依赖任务与并行任务

任务：{任务量、位置信息、子任务DAG依赖图}
子任务：{计算所需CPU数、传输时延}

资源：{处理速度、总带宽、已接收和可接受任务数，位置信息}

任务-子任务-资源：{决策变量、计算资源、传输速率、传输功率、信道增益}

策略	依赖任务调度	并行任务调度
问题	解决必须遵循特定顺序执行的任务序列	将独立任务分散到多个处理器上以同时执行
背景	工作流管理、批处理作业、大数据处理	云计算环境、数据中心、并行计算机
关注点	任务间的依赖性与执行顺序	独立任务的并行性能和资源利用率
最新方法	机器学习预测模型、进化算法混合策略	容器化技术、虚拟化、资源弹性管理
目标	减少总完成时间、优化资源分配、处理任务间依赖关系	提高计算资源利用率、减少等待时间、平衡负载、优化吞吐率
算法	启发式算法、元启发式算法、遗传算法	负载均衡算法、随机算法、最短工作优先

Efficient Dependent Task Offloading for Multiple Applications in MEC-Cloud System

期刊：IEEE TRANSACTIONS ON MOBILE COMPUTING　　日期：2023　　分区：二区

中南大学

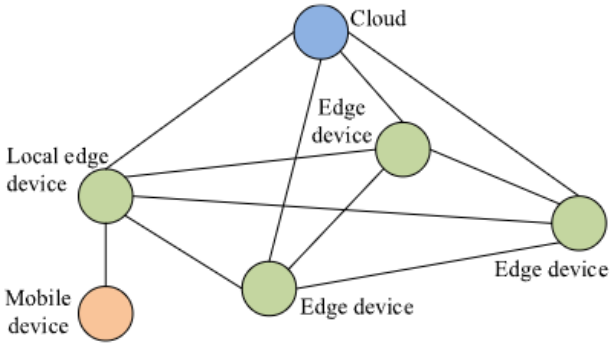
MEC-Cloud系统中多个应用的高效依赖任务卸载

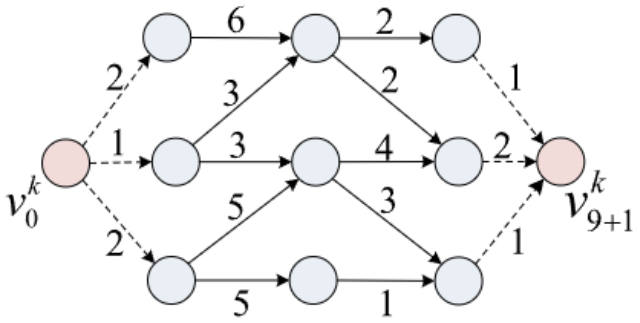
问题：

1.没有考虑任务的依赖性以及MEC和云的编排

研究对象：

任务、边缘设备、云





俩个图模型：

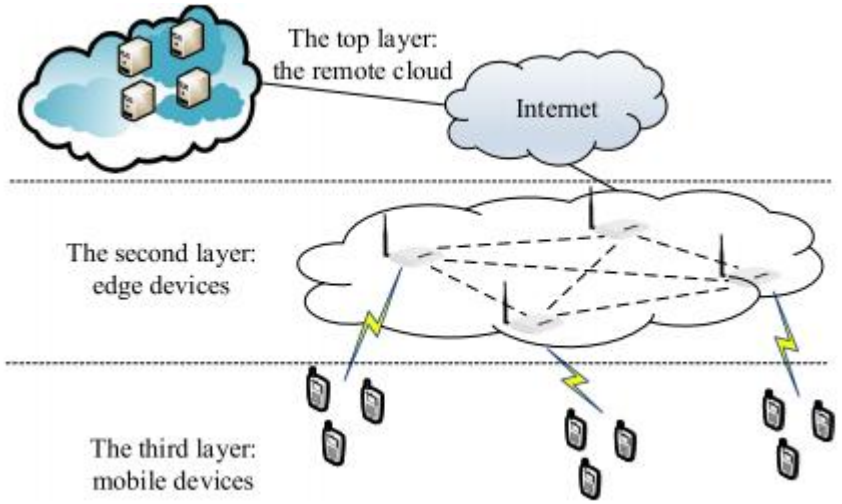
设备建模：从移动、边缘到云设备的无向图

任务建模：从程序任务量大小到返回执行结果数据大小

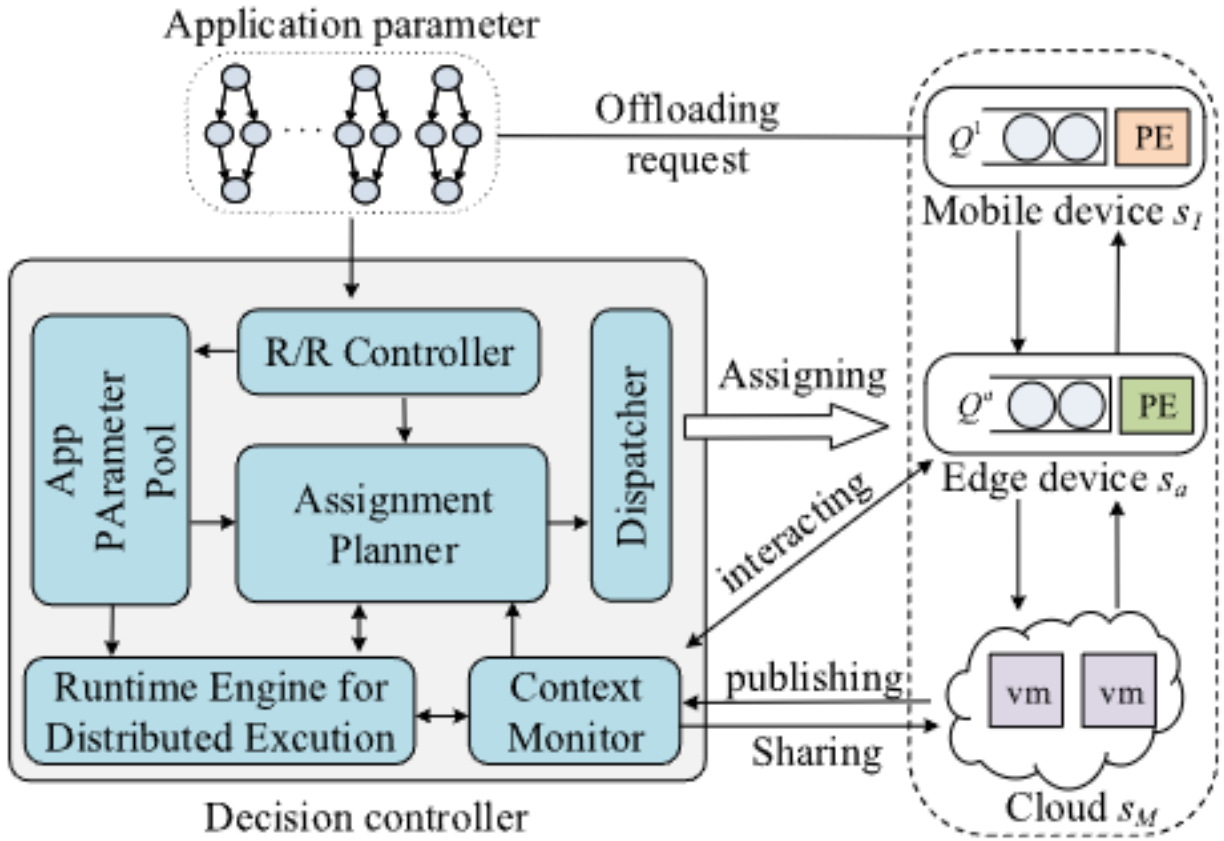
目标：

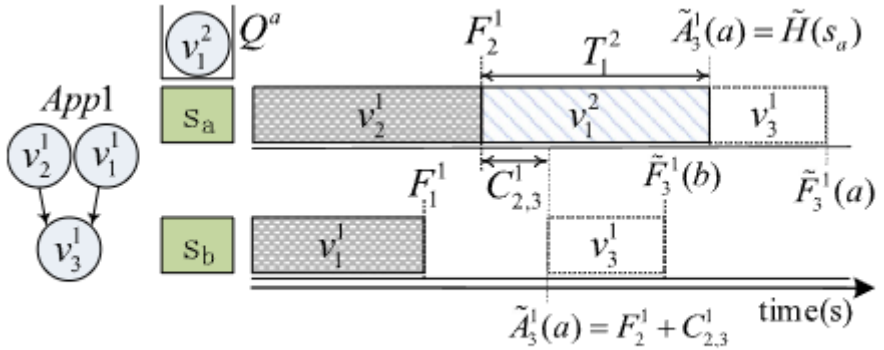
考虑每个设备排队延迟，以最小化应用程序的平均完成时间。

$$\mathbf{P1}: \quad \min_{\mathbf{x}_i^k} \quad \frac{1}{K}\sum_{k=1}^{K} \max_{v_i^k \in \mathcal{V}^k} F(\mathbf{x}_i^k),$$

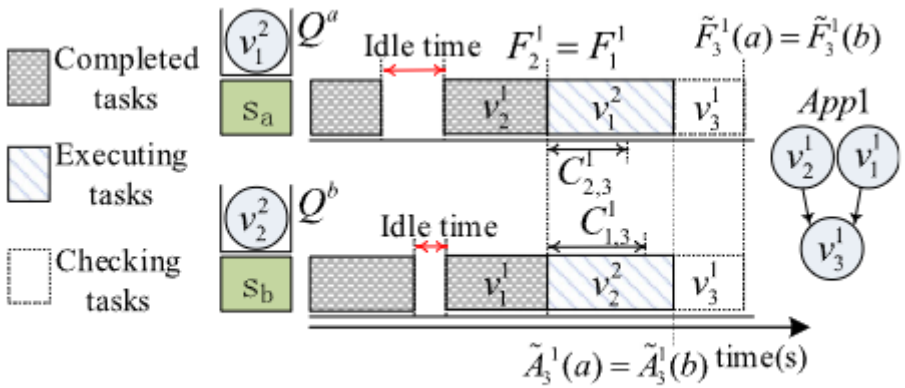


COFE


Application parameter
Offloading request
Mobile device s_1
PE
Assigning
Edge device s_a
interacting
publishing
Sharing
Cloud s_M
vm
vm
App PArameter Pool
R/R Controller
Assignment Planner
Dispatcher
Runtime Engine for Distributed Excution
Context Monitor
Decision controller


App1
time(s)

(a)


Completed tasks
Executing tasks
Checking tasks
Idle time
Idle time
App1
time(s)

(b)

RFID: Towards Low Latency and Reliable DAG Task Scheduling over Dynamic Vehicular Clouds
期刊：IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY　日期：2023　分区：二区
厦门大学

