文章编号：1003-0530(2022)03-0609-10

WSN中基于强化学习的能效优化任务处理机制

张明杰 朱 江
（重庆邮电大学通信与信息工程学院，移动通信教育部工程研究中心，
移动通信技术重庆市重点实验室，重庆 400065）

摘 要：以提高无线传感器网络中任务处理的能效为目标，提出了一种近似最优化的任务处理机制，无线传感器节点可根据任务缓存区的任务数量、信道条件，动态地实现任务向边缘服务器的卸载以及本地处理。将任务处理机制建模为马尔可夫决策过程，因为无线传感器节点不知道此过程的状态转移概率，所以采用A3C算法以实现在环境参数未知情况下的探索和学习，从而得到近似最优的任务处理策略。仿真结果表明，与其他机制相比，所提任务处理机制能提高节点能效，且收敛速度更快。

关键词：无线传感器网络；移动边缘计算；马尔可夫决策过程；强化学习
中图分类号：TN929.5 **文献标识码：**A **DOI：**10.16798/j.issn.1003-0530.2022.03.019

引用格式：张明杰，朱江. WSN中基于强化学习的能效优化任务处理机制[J]. 信号处理，2022，38(3)：609-618. DOI：10.16798/j.issn.1003-0530.2022.03.019.
Reference format：ZHANG Mingjie，ZHU Jiang. Energy efficiency optimization task processing mechanism based on reinforcement learning in WSN[J]. Journal of Signal Processing，2022，38(3)：609-618. DOI：10.16798/j.issn.1003-0530.2022.03.019.

Energy Efficiency Optimization Task Processing Mechanism Based on Reinforcement Learning in WSN

ZHANG Mingjie ZHU Jiang
（Chongqing Key Laboratory of Mobile Communications Technology，Engineering Research Center of Mobile Communications of the Ministry of Education，School of Communication and Information Engineering，Chongqing University of Posts and Telecommunications，Chongqing 400065，China）

Abstract: Aiming at improving the energy efficiency of task processing in wireless sensor networks，a nearly optimal task processing mechanism is proposed，in which wireless sensor nodes can dynamically unload tasks to edge servers and perform local processing according to the number of tasks in the task cache and channel conditions. The task processing mechanism is modeled as a Markov decision process. Since the wireless sensor node does not know the state transition probability of this process，the A3C algorithm is used to realize exploration and learning under unknown environmental parameters，so as to obtain the approximately optimal task processing strategy. Under certain buffer conditions and channel conditions，the optimal task quantity，modulation level and transmission power are selected by this strategy，and the average task processing energy efficiency is improved. Simulation results show that compared with other mechanisms，the proposed task-processing mechanism can improve node energy efficiency and has faster convergence speed.

Key words: wireless sensor network；mobile edge computing；Markov decision process；reinforcement learning

收稿日期：2021-05-31；修回日期：2021-07-20
基金项目：国家自然科学基金(61271260)；重庆市教委科学技术研究项目(KJ1400416)资助课题

1 引言

物联网(internet of things, IoT)时代的来临促进了无线传感器网络(wireless sensor network, WSN)的部署,由于WSN现有以及潜在的广泛应用,使其被确定为当今最重要的技术之一[1]。但因为计算能力和电池容量的限制,无线传感器通常无法有效处理复杂的计算任务,因此,如何提升计算能力和能量效率是WSN研究的热点领域。国内外对于无线传感器能量效率的研究主要分为资源的合理调度[2-4]以及运用能量收集技术(energy harvesting, EH)对无线传感器进行充能[5-8]。

为了满足更多场景下任务处理需求,将边缘计算与WSN相结合。移动边缘计算(mobile edge computing, MEC)能够在网络边缘为用户提供计算卸载和数据缓存,为用户提供更加高效的存储和传输。采用基于边缘计算的任务卸载技术可提高任务处理效率[9]。然而,在边缘计算中,将整个计算任务卸载到边缘计算服务器的方式能效不高,因此某些任务应由本地计算执行[10]。文献[11]定义了一种卸载优先级函数,将部分任务卸载到边缘进行计算。在满足时间延迟的前提下,提出有效的卸载决策以最小化能源成本是一个关键问题[12-15]。通过对缓存器内任务的部署,可以有效提升任务处理过程中的能量利用率,降低网络能耗[16-17]。

将WSN与边缘计算结合后,数据能够得到处理,但二者的自组织能力以及对于环境的适应性有限,为解决相应问题,国内外学者采用人工智能[18]方法对传感器不同的任务卸载场景进行智能调度,降低了系统能耗。文献[19]提出了一种基于强化学习的隐私感知卸载方案,应用Dyna算法框架提供模拟卸载以加快学习过程,从而提高计算性能。文献[20]针对具有EH的IoT设备提出了一种基于强化学习的卸载方案。文献[21]采用二进制卸载策略的无线MEC网络,提出了基于深度强化学习的DROO在线卸载算法框架,降低了计算复杂度。文献[22]提出了一种自适应睡眠/唤醒调度方法,在不牺牲数据包传递效率的情况下,节省每个节点的能量。然而,上述工作主要集中在提升传输速率和降低计算复杂度,较少有学者考虑物理层以及数据链路层参数对系统能量消耗的影响。但相关参数对系统能效的提高有着重要的意义。

为了提高无线传感器设备系统的任务处理能效,本文研究了基于能效的任务处理机制,主要工作如下。

(1) 建立了基于马尔可夫决策过程的任务处理机制。物理层以及数据链路层的最佳控制角度考虑任务本地计算和边缘计算能量的联合优化,通过智能优化代理得到任务到达缓存区后取出的本地计算任务量和卸载任务量,保证任务得到有效处理的同时得到近似最优的系统能效。

