Published:
IEEE Transactions on Intelligent Transportation Systems

Vehicle Selection and Resource Optimization for Federated Learning in Vehicular Edge Computing

Authors:
Huizi Xiao, Jun Zhao, Qingqi Pei, Jie Feng, Lei Liu, and Weisong Shi, Fellow, IEEE

Federated Learning in VEC

Federated learning in vehicular edge computing leverages the concept of collaborative learning to enhance data processing and model training within the vehicular network.

The Pros :

- Decentralized Machine Learning
- Privacy preserving approach
- Efficient knowledge sharing

The Cons

- Vehicles have limited computation capability
- FL involves the training and transmission of model parameters
- Which consumes the vehicles’ precious energy resources and takes up much time
- And the capabilities and data quality of each vehicle are distinct

The Problem

The capabilities and data quality of each vehicle are distinct that will affect the performance of training the model. Therefore it is crucial to,

- Select vehicles with high image quality to join the training is a problem
- Optimize resource allocation under learning time and energy consumption constraints

The Solution Approach

- Taking the vehicle **position** and **velocity** into consideration, we formulate a min-max problem to jointly optimize,
  - The on-board computation capability,
  - The transmission power of updating the local model
  - Local model accuracy
- Position and velocity is taken into consideration so that the vehicles can complete at least one round of FL and update it within the current edge server’s coverage area

Continued

- We propose a **greedy algorithm** for dynamically selecting vehicles so that the vehicles selected for the FL tasks have **higher image quality** data.
- The formulated optimization problem is a nonlinear programming problem, divided into two subproblems,
    - Resource Allocation Problem
    - Local Model Accuracy Problem

System Model

TABLE I: Key Notation Definitions in System Model

Symbol	Definitions	Symbol	Definitions
$\mathcal{N}$	the number of vehicles covered by an edge server	$\mathcal{V}$	a set of vehicles covered by an edge server
v_n	the speed of the vehicle V_n	v^{min} / v^{max}	the minimum / maximum speed on the road
μ / σ	the mean / standard deviation speed of vehicles	$\mathcal{L}$	the coverage diameter of the edge server on the road
l_n	the distance from the vehicle V_n position to the entrance of the coverage area	τ_n	the residence time of the vehicle V_n within the coverage of the edge server
L_n	the motion blur level of the vehicle V_n	$v_n^{'}$	the instantaneous relative speed between the vehicle V_n and observation object
T	the exposure time interval of the on-board cameras	f	the camera focal length
Q	the charge-coupled device pixel size in the horizontal direction	G	the starting pixel position of the object in the image
δ	the angle between the motion direction and the image plane	d	the perpendicular distance from the starting point of the moving object to the pinhole
D_n	training data samples of the vehicle V_n	$\boldsymbol{x}_i$ / $\boldsymbol{y}_i$	the input / output of the training data sample i
$a_n \in \{0,1\}$	whether the vehicle V_n is selected for model training	$\mathcal{M}$	the selected vehicles set for training task
$\boldsymbol{w}_n^k$	local model weight parameters of the vehicle V_n for the k^{th} edge iteration	$L_n(\boldsymbol{w}_n^k, \boldsymbol{x}_i, \boldsymbol{y}_i)$	the model loss function of the vehicle V_n
λ^k	the learning rate of the k^{th} edge iteration	θ / ε	the accuracy of the local / edge model
$\boldsymbol{w}^k$	edge model weight parameters for the k^{th} edge iteration	$\mathcal{I}(\theta)$	the number of edge model iterations
q_n	the average CPU cycles to process a data sample of the vehicle V_n	f_n	on-board CPU calculated frequency of the vehicle V_n
T_n^{com} / E_n^{com}	local computing time / energy consumption of the vehicle V_n	T_n^{tran} / E_n^{tran}	the time / energy consumption to transmit model parameters of the vehicle V_n
k_n	effective switched capacitance of the vehicle V_n	p_n	the transmit power of the vehicle V_n
r_n	transmitting rate of the vehicle V_n	α_n	the state of the V_n vehicular connection
K_0	the performance of the error-recovery system using Forward Error Correction technology	Z_n	the mobility function of V_n in the TCP/IP mobile connection protocol service
C_n	the data size of the local model parameters in V_n	$\mathcal{S}_n$	the total cost of the vehicle V_n
$\lambda^t, \lambda^e \in [0,1]$	the importance weighting indicators of the delay / energy consumption	T_n / E_n	the whole time / energy consumption of the vehicle V_n for one edge iteration