面向动态车辆云的低时延可靠DAG任务调度

问题：

1.动态和不稳定的VC拓扑导致计算资源的可用性随时间变化

研究对象：

子任务建模DAG、车辆

子任务：{调度时间、最早执行时间和完成时间、实际完成时间}

车辆：{数据传输时间}

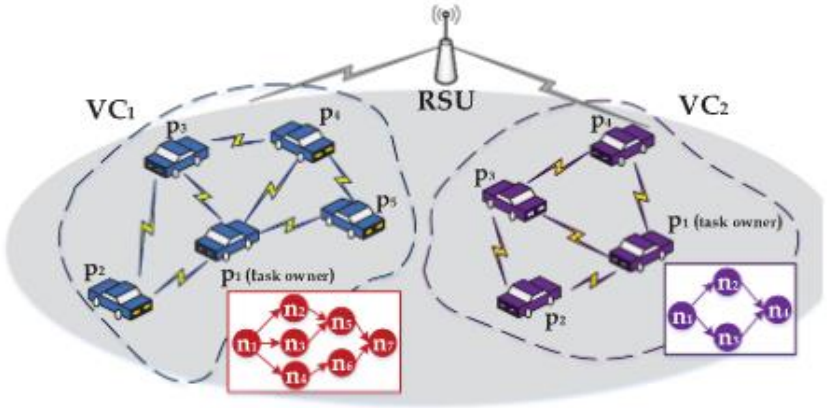
方法：

根据车辆资源的可用性选择性地改变一小部分子任务的调度优先级

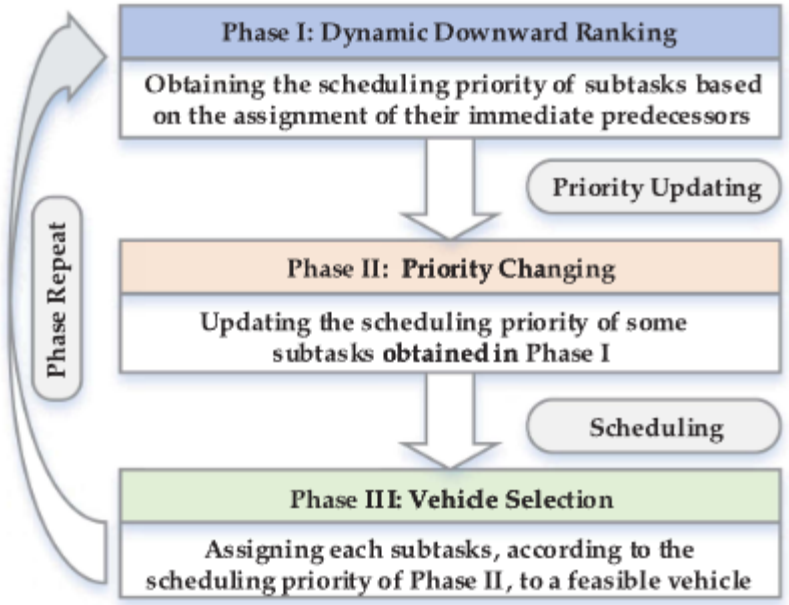
目标：

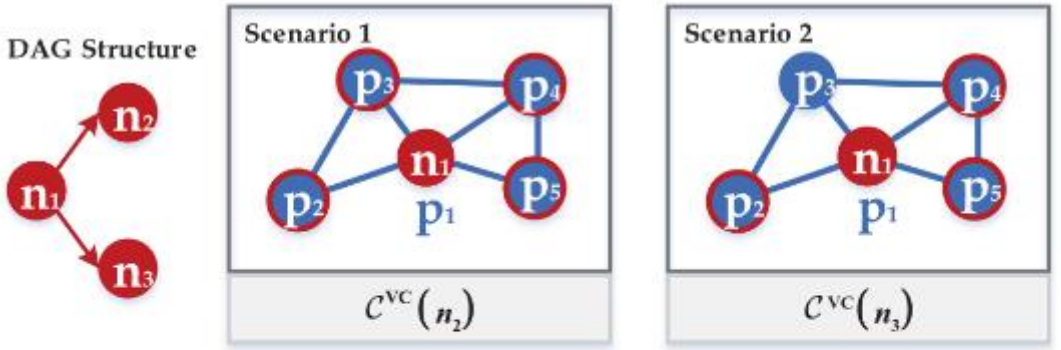
减少DAG任务的整体完成时间

$$P: \quad \arg\min_{\{\xi_{n_i,p_m}, n_i \in V^A,\ p_m \in V^{VC}(st_{n_i})\}} OTC,$$