(2) 利用强化学习代理通过贪婪策略进行动作探索和利用从而获得最佳的卸载策略。针对在线学习收敛速度慢的问题,A3C算法采用异步训练框架加快学习收敛速度。

2 系统模型

图1是多个无线传感器网络的节点进行任务处理的模型,每个无线传感器buffer的处理情况都和节点2相同。每个无线传感节点采集到的数据被封装成待处理的任务,存储在内部的缓存器中。每隔相同的时间段(时隙),节点从缓存器内取出一定数量的任务进行处理:即一部分任务卸载到边缘服务器进行处理,一部分由传感器消耗自身资源进行本地处理。

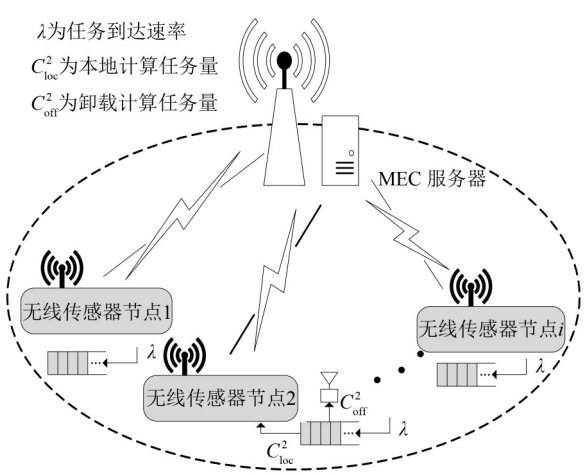

图1 系统模型图

Fig. 1 System model diagram

当有任务卸载到边缘服务器进行处理,在每个时隙内,各个无线传感节点会以CDMA的方式将任务数据发送到边缘服务器端。因此,系统采用的是

以时隙为时间单位的时分多址与 CDMA 混合的多址方式。一个节点经历的无线信道干扰取决于其他节点采用的传输功率(详见 4.3 节)。每个节点根据其缓存区内任务数量和测得的信道状态独立地学习其任务处理策略。

单个无线传感器节点与边缘服务器的任务处理框图如图 2 所示。设缓存器长度为 L，λ 为一帧内任务到达缓存器的平均到达率，λ 服从泊松分布。某一帧开始时节点 i 缓存器内任务量为 n_b^i，b_n^i 为从缓存区内取出的任务数量，取出的任务数量分成本地计算 C_{loc}^i 以及边缘计算 C_{off}^i 两部分，则有 $b_n^i = C_{\text{loc}}^i + C_{\text{off}}^i$。下一帧开始时缓存器内的包的数量 $n_b^{i'}$ 为

$$n_b^{i'} = \min\{n_b^i - C_{\text{loc}}^i - C_{\text{off}}^i I\{\delta = \text{ACK}\} + \lambda\} \quad (1)$$

$\delta = \{\text{ACK}, \text{NACK}\}$ 为边缘服务器反馈的接受确认信息，成功接收为 ACK，失败则返回 NACK，$I\{\cdot\}$ 为指示函数，括号内为真，函数值返回 1，否则返回 0。

在传输之前，可以通过发送导频信息获得边缘计算服务器端反馈的信道状态信息(channel state information，CSI)。智能控制代理观察缓存区内任务量以及前一次传输的信道增益，在此基础上，决定从缓存区取出多少任务进行本地计算和卸载以及获得相应的最佳传输功率和调制级别，使任务得到有效处理的同时近似最大化系统能效。若某信道的任务数据被成功接收，接收机将反馈确认消息 ACK，否则反馈失败消息 NACK。没有被成功发送的任务数据将被重发。

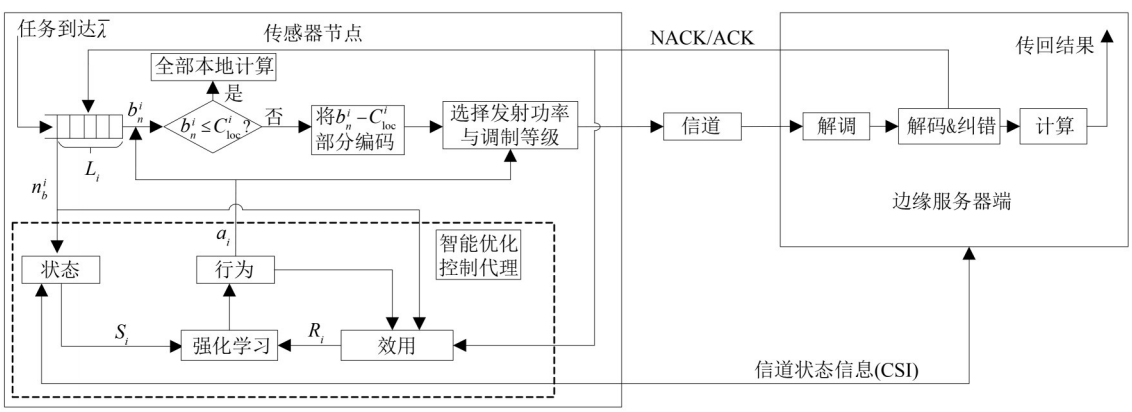

图 2　单个无线传感器节点与边缘服务器的任务处理框图

Fig. 2　Task processing block diagram of a single wireless sensor node and edge server

3　能耗分析

任务处理过程中，所消耗的能量主要由三部分构成，分别为本地计算能量、卸载能量以及从边缘服务器下载任务处理结果所耗能量。由于计算结果下载能量相较于本地计算能量以及计算卸载能量可以忽略不计，本文暂不考虑。

3.1　本地计算能耗

本地计算能量消耗取决于本地计算任务量以及无线传感器节点的计算能力，假设 CPU 频率在每个节点处为固定值，C_i 为节点 i 计算 1 比特数据需要的 CPU 周期数，P_i 为该节点进行本地计算每个 CPU 周期的能量消耗，则节点 i 一帧内本地计算总能耗表达式为

$$E_{\text{loc}}^i = C_{\text{loc}}^i C_i P_i \quad (2)$$

考虑本地计算时延的约束，则有 $C_i C_{\text{loc}}^i / F_i \le T$，其中 F_i 为节点 i 的计算能力，T 为一帧内本地计算时间。