Speed modeling

- Assumptions,
  - The speeds of vehicles to be independent and identically distributed
  - The speed changes on a relatively short distance road covered by an edge server are tiny
- Let v_n be the constant velocity of some vehicle V_n where $v^{max} < v_n < v^{min}$
- The distribution of v_n can be denoted by the following probability distribution function,

$$f(v_n) = \begin{cases} \dfrac{2e^{\frac{-(v_n-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}\left(erf\left(\frac{v^{max}-\mu}{\sigma\sqrt{2}}\right)-erf\left(\frac{v^{min}-\mu}{\sigma\sqrt{2}}\right)\right)}, & v^{min} \leq v_n \leq v^{max}, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Residence time

Let,

L = Coverage diameter of the edge server on the road

$\mathbf{l_n}$ = Distance from the vehicle $\mathbf{V_n}$ to the entrance of the coverage area

The residence time of each vehicle within the coverage of the current edge server can be obtained,

$$\tau_n = \frac{\mathcal{L} - l_n}{v_n}. \quad (2)$$

Image quality

Due to vehicles' mobility, the images acquired by the vehicle will generally show noise, motion blur, and distortion. Let,

- motion blur $\mathbf{L_n}$ is related to the instantaneous relative speed $\mathbf{v`_n}$.
- Image plane is parallel to the moving direction

The motion blur of the image can be denoted using the following equation,

$$L_n = \frac{Tf}{dQ} v_n'. \tag{4}$$

Local model training

- Before an iteration of the edge model training begins, the vehicles selected to participate in the training download the area's current edge model from the edge
- Set $\mathbf{a_n \in \{0, 1\}}$ indicate whether the vehicle is selected for model training
- The main objective is to minimize the loss function to obtain model parameters that best fit the local data.

$$\begin{aligned} \hat{\boldsymbol{w}}_n^k &= \arg\min_{\boldsymbol{w}} \; L_n(\boldsymbol{w}_n^k, \boldsymbol{x}_i, \boldsymbol{y}_i) \\ &= \arg\min_{\boldsymbol{w}} \frac{1}{|D_n|} \sum_{i=1}^{|D_n|} l_n(\boldsymbol{w}_n^k, \boldsymbol{x}_i, \boldsymbol{y}_i). \end{aligned} \tag{5}$$

Local Parameter Update

- where $\mathbf{l_n(\cdot)}$ is the each vehicle’s loss function that evaluates how different the predicted value $\mathbf{\tilde{y}_i = f(w, xi)}$ of the model is from the true one **y**

local parameter update in the k^{th} edge iteration according to the following algorithm to approximate $\hat{\boldsymbol{w}}_n^k$:

$$\boldsymbol{w}_n^k \leftarrow \boldsymbol{w}_n^k - \lambda^k \nabla L_n(\boldsymbol{w}_n^k, \boldsymbol{x}_i, \boldsymbol{y}_i). \quad (6)$$

Edge Aggregation

The edge server generates new edge model **w^k+1** by computing the weighted average of received local models **w^k** n from all connected vehicles set **M**

$$\boldsymbol{w}^{k+1} \leftarrow \frac{1}{|\mathcal{M}|} \sum_{n=1}^{|\mathcal{M}|} \boldsymbol{w}_n^k. \tag{7}$$

The gradient is aggregated as

$$\nabla L(\boldsymbol{w}^{k+1}, \boldsymbol{x}_i, \boldsymbol{y}_i) \leftarrow \frac{1}{|\mathcal{M}|} \sum_{n=1}^{|\mathcal{M}|} \nabla L_n(\boldsymbol{w}_n^k, \boldsymbol{x}_i, \boldsymbol{y}_i). \tag{8}$$

Local Computation Phase

The whole time for the local computation can be described using the following equation,

$$T_n^{com} = \log(\frac{1}{\theta})\frac{|D_n|q_n}{f_n}. \qquad (9)$$

Here, **log(1/ϴ)** denotes the upper bound for local computation iteration

The energy consumption of V_n for local computation is

$$E_n^{com} = p_n^{com}T_n^{com} = \log(\frac{1}{\theta})k_n|D_n|q_n f_n^2. \qquad (10)$$

$\mathbf{p_n^{com}}$ denotes the computation power of the on board CPU.

