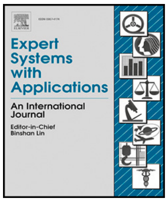
Expert Systems With Applications 213 (2023) 118167

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Spectrum efficiency maximization for multi-hop D2D communication underlaying cellular networks: Machine learning-based methods[☆]

Sengly Muy, Jung-Ryun Lee [*]

School of Electronics and Electrical Engineering (EEE) and Department of Intelligent Energy and Industry at Chung-Ang University, Seoul, 156-756, Republic of Korea

ARTICLE INFO

Keywords:
Multi-hop D2D
ML
DQL
Unsupervised DNN

ABSTRACT

Multi-hop D2D communication has been proposed with the purpose of improving the coverage, quality of service (QoS), and flexibility and adaptability of single-hop D2D communication. However, multi-hop D2D communication often experiences obstacles caused by interference from the shared channel, which makes spectrum efficiency in multi-hop D2D networks an important issue to tackle. In this paper, we study the optimization of spectrum efficiency in multi-hop D2D communication underlaying cellular networks. First, we use iteration-based optimization techniques such as exhaustive search (ES) and gradient search (GS) with barrier function to find the global and local optimal solutions, respectively. More importantly, we propose two machine learning (ML) techniques, the unsupervised deep neural network (DNN) and deep Q-learning (DQL) algorithms and evaluate the performances of both algorithms compared to iteration-based optimization methods. The simulation results verify that both algorithms achieve near-global optimums compared to GS. Moreover, it is verified that the DQL outperforms the unsupervised DNN in terms of optimal spectrum efficiency, while the DQL algorithm has higher time complexity than the unsupervised DNN.

1. Introduction

Device-to-device (D2D) communication enables one user to communicate directly with another in their neighborhood without relaying information via the base station or the core network. The paradigm of D2D has been anticipated as an essential technology for future networks because it can improve throughput, enhance spectrum efficiency and reliability, reduce latency, and extend network coverage (Jameel et al., 2018; Tehrani et al., 2014). Unfortunately, the direct single-hop D2D communication mechanism only performs well within a small area and has difficulty in delivering high-quality transmissions over a wide range. Therefore, multi-hop D2D communication has been proposed in order to improve the coverage, quality of service (QoS), and flexibility and adaptability of single-hop D2D communication (Shaikh & Wismüller, 2018). However, multi-hop D2D communication often experiences obstacles such as heavy data flow, which is caused by interference from the shared channel. Consequently, spectrum efficiency in multi-hop D2D networks has become an important issue to tackle.

Recently, many researchers have studied enhancing the spectrum efficiency of multi-hop D2D communications. The authors of Ali et al. (2017) have examined the performance of the multi-hop D2D communications underlaying cellular networks in the context of the disaster communication system, and compared it with that of the direct mode of D2D communication. Results showed that multi-hop D2D gives higher energy efficiency and better spectrum efficiency than the direct mode. The authors in Wei et al. (2016) have designed a vehicle clustering algorithm to optimize spectrum efficiency of the multi-hop D2D communications underlaying vehicular ad hoc networks (VANETs) by managing the resource allocation and interference based on the clustering mechanism. In Melki et al. (2016) the authors aimed to maximize the spectrum efficiency of multi-hop D2D underlaying a 3GPP LTE-A network by proposing the transmit power control of D2D users based on the interference avoidance approach. In Klaiqi et al. (2018), the authors proposed an energy-efficient optimal adaptive forwarding strategy (OAFS) with a low-complexity sub-optimal adaptive forwarding strategy (SAFS) for multi-hop D2D communications. Analytical and simulation results show that OAFS and SAFS exhibit higher energy efficiency and spectral efficiency in the order of the best relay forwarding (BRF) mode, the cooperative relay beamforming (CRB) mode, direct

[☆] This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC support program (IITP-2022-2018-0-01799) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation), South Korea, by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20214000000280), and by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. NRF-2020R1A2C1010929).

