, M.: A new hybrid particle swarm and simulated annealing stochastic optimization method. Appl. Soft Comput. **60**, 634–654 (2017)