Transmission Phase

Let $\mathbf{p_n}$ be the transmit power of $\mathbf{V_n}$, so the transmitting rate $\mathbf{r_n}$ that the wireless TCP/IP can provide in the steady-state, [with round trip time **RRT**]

$$r_n = \alpha_n \sqrt{p_n}. \tag{11}$$

α_n as the state of the vehicular connection can be defined:

$$\alpha_n \triangleq \frac{K_0 \sqrt{Z_n}}{RRT}. \tag{12}$$

K_0	the performance of the error-recovery system using Forward Error Correction technology

Z_n is the mobility function of V_n in the TCP/IP

Continued

Transmission time and energy consumption can be described as, [**C_n** = Size of the local parameter]

$p_n = (\frac{r_n}{\alpha_n})^2$ can be obtained by equation (11).

$$T_n^{tran} = \frac{C_n}{r_n} = \frac{C_n}{\alpha_n \sqrt{p_n}}. \quad (13)$$

$$E_n^{tran} = p_n T_n^{tran} = \frac{C_n \sqrt{p_n}}{\alpha_n}. \quad (14)$$

We can conclude that the whole time and the energy consumption for the vehicle **V_n** to execute one edge iteration are represented by respectively,

$$T_n = T_n^{com} + T_n^{tran}. \quad (15)$$

$$E_n = E_n^{com} + E_n^{tran}. \quad (16)$$

Problem Formulation

We now formulate an optimization problem to minimize the vehicle's maximum energy consumption and delay costs in FL.

Our optimization problem is organized as follows:

$$\mathcal{S}_n = \mathcal{I}(\theta)(\lambda^t T_n + \lambda^e E_n)$$

$$\begin{aligned}
\min_{a_n, \theta, \boldsymbol{f}, \boldsymbol{p}} \ \max_{n} \ & \mathcal{S}_n \\
\text{s.t.} \ & (\mathrm{C}_1): a_n \in \{0,1\}, \ \forall n \in \mathcal{N}, \\
& (\mathrm{C}_2): a_n \left(T_n^{com} + T_n^{tran}\right) \leq \tau_n, \ \forall n \in \mathcal{N}, \\
& (\mathrm{C}_3): E_n^{tran} \leq E_{max}^{tran}, \ \forall n \in \mathcal{N}, \\
& (\mathrm{C}_4): a_n L_n \leq L_{max}, \ \forall n \in \mathcal{N}, \\
& (\mathrm{C}_5): 0 \leq f_n \leq f_n^{max}, \ \forall n \in \mathcal{N}, \\
& (\mathrm{C}_6): 0 \leq \theta \leq 1.
\end{aligned} \tag{17}$$

DESIGN OF RESOURCE ALLOCATION ALGORITHM

In this section, we address the challenge of resource allocation in vehicular edge computing. We focus on a set M of selected vehicles, each characterized by local model accuracy θ and participation status an = 1.

Objectives:

- **Problem Formulation:** Given the selected vehicles and their accuracy, we formulate a min-max optimization problem.
- **Algorithm Design:** We develop a novel algorithm to solve the min-max problem, ensuring optimal resource allocation.
- **Optimal Solutions:** By solving the problem, we determine the optimal values for on-board CPU frequency (f) and transmission power (p).

Problem Formulation

Our objective is to get the optimal on-board CPU frequency **f** and transmission power **p**.The problem in (17) can be converted to the following min-max optimization problem:

$$
\begin{aligned}
\min_{\boldsymbol{f},\boldsymbol{p}} \ \max_{n} \ & \mathcal{I}(\theta)(\lambda^t T_n + \lambda^e E_n) \\
\text{s.t.} \ & (\mathrm{C}_2^{'}) : \log(\frac{1}{\theta})\frac{|D_n| q_n}{f_n} + \frac{C_n}{\alpha_n \sqrt{p_n}} \leq \tau_n, \ \forall V_n \in \mathcal{M}, \\
& (\mathrm{C}_3^{'}) : p_n \leq \left(\frac{E_{max}^{tran} \alpha_n}{C_n} \right)^2, \ \forall V_n \in \mathcal{M}, \\
& (\mathrm{C}_5) : 0 \leq f_n \leq f_n^{max}, \ \forall V_n \in \mathcal{M}.
\end{aligned} \tag{18}
$$

Problem formulation (Cont.)

