Supplementary Appendix

Talha Bozkus and Urbashi Mitra

I. Proof of Proposition 5

The following expressions are valid for all (s,a) pairs; hence, we drop the (s,a) notation for simplicity.

$$\lim_{t\to\infty}\mathbb{V}[\mathcal{E}_t] = \lim_{t\to\infty}\mathbb{V}\Big[(1-u)\sum_{i=0}^{t-1}u^{t-i-1}\sum_{n=1}^{K}\mathbf{w}_i^{(n)}\mathcal{X}_i^{(n)}\Big]. \tag{1}$$

$$= \lim_{t\to\infty}(1-u)^2\Big[\sum_{i=0}^{t-1}u^{2(t-i-1)}\mathbb{V}\big[\sum_{n=1}^{K}\mathbf{w}_i^{(n)}\mathcal{X}_i^{(n)}\big] + 2\sum_{i=0}^{t-1}\sum_{j=i+1}^{t-1}u^{2(t-i-1)}u^{2(t-j-1)}Cov\big[\sum_{n=1}^{K}\mathbf{w}_i^{(n)}\mathcal{X}_i^{(n)},\sum_{n=1}^{K}\mathbf{w}_j^{(n)}\mathcal{X}_j^{(n)}\big]\Big]. \tag{2}$$

$$\le \lim_{t\to\infty}(1-u)^2\Big[\sum_{i=0}^{t-1}u^{2(t-i-1)}\mathbb{V}\big[\sum_{n=1}^{K}\mathbf{w}_i^{(n)}\mathcal{X}_i^{(n)}\big] + 2\sum_{i=0}^{t-1}\sum_{j=i+1}^{t-1}u^{2(t-i-1)}u^{2(t-j-1)}\sqrt{\mathbb{V}\big[\sum_{n=1}^{K}\mathbf{w}_i^{(n)}\mathcal{X}_i^{(n)}\big]\mathbb{V}\big[\sum_{n=1}^{K}\mathbf{w}_j^{(n)}\mathcal{X}_j^{(n)}\big]}\Big]. \tag{3}$$

$$\le \lim_{t\to\infty}(1-u)^2\Big[\sum_{i=0}^{t-1}u^{2(t-i-1)} + 2\sum_{i=0}^{t-1}\sum_{j=i+1}^{t-1}u^{2(t-i-1)}u^{2(t-j-1)}\Big]\mathbb{V}\big[\sum_{n=1}^{K}\mathbf{w}_i^{(n)}\mathcal{X}_i^{(n)}\big]. \tag{4}$$

$$\le \lim_{t\to\infty}(1-u)^2\Big[\sum_{i=0}^{t-1}u^{2(t-i-1)} + 2\sum_{i=0}^{t-1}\sum_{j=0}^{t-1}u^{2(t-i-1)}u^{2(t-j-1)}\Big]\lambda^2. \tag{5}$$

$$\le \frac{(1-u)}{(1+u)}\lambda^2 + \frac{2\lambda^2}{(1+u)^2}, \tag{6}$$

where (1) follows from (19), (2) follows from the properties of the variance operator, (3) follows from the Cauchy-Schwarz inequality for the variance operator, (4) follows from the fact that variance is independent of time indices from (7), (5) follows from (24), and (6) follows from the infinite geometric sum formula.

II. The details of the wireless networks

A. MISO network with interference channels

We consider the first wireless network model in [1] with T transmitters, a single receiver, and time-slotted communication. The action space of each transmitter is $\mathcal{A} = \{0,1\}$, where 0 is the "silent" action and 1 is the "transmit" action. Each transmitter has a data buffer with size $N-1$ with $N>1$. At the beginning of each slot, data packets arrive at transmitters following i.i.d. Bernoulli distributions with the arrival probability p. If a new packet arrives and the buffer is full, the packet is dropped. Let $\mathbf{B}$ be the buffer probability transition tensor (PTT), whose elements satisfy the following:

$$\mathbf{B}_{i,j,k} = p(b_{t+1}=j|b_t=i,a_t=k), \tag{7}$$

where $b_t \in \{0,1,\ldots,N-1\}$ denotes the buffer state, and a_t is the action taken at time t. The PTT $\mathbf{B}$ is explicitly given by the following:

$$\begin{aligned}\mathbf{B}_{i,i,0} &= 1-p,\ \ 0\le i<N-1,\\ \mathbf{B}_{i,i+1,0} &= p,\ \ 0\le i<N-1,\\ \mathbf{B}_{N-1,N-1,0} &= 1,\\ \mathbf{B}_{0,0,1} &= 1,\\ \mathbf{B}_{i,i,1} &= p,\ \ 0<i\le N-1,\\ \mathbf{B}_{i,i-1,1} &= 1-p,\ \ 0<i\le N-1.\end{aligned} \tag{8}$$

If a packet is transmitted by a transmitter, it may collide with packets being simultaneously transmitted by other transmitters. We employ a generalized Gilbert-Elliot channel model with M states. Let h_t denote the channel state at time t, and $p_{i,j}$ denote the transition probability from channel state i to j, which follows a discretized exponential distribution with a parameter μ, given by the following:

$$\begin{aligned} p_{c,c+1} &= \mu e^{-\mu(c+1)}, \;\; 1 \le c \le M-1, \\ p_{c,c-1} &= \mu e^{-\mu c}, \;\; 2 \le c \le M, \\ p_{c,c} &= \begin{cases} 1-\mu e^{-\mu(c+1)}, \;\; c=1, \\ 1-\mu e^{-\mu c} - \mu e^{-\mu(c+1)}, \;\; 2 \le c \le M-1, \\ 1-\mu e^{-\mu c}, \;\; c=M, \end{cases} \end{aligned} \tag{9}$$

where $c \in \{1, \ldots, M\}$ is the channel index. The channel probability transition matrix (PTM) $\mathbf{C}$ is defined as $\mathbf{C}_{i,j} = p_{i,j}$.

The state of the system at time t is given by a tuple $s_t = (b_t, h_t)$. The PTT of the overall system $\mathbf{P}$ is defined as $\mathbf{P} = \mathbf{B} \otimes \mathbf{C}$ where $\otimes$ is the Kronecker product operator. In a multiple-access network with T transmitters and channels, we use the joint PTT defined as $\mathbf{P}_T = \underbrace{\mathbf{P} \otimes \mathbf{P} ... \otimes \mathbf{P}}_{T \text{ times}}$.