FID

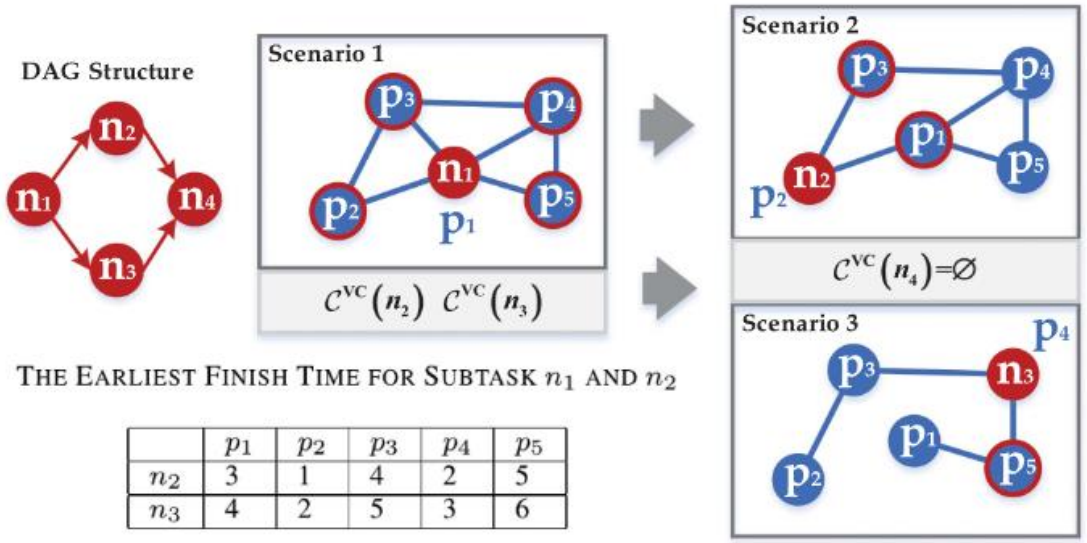




THE ESTIMATED FINISH TIME FOR SUBTASK n_1 AND n_2

	p_1	p_2	p_3	p_4	p_5
n_2	4	8	5	8	8
n_3	5	9	inf.	9	9

改进：最小化ni在第二优选车辆上执行的最早完成时间。



THE EARLIEST FINISH TIME FOR SUBTASK n_1 AND n_2

	p_1	p_2	p_3	p_4	p_5
n_2	3	1	4	2	5
n_3	4	2	5	3	6

改进：不仅考虑将 ni 分配给 pm 时完成时间，而且考虑可以接收的数据传输的车辆数量。

处理器负载较低时，降低电压和频率减少功耗，而负载较高时，提高以提高性能。

问题：
难以预测不断变化的动态网络环境

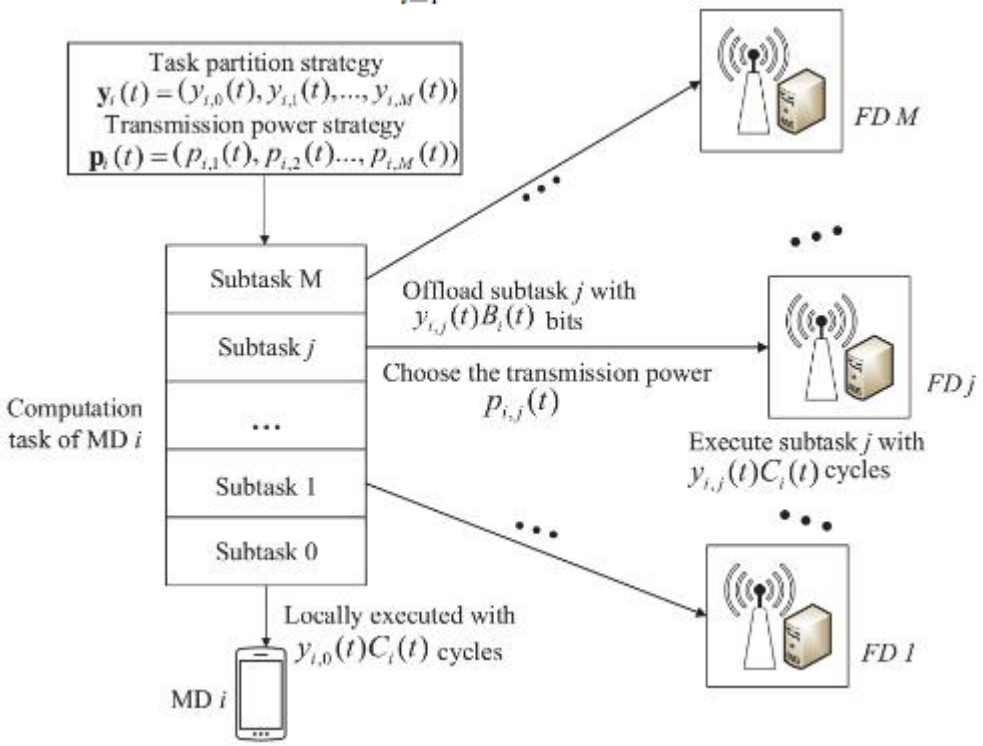
研究对象：
MD、FD
MD：{数据大小、CPU 周期、最大延迟}
FD：{计算、通信资源}
MD-FD：传输功率