3.2　卸载所耗的能量

在本地计算时间不能满足时延约束的情况下，将多余任务卸载到边缘服务器进行计算。任务卸载所采取的传输功率取决于强化学习代理所做的决策。卸载所消耗的能量如下式所示：

$$E_{\text{off}}^i = t_t^i P_{\text{tran}}^i \quad (3)$$

式(3)中 $t_t^i = C_{\text{off}}^i / (R_b \cdot m^i)$ 为节点 i 传输时间，C_{off}^i 为节点 i 的传输任务量，R_b 为传输速率，P_{tran}^i 为节点 i 传输功率。

4　基于 MDP 的任务处理机制

将基于系统能效的任务传输调度机制建模为

马尔科夫决策过程，其状态、行为、状态转移概率以及回报函数被定义为($\boldsymbol{S}, \boldsymbol{A}, \boldsymbol{P}, \boldsymbol{R}$)四元组。

4.1 状态集 $\boldsymbol{S}$

包含所有可能状态的状态空间，在本文中，状态空间定义为关于缓存区任务量、信道增益的聚合状态 $\boldsymbol{S}(n_b^i, \gamma^i)$，其中 n_b^i 为节点 i 缓冲区内的任务量，γ^i 为卸载任务时节点 i 信道的信道增益。

4.2 行为集 $\boldsymbol{A}$

包含所有可能行为的行为空间，在本文中，行为空间定义为智能控制代理根据状态空间选取的调制等级和发射功率以及从缓存区取出的任务量，表示为 $\boldsymbol{A}(b_n^i, m^i, p_{\text{tran}}^i)$。状态空间由信道增益以及缓存器内任务数量构成，每转换到一个新的状态会得到信道增益以及缓存器当前任务数量参数，根据两个参数，行为空间根据奖励函数选择出最优的从缓存区取出的任务数量、调制等级以及发射功率，当缓存器内任务数量多时，需要使用高的调制等级增加传输吞吐量以满足任务处理需求。b_n^i 为节点 i 从缓存器内取出的任务量，m^i 为节点 i 根据状态选择的调制等级(BPSK-8PSK)，p_{tran}^i 为节点 i 相应调制等级的发射功率。

4.3 状态转移概率 $\boldsymbol{P}$

使用有限状态马尔可夫信道对无线信道动态进行建模[23-24]，将等效信道增益划分为有限数 K 个区间，$0 = \Gamma_0 < \Gamma_1 \cdots < \Gamma_K$，信道增益在 Γ_{K-1} 到 Γ_K 间则称为状态 k，在服从瑞利衰落的信道中，γ 呈指数分布，概率密度函数为 $p(\gamma) = 1/\gamma_0 \exp(-\gamma/\gamma_0)$，其中 γ_0 是平均信道增益。

稳态概率为

$$\zeta_k = \int_{\Gamma_{k-1}}^{\Gamma_k} p(\gamma)\mathrm{d}\gamma \qquad k = 1, \cdots, K \tag{4}$$

状态转移概率为

$$p_c(k, k+1) = N(\Gamma_k)\frac{t_t^i}{\pi_k} \qquad k = 1, \cdots, K \tag{5}$$

其中 $N(\Gamma) = \sqrt{2\pi\Gamma/\gamma_0}\, f_d \exp(-\Gamma/\gamma_0)$ 是电平交叉函数，f_d 是最大多普勒频率。π_k 为状态 s_k 下选择行为 a_k 的概率。

用 $P_\Gamma^i(\Gamma(\gamma^i, p_{\text{tran}}^i))$ 表示节点 i 任务数据被成功接收的概率。多节点方案中每个节点在信道中动态交互。根据每个节点的已接收的SIR描述此交互。假设有一些节点想同时与接收器通信。链路的SIR可以表示为

$$\Gamma^i(\gamma^i, p_{\text{tran}}^1, \cdots, p_{\text{tran}}^N) = \gamma^i \times \frac{BA_t^i p_{\text{tran}}^i}{R_b\left(\sum_{j \neq i} A_t^j p_{\text{tran}}^i + \sigma^2\right)} \tag{6}$$

式(6)中 $A_t{}^i = 1/(d^i)^4$ 为链路 i 的路径损耗，d^i 为无线传感器 i 与服务器的距离。σ^2 为热噪声的方差。

任务成功分组传输概率为

$$P_\Gamma(\Gamma^i(\gamma, p_{\text{tran}}^1, \cdots, p_{\text{tran}}^N), m) = 1 - \text{erfc}\left(\sqrt{\Gamma}\sin\left(\frac{\pi}{2^m}\right)\right) \tag{7}$$

令 K 表示成功发送所需的重传次数，设每次传输都是独立的，则 K 的概率质量函数为

$$P_K(k) = P_\Gamma(\Gamma^i(\gamma, p_{\text{tran}}^1, \cdots, p_{\text{tran}}^N), m) \cdot [1 - P_\Gamma(\Gamma^i(\gamma, p_{\text{tran}}^1, \cdots, p_{\text{tran}}^N), m)]^{k-1} \tag{8}$$

得到了成功分组传输概率以及信道状态转移概率，相应的节点状态转移以及转移概率为

传输成功：

$$S_{k+1} = (n_{b,k}^i + q - b_n^i, \gamma_{k+1}^i)$$
$$P_{S_k, S_{k+1}}(k) = P_\Gamma(\Gamma^i(p_t^1, \cdots, p_t^N), m) \cdot P_c(k, k+1) \tag{9}$$

传输失败：

$$S_{k+1} = (n_{b,k}^i + q - C_{\text{loc}}^i, \gamma_{k+1}^i)$$
$$P_{S_k, S_{k+1}}(k) = (1 - P_\Gamma(\Gamma^i(p_{\text{tran}}^1, \cdots, p_{\text{tran}}^N), m)) \cdot P_c(k, k+1) \tag{10}$$

