Distributed Incentives and Digital Twin for Resource Allocation in air-assisted Internet of Vehicles

Peng Wang*, Ning Xu*, Wen Sun†, Gaozu Wang† and Yan Zhang‡
*School of Cyber Engineering, Xidian University, Xi'an, 710126, China.
†School of Cybersecurity, Northwestern Polytechnical University, 127 West Youyi Road, Xi'an Shaanxi, 710072, China.
‡University of Oslo, Oslo, Norway, and also with the Simula Metropolitan Center for Digital Engineering, Oslo, Norway.

2021 IEEE Wireless Communications and Networking Conference (WCNC) | 978-1-7281-9505-6/20/$31.00 ©2021 IEEE | DOI: 10.1109/WCNC49053.2021.9417521

***Abstract*—Internet of Vehicles (IoV) can realize seamless communication connection and computing offloading services with the assistance of air communication. Limited by the high network dynamics of the air-assisted IoV, resource allocation faces great challenges. In this paper, dynamic digital twin of air-assisted IoV is established to capture the time-varying resource supply and demands, so that unified resource scheduling and allocation can be performed. We designed an incentive mechanism for resource allocation based on Stackelberg games to maximize vehicle satisfaction and overall energy efficiency. In the game, the digital twin of air-assisted IoV within the coverage of unmanned aerial vehicles (UAV) are regarded as leaders, while RSUs that provide computing services are followers. At the same time, in order to reduce the delay and reduce the computational burden of the UAV, a distributed incentive mechanism based on the Alternating Direction Multiplier Method (ADMM) was designed to optimize the resource allocation strategy of each RSU. Simulation results show that the proposed scheme can improve the satisfaction of vehicles and the energy efficiency at the same time.**

***Index Terms*—digital twin, Internet of Vehicles, incentives, ADMM, Stackelberg game.**

I. Introduction

The Internet of Vehicles (IoV) realizes seamless communication connection and computing offloading services through air communication. Aerial drones can provide services for delay-sensitive and highly mobile IoV anytime and anywhere, thanks to their advantages in coverage, flexibility, and reliability [1]. However, in air-assisted IoV, the highly mobile and unpredictable network topology poses challenges for efficient resource allocation. As an emerging digital mapping technology, digital twins (DT) provide an excellent solution for intelligent resource allocation in IoV by creating a real-time digital simulation model of physical entities [2]–[4]. It optimizes the overall network service quality by implementing dynamic real-time perception and unified scheduling of resources in dynamic scenarios [5], [6].

In the air-assisted IoV, incentives are desired to encourage communication or computing infrastructures, such as road side units (RSUs), to provide services for vehicles [7]. The existing works on offloading computing tasks in IoV mainly focus on meeting the requirements of low latency and high reliability for diverse applications. Zhou *et al.* [8] proposed an efficient incentive mechanism based on contract theoretical modeling. The paper maximizes the expected utility of the base station while considering the unique characteristics of each vehicle type. Sun *et al.* [9] proposed a mobile edge model based on a new vision of Digital Twin Edge Networks (DITEN). This paper used the Lyapunov method to simplify the formal problem of a dynamic multi-objective optimization problem and solve it with a deep reinforcement learning algorithm.

Note that the existing works assume the mobility of vehicles or the data traffic follow a certain pattern and predictable within a period of time. Moreover, the incentives and resource allocation mechanism itself may be computation-intensive and incurs burden to the limited resources of UAVs, thus should be carried out in a distributed manner. Therefore, this paper proposes dynamic digital twin and distributed incentives for resource allocation in air-assisted IoV. First, the dynamic DT for air-assisted IoV is established to adapt to the high temporal and spatial dynamics of the network. Second, an incentive mechanism based on Stackelberg game is designed to optimize and improve vehicles satisfaction and the overall energy efficiency of RSUs. Specifically, the game achieves this goal through the adjustment of resource allocation strategies and incentive compensation. Finally, we use ADMM to release the computational burden on the UAV in parallel to the RSUs, which minimizes the load of the UAV and reduces the delay. Simulation results show that the proposed scheme can improve the satisfaction of vehicles and the energy efficiency at the same time.

The remainder of this paper is organized as follows. The system model are presented in Section II. Sections III and IV illustrate the incentive of task offloading based on Stackelberg game. The performance evaluation is conducted in Section V. Finally, conclusions are drawn in Section VI.