[*] Corresponding author.
E-mail addresses: muysengly@cau.ca.kr (S. Muy), jrlee@cau.ac.kr (J.-R. Lee).

https://doi.org/10.1016/j.eswa.2022.118167
Received 20 December 2021; Received in revised form 28 June 2022; Accepted 11 July 2022
Available online 21 July 2022

D2D communications, and conventional cellular communications. It is noted that, because the resource management problem in multi-hop D2D communication is usually a non-convex problem, it requires very high computational complexity to obtain the global optimal solution. Therefore, finding more efficient method to solve the problem of the resource allocation in multi-hop D2D communication with low time complexity is an important problem to be tackled.

On the other hand, with the increasing diversity and complexity of mobile network architectures, it has become infeasible to monitor and manage the multitude of network elements. ML has been successfully applied in many areas because it can solve complex problems with large amounts of data, detecting anomalies, predicting future scenarios, and generally discovering patterns that a human can miss. Specifically, ML algorithms based on artificial neural networks have been used as a tool for solving complex non-convex optimization problems in wireless networks (LeCun et al., 2015). Some artificial neural network-based ML algorithms, such as supervised DNN, unsupervised DNN, and deep reinforcement learning (DRL), have been applied to wireless networks. Among these, a supervised DNN usually requires the output labels to update the weights of the DNN. However the output label value in ML signifies the global optimal solution of the optimization model, which is usually difficult to obtain using traditional optimization techniques. On the other hand, an unsupervised DNN does not require output labels. Instead, the loss function is designed according to the optimization target together with optimization constraints (Lee et al., 2018, 2019, 2020; Sun et al., 2017), and the DNN can directly update (train) its weights in a way that minimizes the loss function. DRL combines reinforcement learning (RL) and DNN, where RL handles the problem of a computational agent learning to make decisions by trial and error in interaction with its environment (Sutton & Barto, 2018) and DNN acts as a powerful function approximator. By using DNN for the approximation of state/input states of RL, DRL can handle extremely large state and/or action spaces. One of DRL's most popular algorithms is deep Q-learning (DQL), which has been successfully employed in many environments, such as image processing, Atari games, chess, and so on (Arulkumaran et al., 2017; Li, 2017).

There have been successful studies that have used ML algorithms to solve complex wireless network problems. In Sun et al. (2017), the authors proposed a DNN-based power control algorithm for wireless resource allocation and showed that the proposed DNN algorithm achieves high sum-rate performance with low computational complexity for power control problems over either Gaussian interference or practical multi-cell channels. The authors of Lee et al. (2018, 2019) applied an unsupervised convolutional neural network to optimize the transmit power allocation with the purpose of maximizing energy efficiency. The authors of Lee et al. (2020) proposed an unsupervised DNN for controlling transmit power to maximize spectrum efficiency in a network with co-channel interference. In Zou et al. (2019), DQL was applied to spectrum and power allocation on unlicensed and licensed bands in D2D communication, and the author of Ban (2020) proposed dueling DQL to control power and optimize the sum-rate of D2D pairs.

Up to our knowledge, there is no work which applied ML techniques for the resource management in 'multi-hop' D2D communication based on ML techniques. Moreover, because it is a non-convex issue, solving the resource management problem in multi-hop D2D communication demands a high computing complexity to get the optimal global solution. On the other hand, ML algorithms have been used as a tool for solving the complicated optimization problem with achieving near-global optimum (LeCun et al., 2015). Specifically speaking, we can design ML algorithms so that they can be trained offline to find optimal or sub-optimal solutions for the given problem. After that, the ML-trained model can be deployed with actual input data to obtain an optimal or sub-optimal solutions while reducing computing complexity and enhancing the practical performance (Liang et al., 2019). Therefore, the application of ML algorithms to the spectrum efficiency problem in multi-hop D2D communication can be justified in that we can expect (near-)global-solution of the problem with low computational complexity compared to ES method which is usually used for numerically obtaining global optimal solutions in non-convex problems.

In this paper, we aim to maximize spectrum efficiency based on power control in wireless multi-hop D2D underlay cellular networks using unsupervised DNN and DQL. The contributions of the paper are summarized as follows: (1) We provide a system model and build an optimization model to maximize spectrum efficiency considering a multi-hop environment. (2) We evaluate the performance of our multi-hop D2D underlay cellular network using the optimization-based iterative methods, ES and GS with barrier function. (3) More importantly, we propose two ML algorithms, DNN and DQL, and compare the performance of the proposed algorithms to optimization-based iterative methods, together with time-complexity analysis.