Let $\{s_{t,i}\}_{i=1}^T = \{(b_{t,i}, h_{t,i})\}_{i=1}^T$ be the set of states, where the subscript i denotes the i^{th} transmitter. Given the set of states $\{s_{t,i}\}_{i=1}^T$ and the set of actions $\{a_{t,i}\}_{i=1}^T$, the single-stage cost function at time t is defined as:

$$\mathbf{c}(\{s_{t,i}\}_{i=1}^T, \{a_{t,i}\}_{i=1}^T) = \sum_{i=1}^{T} \left[\alpha_1 \frac{b_{t,i}}{N-1} \mathbf{1}\{a_{t,i}=0\} + \alpha_2 f(h_{t,i}) \mathbf{1}\{a_{t,i}=1\} + \alpha_3 \mathbf{1}\{a_{t,i}=1\} \sum_{j=1}^{T} \mathbf{1}\{a_{t,j}=1\} \frac{1}{\tilde{d}_{ij}^2} \right], \tag{10}$$

where $\mathbf{1}(\cdot)$ is the indicator function and $f(h_{t,i})$ is given by:

$$f(h_{t,i}) = \frac{P_{rec}}{c_0 d_i^{-\sigma} h_{t,i}}, \tag{11}$$

which denotes the power threshold at the transmitter for successful transmission as a function of channel gain $h_{t,i}$ for TX$_i$ at time t, P_{rec} is the power threshold at the receiver, d_i is the distance between the receiver and TX$_i$, $\tilde{d}_{ij}$ is the distance between TX$_i$ and TX$_j$, σ is the path loss component, and c_0 is a constant.

The first component is **buffer cost**, and proportional to how full the buffer is, and incurs when the transmitter remains silent for a long time. The second component is **transmission cost** as a function of the channel conditions, and incurs when the transmitters transmit under unfavorable channel conditions. The third component is **collision cost**, which is inversely proportional to the distance between transmitters transmitting simultaneously. The parameters $\alpha_1, \alpha_2, \alpha_3$ are the weights.

The numerical parameters are given in Table I, and kept constant for different network sizes. The network size is formed by changing N, M and T as follows: the interval [0, 5000] is formed by increasing N, M comparably with $T = 1$, the interval [5000, 10000] is formed similarly with $T = 2$, the interval [10000, 15000] is formed similarly with $T = 3$, the interval [15000, 20000] is formed similarly with $T = 4$, and the interval [20000, 40000] is formed similarly with $T = 5$.

TABLE I: Numerical parameters for MISO network with interference channels

Symbol	Range/Value	Description
T	$2, 3, 4, 5$	Number of transmitters
N	$2, 3, \ldots, 30$	Buffer size
M	$2, 3, \ldots, 10$	Number of states in the channel Markov chain
p	0.9	Arrival probability of data packets at transmitters
μ	1	Mean of the discretized exponential distribution
c_0	$10^{0.17}$	Constant in the power threshold
σ	4.7	Path loss component in the power threshold
P_{rec}	-97 dBm	Power threshold at the receiver
d_i	100	Distance between the receiver and TX$_i$
$\tilde{d}_{ij}$	$50\lvert i-j \rvert$	Distance between TX$_i$ and TX$_j$
α_1	0.4	Weight for the buffer cost
α_2	0.4	Weight for the transmission cost
α_3	0.3	Weight for the collision cost

B. MISO energy harvesting network with multiple relays

We consider the second wireless network model in [1] with 3 transmitters, a single receiver, and 2 relays. Let $h_{t,ik}$ be the channel gain between TX$_i$ and RL$_k$ (k^{th} relay) at time t, $h_{t,Rk}$ be the channel gain between the receiver and RL$_k$ at time t, and $h_{t,iR}$ be the channel gain between TX$_i$ and receiver at time t for $i = 1, 2, 3$ and $k = 1, 2$. Let $x_{t,1}, x_{t,2}, x_{t,3}$ be the inputs to the transmitters at time t, the receiver output at time t is Y_t. The system is governed by the following equations:

$$Y_{t,RL_1} = h_{t,11}x_{t,1}\mathbf{1}\{a_{t,1} = 1\} + h_{t,21}x_{t,2}\mathbf{1}\{a_{t,2} = 1\} + h_{t,31}x_{t,3}\mathbf{1}\{a_{t,3} = 1\} + N(0, \sigma^2). \tag{12}$$

$$Y_{t,RL_2} = h_{t,12}x_{t,1}\mathbf{1}\{a_{t,1} = 2\} + h_{t,22}x_{t,2}\mathbf{1}\{a_{t,2} = 2\} + h_{t,32}x_{t,3}\mathbf{1}\{a_{t,3} = 2\} + N(0, \sigma^2). \tag{13}$$

$$Y_t = h_{t,R1}Y_{t,RL_1} + h_{t,R2}Y_{t,RL_2} + h_{t,1R}x_{t,1}\mathbf{1}\{a_{t,1} = 0\} + h_{t,2R}x_{t,2}\mathbf{1}\{a_{t,2} = 0\} + h_{t,3R}x_{t,3}\mathbf{1}\{a_{t,3} = 0\}. \tag{14}$$

The individual state of TX$_i$ at time t is defined as a tuple $s_{t,i} = (b_{t,i}, h_{t,i})$, where $b_{t,i}$ is the battery state and $h_{t,i}$ is the channel state with $b_{t,i} \in \{0, 1, ..., N-1\}$ $\forall i$. Let $\mathbf{B}$ be the battery PTT of each transmitter storing the probability of transitioning between battery states under different actions. The structure of the battery PTT is the same as the buffer PTT in the previous network model, except that there are now three actions. The joint battery PTT $\bar{\mathbf{B}}$ is obtained as $\bar{\mathbf{B}} = \mathbf{B} \otimes \mathbf{B} \otimes \mathbf{B}$. The channel gains are characterized by the standard Gilbert-Elliot channel model with different probabilities as follows:

$$h_{t,11}, h_{t,12}, h_{t,21}, h_{t,22}, h_{t,31}, h_{t,32} \in \{0, 1\} \text{ with } (p_1, q_1),$$
$$h_{t,1R}, h_{t,2R}, h_{t,3R} \in \{0, 1\} \text{ with } (p_2, q_2),$$
$$h_{t,R1}, h_{t,R2} \in \{0, 1\} \text{ with } (p_3, q_3),$$