To solve the problem in (18) effectively we convert it into the epigraph form by introducing a new variable **ζ**

$$
\begin{aligned}
&\min_{\zeta, \boldsymbol{f}, \boldsymbol{p}} \ \zeta \\
&\text{s.t. } (\mathbf{C}_2^{'}), (\mathbf{C}_3^{'}), (\mathbf{C}_5), \\
&\quad (\mathbf{C}_7) : \mathcal{I}(\theta) \Bigg[\lambda^t \left(\log(\frac{1}{\theta}) \frac{|D_n| q_n}{f_n} + \frac{C_n}{\alpha_n \sqrt{p_n}} \right) \\
&\quad + \lambda^e \left(\log(\frac{1}{\theta}) k_n |D_n| q_n f_n^2 + \frac{C_n \sqrt{p_n}}{\alpha_n} \right) \Bigg] \leq \zeta, \forall V_n \in \mathcal{M}.
\end{aligned}
\tag{19}
$$

epigraph-form: In mathematical optimization, the "epigraph form" is a way to rephrase an optimization problem in a specific format that involves introducing an auxiliary variable to simplify the problem, allowing for more straightforward analysis and solution.

Resource Allocation Solution(Lagrangian dual problem)

To solve the optimization problem in (19) we use the Lagrangian dual problem and the subgradient projection method to approximate the optimal value iteratively

Let λ = {λ1, λ2, · · · , λN } ≥ 0, β = {β1, β2, · · · , βN } ≥ 0, μ = {μ1, μ2, · · · , μN } ≥ 0, and ϕ = {ϕ1, ϕ2, · · · , ϕN } ≥ 0 be dual variables corresponding to the constraints (C2′), (C3′), (C5) and (C7) in (19), respectively, so the Lagrangian dual function of problem (19) can be expressed as

$$G(\boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\varphi}) = \min_{\zeta, 0 \preceq \boldsymbol{f}, \boldsymbol{p}} \mathscr{L}(\zeta, \boldsymbol{f}, \boldsymbol{p}, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\varphi}). \qquad (20)$$

where,

$$\begin{aligned}\mathscr{L}(\zeta, \boldsymbol{f}, \boldsymbol{p}, \boldsymbol{\lambda}, \boldsymbol{\beta}, \boldsymbol{\mu}, \boldsymbol{\varphi}) = \zeta + \sum_{V_n \in \mathcal{M}} \lambda_n \left[\log(\frac{1}{\theta}) \frac{|D_n| q_n}{f_n} + \frac{C_n}{\alpha_n \sqrt{p_n}} - \tau_n \right] + \sum_{V_n \in \mathcal{M}} \beta_n \left[p_n - \left(\frac{E_{max}^{tran} \alpha_n}{C_n} \right)^2 \right] \\ + \sum_{V_n \in \mathcal{M}} \mu_n (f_n - f_n^{max}) + \sum_{V_n \in \mathcal{M}} \varphi_n \left[\mathcal{I}(\theta) \left(\lambda^t \log(\frac{1}{\theta}) \frac{|D_n| q_n}{f_n} + \lambda^t \frac{C_n}{\alpha_n \sqrt{p_n}} + \lambda^e \log(\frac{1}{\theta}) k_n |D_n| q_n f_n^2 + \lambda^e \frac{C_n \sqrt{p_n}}{\alpha_n} \right) - \zeta \right] \end{aligned} \qquad (21)$$

Resource Allocation Solution(Cont.)