2. System model and problem formulation

In this paper, we consider a single cell consisting of a cellular user equipment (CUE), N_1 one-hop D2D networks, and N_2 two-hop D2D networks. The CUE transmits data to the BS via up-link transmission. The CUE, one-hop D2D networks, and two-hop D2D networks are located randomly in the BS coverage area with a radius of R, as in Fig. 1. In this network scenario, we consider the spectrum sharing mode in which D2D links and CUE links share the same spectrum, which causes interference with each other. For the two-hop D2D network, we assume that there are two orthogonal channels with equal bandwidth. It is noted that the first channel is used for communication between the source node and the relay node (link-1), which is interfering with CUE and one-hop networks. Similarly, the second channel is used for the relay node to communicate with the destination node (link-2), also interfering with CUE and one-hop networks. Let $\mathcal{N}_1 = \{1, 2, \ldots, N_1\}$ and $\mathcal{N}_2 = \{1, 2, \ldots, N_2\}$ be the sets of one-hop D2D networks and two-hop D2D networks, respectively. We denote $p_1^{\mathrm{CUE}}, p_{n_1}^{\mathrm{1hop}}, p_{n_2}^{\mathrm{2hop}_1}$, and $p_{n_2}^{\mathrm{2hop}_2}$ as the transmit power of CUE, n_1th one-hop network, link-1 of n_2th two-hop network, and link-2 of n_2th two-hop network, respectively, where $n_1 \in \mathcal{N}_1$ and $n_2 \in \mathcal{N}_2$.

According to the study in Zhao and Pottie (2013), the received signal-to-interference plus noise ratio (SINR) of the CUE's up-link in first and second orthogonal channel can be calculated as

$$\gamma_1^{\mathrm{CUE}_1} = \frac{p_1^{\mathrm{CUE}} g_{1,1}^{\mathrm{CUE\text{-}BS}}}{\sum\limits_{n_1 \in \mathcal{N}_1} I_{n_1,1}^{\mathrm{1hop\text{-}BS}} + \sum\limits_{n_2 \in \mathcal{N}_2} I_{n_2,1}^{\mathrm{2hop_1\text{-}BS}} + \sigma_A^2},$$
$$\gamma_1^{\mathrm{CUE}_2} = \frac{p_1^{\mathrm{CUE}} g_{1,1}^{\mathrm{CUE\text{-}BS}}}{\sum\limits_{n_1 \in \mathcal{N}_1} I_{n_1,1}^{\mathrm{1hop\text{-}BS}} + \sum\limits_{n_2 \in \mathcal{N}_2} I_{n_2,1}^{\mathrm{2hop_2\text{-}BS}} + \sigma_A^2}, \tag{1}$$

where $g_{1,1}^{\mathrm{CUE\text{-}BS}}$ is the channel gain from the CUE to the BS, $\sum_{n_1 \in \mathcal{N}_1} I_{n_1,1}^{\mathrm{1hop\text{-}BS}}$, $\sum_{n_2 \in \mathcal{N}_2} I_{n_2,1}^{\mathrm{2hop_1\text{-}BS}}$ and $\sum_{n_2 \in \mathcal{N}_2} I_{n_2,1}^{\mathrm{2hop_2\text{-}BS}}$ are the interferences from the one-hop networks, from link-1 of the two-hop networks, and from link-2 of the two-hop networks to the BS, respectively. By employing the Shannon capacity formula, the achievable spectrum efficiency of CUE's up-link can be calculated as

$$R_1^{\mathrm{CUE}} = \frac{1}{2} \log_2 \left(1 + \gamma_1^{\mathrm{CUE}_1}\right) + \frac{1}{2} \log_2 \left(1 + \gamma_1^{\mathrm{CUE}_2}\right). \tag{2}$$