where p_1, p_2, p_3 are the probabilities of transitioning from state 0 (good) to 1 (bad), and q_1, q_2, q_3 are the probabilities of transitioning from state 1 (bad) to 0 (good). Different (p, q) can fully characterize the different channel gains. For example, the channel conditions between the transmitter and relay, as well as the relay and receiver, are likely to be better than that of a direct channel between the transmitter and receivers due to shorter distance and less interference. Hence, we can assume that $p_1 = p_3 \ll p_2$ and $q_2 \ll q_1 = q_3$ (the channel is more likely to be bad for the direct channel). We then construct three channel PTMs: $\mathbf{C}_1$, $\mathbf{C}_2$ and $\mathbf{C}_3$ for each three different distributions, and obtain the joint channel PTT $\bar{\mathbf{C}}$ as $\bar{\mathbf{C}} = \mathbf{C}_1 \otimes \mathbf{C}_2 \otimes \mathbf{C}_3$. The PTT of the overall system $\mathbf{P}$ is defined as $\mathbf{P} = \bar{\mathbf{B}} \otimes \bar{\mathbf{C}}$. Given the set of states $\{s_{t,i}\}_{i=1}^3$ and the set of actions $\{a_{t,i}\}_{i=1}^3$, the single-stage cost function at time t is defined as:

$$\mathbf{c}(\{s_{t,i}\}_{i=1}^3, \{a_{t,i}\}_{i=1}^3) = -\alpha_1 Y_t + \alpha_2 \sum_{i=1}^{3}\sum_{j=1}^{2} x_{t,i}\mathbf{1}\{a_{t,i} = j\} + \alpha_3 \sum_{i=1}^{3}(1 - \frac{b_{t,i}}{N}), \tag{15}$$

where the first term is **negative throughput**, the second term is **drop cost** that balances the load on the relays (if multiple transmitters choose to use the same path through relay-1 or relay-2, there may be performance degradation), and the third term is **total amount of battery consumed** (if $b_{t,i} = 0$, it means the battery is empty, and the corresponding cost is very large), with $\alpha_1, \alpha_2, \alpha_3$ being the weights.

The numerical parameters are given in Table II, and kept constant across different network sizes. The network size is formed as follows: the interval [0, 5000] is formed by increasing N with 3 transmitters and 1 relay, [5000, 10000] is formed by increasing N with 3 transmitters and 2 relays, [10000, 15000] is formed by increasing N with 4 transmitters and 2 relays, [15000, 20000] is formed by increasing N with 4 transmitters and 3 relays, [20000, 30000] is formed by increasing N with 5 transmitters and 2 relays, and [30000, 40000] is formed by increasing N with 5 transmitters and 3 relays.

TABLE II: Numerical parameters for MISO energy harvesting network with multiple relays

Symbol	Range/Value	Description
T	$3, 4, 5$	Number of transmitters
R	$1, 2, 3$	Number of relays
N	$2, 3, \ldots, 30$	Buffer size
σ^2	1	Variance of the noise
$x_{t,1}$, $x_{t,2}$, $x_{t,3}$	1	Inputs to the transmitters
p_1, q_2, p_3	0.2	Transition probabilities for channel gains
q_1, q_3, p_2	0.8	Transition probabilities for channel gains
α_1	0.2	Weight for negative throughput
α_2	0.3	Weight for drop cost
α_3	0.5	Weight for battery consumption cost

C. MIMO network with interference channels

We generalize the MISO wireless network model to the multiple receivers case. Assume there are T transmitters and R receivers. The channels between transmitters and receivers are modeled with a standard Gilbert-Elliot model, and the channels between closer TX-RX pairs are more likely to be in a good state than those between farther TX-RX pairs due to shorter distance and less interference. For a network with $T = 3$ transmitters and $R = 4$ receivers, the channels between TX$_1$ and RX$_1$, TX$_1$ and RX$_2$, TX$_2$ and RX$_2$, TX$_2$ and RX$_3$, TX$_3$ and RX$_3$, TX$_3$ and RX$_4$ have the probability of transitioning from good to bad state $p_1 = 0.2$ and the probability of transitioning from bad to good state $1-p_1 = 0.8$. The other channels have the probability of transitioning from good to bad state $p_2 = 0.4$ and the probability of transitioning from bad to good state $1 - p_2 = 0.6$. The channel PTM between TX$_i$ and RX$_j$ is denoted by $\mathbf{C}_{ij}$. The overall channel PTM $\mathbf{C}$ is formed by taking the Kronecker product between $\mathbf{C}_{ij}$ for $i \in \{1, 2, 3\}$ and $j \in \{1, 2, 3, 4\}$.

There are $R+1$ different actions each transmitter can take: remain silent or transmit j packets to j different receivers for $j = 1, 2, 3, ..., R$. The buffer PTT $\mathbf{B}$ of a generic transmitter with $T = 3$ and $R = 4$ is defined as:

$$\begin{aligned}
\mathbf{B}_{i,i,0} &= 1-p, \;\; 0 \le i < N-1,\\
\mathbf{B}_{i,i+1,0} &= p, \;\; 0 \le i < N-1,\\
\mathbf{B}_{N-1,N-1,0} &= 1,\\
\mathbf{B}_{0,0,1} &= 1,\\
\mathbf{B}_{i,i,1} &= p, \;\; 0 < i \le N-1,\\
\mathbf{B}_{i,i-1,1} &= 1-p, \;\; 0 < i \le N-1.\\
\mathbf{B}_{i,0,2} &= 1, \;\; 0 \le i \le 1,\\
\mathbf{B}_{i+1,i,2} &= p, \;\; 1 \le i \le N-2,\\
\mathbf{B}_{i+2,i,2} &= 1-p, \;\; 0 \le i \le N-3.\\
\mathbf{B}_{i,0,3} &= 1, \;\; 0 \le i \le 2,\\
\mathbf{B}_{i+2,i,3} &= p, \;\; 1 \le i \le N-3,\\
\mathbf{B}_{i+3,i,3} &= 1-p, \;\; 0 \le i \le N-4.\\
\mathbf{B}_{i,0,4} &= 1, \;\; 0 \le i \le 3,\\
\mathbf{B}_{i+3,i,4} &= p, \;\; 1 \le i \le N-4,\\
\mathbf{B}_{i+4,i,4} &= 1-p, \;\; 0 \le i \le N-5.
\end{aligned} \tag{16}$$