方法：
在动态环境下联合任务划分和功率控制（多智能体深度强化学习）

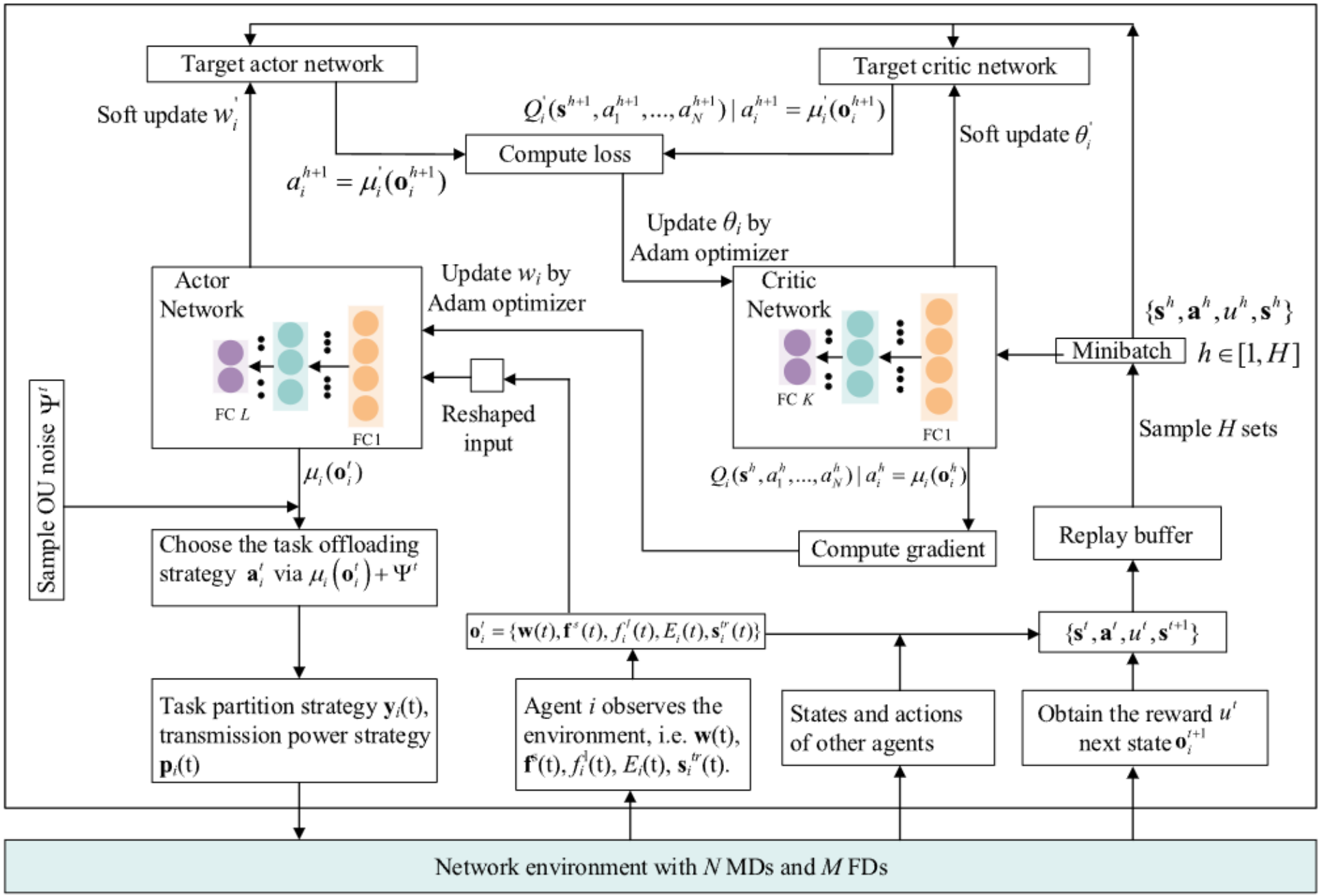
目标：连续决策，最小化MD的执行延迟和能量消耗

$$u_i(t) = \alpha_b B_i(t) - \alpha_d \tau_i(t) - \alpha_e e_i(t)$$

$$\Phi_i(t) = \sum_{t=1}^{t=T} \gamma^{(t-1)} u_i(t)$$



MADDPG


Target actor network
Target critic network
Soft update $w_i^{'}$
$Q_i^{'}(\mathbf{s}^{h+1}, a_1^{h+1}, ..., a_N^{h+1}) \mid a_i^{h+1} = \mu_i^{'}(\mathbf{o}_i^{h+1})$
Soft update $\theta_i^{'}$
Compute loss
$a_i^{h+1} = \mu_i^{'}(\mathbf{o}_i^{h+1})$
Update θ_i by Adam optimizer
Actor Network
FC L
FC1
Update w_i by Adam optimizer
Critic Network
FC K
FC1
$\{\mathbf{s}^h, \mathbf{a}^h, u^h, \mathbf{s}^h\}$
Minibatch
$h \in [1, H]$
Sample OU noise Ψ^t
Reshaped input
Sample H sets
$\mu_i(\mathbf{o}_i^t)$
$Q_i(\mathbf{s}^h, a_1^h, ..., a_N^h) \mid a_i^h = \mu_i(\mathbf{o}_i^h)$
Choose the task offloading strategy $\mathbf{a}_i^t$ via $\mu_i(\mathbf{o}_i^t) + \Psi^t$