Similarly, the SINR of the n_1th one-hop networks in the first and second orthogonal channel can be calculated as

$$\gamma_{n_1}^{\mathrm{1hop}_1} = \frac{p_{n_1}^{\mathrm{1hop}} g_{n_1,n_1}^{\mathrm{1hop\text{-}1hop}}}{I_{1,n_1}^{\mathrm{CUE\text{-}1hop}} + \sum\limits_{n_1' \in \mathcal{N}_1 \setminus n_1} I_{n_1',n_1}^{\mathrm{1hop\text{-}1hop}} + \sum\limits_{n_2 \in \mathcal{N}_2} I_{n_2,n_1}^{\mathrm{2hop_1\text{-}1hop}} + \sigma_A^2},$$
$$\gamma_{n_1}^{\mathrm{1hop}_2} = \frac{p_{n_1}^{\mathrm{1hop}} g_{n_1,n_1}^{\mathrm{1hop\text{-}1hop}}}{I_{1,n_1}^{\mathrm{CUE\text{-}1hop}} + \sum\limits_{n_1' \in \mathcal{N}_1 \setminus n_1} I_{n_1',n_1}^{\mathrm{1hop\text{-}1hop}} + \sum\limits_{n_2 \in \mathcal{N}_2} I_{n_2,n_1}^{\mathrm{2hop_2\text{-}1hop}} + \sigma_A^2}, \tag{3}$$

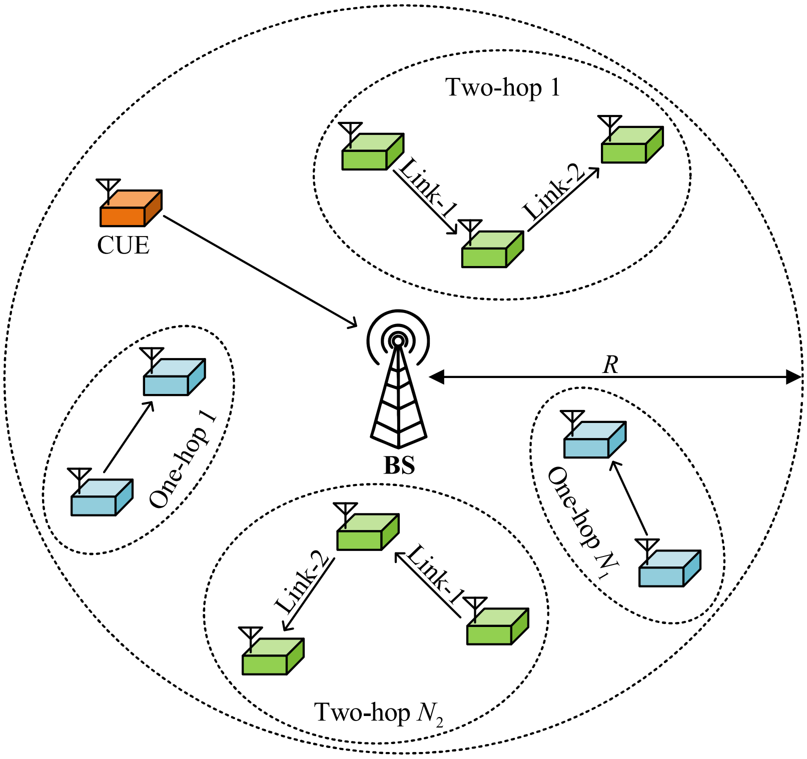


Fig. 1. System Model.

where $g_{n_1,n_1}^{\text{1hop-1hop}}$ is the channel gain from n_1th D2D transmitter (DTx) in one-hop networks to n_1th D2D receiver (DRx) in one-hop networks. $I_{1,n_1}^{\text{CUE-1hop}}$, $\sum_{n_1'\in\mathcal{N}_1\setminus n_1} I_{n_1',n_1}^{\text{1hop-1hop}}$, $\sum_{n_2\in\mathcal{N}_2} I_{n_2,n_1}^{\text{2hop}_1\text{-1hop}}$ and $\sum_{n_2\in\mathcal{N}_2} I_{n_2,n_1}^{\text{2hop}_2\text{-1hop}}$ are the interferences to the n_1th one-hop networks from the CUE, from other one-hop networks, from link-1s of the two-hop networks, and from link-2s of the two-hop networks, respectively. The achievable spectrum efficiency of the n_1th one-hop networks can be calculated as following

$$R_{n_1}^{\text{1hop}} = \frac{1}{2}\log_2\left(1+\gamma_{n_1}^{\text{1hop}_1}\right) + \frac{1}{2}\log_2\left(1+\gamma_{n_1}^{\text{1hop}_2}\right). \tag{4}$$