The overall buffer PTT is obtained by taking the Kronecker product of $\mathbf{B}$ with itself T times. Let s_t be the state at time t defined as a tuple: $s_t = \{b_t, h_t\}$, where b_t and h_t are the buffer and channel state of the overall system defined for $T = 3$ and $R = 4$ as follows: $b_t = [b_{t,1}, b_{t,2}, b_{t,3}]$, where $b_{t,i} \in \{0, 1, ..., N-1\}$ is the buffer state of the TX$_i$ for $i = 1, 2, 3$, and $h_t = [h_{t,11}, h_{t,12}, h_{t,13}, h_{t,21}, h_{t,22}, h_{t,23}, h_{t,31}, h_{t,32}, h_{t,33}, h_{t,41}, h_{t,42}, h_{t,43}]$, where $h_{t,ij} \in \{0, 1, ..., M-1\}$ is the channel state of the channel between TX$_i$ and RX$_j$. Given the state s_t and the set of actions $\{a_{t,i}\}_{i=1}^{3}$, the single-stage cost function at time t is defined as:

$$\mathbf{c}(s_t, \{a_{t,i}\}_{i=1}^{3}) = \alpha_1 \sum_{i=1}^{3} \frac{b_{t,i}}{N-1} + \alpha_2 \sum_{i=1}^{3}\sum_{j=1}^{4} f(h_{t,ij}) a_{t,i} + \alpha_3 \sum_{i=1}^{3} \mathbf{1}\{a_{t,i}=1\} \sum_{j=1}^{3} \mathbf{1}\{a_{t,j}=1\} \frac{1}{\tilde{d}_{ij}^{2}} + \alpha_4 \sigma\{r_{t,1}, r_{t,2}, r_{t,3}, r_{t,4}\}, \tag{17}$$

where $\mathbf{1}(\cdot)$ is the indicator function and $f(h_{t,ij})$ is defined as:

$$f(h_{t,ij}) = \frac{P_{j,rec}}{c_0 d_{ij}^{-\sigma} h_{t,ij}}, \tag{18}$$

which denotes the power threshold at the transmitter for successful transmission as a function of channel gain $h_{t,ij}$ between TX$_i$ and RX$_j$ at time t, $P_{j,rec}$ is the power threshold at RX$_j$, d_{ij} is the distance between RX$_j$ and TX$_i$, σ is the path loss component, and c_0 is a constant, $r_{t,j}$ is the number of data packets that RX$_j$ receives at time t, and $\sigma\{x_1, ..., x_N\}$ is the sample standard deviation of the given list of numbers. The distance between TX$_i$ and TX$_j$ $\tilde{d}_{ij}$ is given by: $\tilde{d}_{ij} = 50|i-j|$.

The first three components are **buffer cost**, **transmission cost**, and **collision cost** as defined previously. The fourth component is the **receiver load cost**. It occurs when the receivers receive a relatively unequal number of data packets. For example, if $r_{t,1} = r_{t,2} = 3$, $r_{t,3} = r_{t,4} = 1$, then there will be an unbalanced load on the receivers ($\sigma\{3, 3, 1, 1\} = 1$) than the case where $r_{t,1} = r_{t,2} = r_{t,3} = r_{t,4} = 2$ ($\sigma\{2, 2, 2, 2\} = 0$). The parameters $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are the weights.

The numerical parameters are given in Table III, and kept constant across different network sizes. For settings with different $R = 4$ and $T = 3$, the following distances are used: $d_{11} = d_{12} = d_{22} = d_{23} = d_{33} = d_{34} = 100$ and $d_{13} = d_{14} = d_{21} = d_{24} = d_{31} = d_{32} = 200$. For the other cases, the closer TX-RX pairs have a distance of 100, while the others have a distance of 200. The network size is formed as follows: the interval [0, 5000] is formed by increasing N with 2 transmitters and 2 receivers, [5000, 10000] is formed by increasing N with 2 transmitters and 3 receivers, [10000, 15000] is formed by increasing N with 3 transmitters and 3 receivers, [15000, 20000] is formed by increasing N with 3 transmitters and 4 receivers, [20000, 30000] is formed by increasing N with 4 transmitters and 4 receivers, and [30000, 40000] is formed by increasing N with 4 transmitters and 5 receivers.

TABLE III: Numerical parameters for MIMO network with interference channels

Symbol	Range/Value	Description
T	$2, 3, 4$	Number of transmitters
R	$2, 3, 4, 5$	Number of receivers
N	$2, 3, \ldots, 30$	Buffer size
p	0.8	Arrival probability for incoming data packets
$P_{j,rec}$	$10^{-9.7}$	Power threshold at receiver j
c_0	$10^{0.17}$	Constant in the power threshold
σ	4.7	Path loss component in the power threshold
α_1	0.3	Weight for buffer cost
α_2	0.2	Weight for transmission cost
α_3	0.3	Weight for collision cost
α_4	0.2	Weight for receiver load cost
d_{ij}	100 or 200	Distance between TX$_i$ and RX$_j$

D. MIMO network with mobile transmitters

We generalize the MIMO network to the case where the transmitters are mobile. Assume the transmitters are constrained within L x L area, where L is an integer multiple of 10. The boundary coordinates are (0,0), (0,L), (L,0), (L,L). Each transmitter can be present in the coordinates that are multiples of 10s and move multiple units of 10s each time. Assume there are $T = 4$ transmitters and $R = 3$ receivers. Transmitters TX$_1$ and TX$_2$ are faster and can move either 10m or 20m each time or remain stationary, whereas the transmitters TX$_3$ and TX$_4$ are slower and can only move 10m each time or remain stationary. The three receivers are stationary, and their locations are (0,0), (0.6L,0.2L), and (0.2L,0.6L). The initial location of each transmitter is uniformly randomly assigned with the constraint that each has a different initial location and is not allowed to occupy the same location as receivers.

The state is defined by the explicit locations of each transmitter at time t as $s_t = [x_{t,1}, y_{t,1}, x_{t,2}, y_{t,2}, x_{t,3}, y_{t,3}, x_{t,4}, y_{t,4}]$, where $(x_{t,i}, y_{t,i})$ are the location of the i^{th} transmitter for $i = 1, 2, 3, 4$ and $x_{t,i}, y_{t,i} \in \{0, 10, ..., L\}$ at time t. The action space of each transmitter contains the indices of the receivers: $\mathcal{A} = \{1, 2, ..., R\}$. The PTT of the overall system is constructed by concatenating the probability that the transmitters move from the locations at time t (defined by s_t) to their next locations at time $t + 1$ (defined by s_{t+1}). Each transmitter has an equal probability of either moving in any direction 10m or 20m (for fast transmitters such as TX$_1$ and TX$_2$) or remaining stationary. For example, the following equation describes the probability of transitioning from s_t to s_{t+1} for TX$_1$:

$$p(s_{t+1} = [x_{t,1} + 20, y_{t,1}, x_{t,2}, y_{t,2}, x_{t,3} + 10, y_{t,3}, x_{t,4} - 10, y_{t,4}] \Big| s_t = [x_{t,1}, y_{t,1}, x_{t,2}, y_{t,2}, x_{t,3}, y_{t,3}, x_{t,4}, y_{t,4}]) = (\frac{1}{9})^8. \tag{19}$$

