3

We define a task offloading policy $X = \{x_n^k | n \in N,\ k \in K\}$ to describe the mode & channel selection of each VT_n, Here, n from 1 to N denotes the set of video tasks, and k from 1 to K denotes the K optional channels. The binary variable x_n^k indicates the transmission state of VT_n. We introduce the transmit power policy P, where p_n^k denotes the transmit power. We denote the local computational resource allocated to VT_n as f_n^k, The energy consumption for processing the video task locally is E_n^{loc}, the time consumption is T_n^{loc}, the video processing quality is ϕ_n^{loc}. For MEC computing, they are T_n^{off}, E_n^{off} and ϕ_n^{off}. And T_n, E_n, ϕ_n equals $(1 - x_n) * local + x_n * offload$. So our utility R_n equals the weighting of ϕ and E. The problem can be formulated as a maximization with some constraints

4

The environment state of the cooperative network is task state $S_{task}(t)$, mode & channel selection state $S_{channel}(t)$, power transmission state $S_{pow}(t)$, and local resource allocation state $S_{res}(t)$.

S_task is [Sn, Cn], $S_n(t)$ represents the task size of VT_n, and $C_n(t)$ represents the required CPU cycle to compute VT_n.

The action space can be expressed as:

$\mathcal{A}(t) = \{x_j^k(t), k(t), f_j^k(t), \beta_j^k(t)\}$.

Task offloading decision : xnk(t) is based on $S_{task}(t)$.

Channel selection: is based on $S_{channel}(t)$.

local computational resource allocation: $f_n^k(t)$ Based on $S_{res}(t)$ and $S_{task}(t)$,

Compression rate selection: $\beta_n^k(t)$.

5

We use the dataset for numeric simulation and to determine our hyperparameters.

The first three graphs are the comparison of convergence process of the system average reward in our MADDPG. They change number of agents or we say number of video tasks N, time constraint τ_n, and quality constraint ϕ_n separately. We use the method of Lagrangian function to treat the constraints.

The second three graphs are the comparison of average converged rewards in different algorithms. Here MAA2C and MAPPO are another two MADRL algorithms, and NAC and ALLES are two baselines, NAC is the algorithm that set action β to a fixed value 0.2, ALLES is the algorithm that all x_n^k = 1, which means all agents would use MEC mode and no local mode.

6

From graphs, we can see the average system rewards would increase as time constraint increase, and would decrease as quality constraint increase because when time constraint is larger or quality constraint is smaller, the penalty caused by longer processing time or weaker quality would be smaller, so the reward would increase. The reward increases then decreases as number of agents increases, because when number of agents is smaller then a critical point, the channels are not in saturation, so more agents would cause the system use more offloading resources, but when agents are more and more, the channels are in saturation and each agents would has less resources to compute, they tend to compute locally, so the reward decreases.

For different algorithms, we can see MADDPG is a little better than MAA2C and MAPPO if we choose the best choice of parameters.

Change the value of power s_n^k of one agent, the energy and quality of it would increase, but other agents' energy would increase and reward would decrease.

This is another of our experiment to see the influences of agent to agent