The SINR of link-1 in the n_2-th two-hop networks is given by

$$\gamma_{n_2}^{\text{2hop}_1} = \frac{p_{n_2}^{\text{2hop}_1} g_{n_2,n_2}^{\text{2hop}_1\text{-2hop}_1}}{I_{1,n_1}^{\text{CUE-2hop}_1} + \sum\limits_{n_1\in\mathcal{N}_1} I_{n_1,n_2}^{\text{1hop-2hop}_1} + \sum\limits_{n_2'\in\mathcal{N}_2\setminus n_2} I_{n_2',n_2}^{\text{2hop}_1\text{-2hop}_1} + \sigma_A^2}, \tag{5}$$

where $g_{n_2,n_2}^{\text{2hop}_1\text{-2hop}_1}$ is the channel gain from n_2-th link-1 DTx in one-hop networks to n_2-th link-1 DRx in two-hop networks. $I_{1,n_1}^{\text{CUE-2hop}_1}$, $\sum_{n_1\in\mathcal{N}_1} I_{n_1,n_2}^{\text{1hop-2hop}_1}$ and $\sum_{n_2'\in\mathcal{N}_2\setminus n_2} I_{n_2',n_2}^{\text{2hop}_1\text{-2hop}_1}$ are the interferences to n_2-th of the two-hop networks from the CUE, from one-hop networks, and from other link-1s of two-hop networks, respectively. The SINR of link-2 in the n_2-th two-hop networks is given by

$$\gamma_{n_2}^{\text{2hop}_2} = \frac{p_{n_2}^{\text{2hop}_2} g_{n_2,n_2}^{\text{2hop}_2\text{-2hop}_2}}{I_{1,n_1}^{\text{CUE-2hop}_2} + \sum\limits_{n_1\in\mathcal{N}_1} I_{n_1,n_2}^{\text{1hop-2hop}_2} + \sum\limits_{n_2'\in\mathcal{N}_2\setminus n_2} I_{n_2',n_2}^{\text{2hop}_2\text{-2hop}_2} + \sigma_A^2}, \tag{6}$$

where $g_{n_2,n_2}^{\text{2hop}_2\text{-2hop}_2}$ is the channel gain from n_2-th link-2 DTx in one-hop networks to n_2-th link-2 DRx in two-hop networks. $I_{1,n_1}^{\text{CUE-2hop}_2}$, $\sum_{n_1\in\mathcal{N}_1} I_{n_1,n_2}^{\text{1hop-2hop}_2}$ and $\sum_{n_2'\in\mathcal{N}_2\setminus n_2} I_{n_2',n_2}^{\text{2hop}_2\text{-2hop}_2}$ are the interference from CUE, one-hop networks and other link-2s of the two-hop networks, respectively. The achievable spectrum efficiency of link-1 and link-2 in the n_2-th two-hop networks can be calculated as following

$$R_{n_2}^{\text{2hop}_1} = \frac{1}{2}\log_2\left(1+\gamma_{n_2}^{\text{2hop}_1}\right),$$

$$R_{n_2}^{\text{2hop}_2} = \frac{1}{2}\log_2\left(1+\gamma_{n_2}^{\text{2hop}_2}\right). \tag{7}$$

In our work, we consider the decode-and-forward (DF) relaying scheme for two-hop D2D communication because DF relaying scheme significantly reduces noise and interference from the source node to the destination node (Levin & Loyka, 2012). Then, the spectrum efficiency of the network is given by

$$\begin{aligned} \text{SE}\left(\vec{p}\right) = {} & R_1^{\text{CUE}} + \sum_{n_1\in\mathcal{N}_1} R_{n_1}^{\text{1hop}} \\ & + \sum_{n_2\in\mathcal{N}_2} \min\left\{R_{n_2}^{\text{2hop}_1}, R_{n_2}^{\text{2hop}_2}\right\}, \end{aligned} \tag{8}$$

where $\vec{p} = \left(p_1^{\text{CUE}}, p_1^{\text{1hop}}, \ldots, p_{n_1}^{\text{1hop}}, p_1^{\text{2hop}_1}, \ldots, p_{n_2}^{\text{2hop}_1}, p_1^{\text{2hop}_2}, \ldots, p_{n_2}^{\text{2hop}_2}\right)$ is the vector of transmit powers of the CUE, one-hop D2D networks, link-1 of two-hop D2D networks, and link-2 of two-hop D2D networks, respectively.