Given the state s_t and the set of actions $\{a_{t,i}\}_{i=1}^4$, the single-stage cost function at time t is defined as:

$$\mathbf{c}(s_t, \{a_{t,i}\}_{i=1}^4) = -\alpha_1 \sum_{i=1}^{4} log_2(1 + \frac{\sigma_i}{d_{t,ia_{t,i}}^2}) + \alpha_2 \sigma\{r_{t,1}, r_{t,2}, r_{t,3}\} + \alpha_3 \sum_{i,j:a_{t,i}=a_{t,j}} \frac{P_{i,tra} + P_{j,tra}}{\tilde{d}_{t,ij}^2} \tag{20}$$

where $d_{t,ia_{t,i}}$ is the distance between the TX$_i$ and RX$_{a_{t,i}}$ (the distance between TX$_i$ and the receiver it is transmitting to at time t), σ_i is the path loss coefficient, $r_{t,j}$ is the number of transmitters transmitting to RX$_j$ at time t, $P_{i,tra}$ is the power threshold at TX$_i$, and $\tilde{d}_{t,ij}$ is the distance between TX$_i$ and TX$_j$ at time t, and α_1, α_2 and α_3 are the weighting coefficients. The first component is the **negative throughput**, which is proportional to the squared distance between the pair of each connected transmitter and receiver, discouraging transmitters from connecting to the far receivers. The second component is the **receiver load cost**, which prevents the overload on receivers by encouraging transmitters to connect to different receivers. The third component is the **transmitter interference cost**, which is inversely proportional to the distance between transmitters transmitting to the same receiver.

The numerical parameters are given in Table IV and kept constant across different network sizes. A larger α reduces the effect of path loss, and we assume that the characteristics of the fast transmitters TX_1 and TX_2 allow them to handle the path loss easier than TX_3 and TX_4. The network size is formed as follows: the interval [0, 5000] is formed by increasing N with 2 transmitters, and 2 receivers [5000, 10000] is formed by increasing N with 2 transmitters and 3 receivers, [10000, 20000] is formed by increasing N with 3 transmitters and 3 receivers, [20000, 30000] is formed by increasing N with 3 transmitters and 4 receivers, and [30000, 40000] is formed by increasing N with 4 transmitters and 4 receivers with slowly increasing the size of the predefined area L.

TABLE IV: Numerical Parameters for MIMO network with mobile channels

Symbol	Range/Value	Description
L	Integer multiple of 10	Area size constraint for transmitters
T	$2, 3, 4$	Number of transmitters
R	$2, 3, 4$	Number of receivers
σ_1, σ_2	1	Path loss coefficient for TX_1 and TX_2
σ_3, σ_4	10	Path loss coefficient for TX_3 and TX_4
α_1	0.4	Weight for negative throughput cost
α_2	0.4	Weight for receiver load cost
α_3	0.3	Weight for transmitter interference cost
$P_{i,tra}$	$10^{-9.7}$	Transmitter power of TX_i
$\tilde{d}_{t,ij}$	$50\lvert i-j\rvert$	Distance between TX_i and TX_j at time t

E. Practicality of the wireless network models

The presented four wireless network models vary in their topology, scale, complexity, objective, and implementations and serve as representative examples that accurately depict the complexities and challenges of real-world wireless networks:

1) MISO network with interference channels: This network model employs a dynamic multi-state Gilbert-Elliot channel model, which captures the evolution of channel dynamics of real-world wireless networks. In addition, the cost function is designed to handle interference, addressing a critical aspect of real-world wireless communications.

2) MISO energy harvesting network with multiple relays: This network model offers flexibility by allowing the algorithm to choose between traditional (TX to RX) and relay-based optimization schemes. Similarly, the consideration of channel parameters favoring relay channels over direct channels aligns with practical scenarios, where relays can enhance communication reliability. On the other hand, integrating the additive white Gaussian noise in relay outputs enhances model practicality by accounting for realistic noise effects.

3) MIMO network with interference channels: This network model adds complexity to the previous MISO models by integrating multiple receivers. In addition, finite-sized data buffers reflect the practical limitations of real-world transmitters. On the other hand, integrating the receiver load cost component reflects a practical concern in optimizing MIMO networks, ensuring a balanced distribution of load among receivers.

4) MIMO network with mobile transmitters: In real-world scenarios, mobile transmitters like smartphones are prevalent. This network model accurately reflects this reality, capturing the practical constraint that transmitters operate within a defined area. Consequently, outside this area, there is no coverage—a common real-world limitation. Moreover, random movement with constant speeds and directions reflects the unpredictability of real-world mobile devices. On the other hand, the channel between transmitters and receivers follows a path-loss model, providing a realistic representation of signal attenuation over distance in wireless communication. Finally, the objective of the network is to determine the optimal association between transmitters and receivers, which reflects practical considerations in MIMO networks, where choosing the right associations can significantly impact overall system performance.

III. Parameter Tuning for Proposed Algorithm

We herein discuss how to choose the parameters v, l, K, α_t and ϵ_t for small ($|\mathcal{S}| \leq 10000$), modest ($|\mathcal{S}| \in [10000, 40000]$), and large networks ($|\mathcal{S}| \geq 40000$). Table V summarizes the parameters that yield near-optimal APE and can be used for different simulations. However, an optimal parameter selection requires cross-validation.

The number of visits (v) needed for each state-action pair is independent of network size, and near-optimal performance can be achieved with a small v value, around $v \approx 50$. Likewise, the proposed algorithm allows us to have short trajectories (l) while ensuring several key factors: (i) achieving near-optimal APE performance, (ii) minimizing runtime and computational

Params	Small networks	Modest-sized networks	Large networks
l	$l \in [5, 10]$	$l \in [10, 15]$	$l \in [15, 30]$
K	$K \in \{2, 3\}$	$K \in \{3, 4, 5\}$	$K \in \{5, 6, 7, 8\}$
α_t	$c_1 \in \{10^2, 5 \cdot 10^2\}$	$c_1 \in \{10^2, 10^3\}$	$c_1 \in \{10^3, 10^4\}$
ϵ_t	$c_2 \in \{0.9, 0.95\}$ $c_3 \in \{0.01, 0.1\}$	$c_2 \in \{0.95, 0.99\}$ $c_3 \in \{0.01, 0.05\}$	$c_2 \in \{0.99, 0.999\}$ $c_3 \in \{0.005, 0.01\}$

TABLE V: Near-optimal hyper-parameters of Algorithm 1

requirements per trajectory, (iii) capturing sufficient samples from neighboring states, (iv) preserving the significance of the initial state despite discounting in longer trajectories, and (v) avoiding redundant paths and loops that provide no new samples.