II. System Model

A. Dynamic Digital Twin for air-Assisted IoV

Fig. 1 shows a digital twin driven air-assisted IoV, consisting of vehicles, RSUs, UAVs, and digital twins. UAVs provide supplementary capabilities for terrestrial communications in areas where RSUs cannot cover. We use the set $\mathcal{M} = \{1, \cdots, m, \cdots, M\}$ to represent RSUs in the network. The set of vehicles that require offloading tasks at time slot t is expressed as $\mathcal{N} = \{1, \cdots, n, \cdots, N\}$. In the digital layer,

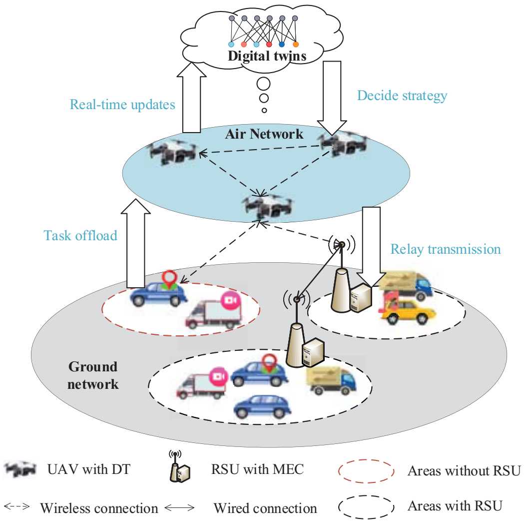

Fig. 1. The influence of preferences on the contribution of RSUs in the first game.

the digital representation of physical entities, e.g., vehicles, RSUs, UAVs, provides the dynamics of how an air-assisted IoV system operates, which can be represented as follows:

$$\mathcal{D} = \{G, L, \mathcal{F}, \mathcal{C}, \mathcal{Q}\}. \tag{1}$$

where G is a digital model of network topology , and L is the network transmission load of system entities, $\mathcal{F} = \{f_1, f_2, \cdots, f_M\}$ is a vector describing idle computing resource status of RSUs, $\mathcal{C} = \{c_1, c_2, \cdots, c_N\}$ represents the demand information of vehicles at this time, $\mathcal{Q} = \{Q_1, Q_2, \cdots, Q_N\}$ is the preference of the vehicle to the resource provider. It is caused by differences in the types of applications supported by computing resources on different RSUs and previous qualities of service.

B. Offloading in DT-driven air-assisted IoV

In the air-assisted IoV, as shown in Fig. 1, the UAV can relay computing tasks such as sensor information processing and automatic navigation to other entities with rich resources, such as RSU. For vehicle $n \in \mathcal{N}$, its task offloading request at time t is expressed as $\{c_n, l_n, Q_n\}$, where c_n is the total number of CPU cycles required to complete the task, l_n is the size of the task request packet, and $Q_n = \{q_{n,1}, ..., q_{n,m}, ...q_{n,M}\}$ is the preference of the vehicle over different RSUs and $q_{n,m}$ represents the preference of vehicle n to RSU m. The digital twin of vehicles is responsible to provide Q_n, which reflects the locality awareness and time constraint of the RSUs. If a RSU is far away from the vehicle and cannot make it within a certain time constraint $q_{n,m} = 0$. For RSU $m \in \mathcal{M}$, its available CPU frequency resources can be expressed as $\{f_m^{\text{idle}}, \beta_m\}$, where f_m^{idle} is the number of CPU frequency, and β_m is the cost per CPU frequency. Therefore, in the resource offload, we finally determine the CPU resources provided by each RSU for the vehicle.

The DT on the UAV can get the task request information, vehicle preference information and idle resource information in the system in real time. Therefore, DT can develop the best strategy $\mathcal{F}^* = \{f_1^*, f_2^*, ..., f_m^*\}$ to determine the actual contribution resources of each RSU. Subsequently, the problem is to determine the resource allocation policy $\mathcal{P} = \{P^m, m \in \mathcal{M}\}$ to maximize vehicle satisfatction and the energy efficiency ratio of RSUs.

C. Computation Delay Model in DT-driven air-Assisted IoV

The computation delay depends on two aspects: how busy the RSU is in the system and how the computational tasks are distributed. After deciding the task offloading policy $\mathcal{P}$, the computation delay for vehicle n is $t_{m,n} = c_{n,m}/p_{m,n}$, where $c_{n,m}$ is the CPU cycles required for RSU m to calculate tasks of RSU m for vehicle n, and $p_{m,n}$ is the CPU frequency provided by RSU m for vehicle n. Therefore, the total computation delay of RSU m is calculated as

$$T_m(P_m) = \sum_{n \in \mathcal{N}} t_{m,n}. \tag{2}$$

Furthermore, the global delay is the sum of the delays of all vehicles as follows:

$$T(\mathcal{P}) = \sum_{m \in \mathcal{M}} T_m(P_m). \tag{3}$$

D. Utility Function

For vehicle n, it tends to maximize its service satisfaction, which is the accumulated satisfaction the vehicle achieves from various RSUs. Satisfaction of vehicle is defined as the ratio of the cumulative satisfaction of the vehicle from each RSU to the total number of resources received by the vehicle. Thus the satisfaction of vehicle n is