In order to simplify the channel gain model, we construct the channel matrix as

$$\mathbf{G} = \begin{bmatrix} g_{1\times 1}^{\text{CUE-BS}} & g_{1\times N_1}^{\text{1hop-BS}} & g_{1\times N_2}^{\text{2hop}_1\text{-BS}} & g_{1\times N_2}^{\text{2hop}_2\text{-BS}} \\ g_{N_1\times 1}^{\text{CUE-1hop}} & g_{N_1\times N_1}^{\text{1hop-1hop}} & g_{N_1\times N_2}^{\text{2hop}_1\text{-1hop}} & g_{N_1\times N_2}^{\text{2hop}_2\text{-1hop}} \\ g_{N_2\times 1}^{\text{CUE-2hop}_1} & g_{N_2\times N_1}^{\text{1hop-2hop}_1} & g_{N_2\times N_2}^{\text{2hop}_1\text{-2hop}_1} & 0_{N_2\times N_2} \\ g_{N_2\times 1}^{\text{CUE-2hop}_2} & g_{N_2\times N_1}^{\text{1hop-2hop}_2} & 0_{N_2\times N_2} & g_{N_2\times N_2}^{\text{2hop}_2\text{-2hop}_2} \end{bmatrix}. \tag{9}$$

It is noticed that $g_{A\times B}$ is the channel gain matrix of size $A \times B$, and $0_{A\times B}$ is the zero matrix of size $A \times B$. Here, $g_{i,j} = \frac{\hat{g_{i,j}}}{d_{i,j}^m}$ is defined as the channel gain between two nodes where $\hat{g_{i,j}}$, $d_{i,j}$, and m are Rician fading, the distance between the two nodes, and the path-loss exponent, respectively.

Finally, we formulate the optimization problem as

$$\begin{aligned} & \max_{\vec{p}} \text{SE}\left(\vec{p}\right) \\ \text{s.t.}\quad & P_{\min} \le \vec{p} \le P_{\max} \end{aligned} \tag{10}$$

for $n_1 \in \mathcal{N}_1$ and $n_2 \in \mathcal{N}_2$

where $P_{\min}$ and $P_{\max}$ are the minimum and maximum transmit power, respectively.

3. Optimization-based resource allocation

3.1. Exhaustive search

ES is an algorithmic technique to find the global optimum by testing every possible candidate solution. In our problem, the transmit power $\vec{p}$ is uniformly quantized with D levels. Then ES tests over all possible combinations of the quantized parameters. Although the ES technique can achieve a global solution, it requires a long computation time to complete all iterations.

3.2. Gradient search with barrier

GS is a first-order iterative optimization algorithm for finding a local minimum for the non-constrained optimization problem. However, our problem is the constrained optimization problem because of the limited transmit power $P_{\min} \le \vec{p} \le P_{\max}$. In order to eliminate the constraint, we use a logarithmic barrier function, a continuous function whose value increases to infinity at the boundary points. With the appropriate logarithmic barrier function, the optimization is no longer constrained and, is therefore easier to handle. The logarithmic barrier function is given as follows:

$$\psi\left(\vec{p}\right) = \frac{1}{t}\sum_{p\in\vec{p}}\left[\ln\left(P_{\max} - p\right) + \ln\left(p - P_{\min}\right)\right], \tag{11}$$

where $t > 0$ is a parameter of the barrier. With the logarithmic barrier function in (11), our problem becomes an optimization problem without constraints, given as

$$\max_{\vec{p}} \mathcal{B}\left(\vec{p}\right) = \max_{\vec{p}}\left[U\left(\vec{p}\right) + \psi\left(\vec{p}\right)\right]. \tag{12}$$

To solve the problem in (12), we use the simple GS technique. Algorithm 1 explains the GS for searching the maximum $\mathcal{B}\left(\vec{p}\right)$, where β is the learning rate and ϵ is the error tolerance.