Increasing the number of Markovian environments (K) reduces the average runtime complexity of the algorithm [2]. However, there are several points to consider: (i) Emphasizing high-order node relationships may lead to the loss of low-order node relationships. (ii) $\mathbf{L}^{(n)}$ converges to a fixed tensor as n increases, rendering samples from corresponding SMEs redundant and potentially degrading performance. (iii) Memory requirements increase. Hence, K should be small enough to avoid these drawbacks while also increasing with the network size to prevent an increase in other parameters (specifically v, l) and reduce runtime complexity. On the other hand, the order n also needs to be optimized. To this end, the theoretical approach based on the partial ordering of Q-functions in our prior work [3] can be employed. We can also numerically optimize the order n using an efficient Bellman error search based on our prior work [1]. Please see the main paper for details.

The learning rate (α_t) must adhere to the convergence conditions of Q-learning [4] and have a suitable decay to adjust the learning speed. We assume the form $\alpha_t = \frac{1}{1+\frac{t}{c_1}}$, where $c_1 > 0$ determines the decay rate and should increase with $|\mathcal{S}|$, v, l, and K. On the other hand, the parameter ϵ_t is essential for balancing exploration and exploitation. We use the form $\epsilon_t = \max((c_2)^t, c_3)$, where $c_2 > 0$ adjusts the decay rate, and $0 < c_3 \ll 1$ determines the minimum exploration probability. As the system parameters increase, c_2 should also increase to ensure sufficient exploration. Furthermore, c_3 should be small and positive, allowing for exploration with a low probability when the policy is nearly converged.

In general, a small u encourages the algorithm to rely on the Q-functions of different Markovian environments, promoting extensive exploration of the state-action space. This not only accelerates learning in the early stages by promoting exploration and yielding rapid initial progress but also facilitates a better trade-off between exploration and exploitation. In contrast, a large u prioritizes the ensemble Q-function output, enabling the algorithm to leverage the knowledge acquired from prior experiences, thereby expediting the overall convergence process. The emphasis on the ensemble output also contributes to stability throughout the learning process, resulting in more consistent Q-functions. Although different choices are possible, we find that a constant update ratio $u_t = 0.5$ gives a favorable balance between performance and complexity, facilitating convergence of Algorithm 1 across different network sizes. Our proposed approach is easier to tune compared to the algorithms presented in our prior work, particularly regarding the parameter u_t, which does not require extensive fine-tuning.

IV. Details on the reproducibility of results and figures

We simulate our simulations in Python-3, running on MacBook Pro with Intel Core i7 2.5GHz CPU and 16GB RAM. The proposed algorithm, along with several others, is implemented using the Numpy library. For neural network-based algorithms like ADQN, the TensorFlow library is employed.

A. *Figure 4*

The PTT of the original environment $\mathbf{P}$ is constructed using sample averaging. We keep sampling and updating $\mathbf{P}$ till the estimation error defined below is less than 0.05.

$$\text{estimation error} = \frac{1}{|\mathcal{A}|} \sum_{a \in \mathcal{A}} \|\mathbf{P}_a - \hat{\mathbf{P}}_a\|_2. \tag{21}$$

We run five different Q-learning algorithms on five different environments using the parameters in Section IV-B. We terminate each Q-learning algorithm when the averaged Q-function updates defined below falls below 0.05.

$$\frac{1}{|\mathcal{S}||\mathcal{A}|}\sum_{s\in\mathcal{S},a\in\mathcal{A}}\mathbf{Q}_{t+1}(s,a)-\mathbf{Q}_t(s,a)<0.05 \tag{22}$$

B. *Figure 5 and 6*

Since each experiment is run multiple times for different network sizes, there is no single set of parameters that optimize the performance for each different network size. Hence, we herein provide the range of each parameter for small ($|\mathcal{S}|\leq 10000$), modest ($|\mathcal{S}|\in[10000,40000]$), and large networks ($|\mathcal{S}|\geq 40000$), and then we do cross-validation to pick the best parameters.

- For ESQL, the parameters $v=50,\gamma=0.99,\lambda=1,u_t=0.5$ are fixed across different network sizes. The other parameters K,l,α_t,ϵ_t are chosen from Table V.
- For MMQ, the following parameters are used: $\epsilon=0.01$ and α is chosen from {0.005, 0.01, 0.02, 0.04, 0.08}. The buffer size is 100 for small networks, 500 for modest-sized networks, and 1000 for large networks. The number of action-value functions N is chosen from {1,2,3} for small networks, {3,4,5,6} for modest-sized networks, {5,6,7,8} for large networks.
- For SQ, the following parameters are used: $\gamma=0.9$ and $T=10^3$. They are kept constant for different network sizes.
- For DQ, the learning rate depends on the number of visits to state-action pair (s,a) as $\alpha_t(s,a)=\frac{1}{n_t(s,a)}$ and $\gamma=0.95$. They are kept constant for different network sizes.
- For EBQ, the following parameters are used: $\alpha\left(s_t,a_t\right)=1/n_t\left(s_t,a_t\right)^{0.8},\gamma=0.99$, and kept constant for different network sizes. The ensemble size K is chosen from {2,3,4} for small networks, {4,5,6,7} for modest-sized networks and {6,7,8,9,10} for large networks.
- For ADQN, the number of estimators is chosen from $\{2,3,4\}$ for small networks, $\{3,4,5,6\}$ for modest-sized networks and $\{5,6,7,8,9,10\}$ for large networks. The following parameters are selected via cross-validation from the following sets: batch size: $\{16,32,64,128\}$, replay buffer memory: $\{5{\cdot}10^3,10^4,2{\cdot}10^4\}$. In addition, the following implementations are used: for small networks: 3-layer fully connected NN with 48 nodes in each layer followed by ReLU; for modest-sized networks: 4-layer fully connected NN with 48 nodes in each layer followed by ReLU; for large networks: 4-layer fully connected NN with 96 nodes in each layer followed by ReLU. For model-3, the pair of the normalized buffer and channel states is used as input (*i.e.* the input size is 2). For model-4, the pair of the normalized battery and channel states for different transmitters are concatenated and used as input (*i.e.* the input size is two times the number of transmitters). The output layer has $|\mathcal{A}|$ nodes, followed by a softmax function, where each node represents the probability of a specific action being the optimal one. We use a dropout with a probability of 0.2 after each layer except the final layer. The l_2 regularization is employed with the corresponding weight chosen from $\{10^{-4},10^{-3},10^{-2},10^{-1}\}$. The Adam optimizer is used to update the weight of neural networks.
- For PSR, the PTT of the original environment $\mathbf{P}$ is being estimated until the estimation error (21) falls below 0.001. For PSR, the following parameters are used $\Psi=0.2,M=10,K=5,T=100,\gamma=0.9,\alpha=10^{-5},\theta=0.1$, and kept constant for different network sizes. For the policy iteration stage, equation (20) is employed.
- For ISARSA, the following parameters are used: $\epsilon=0.1,\alpha=0.2,\lambda=0.5$, and kept constant for different network sizes. The agent performs 250 episodes.
- For METRPO, the following parameters are used: 2-layer fully connected NN with 512 nodes in each layer followed by ReLU for small networks, 2-layer fully connected NN with 1024 nodes in each layer followed by ReLU for modest-sized networks, and 3-layer fully connected NN with 1024 nodes in each layer followed by ReLU for large networks. The Adam optimizer with a learning rate 0.001 is used. The batch size of 1000. The number of learned models is chosen from the following: {1,2,3,4,5} for small networks, {4,5,6,7,8} for modest-sized networks, and {7,8,9,10} for large networks.
- For A3C, the same experimental setting in [5] is employed for all network sizes.