$$\mathcal{S}_n = \frac{\sum_{m \in \mathcal{M}} \left\{ q_{n,m} p_{m,n} - \frac{q_{n,m} p_{m,n}^2}{2\tilde{f}} \right\}}{\sum_{m \in \mathcal{M}} p_{m,n}}, \tag{4}$$

where $\tilde{f} = \kappa f^{max}/(NM)$ is the maximum expected value of resources from RSUs, $f^{max} = \sum_{m \in \mathcal{M}} f_m^{idle}$ is calculated based on the idle CPU size reported by RSUs before starting the game. κ is a constant. This means the satisfaction $\mathcal{S}_n$ is positively correlated with preference $q_{m,n}$, and $\mathcal{S}_n$ achieves the maximum when the allocated CPU frequency $p_{m,n}$ equals to $\tilde{f}$, and the marginal benefit is guaranteed to decrease.

For the DT, it considers how to reduce energy consumption of all RSUs. According to [10], the energy consumption on the RSU is related to the frequency and duration of the CPU used. We can express energy consumption as:

$$E(\mathcal{P}) = \sum_{m \in \mathcal{M}} \sum_{m \in \mathcal{N}} \omega p_{m,n}^2 c_{n,m}. \tag{5}$$

where ω is a constant determined by the CPU architecture.

In order to find an optimal computing resource allocation scheme and meet the goal of minimizing the energy consumption of RSUs and maximizing the satisfaction of vehicles, we construct an incentive scheme based on Stackelberg game to make the DT motivates RSUs.

III. DISTRIBUTED INCENTIVES FOR MAXIMIZING VEHICLE SATISFACTION AND ENERGY EFFICIENCY

We first deduce the optimization problem of maximizing satisfaction and minimizing energy consumption before constructing the Stackelberg game. We use the classic-ADMM with two blocks and Jacobian-ADMM to solve the above two problems in Section III-A and Section III-B respectively. Moreover, we model the two problems as a complete Stackelberg game and give the solution of it in Section III-C.

A. Average Satisfaction Maximization Problems For Vehicles

The goal of vehicles is to get the best quality of service. However, vehicle satisfaction is related to the resource allocation decision of each RSU, which is not determined by the vehicle. Therefore, we turn the problem into maximizing the satisfaction of each RSU to vehicles. According to Eqn. (4) and related constraints, for each RSU, the problem of maximizing average satisfaction can be described as

$$\mathbf{P1}': \quad \underset{P_m=\{p_{m,n}\}}{maximize} \quad \frac{\sum\limits_{n\in\mathcal{N}}\left(q_{n,m}p_{m,n}-\frac{q_{n,m}}{2\tilde{f}}p_{m,n}^2\right)}{\sum\limits_{n\in\mathcal{N}}p_{m,n}}$$
$$\text{s.t.} \quad \sum_{n\in\mathcal{N}}p_{m,n}=f_m,\ m\in\mathcal{M} \qquad (C1)$$

where $C1$ indicates that the upper bound of the computing resources that can be allocated by RSUs.

Obviously, $\mathbf{P1}'$ is a non-convex fractional programming problem. We can use Dinkelbach algorithm to convert such non-convex fractional optimization problem into convex optimization problem. Therefore, $\mathbf{P1}'$ is equivalent to minimizing the following function

$$h(P_m,\eta_m)=\sum_{n\in\mathcal{N}}\left(\frac{q_{n,m}}{2\tilde{f}}p_{m,n}^2-q_{n,m}p_{m,n}\right)+\eta_m\sum_{n\in\mathcal{N}}p_{m,n}. \tag{6}$$

It is easy to see from (6) that $h(P_m,\eta_m)$ is a strictly monotonic increasing function with respect to η_m, and $h(P_m,\eta_m)$ is a convex function with respect to P_m.

Theorem 1: The maximum average satisfaction could be obtained by vehicle n, if and only if $h(P_m^*,\eta_m^*)=0$.

The proof of $Theorem\,1$ can refer to [11]. Therefore, $\mathbf{P1}'$ can be transformed into $\mathbf{P1}$ as follows:

$$\mathbf{P1}: \quad \underset{P_m}{minimize} \quad h(P_m,\eta_m)$$
$$\text{s.t.} \quad \sum_{n\in\mathcal{N}}p_{m,n}=f_m,\ m\in\mathcal{M} \qquad (C1)$$