Compute gradient
Replay buffer
$\mathbf{o}_i^t = \{\mathbf{w}(t), \mathbf{f}^s(t), f_i^f(t), E_i(t), \mathbf{s}_i^{tr}(t)\}$
$\{\mathbf{s}^t, \mathbf{a}^t, u^t, \mathbf{s}^{t+1}\}$
Task partition strategy $\mathbf{y}_i$(t), transmission power strategy $\mathbf{p}_i$(t)
Agent i observes the environment, i.e. $\mathbf{w}$(t), $\mathbf{f}^s$(t), f_i^f(t), E_i(t), $\mathbf{s}_i^{tr}$(t).
States and actions of other agents
Obtain the reward u^t next state $\mathbf{o}_i^{t+1}$
Network environment with N MDs and M FDs

Multi-level parallel scheduling of dependent-tasks using graph-partitioning and hybrid approaches over edge-cloud

期刊：Soft Computing　　日期：2022　　分区：三区

印度浦那大学

在边缘云上使用图分区和混合方法对依赖任务的多级并行调度

将每个工作流划分为子任务集群，目的是本地化每个集群中的子任务之间的通信。

问题：

子任务的数量与算法复杂度呈指数级增长

研究对象：

子任务集、分布式计算资源

方法：

1.图分区 Brandes 算法减少了需要调度的任务数量

2.介数中心性（BC）衡量了节点在图中作为桥梁的程度，即节点在图中的所有最短路径中出现的频率。

目标：

最小化总执行时间