After finding the best set of hyper-parameters for each algorithm, the runtimes are measured. The runtime of Algorithm 1 is measured as explained in Section IV-C. For others, each algorithm is run until the corresponding convergence condition is satisfied (or the maximum number of iterations is achieved.)

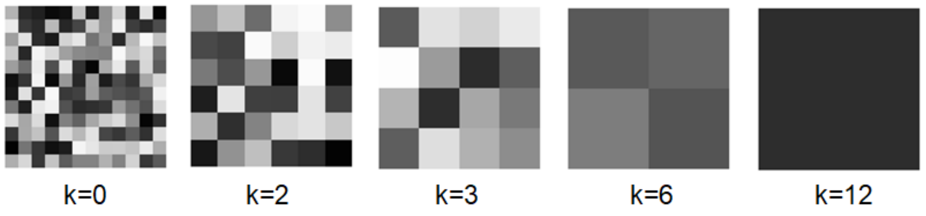

Fig. 1: Structural state-action aggregation with different k

C. *Figure 7*

We give an example case of the structural state-action aggregation algorithm presented in [6] in Fig.1. Assume $|\mathcal{S}| = 12$ and $k \in \{2, 3, 6, 12\}$. Although there are a variety of possible aggregations for each k, we adopt a straightforward approach, aggregating Q-functions starting from the first state-action. We find the k-nearest neighbors and aggregate them in a single Q-function and subsequently move on to the next unaggregated state. Fig.1 show the aggregated PTTs for different k. For instance, when $k = 2$, the 1^{st} and 2^{nd} state-actions are grouped together in the same aggregation. Consequently, their Q-functions are represented by a single Q-function.

For each k, we first find the aggregations and obtain the reduced-dimensional (aggregated) PTT. Then, we run Algorithm 1 using the same parameters in Section IV-B and obtain the aggregated policy. We finally interpolate the aggregated policy to obtain the estimated policy for the original system. The aggregation process works by simply assigning the policy of the aggregation to each state in the aggregation. APE is calculated by comparing the optimal policy with the interpolated policy for each different k.

D. *Figure 8*

For all sub-figures, Algorithm 1 is run using the same settings in Section IV-B.

- For Fig.8a, the following quantity is computed:
$$\frac{1}{|\mathcal{S}||\mathcal{A}|} \sum_{s\in\mathcal{S}, a\in\mathcal{A}} |\Delta_t^{it}(s,a) - \Delta_{t-1}^{it}(s,a)|, \tag{23}$$
for $t = [0, 4000]$. The quantity is calculated 50 times and averaged, represented by the blue curve. In each iteration, the shaded area's lower and upper bounds correspond to the smallest and largest values obtained from the 50 results.
- For Fig.8b, the blue curve is obtained by computing $|\Delta_t^{it}(7,1) - \Delta_{t-1}^{it}(7,1)|$. The quantity is calculated 50 times and averaged. In each iteration, the shaded area's lower and upper bounds correspond to the smallest and largest values obtained from the 50 results. The orange line is the value of $\sum_{n=1}^{5} \theta^{(n)}(7,1)$, where $\theta^{(n)}$ is as defined in Proposition 2.
- For Fig.8d, the following quantity is computed:
$$\frac{1}{40} \sum_{t'=t-20}^{t+20} \left(\mathbf{Q}_t^{it}(7,1) - \mathbf{Q}^*(7,1)\right), \tag{24}$$
where the optimal Q-functions $\mathbf{Q}^*(7,1)$ is computed by running a different Q-learning with the same parameters as in Section IV-B until the averaged Q-function updates defined in (22) below falls below 0.0001. The quantity is calculated 50 times. In each iteration, the lowest and highest points of the curve are the smallest and largest values obtained from the 50 results.
- For Fig.8e, the blue curve in the plot is obtained by computing the following:
$$\frac{1}{40} \sum_{t'=t-20}^{t+20} \left(\mathbf{Q}_t^{it}(7,1) - \mathbf{Q}^*(7,1)\right)^2 - \left[\frac{1}{40} \sum_{t'=t-20}^{t+20} \left(\mathbf{Q}_t^{it}(7,1) - \mathbf{Q}^*(7,1)\right)\right]^2, \tag{25}$$
where the optimal Q-functions are calculated as previously. The quantity is calculated 50 times. In each iteration, the lowest and highest points of the curve are the smallest and largest values obtained from the 50 results. The orange dotted curve is the value of $\frac{(1-u)}{(1+u)}\frac{\lambda^2}{3}$ with $u = 0.5$ and $\lambda = 1$, which is $\frac{1}{9}$. The green dotted curve is the value of $\frac{(1-u)}{(1+u)}\lambda^2$, which is $\frac{1}{3}$. The red dotted curve is the value of $\frac{2\lambda^2}{(1+u)^2} + \frac{(1-u)}{(1+u)}\lambda^2$ with $u = 0.5$ and $\lambda = 1$, which is $\frac{11}{9}$.