$\mathbf{P1}$ is a convex optimization problem. If we want to use ADMM to solve it, we need to further transform $\mathbf{P1}$. First, we divide the purchase decision P_m of vehicle m into two parts, that is $\mathbf{x}=\{(P_{m,1}),\cdots,(P_{m,k})\}^T$ and $\mathbf{z}=\{(P_{m,k+1}),\cdots,(P_{m,N})\}^T$. Based on this, $\mathbf{P1}$ can be transformed into the following problem:

$$\tilde{\mathbf{P1}}: \quad \underset{\mathbf{x}}{minimize} \quad \Gamma(\mathbf{x})+\Psi(\mathbf{z})$$
$$\text{s.t.} \quad \mathbf{E_x x}+\mathbf{E_z z}=f_m \tag{7}$$

where $\mathbf{E_x x}\in\mathbf{R}^{1\times k}$, $\mathbf{E_z z}\in\mathbf{R}^{1\times(N-k)}$. Both are unit vectors. Therefore, $\Gamma(\mathbf{x})$ and $\Psi(\mathbf{z})$ are represented as

$$\Gamma(\mathbf{x})=\sum_{n=1}^{k}\left(\frac{q_{n,m}}{2\tilde{f}}p_{m,n}^2-q_{n,m}p_{m,n}\right)+\eta_m\sum_{n=1}^{k}p_{m,n}. \tag{8}$$

$$\Psi(\mathbf{z})=\sum_{n=k+1}^{N}\left(\frac{q_{n,m}}{2\tilde{f}}p_{m,n}^2-q_{n,m}p_{m,n}\right)+\eta_m\sum_{n=k+1}^{N}p_{m,n}. \tag{9}$$

The corresponding augmented Lagrangian expression is

$$\mathcal{L}_\delta(\mathbf{x},\mathbf{z},\mathbf{y})=\Phi(\mathbf{x})+\Psi(\mathbf{z})+\frac{\delta}{2}\left\|\mathbf{r}+\boldsymbol{\mu}\right\|_2^2-\frac{\delta}{2}\left\|\boldsymbol{\mu}\right\|_2^2, \tag{10}$$

where $\mathbf{r}=\mathbf{E_x x}+\mathbf{E_z z}-f_m$ is primal residual. δ is a positive constant and represents the penalty parameter. Note that $\mathbf{y}$ is a vector of Lagrange multipliers, and $\boldsymbol{\mu}=\frac{\mathbf{y}}{\delta}$ is a vector of the scaled dual variables.

According to [12], the variable update steps at iteration $l+1$ are presented as

$$\mathbf{x}^{l+1}=\arg\min\Gamma(\mathbf{x})+\frac{\delta}{2}\left\|\mathbf{E_x x}+\mathbf{E_z z}^l-f_m+\boldsymbol{\mu}^l\right\|_2^2, \tag{11}$$

$$\mathbf{z}^{l+1}=\arg\min\Psi(\mathbf{z})+\frac{\delta}{2}\left\|\mathbf{E_x x}^{l+1}+\mathbf{E_z z}-f_m+\boldsymbol{\mu}^l\right\|_2^2, \tag{12}$$

$$\boldsymbol{\mu}^{l+1}=\boldsymbol{\mu}^l+\mathbf{E_x x}^{l+1}+\mathbf{E_z z}^{l+1}-f_m. \tag{13}$$

Before giving the termination criterion, we need introduce a variable of the dual residual $\mathbf{s}^{l+1}$ expressed as

$$\mathbf{s}^{l+1}=\delta\mathbf{E_x^T E_z}\left(\mathbf{z}^{l+1}-\mathbf{z}^l\right). \tag{14}$$

Therefore, the termination criteria for ADMM is given as follows:

$$\left\|\mathbf{r}^{l+1}\right\|_2^2\le\epsilon_1 \quad and \quad \left\|\mathbf{s}^{l+1}\right\|_2^2\le\epsilon_2, \tag{15}$$

where $\epsilon_1>0$ and $\epsilon_2>0$ are very small, denoting feasibility tolerances with respect to primal conditions and dual conditions. Moreover, the termination criteria for Dinkelbach algorithm is denoted as

$$\Delta^{l+1}=\left|\left(\Gamma(\mathbf{x}^{l+1})+\Psi(\mathbf{z}^{l+1})\right)-\left(\Gamma(\mathbf{x}^l)+\Psi(\mathbf{z}^l)\right)\right|\le\varepsilon, \tag{16}$$

where ε is a positive constant that close to 0. Thus, the iteration is terminated when (15) and (16) are satisfied at the same time.