We take the partial derivatives of L (ζ, f, p,λ, β, μ, ϕ) in (21) with respect to variables fn and pn and make them equal to 0 in order to minimize (20)

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial f_n} &= \mu_n + 2\varphi_n \mathcal{I}(\theta)\lambda^e \log(\frac{1}{\theta}) k_n |D_n| q_n f_n \\ &- \log(\frac{1}{\theta}) \frac{|D_n| q_n}{f_n^2} \left(\lambda_n + \lambda^t \varphi_n \mathcal{I}(\theta)\right) = 0.\end{aligned} \tag{22}$$

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial p_n} =& \beta_n + \frac{1}{2}\varphi_n \mathcal{I}(\theta)\lambda^e \frac{C_n}{\alpha_n \sqrt{p_n}} \\ &- \frac{1}{2} \frac{C_n}{\alpha_n \sqrt{p_n^3}} \left(\lambda_n + \lambda^t \varphi_n \mathcal{I}(\theta)\right) = 0.\end{aligned} \tag{23}$$

Resource Allocation Solution(Cont.)

Solving (22) and (23) we obtain the solution for fn and pn

$$f_n^* = \begin{cases} \frac{-\mu_n - (\sqrt[3]{Y_1} + \sqrt[3]{Y_2})}{6\mathcal{I}(\theta)\lambda^e \varphi_n k_n \log(\frac{1}{\theta})|D_n| q_n}, & \mu_n \in (0, -(\sqrt[3]{Y_1} + \sqrt[3]{Y_2})] \\ & \cap(0, \Delta), \\ \frac{B}{4A}, & \mu_n = \Delta, \\ \frac{\mu_n(\cos\frac{\arccos G}{3} + \sqrt{3}\sin\frac{\arccos G}{3} - 1)}{6\mathcal{I}(\theta)\lambda^e \varphi_n k_n \log(\frac{1}{\theta})|D_n| q_n}, & \mu_n > \Delta, \\ \text{no solution}, & \text{otherwise}. \end{cases} \quad (24)$$

Let,

$$x_n = \sqrt{p_n}$$

$$x_n^* = \begin{cases} \frac{-\sqrt{\bar{A}} - (\sqrt[3]{\bar{Y_1}} + \sqrt[3]{\bar{Y_2}})}{3\beta_n}, & \Delta_1 \in (0, -\frac{2\alpha_n}{C_n}(\sqrt[3]{\bar{Y_1}} + \sqrt[3]{\bar{Y_2}})] \\ & \cap(0, \Delta_2), \\ \frac{\Delta_1(\frac{4}{3}\alpha_n^4 - C_n^4)}{2\beta_n C_n^3 \alpha_n}, & \Delta_1 = \Delta_2, \\ \frac{\sqrt{\bar{A}}(\cos\frac{\arccos \bar{G}}{3} + \sqrt{3}\sin\frac{\arccos \bar{G}}{3} - 1)}{3\beta_n}, & \Delta_1 > \Delta_2, \\ \text{no solution}, & \text{otherwise}. \end{cases} \quad (25)$$

Lagrangian Multiplier Update

If the Lagrange dual variables λ, β, μ and ϕ are known in advance, we can obtain the optimal solutions of fn* and pn* from (24) and (25).

Due to the convexity of the problem, the subgradient projection method can be applied to update the values to approximate the optimal one.

Subgradients of G(λ, β, μ, ϕ)

$$
\begin{aligned}
\nabla\lambda_n &= \log(\frac{1}{\theta})\frac{|D_n|q_n}{f_n^*} + \frac{C_n}{\alpha_n\sqrt{p_n^*}} - \tau_n,\ \forall V_n \in \mathcal{M} \\
\nabla\beta_n &= p_n^* - \left(\frac{E_{max}^{tran}\alpha_n}{C_n}\right)^2,\ \forall V_n \in \mathcal{M} \\
\nabla\mu_n &= f_n^* - f_n^{max},\ \forall V_n \in \mathcal{M} \\
\nabla\varphi_n &= \mathcal{I}(\theta)\Bigg(\lambda^t \log(\frac{1}{\theta})\frac{|D_n|q_n}{f_n^*} + \lambda^t \frac{C_n}{\alpha_n\sqrt{p_n^*}} \\
&\quad + \lambda^e \log(\frac{1}{\theta})k_n|D_n|q_n(f_n^*)^2 + \lambda^e \frac{C_n\sqrt{p_n^*}}{\alpha_n}\Bigg) - \zeta, \forall V_n \in \mathcal{M}
\end{aligned}
\tag{28}
$$

Lagrangian Multiplier Update(Cont.)