4.4 奖励函数 $\boldsymbol{R}$

在WSN的应用中，能耗、吞吐量以及时延都是非常关键的因素，在最小化能源消耗的同时，如果任务处理数量太少或延时太大都是不可接受的。因此，采用总消耗的能量成功处理的任务量作为目标函数。从缓存区内取出的任务 b_n^i 分为本地计算 C_{loc}^i 以及卸载到边缘 $C_{\text{off}}^i = b_n^i - C_{\text{loc}}^i$ 两部分分别进行处理。系统的目标函数为每总消耗能量的成功任务处理数量，则一帧内效用函数表达式为

$$\frac{\text{一帧内任务处理量}}{\text{一帧内总消耗能量}} = \frac{C_{\text{loc}}^i + C_{\text{off}}^i \cdot P_\Gamma(\Gamma^i, m)}{E_{\text{loc}}^i + E_{\text{off}}^i} \tag{11}$$

式(11)中 $P_\Gamma(\Gamma^i, m)$ 为任务成功分组传输概率，效用函数的单位为任务处理数量每焦耳。假设卸载一个任务所包含数据的信息比特为 L_b 与添加错误解码代码后的任务数据信息比特为 L。传输速率为 R_b bit/s。传输消耗能量为 $t_t^i p_t^i = (C_{\text{off}}^i/(R_b \cdot m^i))p_{\text{tran}}^i$。

由于任务到达速率的不同，不可预知下一状态即将到达多少任务量，如果任务量过多，而采用的卸载策略卸载任务量小，则根据当前缓存区内所剩

余的任务数量对当前奖励函数进行惩罚，迫使当前状态时，选择卸载任务量更大的行为。为了最小化缓存区溢出的可能性，将缓存处理成本$f(n_b^i)$合并到奖励函数中，提高服务质量（Quality of Service，QoS）。奖励函数表示为

$$R(S,A)=R((n_b{}^i,\gamma^i),(b_n{}^i,m^i,p_{\text{tran}}{}^i))=\frac{L_b}{L}\cdot\left(\frac{C_{\text{loc}}^i+C_{\text{off}}^i\cdot P_\Gamma(\Gamma^i,m)}{C_{\text{loc}}^i C_i P_i+(C_{\text{off}}^i/(R_b\cdot m^i))P_{\text{tran}}^i}\right)\times 10^{-3}-\varepsilon f^i(n_b^i)\quad(12)$$

奖励函数前一项为每单位总能量消耗后任务成功处理量。后一项是缓存区处理成本，权重ε在防止缓冲溢出的同时，使得缓存处理成本降低。奖励函数由贝尔曼方程驱动，通过循环迭代的方式获得计算回报值$R(s,a)$。

由于需要的是最大化每个用户总消耗能量的长期平均任务处理量，因此关注于每个阶段的平均奖励，表示为：

$$\rho^\pi(s_0)=\lim_{n\to\infty}\frac{1}{n}E_\pi\left[\sum_{k=0}^{n-1}R(s_k,\pi(s_k))\right],\quad s_k\in S,\pi(s_k)\in A\quad(13)$$

式(13)中$\pi(s_k)$为s_k状态下采取的策略，$E(\cdot)$为取均值。

5　求近似最优解

在已知状态转移概率以及信道状态完美的情况下，对于任何马尔科夫决策过程，存在一个最优策略π^*，优于或至少不差于所有其他策略，根据已知的缓存区任务数量通过策略迭代选择出最佳即使得价值函数最大的调制等级以及发射功率，带入效用函数中可以得到相应状态下的最优能效值，不同状态下的得到的能效值不同，对所有状态下的最优能效值求平均，得到平均最优能效。本文仿真之所以能得到最优值，其实是假设预知某个状态时，选择相应行为能获得最优能效，然而在大部分场景中，这个假设难以成立，即难以获得状态转移概率，以及完美的信道状态，故采用强化学习算法得到接近最优的平均能效值。

本节中运用 Asynchronous Advantage Actor-critic 算法解决 MDP 每个阶段的平均回报值。相较于 AC 算法、Dyna 算法以及 DROO 算法，A3C 算法利用同时在多个线程里面分别和环境进行交互学习的方式，算法框架如图 3 所示，每个线程即代表根据不同

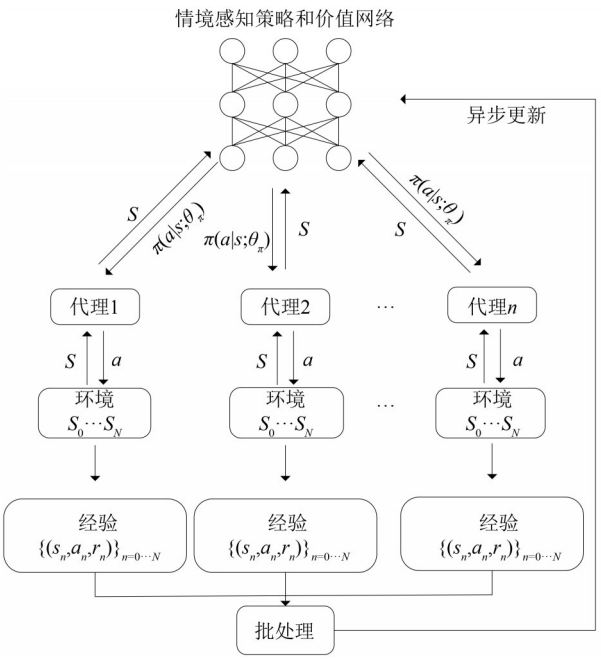

图 3　A3C 算法框架

Fig. 3　Algorithm framework

的信道增益以及缓存区任务数量，选择不同的从缓存区取出任务数量、调制等级以及发射功率后，与环境交互得到该线程的回报值。每个线程都学习出自己的成果，最后把所有子网络的学习成果汇总起来，整理保存在一个公共的全局网络中，并且，子网络会定期从公共网络中学习成果回来指导自己和环境之后的学习交互。回避了相关性过强的问题的同时还做到了异步并发的学习效果。由于 A3C 算法中各个子线程相互独立，对单个子线程进行分析可推广至全局。