- For Fig.8f, for given n_1 and n_2, we compute the following:

$$ADC = \frac{1}{24995000} \sum_{t_1 \neq t_2 \in [0,5000]} \text{Distance Correlation}([\mathcal{X}_{t_1}^{(n_1)}(7,1)], [\mathcal{X}_{t_2}^{(n_2)}(7,1)]), \tag{26}$$

  where $[\mathcal{X}_{t_1}^{(n_1)}(7,1)]$ is the distribution vector of $\mathcal{X}_{t_1}^{(n_1)}(7,1)$ at time t_1 of size 50, which is obtained by running Algorithm 1 50 times. The vector $[\mathcal{X}_{t_2}^{(n_2)}(7,1)]$ is obtained similarly. The distance correlation between these two vectors can be obtained by the Python library "scipy.spatial.distance.correlation". In the summation, there are 25000000-5000=24995000 terms. After obtaining the ADC for each (n_1, n_2) pair, we draw the resulting by Python "imshow" library using bicubic interpolation.
- For Fig.8g, we plot the histogram of $\mathbf{Q}_t^{(n)}(7,1) - \mathbf{Q}^*(7,1)$ over $t \in [0, 5000]$ for different n. Then, we fit a Gaussian curve to data using the Python library "from scipy.stats.norm.fit".

E. Figure 9

- For Fig.9a, we first estimate the original PTT and cost matrix such that the estimation error < 0.05 using sample averaging. We then employ the HOLA algorithm in [1] (with $\Delta = 0.5, n = 2$) to obtain the estimated policy. Then we compute the Bellman order $BE(\mathbf{Q}_{\boldsymbol{\pi}}^{(n)}) = \mathbf{c}_{\boldsymbol{\pi}} + \gamma \mathbf{P}_{\pi} \mathbf{Q}_{\boldsymbol{\pi}}^{(n)} - \mathbf{Q}_{\boldsymbol{\pi}}^{(1)}$ for $n \in \{1, 2, ..., 10\}$ using the estimated policy. This result is given in the blue curve. For the orange curve, we search all possible different policies and pick the one that minimizes APE.
- For Fig.9b, we run Algorithm 1 with the following parameters: $|\mathcal{S}| = 40000, K = 10, u_t = 0.5, v = 50, l = 15, \gamma = 0.99, \alpha_t = \frac{1}{1+\frac{t}{1000}}, \epsilon_t = \min(0.99^t, 0.01)$. Then we obtain the estimated policy and compute the corresponding APE. Then, for each $n \in \{1, 2, .., 10\}$, we remove the n^{th} environment, and run Algorithm 1 with $K = 9$ environments using the same parameters and compute APE. The plot shows the absolute difference between the APE with $K = 10$ and the APE with $K = 9$ when n^{th} environment is removed.

F. Comparison with prior work

In Section IV-B, we give a performance comparison between the proposed algorithm and our prior work [6], [2], [3]. All algorithms are simulated on the MIMO wireless network with interference channels with a network size of 40000. The parameters in Section IV-B are employed for the proposed algorithm. We use the following parameters for the other algorithms:

- For [6]: $K = 5, p = 0.9, l = 5, v = 5, u = 0.5, \epsilon = 0.25, \gamma = 0.9, \alpha_t = \frac{0.5}{t}$.
- For [2]: $K = 5, p = 0.9, l = 15, v = 20, u_t = 1 - e^{\frac{-t}{5000}}, \epsilon_t = \max((0.95)^t, 0.01), \gamma = 0.9, \alpha_t = \frac{1}{1+\frac{t}{100}}$.
- For [3]: $K = 5, p = 0.9, l = 10, v = 40, u_t = 1 - e^{\frac{-t}{1000}}, \epsilon_t = \max(0.99^t, 0.01), \gamma = 0.9,\ \alpha_t = \frac{1}{1+\frac{t}{100}}$.

We note that the parameters K, p, γ are chosen to be the same for all algorithms for a fair performance comparison. The other parameters are individually optimized for each algorithm. To estimate the original PTT, sample averaging is used (with estimation error < 0.5). No state-aggregation algorithm is employed.

For the robustness analysis, we employ the following ranges for each parameter. The APE results are averaged at the end. The changes in APE reported in Section IV-B are the difference between the largest and smallest APE among all different settings.

- For [6]: $K \in \{2, 3, 4, 5\}, p \in \{0.9, 0.95\}, l \in \{5, 10, 20\}, v \in \{5, 10, 20\}, u = \{0.4, 0.5, 0.6\}, \epsilon \in \{0.25, 0.40\}, \gamma \in \{0.9, 0.95\}, \alpha_t \in \{\frac{0.5}{t}, \frac{1}{t}\}$.
- For [2]: $K \in \{2, 3, 4, 5, 6, 7\}, p \in \{0.9, 0.95\}, l \in \{10, 15, 20\}, v \in \{10, 20, 40\}, u_t \in \{1 - e^{\frac{-t}{1000}}, 1 - e^{\frac{-t}{5000}}\}, \epsilon_t \in \{\max((0.95)^t, 0.01), \max((0.99)^t, 0.001)\}, \gamma \in \{0.9, 0.95\}, \alpha_t \in \{\frac{1}{1+\frac{t}{100}}, \frac{1}{1+\frac{t}{500}}\}$.
- For [3]: $K \in \{3, 4, 5, 6, 7\}, p \in \{0.9, 0.95\}, l \in \{5, 10, 15\}, v \in \{20, 30, 40\}, u_t \in \{1 - e^{\frac{-t}{1000}}, 1 - e^{\frac{-t}{2500}}\}, \epsilon_t \in \{\max((0.95)^t, 0.01), \max((0.99)^t, 0.01)\}, \gamma \in \{0.9, 0.95\}, \alpha_t \in \{\frac{1}{1+\frac{t}{100}}, \frac{1}{1+\frac{t}{500}}\}$.

References

[1] Talha Bozkus and Urbashi Mitra. Link analysis for solving multiple-access mdps with large state spaces. *IEEE Transactions on Signal Processing*, 71:947–962, 2023.

[2] Talha Bozkus and Urbashi Mitra. Ensemble graph Q-learning for large scale networks. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[3] Talha Bozkus and Urbashi Mitra. Multi timescale ensemble Q-learning for markov decision process policy optimization. https://github.com/talhabozkus/Multi-Timescale-Ensemble-Q-learning-for-Markov-Decision-Process-Policy-Optimization.git, Submitted for publication to IEEE Transactions on Signal Processing, 2023.

[4] Francisco S Melo. Convergence of Q-learning: A simple proof. *Institute Of Systems and Robotics, Tech. Rep*, pages 1–4, 2001.

[5] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PMLR, 2016.

[6] Talha Bozkus and Urbashi Mitra. Ensemble link learning for large state space multiple access communications. In *2022 30th European Signal Processing Conference (EUSIPCO)*, pages 747–751, 2022.