Algorithm 1 Gradient search algorithm

```
initialize $\vec{p}$ randomly in feasible region
set $\beta$ and $\epsilon$
repeat :
    $\vec{p}_{new} = \vec{p}_{old} + \beta \times \frac{\partial\mathcal{B}}{\partial\vec{p}}$
until $\left|\mathcal{B}\left(\vec{p}_{new}\right) - \mathcal{B}\left(\vec{p}_{old}\right)\right| < \epsilon$
```

4. Machine-learning based resource allocation

In this section, we propose two machine learning designs, unsupervised DNN and DRL, to solve the optimization problem in (10).

4.1. Unsupervised DNN

The design of the proposed unsupervised DNN architecture is illustrated in Fig. 2. We assume that the BS is equipped with the unsupervised DNN model, and the mathematical DNN model can be expressed as

$$\vec{z} = f_{\mathbf{W},\mathbf{B}}\left(\vec{x}\right) \tag{13}$$

where $\mathbf{W} = \left\{\mathbf{w}^{(0)}, \mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(L_{\mathrm{DNN}})}\right\}$ is the set of weight matrices, $\mathbf{B} = \left\{\vec{b}^{(0)}, \vec{b}^{(1)}, \ldots, \vec{b}^{(L_{\mathrm{DNN}})}\right\}$ is the set of bias vectors, and $\vec{x}$ represents the input vector. Here, the input vector $\vec{x}$ is the reshaped channel gains matrix $\mathbf{G} \in \mathbb{R}^{(N_1+2N_2+1)^2}$ with $N = \left(N_1 + 2N_2 + 1\right)^2$ elements. With the input vector $\vec{x}$, we can calculate the output of the first hidden layer given by

$$\vec{y}^{(0)} = \left[\mathbf{w}^{(0)}\vec{x} + \vec{b}^{(0)}\right]^+ \tag{14}$$

where $\mathbf{w}^{(0)} \in \mathbb{R}^{N\times N}$ and $\vec{b}^{(0)} \in \mathbb{R}^{N\times 1}$ are the first weight matrix and bias vector of the DNN, respectively. $[\cdot]^+ = \max(0,\cdot)$ is the rectified linear unit (ReLU) activation function. Similarly, $\vec{y}^{(l)}$ for $2 \le l \le L_{\mathrm{DNN}}$ is computed by

$$\vec{y}^{(l)} = \left[\mathbf{w}^{(l-1)}\vec{y}^{(l-1)} + \vec{b}^{(l-1)}\right]^+, \tag{15}$$

where $\mathbf{w}^{(l-1)} \in \mathbb{R}^{N\times N}$ and $\vec{b}^{(l-1)} \in \mathbb{R}^{N\times 1}$ are the lth weight matrix and bias vector of the DNN, respectively. $\vec{y}^{(l)}$ is forwarded to the next hidden layer. Based on the forward value of the last hidden layer, $\vec{y}^{(L_{\mathrm{DNN}})}$, we can determine the output of DNN as

$$\vec{z} = g\left(\mathbf{w}^{(L_{\mathrm{DNN}})}\vec{y}^{(L_{\mathrm{DNN}})} + \vec{b}^{(L_{\mathrm{DNN}})}\right), \tag{16}$$

where $\mathbf{w}^{(L_{\mathrm{DNN}})} \in \mathbb{R}^{N\times N}$ and $\vec{b}^{(L_{\mathrm{DNN}})} \in \mathbb{R}^{N\times 1}$ are the last weight matrix and bias vector of the DNN, respectively. The sigmoid function $g(x) = \frac{1}{1+e^{-x}}$ is used as the activation function at the output layer. The output $\vec{z}$ has a value between 0 to 1 because of the activation function g; therefore, we map the output with the transmit power, which is given by

$$\vec{p} = \vec{z}P_{\max} + \left(1 - \vec{z}\right)P_{\min}. \tag{17}$$

Normally, the loss function is defined as the negative of the objective function, given by $L_{\mathrm{old}}\left(\vec{p}\right) = -\mathrm{SE}\left(\vec{p}\right)$. However, based on numerous simulation experiments, we found that the performance of the unsupervised DNN using this conventional negative-objective-function DNN is not so attractive compared with ES. Therefore, we develop a new objective function suitable for the interference-limited environment of our problem. For this purpose, we design weighting parameters to weight the loss function according to the interference level of each pair, given as