算法的本质为更新相对状态值函数$h(s)$以及平均奖励值ρ，建立本文贝尔曼驱动方程为

$$B(h(s))=R(s,\pi(s))+\sum_{s'=1}^{S}P_{s,s'}(\pi(s))h(s')\quad(14)$$

状态值更新与奖励值更新可表示为

$$\begin{aligned}h_{k+1}(s_{k+1})&=B(h_k(s_k))-\rho_k\\ \rho_{k+1}&=B(h_k(s_k))-h_k(s_k)\end{aligned}\quad(15)$$

通过贝尔曼驱动方程，可以消除对状态转移概率的需求，相对状态值函数具体更新过程为

$$\begin{aligned}h_{k+1}(s_{k+1})&=(1-\alpha_k)h_k(s_k)+\alpha_k h_{k+1}(s_{k+1})=\\ &(1-\alpha_k)h_k(s_k)+\alpha_k(B(h_{k+1}(s_{k+1}))-\rho_k)=\\ &h_k(s_k)+\alpha_k(R(s_k,\pi(s_k))+h_{k+1}(s_{k+1})-\\ &h_k(s_k)-\rho_k)\end{aligned}\quad(16)$$

平均奖励值更新与上式相似

$$\rho_{k+1}=\rho_k+\beta_k[R(s_k,\pi(s_k))+h_{k+1}(s_{k+1})-h_k(s_k)-\rho_k] \quad (17)$$

式(16)、(17)中α_k与β_k决定了状态值函数和平均回报值的当前和未来预估的权重。式$R(s_k,\pi(s_k))+h_{k+1}(s_{k+1})-h_k(s_k)-\rho_k$称为TD error (time slot error),其指导着学习过程,确定状态值函数和平均奖励的学习率。

算法初始化时随机进入一个状态s_k,则选择行为的概率服从均匀分布,为了选出奖励函数更大的行为,算法设定了偏好值,其更新公式为

$$p(s_k,a_k)=p(s_k,a_k)+\eta_k[R(s_k,\pi(s_k))+h_{k+1}(s_{k+1})-h_k(s_k)-\rho_k] \quad (18)$$

式中η_k决定了偏好值的学习率。算法中初始偏好值$p(s_k,a_k)=0,\forall s\in \boldsymbol{S},\forall a\in \boldsymbol{A}$,算法最初会统一选择每个动作的概率,随着迭代的进行,通过增加选择该特定动作的偏好值来确定导致相对状态值函数增加的行为的优先级。相反情况下,如果TD error为负时,通过降低其偏好值而受到惩罚,导致相对状态值函数减小。

完整的A3C算法如表1所示,由于A3C是异步多线程的,在此给出任意一个线程的算法流程。首先初始化全局网络参数,然后将全局网络同步到所有子线程Actor与Critic网络,子线程经过参数学习迭代后,将最优的学习效果同步到全局网络,同时,更新所有子线程的学习参数为当前最优全局参数。

表1 A3C算法

Tab. 1 A3C algorithm

算法1:Asynchronous Advantage Actor-critic(A3C)
初始化参数:$\alpha_k,\beta_k,\eta_k,\varepsilon,h(s_k)=0,\rho_k=0$,偏好值$p(s,a)=0,\forall s\in \boldsymbol{S},\forall a\in \boldsymbol{A}$,全局共享的迭代轮数$T$,全局最大的迭代次数$T_{\max}$,本线程最大长度$T_{\text{local}}$
1. 初始化全局网络参数
2. 从公共部分的神经网络同步参数到本线程
3. for $k=0,1,2,\cdots$
4. 初始化状态S_k,更新缓存区内任务量n_b^i以及信道增益γ^i
5. 基于策略$\pi(s_k,a_k)$选出动作a_k,计算卸载任务量$C_{\text{off}}^i=b_n^i-C_{\text{loc}}^i$并选取调制等级$m^i$以及发射功率$p_{\text{tran}}^i$
6. 根据式(2)计算本地计算总能耗,根据式(3)计算卸载能耗,根据式(4)~(7)计算任务成功分组传输概率
7. 执行动作a_k,根据式(12)得到的奖励$r=R(s_k,a_k)$并更新状态S_{k+1}
8. 计算TD error:$\delta=r+h_{k+1}(s_{k+1})-h_k(s_k)-\rho_k$
9. 更新相对状态值以及平均回报值:$h_{k+1}(s_{k+1})=h_k(s_k)+\alpha_k\delta,\rho_{k+1}=\rho_k+\beta_k\delta$
10. 更新actor偏好值:$p(s_k,a_k)=p(s_k,a_k)+\eta_k\delta$
11. $k\leftarrow k+1,T\leftarrow T+1$
12. $T-T_{\text{start}}==T_{\text{local}}$则进入步骤13,否则continue
13. 将获得参数与全局参数对比,如全局平均能效值优于本线程学习值,则回到步骤2,否则进入步骤14
14. End for
15. 更新全局神经网络的参数模型
16. 如果$T>T_{\max}$,则算法结束,否则进入步骤2

Actor根据概率选择决策,条件概率越大的状态行为被选择到的可能性就越大。算法最初开始时每个动作被选中的概率均等,Actor在初始阶段可能选择任何可用的动作,这也称作探索阶段,与所有的强化学习算法相似,A3C算法在学习过程也需要经过平衡探索和利用的步骤。利用阶段的意义在于搜索平均奖励最大化的决策,而探索步骤的意义在于尝试所有可能的最佳决策,避免陷入局部最优解。