B. Global Energy Efficiency Maximization Problems For DT

Based on (5), the global efficiency maximization problemsm for DT can be formulated as

$$\begin{aligned}\mathbf{P2}: \quad & \underset{\mathcal{P}}{minimize} \quad E(\mathcal{P}) \\ & \text{s.t.} \quad \sum_{n\in\mathcal{N}} p_{m,n} = f_m, \forall m \in \mathcal{M} \quad (C1)\end{aligned} \tag{17}$$

where $C1$ is available computing resource constraints for all RSUs. $\underset{\mathcal{P}}{minimize}\, E(\mathcal{P})$ is equivalent to $\underset{\mathcal{P}}{minimize} \sum_{m\in\mathcal{M}} E_m(P_m)$, and $E_m(P_m)$ is a convex function, which could be proved by deriving the Hessian matrix. Thus, $\mathbf{P2}$ is a convex optimization problem. However, the classical two block ADMM algorithm can not solve the convex optimization problem of high-dimensional variables. To solve this problem, we adopt Jacobian ADMM, which can deal with the problem in parallel. Jacobian ADMM is essentially an ADMM with N blocks ($N > 2$), we need divide the matrix of computing resources allocation into M parts. Base on this, $\mathbf{P2}$ can be rewritten as

$$\begin{aligned}\tilde{\mathbf{P2}}: \quad & \underset{\mathcal{P}}{minimize} \quad \sum_{m=1}^{M} E(P_m) \\ & \text{s.t.} \quad \sum_{m=1}^{M} \mathbf{A}_m P_m = \mathbf{W}\end{aligned} \tag{18}$$

where $\mathcal{P} = \{P_1^T; \cdots; P_M^T\}^T$, $\mathbf{A}_m \in \mathbf{R}^{M\times N}$, and $\mathbf{W} = \{f_1, \cdots, f_m, \cdots, f_M\}^T$.

The augmented Lagrangian function of $\tilde{\mathbf{P2}}$ is

$$\begin{aligned}\mathcal{L}_\delta(\mathcal{P}, \boldsymbol{\lambda}) = & \sum_{m=1}^{M} E_m(P_m) - \boldsymbol{\lambda}^T \left(\sum_{m=1}^{M} \mathbf{A}_m P_m - \mathbf{W}\right) \\ & + \frac{\delta}{2}\left\|\sum_{m=1}^{M} \mathbf{A}_m P_m - \mathbf{W}\right\|_2^2.\end{aligned} \tag{19}$$

Similar to the augmented Lagrangian function in the classic ADMM with two blocks i.e., Eqn. (10) , $\delta > 0$ represents the penalty parameter, $\boldsymbol{\lambda}$ is a cloumn vector of Lagrange multipliers. Then, we can update primal variables at $(k+1)-th$ iteration as

$$\begin{aligned}P_1^{k+1} = & \arg\min \; E_1(P_1) \\ & + \frac{\delta}{2}\left\|\mathbf{A}_1 P_1 + \sum_{m=2}^{M} \mathbf{A}_m P_m^k - \mathbf{W} - \frac{\boldsymbol{\lambda}^k}{\delta}\right\|_2^2, \\ & \vdots \\ P_M^{k+1} = & \arg\min \; E_M(P_M) \\ & + \frac{\delta}{2}\left\|\sum_{m=1}^{M-1} \mathbf{A}_m P_m^k + \mathbf{A}_M P_M - \mathbf{W} - \frac{\boldsymbol{\lambda}^k}{\delta}\right\|_2^2.\end{aligned}$$

Furthermore, the update rule for dual variables is

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k - \delta\left(\sum_{m=1}^{M} \mathbf{A}_m P_m^{k+1} - \mathbf{W}\right). \tag{20}$$

The termination criteria is

$$\Delta^{k+1} = |\mathcal{L}_\delta(\mathcal{P}^{k+1}, \boldsymbol{\lambda}^{k+1}) - \mathcal{L}_\delta(\mathcal{P}^k, \boldsymbol{\lambda}^k)| \le \varepsilon. \tag{21}$$

C. DT-driven Game-Based Task Offloading Problem

Based on Section III-B, we propose a mechanism based on Stackelberg game and Jacobian ADMM to allocate computing resources, so that DT and RSUs can reach a consensus on the allocation scheme, and solve the whole problem in a distributed and parallel way.

In the Stackelberg game, the DT as leader provides incentives $\boldsymbol{\theta_m} = \{(\theta_{m,1}), \cdots, (\theta_{m,N})\}$ to RSU m, which indicates the additional compensation of DT to RSU m. According to [13], the incentive function of RSU m can be defined as

$$\Phi_m\left(h_m(P_m, \eta_m), \boldsymbol{\theta_m}\right) = \mathcal{L}_m(P_m, \boldsymbol{\lambda}_m) + \mathcal{H}_m(P_m, \eta_m, \boldsymbol{\theta_m}), \tag{22}$$

where

$$\mathcal{L}_m(P_m, \boldsymbol{\lambda}_m) = E_m(P_m) - \boldsymbol{\lambda}_m \mathbf{A}_m P_m. \tag{23}$$

$$\mathcal{H}_m(P_m, \eta_m, \boldsymbol{\theta_m}) = h(P_m, \eta_m) - \boldsymbol{\theta_m} P_m. \tag{24}$$