From (28), the subgradients of G(λ, β, μ, ϕ) can be obtained. The subgradient projection method for acquiring Lagrange multipliers to solve (27) is iterated in the following way:

$$
\begin{aligned}
\lambda_n(t+1) &= [\lambda_n(t) - i(t) \bigtriangledown \lambda_n(t)]^+, \\
\beta_n(t+1) &= [\beta_n(t) - j(t) \bigtriangledown \beta_n(t)]^+, \\
\mu_n(t+1) &= [\mu_n(t) - k(t) \bigtriangledown \mu_n(t)]^+, \\
\varphi_n(t+1) &= [\varphi_n(t) - o(t) \bigtriangledown \varphi_n(t)]^+.
\end{aligned}
\qquad (29)
$$

Resource Allocation Solution(Cont.)

The optimal solutions can be brought to (28) and (29) to obtain the new dual variables. This is a cyclic iterative approximation process, which is explained in Algorithm 1.

Algorithm 1 On-Board CPU Frequency and Transmission Power Optimization Algorithm

---

Initialization:

- Set the initial value of dual variables to $\boldsymbol{\lambda}(0)$, $\boldsymbol{\beta}(0)$, $\boldsymbol{\mu}(0)$, $\boldsymbol{\varphi}(0)$, maximum number of iterations t_{max} and the specified precision ϵ.
- Let $t = 0$.

Iteration:

```
while t ≤ t_max do
    Substitute the dual variables λ(t), β(t), μ(t) and φ(t)
    into (24) and (25) to obtain f_n(t) and p_n(t) respectively.
    Update new dual variables λ(t+1), β(t+1), μ(t+1)
    and φ(t+1) using (29), according to the new f_n(t)
    and p_n(t).
    if ||λ(t+1) − λ(t)|| < ε, ||β(t+1) − β(t)|| < ε, ||μ(t+1) − μ(t)|| < ε and ||φ(t+1) − φ(t)|| < ε
     then
       f_n^* = f_n(t) and p_n^* = p_n(t).
       break.
    else
       t = t + 1.
    end if
end while
```

Output: $\boldsymbol{f}^* = (f_1^*, f_2^*, \cdots, f_N^*)$ and $\boldsymbol{p}^* = (p_1^*, p_2^*, \cdots, p_N^*)$.

LOCAL MODEL ACCURACY

In this section, the on-board CPU frequency f* and transmission power p* obtained from the above section are regarded as fixed values, and we design the heuristic search algorithms to solve the min-max problem.

$$
\begin{aligned}
&\min_{\theta}\ \max_{n}\ \frac{\log(\frac{1}{\theta})H_n + G_n}{1-\theta} \\
&\text{s.t. } (\bar{\mathrm{C}}_2) : \log(\frac{1}{\theta})\frac{|D_n|q_n}{f_n} \le \tau_n^{\theta},\ \forall V_n \in \mathcal{M}, \\
&\qquad (\mathrm{C}_6) : 0 \le \theta \le 1.
\end{aligned} \tag{30}
$$

We obtain the optimal local accuracy value for each vehicle by solving the optimization problem in (30) using self-adaptive global best harmony search (SGHS) algorithm.

Joint Algorithm for Resource Allocation and Local Model Accuracy

Algorithm 3 A Joint Algorithm for Resource Allocation and Local Model Accuracy

Initialization:

- Initialize local model accuracy $\theta(0) \leftarrow [0, 1]$.
- Set the maximum number of iterations l_{max} and the specified precision ε.
- Let $l = 1$.

```
Allocate on-board CPU frequency f(0) and the transmission power p(0) by calling Algorithm 1 based on θ(0).
Substitute θ(0), f(0) and p(0) into (26) to obtain ζ(0) among |M| vehicles.
```

Iteration:

```
while l ≤ l_max do
    Search θ(l) by calling Algorithm 2 based on f(l − 1) and p(l − 1).
    Compute f(l) and p(l) by calling Algorithm 1 based on θ(l).
    Substitute θ(l), f(l) and p(l) into (26) to obtain ζ(l).
    if |ζ(l) − ζ(l − 1)| ≤ ε then
        break.
    end if
    l = l + 1.
end while
```

Output: the optimal resource allocation $\hat{\boldsymbol{f}}^*$, $\hat{\boldsymbol{p}}^*$ and local model accuracy $\hat{\theta}^*$ in vehicle set $\mathcal{M}$.