6 仿真结果与分析

6.1 参数设置

本次实验使用的电脑主频为3 GHz,内存为8 GB,处理器为Intel(R) CORE(TM) i5-8500(四核),故仿真设置子线程数为4,电脑的操作系统为Windows 10,使用的仿真平台为Matlab R2019a以及Python 3.7.9版本,神经网络模型为使用Python中的torch库搭建的网络模型,模型中参数如表2所

示。本次实验将本文所提算法与Dyna算法、DROO算法、AC算法进行对比。在对比算法中，系统参数与本文算法采用相同设置，采用自适应调制方式以及固定发射功率p_t = 0.8 Watt。

表2　神经网络参数设置

Tab. 2　Neural network parameter setting

参数	参数值
隐藏层个数	3
神经网络大小	256 × 128 × 64
激活函数	Tanh
优化器	Adam
学习率	0.01
批量经验样本大小	64
回放记忆单元大小	200
目标值网络更新频率	100

本节中使用表3中所设的参数来构建仿真，仿真过程中状态集为$\boldsymbol{S}$=[$(n_b^1, n_b^2, n_b^3, n_b^4)\times(\gamma_i^1, \gamma_i^2, \gamma_i^3, \gamma_i^4)$]，行为集为$\boldsymbol{A}$ = [$(b_n^1, b_n^2, b_n^3)\times(m^1, m^2, m^3)\times(p_{\text{tran}}^1, p_{\text{tran}}^2, p_{\text{tran}}^3)$]，A3C算法初始化参数值$\alpha_k$ = 0.05，β_k = 0.005，η_k = 0.01，ε = 0.6，计算1比特数据需要的CPU周期数$C_i \in$ [500, 1500] cycle/bit，计算每个CPU周期的能量消耗$P_i \in (0, 20\times10^{-11})$ J/cycle，二者均服从均匀分布，节点计算能力$F_i \in \{0.1, 0.2, \cdots, 1\}$GHz。将最佳解决方案与A3C算法学习的策略作比较。一个任务所包含数据的信息比特为L_b=80 bit，添加错误解码代码后的任务数据信息比特为L=100 bit。将强化学习策略与最佳策略进行比较，证实了学习策略接近最优策略。将学习策略与简单策略也进行了简单比较，可以看出强化学习算法显著提高了平均任务处理量能效。将本文任务处理机制与其他文献中的任务处理机制相比较，本文提出的任务处理机制得到的平均能效明显更高。

表3　仿真参数

Tab. 3　Simulation parameters

参数	参数值
信道带宽	W = 10 MHz
热噪声方差	$\sigma^2 = 5\times10^{-15}$ W
最大多普勒频率	f_D = 50 Hz
路径损耗	$A_t^i = 0.097/(d^i)^4$
传输功率	p_t = [0,0.4,0.8] Watt
本地计算时间约束	T = 100 ms
信道增益	$\gamma \in$ = [−2,0,2,4] dB
节点距服务器距离	d = [320,460,570] m
缓存处理成本	$f(n_b) = 0.05(n_b + 4)$ if $n_b \neq \max(n_b)$
SIR范围	Γ = [0,1,⋯,10] dB

6.2　实验结果分析

图4表示了多节点情况下A3C算法在平均任务到达速率λ = 2.0时学习到的每消耗1毫焦耳能量平均处理的任务数，简称为节点学习值，A3C算法能跟踪控制概率的变化，从而获得接近最优策略的能力。

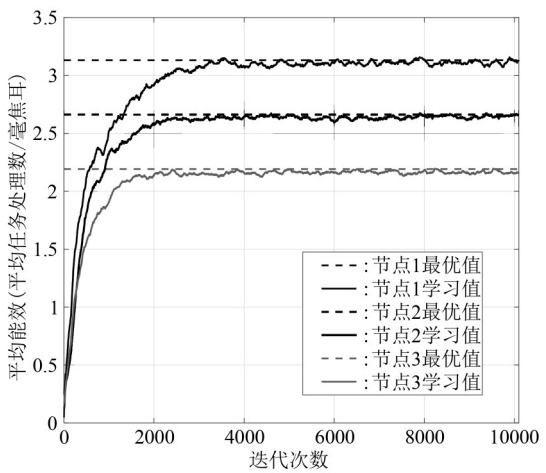

图4　每个节点学习值与最优值对比图

Fig. 4　Comparison diagram of learning value and optimal value of each node

在图4中可以看出，学习到的毫每焦耳平均任务处理数非常接近最佳每毫焦耳平均卸载任务数。评估了多节点方案中独立A3C算法的性能，用3个节点与一个边缘服务器通信来模拟多节点系统，节点1是最近节点，节点3是最远节点，在此方案中，靠近边缘服务器的节点将具有更高的任务处理能效，因为它实现相同的任务处理量所需要的能量较少。

图5表示了在不同的任务到达速率λ的情况下，A3C学习策略与简单策略分别所实现每消耗1毫焦耳能量平均卸载的任务数。运用简单策略所得到的平均能效值称之为简单值。在简单策略中，代理选择最高可能的调制方式同时选择出给定调制方式的情况下达到预定义信号干扰比(SIR)的发射功率。应用简单策略情况下，当缓冲器中只有

1个任务时，发射机选择二进制相移键控（BPSK）进行发送，当队列中分别有2个任务和多于3个任务时，发射机选择正交相移键控（QPSK）和8PSK进行发送。对于每个调制，发射机选择发射功率以实现固定的预定义SIR。对于BPSK到8PSK，分别使用（6，10，15）dB作为预定义的链路信噪比。由于服务器端接收任务数据的概率并非100%，传感器端未收到服务器端返回的ACK信令，则需要重新发送任务数据，如果任务成功接收概率过低，重发次数过多，消耗能量越大，故需要满足特定任务成功接收概率，而预定义的SIR的任务分组成功正确接收概率可以达到80%以上。