We can see from the incentive function, first of all, $\mathcal{L}_m(P_m, \boldsymbol{\lambda}_m)$ is the information from DT , which allows RSU m to know the optimization direction of DT and thus schedule its own resources. For $\mathcal{H}_m(P_m, \eta_m, \boldsymbol{\theta_m})$, which iconsists of its own original goal and compensation from DT. Furthermore, we have shown that both $E_m(P_m)$ and $h(P_m, \eta_m)$ are strongly convex functions of p_m, so $\Phi_m\left(h_m(P_m, \eta_m), \boldsymbol{\theta_m}\right)$ is also a strongly convex function of P_m. Therefore, we can formulate the Stackelberg game as

$$\begin{aligned}\mathbf{P3}: \quad Leader: \quad & \underset{\mathcal{P}}{minimize} \quad E(\mathcal{P}) \\ Follower: \quad & \underset{P_m, Q_m}{minimize} \quad \Phi_m\left(h(P_m, \eta_m), \boldsymbol{\theta}_m\right) \\ & \text{s.t.} \quad \sum_{n\in\mathcal{N}} p_{m,n} = f_m, \forall m \in \mathcal{M} \quad (C1)\end{aligned} \tag{25}$$

To solve this game, the Jacobian-ADMM-based two-layer iteration algorithm is proposed as follows. At the beginning, the leader sends $\mathcal{L}_m(P_m, \boldsymbol{\lambda}_m)$ to the corresponding RSU m. We define the number of iterations of the outer loop as k. At iteration k, given incentive parameters $\{\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_m, \cdots, \boldsymbol{\theta}_M\}$ from leader, each RSU updates their own incentive function, and then the leader and the followers would reach a current optimal scheme. At next iteration $k+1$, leader will adjust the incentive parameters based on the updated $P_m, \forall m \in \mathcal{M}$. Then, a new current optimal scheme can be reached. Note that, Each RSU updates its own computing resource allocation scheme $P_m, \forall m \in \mathcal{M}$ at each iteration k is called inner loop, where t is the iteration times for the inner loop. The specific calculation process is as follows.

1) **Inner loop**
Follower's update:

$$\begin{aligned} P_m^{t+1}\{k\} =& \arg\min \ \mathcal{L}_m\left(P_m, \boldsymbol{\lambda}_m\{k\}\right) \\ &+ \mathcal{H}_m\left(P_m, \eta_m^t\{k\}, \boldsymbol{\theta_m}\{k\}\right) \\ &+ \frac{\delta}{2}\left\|\mathbf{A}_m P_m + \sum_{i\neq m}^{M} \mathbf{A}_i P_i^t\{k\} - \mathbf{W}\right\|_2^2 \end{aligned} \tag{26}$$

Leader's dual update:

$$\boldsymbol{\lambda}^{t+1}\{k\} = \boldsymbol{\lambda}^t\{k\} - \delta\left(\sum_{m=1}^{M} \mathbf{A}_m P_m^{t+1}\{k\} - \mathbf{W}\right) \tag{27}$$

where $\boldsymbol{\lambda} = \{\boldsymbol{\lambda}_1, \cdots, \boldsymbol{\lambda}_m, \cdots, \boldsymbol{\lambda}_M\}^T$.
η_m update based on Eqn. (4), i.e.,

$$\eta_m^{t+1}\{k\} = \frac{\sum\limits_{n\in\mathcal{N}} \log_2(1+q_{n,m}p_{m,n}^{t+1}\{k\})}{\sum\limits_{n\in\mathcal{N}} p_{m,n}^{t+1}\{k\}}. \tag{28}$$

2) **Outer loop**
Leader's incentive parameter update: At the end of the inner loop, each RSU feeds back its marginal cost $\nabla_{P_m^t\{k\}} h\left(P_m^t\{k\}, \eta_m\{k+1\}\right)$ to the leader. Then the leader adjusts its strategy of incentive parameters based on these information. Therefore , we have

$$\boldsymbol{\theta}_m\{k+1\} = \nabla_{P_m^t\{k\}} h\left(P_m^t\{k\}, \eta_m\{k+1\}\right). \tag{29}$$

Note that the termination criteria for the inner loop can be defined as

$$\begin{aligned} \Delta_\Phi\{k\} =& \left|\Phi_m\left(h_m(P_m^{t+1}\{k\}, \eta_m^{t+1}\{k\}), \boldsymbol{\theta}_m\{k+1\}\right)\right. \\ &- \Phi_m\left(h_m(P_m^t\{k\}, \eta_m^t\{k\}), \boldsymbol{\theta}_m\{k\}\right)\left.\right| < \varepsilon. \end{aligned} \tag{30}$$

