

Deep Q-Learning based Joint Dynamic Network Slicing and Resource Allocation in Multi-Tenant Edge Computing System

Network slice parameter α, β

Task format:

$J_i = (\rho_i, a_i, d_i, T_i^*)$

Local execution time:

$$T_i^l = \frac{(1 - x_i)d_i}{F_i} \quad (1)$$

Data rate:

$$r_i = b_i \log_2 \left(1 + \frac{g_i p_i}{\sigma^2} \right) \quad (2)$$

Transmission time:

$$T_i^{tx} = \frac{x_i a_i}{r_i} \quad (3)$$

Execution time:

$$T_i^{ex} = \frac{x_i d_i}{f_i} \quad (4)$$

Remote execution time:

$$T_i^o = T_i^{tx} + T_i^{ex} \quad (5)$$

Utility:

$$U_i(x, b, f) = \begin{cases} \Psi & \text{if } T_i > T_i^*, \\ \max(T_i^o, T_i^l) & \text{otherwise.} \end{cases} \quad (6)$$

Revenue:

$$Rev_z = \sum_i \rho_i U_i \quad (7)$$

Cost:

$$C_z = \zeta \alpha + \xi \beta \quad (8)$$

Problem formulation:

$$\begin{aligned} & \max \frac{1}{n} \sum_{z=0}^n Rev_z - Cost_z \\ \text{s.t. } & C1 : 1 \\ & C2 : 2 \end{aligned} \quad (9)$$

SP1:

$$\begin{aligned} & \max_{\alpha, \beta} E[Rev_z(\alpha, \beta) - Cost_z(\alpha, \beta)] \\ \text{s.t. } & C1 : \alpha \in [0, 1] \\ & C2 : \beta \in [0, 1] \end{aligned} \quad (10)$$

RL:

State:

current slice scale, traffic history in last k time slots

Action:

increase/ decrease slice scale

Reward:

$$r_{s,a} = \begin{cases} \Phi, & \text{if } \sum_{n \in F} \sum f_{n,w} > F, \\ \sum_{i=0}^n \varphi C_i(x_i, f_i, w_i, e_i) & \text{otherwise.} \end{cases} \quad (11)$$

$$V^\pi = E[\sum_{i=0}^{\infty} \gamma r_i | s_0 = s] \quad (12)$$

$$\pi^* = \arg \max_{\pi} Q(s, a), \forall s \in S \quad (13)$$

$$Q^*(s, a) = Q(s, a) + \alpha(r + \beta \max Q(s, a) - Q(s, a)) \quad (14)$$

$$Loss(\theta) = E[y - Q(s, a, \theta)^2] \quad (15)$$

SP2:

$$\begin{aligned} & \max_{x,b,f} \sum_i \rho_i U_i(x_i, b_i, f_i) \\ \text{s.t. } & C1 : x_i \in [0, 1] \\ & C2 : b_i \in [0, B] \\ & C3 : f_i \in [0, F] \\ & C4 : \sum_i b_i = B \\ & C5 : \sum_i f_i = F \end{aligned} \quad (16)$$

Theorem 1: When minimum occur,

$$x_i = \frac{d_i}{F_i \left(\frac{a_i}{b_i \log_2 \left(1 + \frac{g_i p_i}{\sigma^2} \right)} + \frac{d_i}{f_i} + \frac{d_i}{F_i} \right)} \quad (17)$$

Proof: For i

$$U_i = \max(T_i^o, T_i^l) = \max(x_i \left(\frac{a_i}{b_i \log_2 \left(1 + \frac{g_i p_i}{\sigma^2} \right)} + \frac{d_i}{f_i} \right), \frac{d_i}{F_i}) \quad (18)$$

$$T_i^o = T_i^l, x_i \left(\frac{a_i}{b_i \log_2 \left(1 + \frac{g_i p_i}{\sigma^2} \right)} + \frac{d_i}{f_i} \right) = \frac{(1 - x_i) d_i}{F_i} \quad (19)$$

After rearrange equation (), we can obtain $x_i = \frac{d_i}{F_i \left(\frac{a_i}{b_i \log_2 \left(1 + \frac{g_i p_i}{\sigma^2} \right)} + \frac{d_i}{f_i} + \frac{d_i}{F_i} \right)}$.