Greedy vehicle selection algorithm

Algorithm 4 Greedy vehicles selection algorithm

Initialization:

- Initialize $\mathcal{M} \leftarrow \emptyset$.

Iteration:

while $\mathcal{V} \neq \emptyset$ **do**

Solve $\Psi_n = \min\limits_{\theta, \boldsymbol{f}, \boldsymbol{p}} \max\limits_{m} \mathcal{S}_m$ with $V_m \in \{\mathcal{M} \cup V_n\}$ by calling Algorithm 3 for each vehicle V_n which meet the condition $L_n \leq L_{max}$ in the vehicle set $\mathcal{V}$.

Set $\Psi_{\hat{n}}^{min} = \min\{\Psi_n\}$.

Update $\mathcal{M} \leftarrow \mathcal{M} \cup V_{\hat{n}}$ and $\mathcal{V} \leftarrow \mathcal{V} \setminus V_{\hat{n}}$.

end while

Output: the set $\mathcal{M}^*$ of the selected vehicles.

Algorithm 4 is a dynamic addition process. When a new vehicle enters the current edge server's coverage area, it will turn to the vehicle set V automatically that can be selected

SIMULATION RESULTS AND ANALYSIS

In the case of optimizing resource allocation and local model accuracy, the experiments are conducted to verify the effectiveness of Algorithm 4 to select vehicle set M.

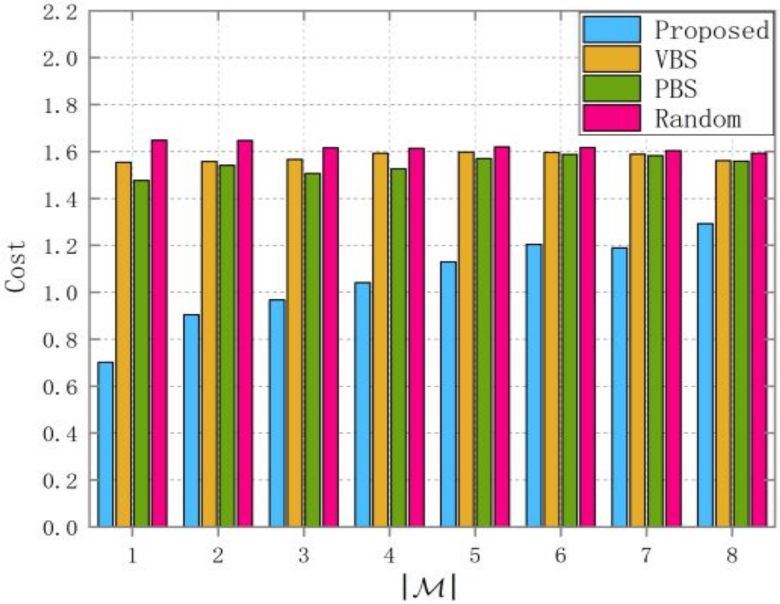
We set up the following schemes to compare with our proposed algorithm:

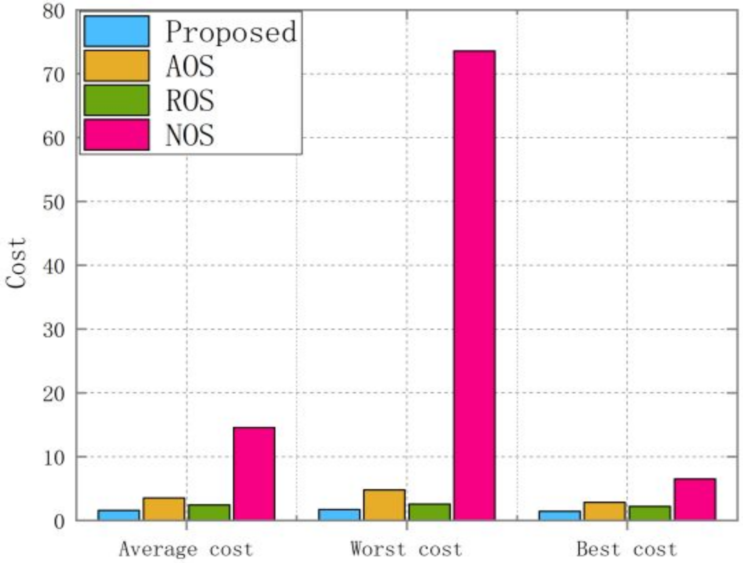
- Velocity-based selection scheme (VBS)
- The position-based selection scheme (PBS)
- Random selection scheme.