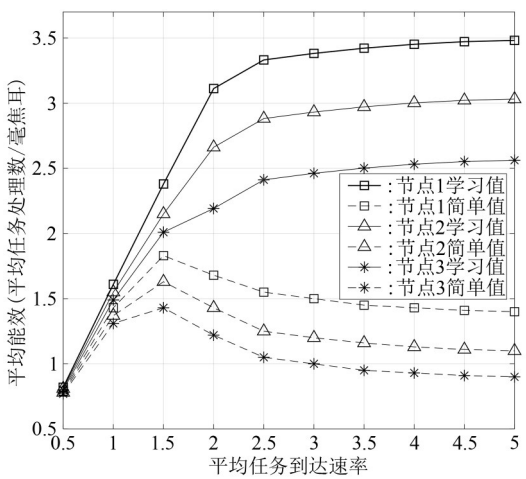

图5 不同任务到达速率下各节点平均能效对比图

Fig. 5 Comparison chart of average energy efficiency of each node under different task arrival rates

从图5中可以看出，在任务到达速率低时（$\lambda \leqslant 1$），两种策略的所达到的平均任务处理能效差距不大，但随着任务到达率提升，A3C算法相较于简单策略所提升的平均能效显而易见。

图6表示了在不同任务到达率情况下，采用A3C学习策略、Dyna算法框架、DROO算法框架以及简单策略所达到的平均能效值。

从图6中可以看出，在相同的平均任务到达速率情况下，A3C算法学习到的值非常接近最优策略值，并优于DROO、Dyna框架所学习到的值，并在高任务到达速率时，每消耗总能量能达到2~3倍简单策略所能达到的任务处理数量。

因此，所提任务处理机制具有更高的能量效率。最优策略由于需要了解信道转换概率和数据包到达率，在实际应用中可能不可行，A3C算法不需要获得相应转换概率却仍然能获得近似最优的任务处理量。

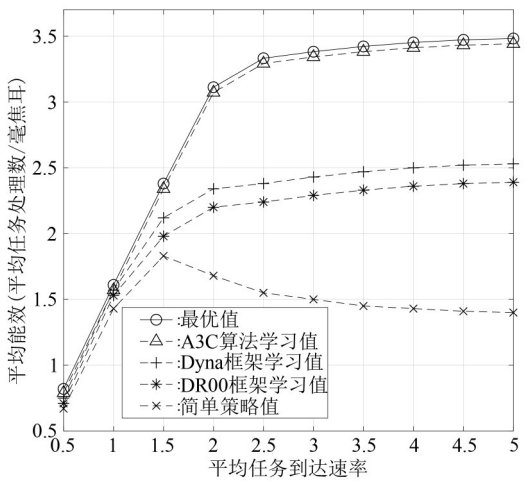

图6 不同任务到达率下各算法平均能效对比图

Fig. 6 Comparison of average energy efficiency of the next algorithm with different task arrival rates

图7表示了相较于传统算法，A3C算法带来的收敛速度上的提升。为了方便比较，在传统算法中也采用本文的任务处理机制，便于观察在能达到相同的任务处理平均能效的情况下的收敛速率。

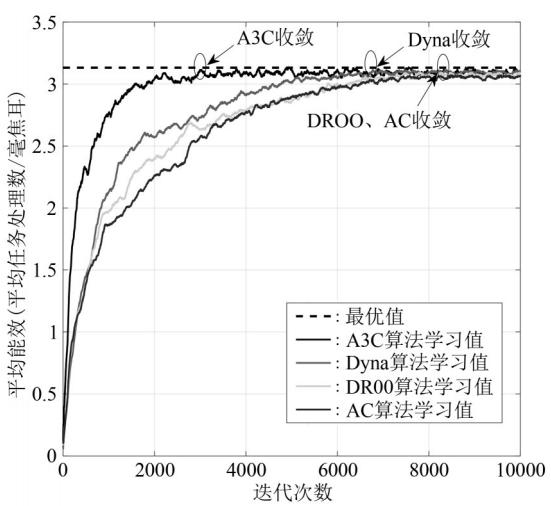

图7 各算法收敛速度对比图

Fig. 7 Comparison of convergence speed of each algorithm

考虑单节点的情况下AC算法、DROO算法、Dyna算法以及A3C算法收敛速度，从图中可以看出，在相同的任务处理数量以及相同的参数设置下，A3C学习到的平均能效在迭代了3000次后开始收敛，而Dyna算法学习到的平均能效收敛在7000次

左右，而 DROO 以及 AC 算法则需要接近 8200 次迭代才能收敛，从图中可以看出，A3C 算法的收敛速度相较于传统算法得到了明显的提升。各算法达到行为收敛的时间如表 4 所示。

表 4　各算法仿真耗时

Tab. 4　The simulation of each algorithm takes time

算法	耗时/s
A3C 算法	7341.86
Dyna 算法	18664.54
DROO 算法	21355.33
AC 算法	21732.02

从表 4 中亦能看出，A3C 算法收敛时间快于其他算法。A3C 算法虽然提升了学习收敛速率，但在计算方面也更加复杂，A3C 算法计算复杂度为 $O(N)$，子线程越多，收敛速度越快，但复杂度也会越大，应根据实际情况调整子线程的数量，以满足用户需求。

7　结论

本文将无线传感器网络中任务处理的能效问题建模为 MDP，从物理层以及数据链路层参数最佳控制角度着眼于无线传感器将任务本地计算以及卸载到边缘进行计算过程，保证任务得到有效处理的同时近似最大化系统能效，用强化学习方法进行求解。文章比较了本文方案与其他方案的能量效率，有明显的提升。在未来的工作中，将考虑把多节点系统中节点间的独立学习扩展为联合学习以及无线传感器的能量供应情况。

参考文献

[1] PANDANA C, LIU K J R. Near-optimal reinforcement learning framework for energy-aware sensor communications[J]. IEEE Journal on Selected Areas in Communications, 2005, 23(4): 788-797.