And according (19) and (21), the termination criteria for the outer loop is

$$\Delta_{\mathcal{L}}\{k\} = \left\|\mathcal{L}\left(P_m^t\{k+1\}, \boldsymbol{\lambda}_m^{k+1}\right) - \mathcal{L}\left(P_m^t\{k\}, \boldsymbol{\lambda}_m^k\right)\right\| \le \varepsilon, \tag{31}$$

where

$$\begin{aligned} \mathcal{L}\left(P_m^t\{k\}, \boldsymbol{\lambda}_m^k\right) =& \sum_{m=1}^{M} \mathcal{L}_m\left(P_m^t\{k\}, \boldsymbol{\lambda}_m^k\right) + (\boldsymbol{\lambda}\mathbf{E})^T\mathbf{W} \\ &+ \frac{\delta}{2}\left\|\sum_{m=1}^{M} \mathbf{A}_m P_m^t\{k\} - \mathbf{W}\right\|_2^2. \end{aligned} \tag{32}$$

The proposed DT-driven game-ADMM jointly considers the overall energy efficiency and the satisfaction of RSUs. On the premise of ensuring the satisfaction of RSUs, the global energy consumption can be minimized, and the algorithm can converge in a fast manner. The proof can be seen in [14].

IV. SIMULATION RESULTS

The scenario of DT-driven air-assisted IoV is set in a simulator based on the CRAWDAD dataset. In particular, the idle computing resources of RSUs are uniformly distributed and the values are between [30,60] GHz. The unit cost of RSUs also follows the uniform distribution in [1,2]. Offloading tasks obeys the Poisson distribution, the size of the computational offloading request packet follows the uniform distribution. Vehicles have their preference for RSUs, which is randomly distributed on [0, 10]. The DT is established on the UAV, which reflects the dynamic resource demand and supply and input to the resource allocation scheme.

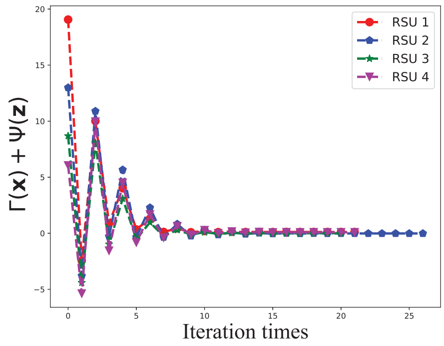

Fig. 2. The classic-ADMM iterative convergence.

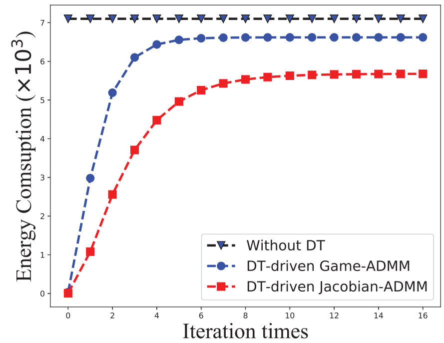

Fig. 3. Energy consumption of RSUs varies with the number of iterations under the three scenarios.

Fig. 2 shows the convergence of the DT-driven ADMM scheme of different RSUs over the iteration times. The vertical axis is the objective function of the optimization problem $\tilde{\mathbf{P1}}$. As can be seen, after 12 iteration times, all the RSUs can converge to a constant that is very close to zero, which is consistent with *Theorem* 1. This means that the objective functions of different RSUs can eventually converge to achieve the goal of maximizing satisfaction. This shows the feasibility of the proposed DT-driven ADMM scheme.

Fig. 3 compares the energy consumption of three schemes, i.e., the DT-driven Jacobian-ADMM, the DT-driven game-ADMM, and the scheme without DT, over the iteration times. The scheme without DT has the highest energy consumption and remains unchanged, because the CPU frequency and tasks can only be randomly assigned. Although the energy consumption of the DT-driven Jacobian ADMM is the lowest, this comes at the cost of lower vehicle satisfaction, because the minimum total energy consumption is its only goal. The proposed DT-driven game-ADMM jointly considers the over-

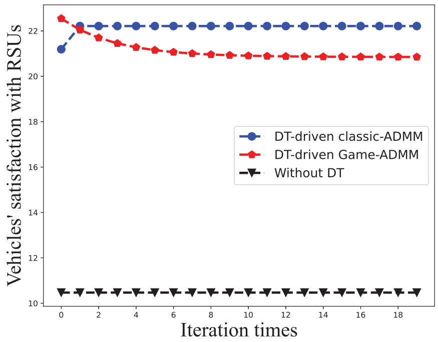

Fig. 4. The comparison of satisfaction varies with the number of iterations under the three schemes.

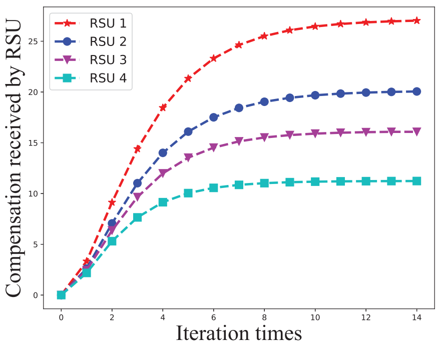

Fig. 5. The comparison of incentives received by RSU varies with the number of iterations under the three schemes.

all energy efficiency and the satisfaction of RSUs. Therefore, its energy consumption is among other schemes.