Reformulate:
relax U_i to $\hat{U}_i = \dots$ no penalty

$$\begin{aligned} \max_{x,b,f} \quad & \sum_i \rho_i x_i \left(\frac{a_i}{b_i \log_2 \left(1 + \frac{g_i p_i}{\sigma^2}\right)} + \frac{d_i}{f_i} \right) \\ \text{s.t.} \quad & C2 - C5 \end{aligned} \quad (20)$$

Proposition 2: Objective is convex

Proof: The Hessian matrix of (12) consists of, where. Thus, the Hessian matrix of (12) is a positive definite matrix, and hence is convex.

$$\frac{\partial^2}{\partial^2 b_i^2} = \dots \geq 0 \quad (21)$$

$$\frac{\partial^2}{\partial^2 f_i^2} = \dots \geq 0 \quad (22)$$

Theorem 2: Given a set of offloading request, the optimal bandwidth and computing resource allocation is

$$b_i = \frac{\sqrt{\frac{\rho_i x_i a_i}{\log_2 \left(1 + \frac{g_i p_i}{\sigma^2}\right)}}}{\sum_i \sqrt{\frac{\rho_i x_i a_i}{\log_2 \left(1 + \frac{g_i p_i}{\sigma^2}\right)}}} B, f_i = \frac{\sqrt{\rho_i x_i d_i}}{\sum_i \sqrt{\rho_i x_i d_i}} F \quad (23)$$

Proof: We introduce Lagrange multiplier to solve the problem (12). The Lagrange dual function of (12) is

$$L = \sum_i \rho_i x_i \left(\frac{a_i}{b_i \log_2 \left(1 + \frac{g_i p_i}{\sigma^2}\right)} + \frac{d_i}{f_i} \right) + \lambda \left(\sum_i b_i - B \right) + \mu \left(\sum_i f_i - F \right) \quad (24)$$

$$\begin{cases} \frac{\partial L}{\partial b_i} = +\lambda = 0, \forall i \\ \frac{\partial L}{\partial \lambda} = \sum_i b_i - B = 0 \end{cases} \quad (25)$$

$$\begin{cases} \frac{\partial L}{\partial f_i} = -\frac{\rho_i x_i d_i}{f_i^2} + \mu = 0, \forall i \\ \frac{\partial L}{\partial \mu} = \sum_i f_i - F = 0 \end{cases} \quad (26)$$

After solve equation (), we can obtain $b_i = \frac{\sqrt{\frac{\rho_i x_i a_i}{\log_2 \left(1 + \frac{g_i p_i}{\sigma^2}\right)}}}{\sum_i \sqrt{\frac{\rho_i x_i a_i}{\log_2 \left(1 + \frac{g_i p_i}{\sigma^2}\right)}}} B, f_i = \frac{\sqrt{\rho_i x_i d_i}}{\sum_i \sqrt{\rho_i x_i d_i}} F$.

Alg:

init: $x_i = \min(1, \frac{T_i^* F_i^l}{d_i}), \forall i$

Algorithm 1 QoE-Driven Iterative Video Adaptation

Input: The set of

Initialization: replay memory $D = \emptyset$, policy network Q , target network Q

```
1: for episode = 1, M do
2:   Initialize state
3:   while current state is not a terminal state do
4:     Randomly select an action with probability  $\varepsilon$ 
5:     Otherwise, select the action  $a = \max Q$  according to policy network  $Q$ 
6:     Execute the action observe reward  $r$ 
7:     Store transition  $(s_i, s_{i+1}, a, r)$  in  $D$ 
8:     Sample random a minibatch of transitions from  $D$ 
9:      $y =$ 
10:    Update weights  $\theta$  of policy network  $Q$  via a learning step with optimizer
11:    Reset target network  $Q \leftarrow Q$  every  $C$  steps
12:   end while
13: end for
```

Output: The set of