[2] AZARHAVA H, MUSEVI NIYA J. Energy efficient resource allocation in wireless energy harvesting sensor networks[J]. IEEE Wireless Communications Letters, 2020, 9(7): 1000-1003.

[3] ZHOU Fuhui, WU Yongpeng, HU R Q, et al. Computation rate maximization in UAV-enabled wireless-powered mobile-edge computing systems[J]. IEEE Journal on Selected Areas in Communications, 2018, 36(9): 1927-1941.

[4] AKBAS A, YILDIZ H U, TAVLI B, et al. Joint optimization of transmission power level and packet size for WSN lifetime maximization[J]. IEEE Sensors Journal, 2016, 16(12):5084-5094.

[5] ZHANG Yang, GAO Hong, CHENG Siyao, et al. An efficient EH-WSN energy management mechanism[J]. Tsinghua Science and Technology, 2018, 23(4):406-418.

[6] SHARMA P K, JEONG Y S, PARK J H. EH-HL: effective communication model by integrated EH-WSN and hybrid LiFi/WiFi for IoT[J]. IEEE Internet of Things Journal, 2018, 5(3):1719-1726.

[7] WANG Feng, XU Jie, CUI Shuguang. Optimal energy allocation and task offloading policy for wireless powered mobile edge computing systems[J]. IEEE Transactions on Wireless Communications, 2020, 19(4): 2443-2459.

[8] CLERCKX B, ZHANG Rui, SCHOBER R, et al. Fundamentals of wireless information and power transfer: From RF energy harvester models to signal and system designs[J]. IEEE Journal on Selected Areas in Communications, 2019, 37(1): 4-33.

[9] PAN Jianli, MCELHANNON J. Future edge cloud and edge computing for Internet of Things applications[J]. IEEE Internet of Things Journal, 2018, 5(1): 439-449.

[10] BALDOVINO R G, VALENZUELA I C, DADIOS E P. Implementation of a low-power wireless sensor network for smart farm applications[C]//2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM). Baguio City, Philippines. IEEE, 2018: 1-5.

[11] YOU Changsheng, HUANG Kaibin, CHAE H, et al. Energy-efficient resource allocation for mobile-edge computation offloading[J]. IEEE Transactions on Wireless Communications, 2017, 16(3): 1397-1411.

[12] YANG Yingjie, CHEN Xin, CHEN Ying, et al. Green-oriented offloading and resource allocation by reinforcement learning in MEC[C]//2019 IEEE International Conference on Smart Internet of Things (SmartIoT). Tianjin, China. IEEE, 2019: 378-382.

[13] CAO Xiaowen, WANG Feng, XU Jie, et al. Joint computation and communication cooperation for energy-efficient mobile edge computing[J]. IEEE Internet of Things Journal, 2019, 6(3): 4188-4200.

[14] WANG Feng, XING Hong, XU Jie. Real-time resource allocation for wireless powered multiuser mobile edge

computing with energy and task causality [J]. IEEE Transactions on Communications, 2020, 68(11): 7140-7155.

[15] CHEN Xianfu, ZHANG Honggang, WU C, et al. Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning [J]. IEEE Internet of Things Journal, 2019, 6(3): 4005-4018.

[16] 朱江,徐斌阳,李少谦.一种基于马尔可夫决策过程的认知无线电网络传输调度方案[J].电子与信息学报,2009,31(8): 2019-2023.
ZHU Jiang, XU Binyang, LI Shaoqian. A transmission and scheduling scheme based on Markov decision process in cognitive radio networks[J]. Journal of Electronics & Information Technology, 2009, 31(8): 2019-2023.(in Chinese)

[17] LIAO Yangzhe, QIAO Xinhui, SHOU Liqing, et al. Caching-aided task offloading scheme for wireless body area networks with MEC[C]//2019 NASA/ESA Conference on Adaptive Hardware and Systems (AHS). Colchester, UK. IEEE, 2019: 49-54.

[18] 柏果,程郁凡,唐万斌.基于深度学习的单载波频域均衡算法研究[J].信号处理,2021,37(6): 922-931.
BAI Guo, CHENG Yufan, TANG Wanbin. Research on single-carrier frequency-domain equalization algorithm based on deep learning [J]. Journal of Signal Processing, 2021, 37(6): 922-931. (in Chinese)

[19] MIN Minghui, WAN Xiaoyue, XIAO Liang, et al. Learning-based privacy-aware offloading for healthcare IoT with energy harvesting[J]. IEEE Internet of Things Journal, 2019, 6(3): 4307-4316.

[20] MIN Minghui, XIAO Liang, CHEN Ye, et al. Learning-based computation offloading for IoT devices with energy harvesting[J]. IEEE Transactions on Vehicular Technology, 2019, 68(2): 1930-1941.

[21] HUANG Liang, BI Suzhi, ZHANG Y J A. Deep reinforcement learning for online computation offloading in wireless powered mobile-edge computing networks [J]. IEEE Transactions on Mobile Computing, 2020, 19(11): 2581-2593.

[22] YE Dayong, ZHANG Minjie. A self-adaptive sleep/wake-up scheduling approach for wireless sensor networks[J]. IEEE Transactions on Cybernetics, 2018, 48(3): 979-992.

[23] RAZAVILAR J, LIU K J Ray, MARCUS S I. Jointly optimized bit-rate/delay control policy for wireless packet networks with fading channels [J]. IEEE Transactions on Communications, 2002, 50(3):484-494.

[24] WANG Shenhong, MOAYERI Nader. Finite-state Markov channel-a useful model for radio communication channels[J]. IEEE Transactions on Vehicular Technology, 1995, 44(1):163-171.

作者简介

张明杰 男,1995年生,重庆潼南人。重庆邮电大学硕士研究生,主要研究方向为无线传感器网络能效。
E-mail: 15823109239@163.com

朱 江 男,1977年生,湖北荆州人。重庆邮电大学教授,博士,主要研究方向为通信理论与技术、信息安全技术。
E-mail: zhujiang@cqupt.edu.cn