Fig. 4 compares the vehicles' satisfaction with RSUs of three schemes, i.e., the DT-driven classic-ADMM, the DT-driven game-ADMM, and the scheme without DT. The solution without DT is the lowest. This is because without the assistance of DT, the vehicle's preference for RSU is unknown, and the allocation cannot fully meet the actual needs of the vehicle. Thus, the satisfaction of DT-driven game-ADMM is a bit lower than that of classic-ADMM. This is because in order to better serve vehicles, RSUs would allocate more resource to vehicles that have higher preference.

Fig. 5 shows the variation of the incentives received by different RUSs with the number of iterations. After 12 iterations, the excitation value of RSUs converged. RSU 1 receives the highest compensation among the RSUs. Because RSU1 allocates a portion of CPU frequency to vehicles. RUS1 has the largest increase in energy consumption, so it gets the most compensation. This reflects the effectiveness of the proposed scheme in balancing energy consumption and satisfaction.

V. CONCLUSION

In this paper, we proposed dynamic digital twin and distributed incentives for resource allocation in air-assisted IoV. This incentive scheme achieves the best CPU frequency allocation for each RSU considering the high dynamics of IoV. It also can minimize system energy consumption and maximize service quality. Distributed ADMM is used to solve the optimal solution of incentive problem to adapt to the lightweight processing capability of UAV. The simulation shows the feasibility and effectiveness of our proposed scheme.

ACKNOWLEDGMENT

This work was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 824019, in part by Special Funds for Central Universities Construction of World-Class Universities (Disciplines), and in part by China 111 Project (B16037).

REFERENCES

[1] N. Cheng, W. Quan, W. Shi, H. Wu, Q. Ye, H. Zhou, W. Zhuang, X. S. Shen, and B. Bai, "A Comprehensive Simulation Platform for Space-Air-Ground Integrated Network," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 178–185, 2020.

[2] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Low-latency federated learning and blockchain for edge association in digital twin empowered 6g networks," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[3] Y. Lu, X. Huang, K. Zhang, S. Maharjan, and Y. Zhang, "Communication-efficient federated learning and permissioned blockchain for digital twin edge networks," *IEEE Internet of Things Journal*, pp. 1–1, 2020.

[4] Y. Dai, K. Zhang, S. Maharjan, and Y. Zhang, "Deep reinforcement learning for stochastic computation offloading in digital twin networks," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[5] C. Gehrmann and M. Gunnarsson, "A Digital Twin Based Industrial Automation and Control System Security Architecture," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 669–680, 2020.

[6] W. Sun, S. Lei, L. Wang, Z. Liu, and Y. Zhang, "Adaptive federated learning and digital twin for industrial internet of things," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020.

[7] M. Liwang, S. Dai, Z. Gao, X. Du, M. Guizani, and H. Dai, "A Computation Offloading Incentive Mechanism with Delay and Cost Constraints under 5G Satellite-Ground IoV Architecture," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 124–132, 2019.

[8] Z. Zhou, P. Liu, J. Feng, Y. Zhang, S. Mumtaz, and J. Rodriguez, "Computation Resource Allocation and Task Assignment Optimization in Vehicular Fog Computing: A Contract-Matching Approach," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3113–3125, 2019.

[9] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing Offloading Latency for Digital Twin Edge Networks in 6G," *IEEE Transactions on Vehicular Technology*, 2020.

[10] H. Chai, S. Leng, Y. Chen, and K. Zhang, "A Hierarchical Blockchain-Enabled Federated Learning Algorithm for Knowledge Sharing in Internet of Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2020.

[11] Z. Zhou, M. Dong, K. Ota, J. Wu, and T. Sato, "Energy efficiency and spectral efficiency tradeoff in device-to-device (D2D) communications," *IEEE Wireless Communications Letters*, vol. 3, no. 5, pp. 485–488, 2014.

[12] Y. He, S. Zhang, L. Tang, and Y. Ren, "Large Scale Resource Allocation for the Internet of Things Network Based on ADMM," *IEEE Access*, vol. 8, pp. 57 192–57 203, 2020.

[13] Z. Zheng, L. Song, and Z. Han, "Bridge the gap between ADMM and Stackelberg game: Incentive mechanism design for big data networks," *IEEE Signal Processing Letters*, vol. 24, no. 2, pp. 191–195, 2017.

[14] Y. Yu, X. Bu, K. Yang, Z. Wu, and Z. Han, "Green large-scale fog computing resource allocation using joint benders decomposition, dinkelbach algorithm, admm, and branch-and-bound," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4106–4117, 2018.