Besides these we take a few more schemes where we optimize a specific variable and randomly select the remaining variables:

- The accuracy-optimized scheme (AOS)
- The resource-optimized scheme (ROS)
- The non-optimized scheme (NOS)

Optimizing resource allocation and local model accuracy





Convergence of the Algorithms

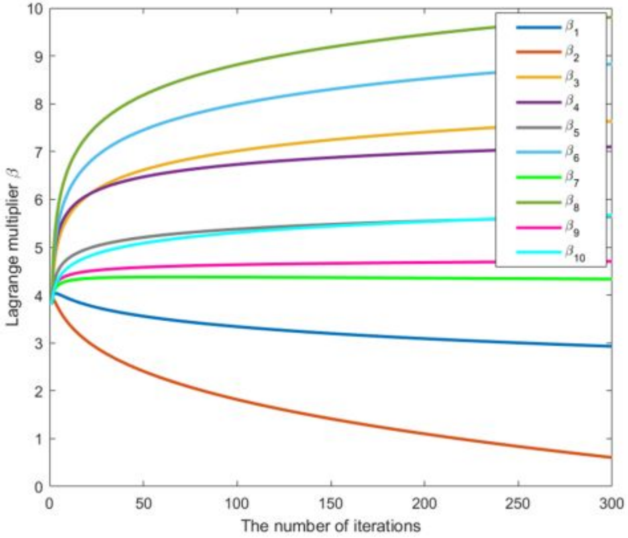


Fig. 2: Convergence of Algorithm 1.

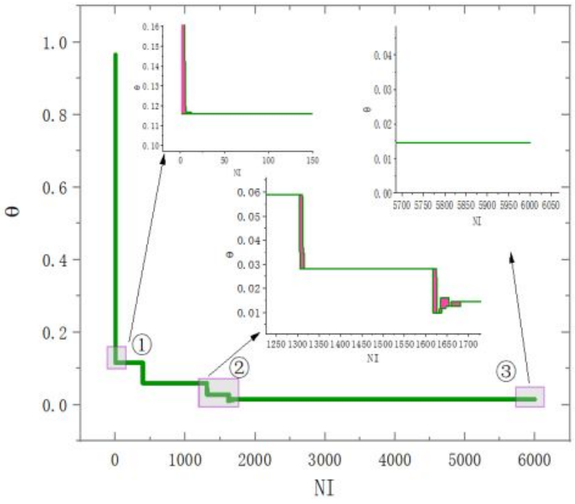


Fig. 3: Convergence of Algorithm 2.

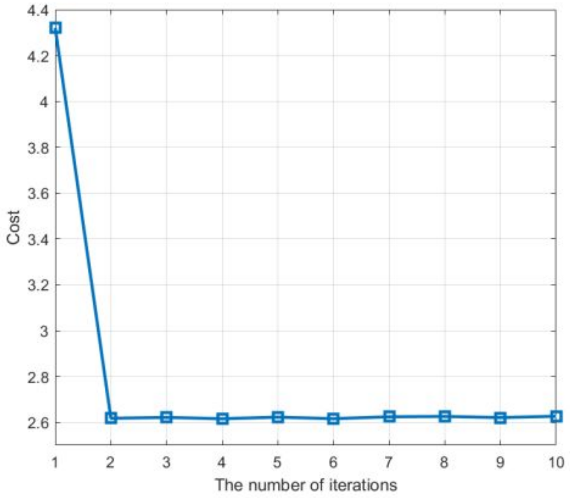


Fig. 4: Convergence of Algorithm 3.