$$\omega_i = \left(\frac{g_{i,i}}{\sum_{j=1,j\neq i}^{N_1+2N_2+1} g_{j,i}}\right) \Big/ \left(\frac{\mu_{i,i}}{\sum_{j=1,j\neq i}^{N_1+2N_2+1} \mu_{j,i}}\right), \tag{18}$$

where $g_{i,i}$ and $\sum_{j=0,j\neq i}^{N_1+2N_2} g_{j,i}$ represent the channel gains of the direct link and the interference links, respectively; and $\mu_{i,i}$ is the mean channel gains. Therefore, ω_i represents the ratios of the channel gains of the direct channels to interference channels. Applying these weights ω_i to each data rate, we can expect that ω_i decrease (increase) the rate of a node having strong (weak) interference, compared to the mean channel gains $\mu_{i,i}$. We then define the modified loss function as

$$\begin{aligned} L_{\mathrm{new}}\left(\vec{p}\right) = {} & \omega_1 \log_2\left(1 + \gamma_1^{\mathrm{CUE}}\right) \\ & + \sum_{n_1\in\mathcal{N}_1} \omega_{1+n_1} \log_2\left(1 + \gamma_{n_1}^{\mathrm{1hop}}\right) \\ & + \frac{1}{2}\min\left\{\sum_{n_2\in\mathcal{N}_2} \omega_{1+N_1+n_2} \log_2\left(1 + \gamma_{n_2}^{\mathrm{2hop_1}}\right),\right. \\ & \left.\sum_{n_2\in\mathcal{N}_2} \omega_{1+N_1+N_2+n_2} \log_2\left(1 + \gamma_{n_2}^{\mathrm{2hop_2}}\right)\right\}. \end{aligned} \tag{19}$$

To determine the parameters that minimize the loss function, we use a gradient descent method given by

$$\mathbf{W}_{\mathrm{new}} = \mathbf{W}_{\mathrm{old}} - \delta\nabla_{\mathbf{W}_{\mathrm{old}}} L_{\mathrm{new}}\left(\vec{p}\right), \tag{20}$$

where δ denotes a learning rate. To derive the gradient of the $\min\left(f(x), g(x)\right)$ function in (19), we convert it as

$$\min\left(f(x), g(x)\right) = \frac{f(x) + g(x) - \left|f(x) - g(x)\right|}{2}. \tag{21}$$

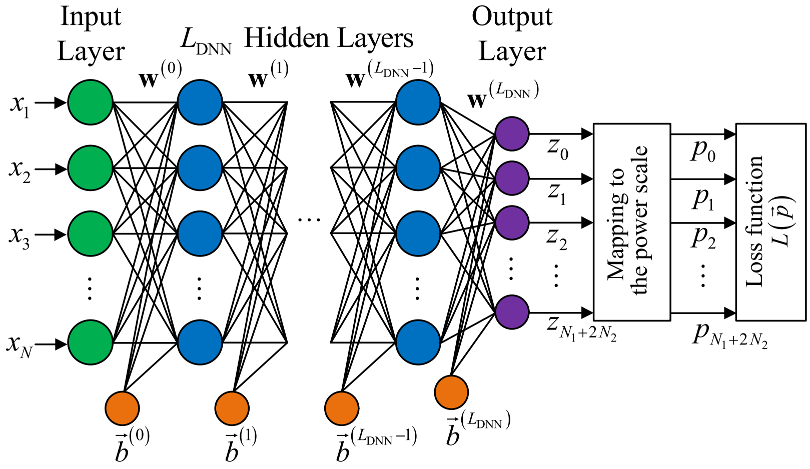

Input Layer
L_{DNN} Hidden Layers
Output Layer
$\mathbf{w}^{(0)}$
$\mathbf{w}^{(1)}$
$\mathbf{w}^{(L_{\text{DNN}}-1)}$
$\mathbf{w}^{(L_{\text{DNN}})}$
x_1
x_2
x_3
x_N
z_0
z_1
z_2
$z_{N_1+2N_2}$
Mapping to the power scale
p_0
p_1
p_2
$p_{N_1+2N_2}$
Loss function $L(\vec{p})$
$\vec{b}^{(0)}$
$\vec{b}^{(1)}$
$\vec{b}^{(L_{\text{DNN}}-1)}$
$\vec{b}^{(L_{\text{DNN}})}$

Fig. 2. Unsupervised DNN's scheme.

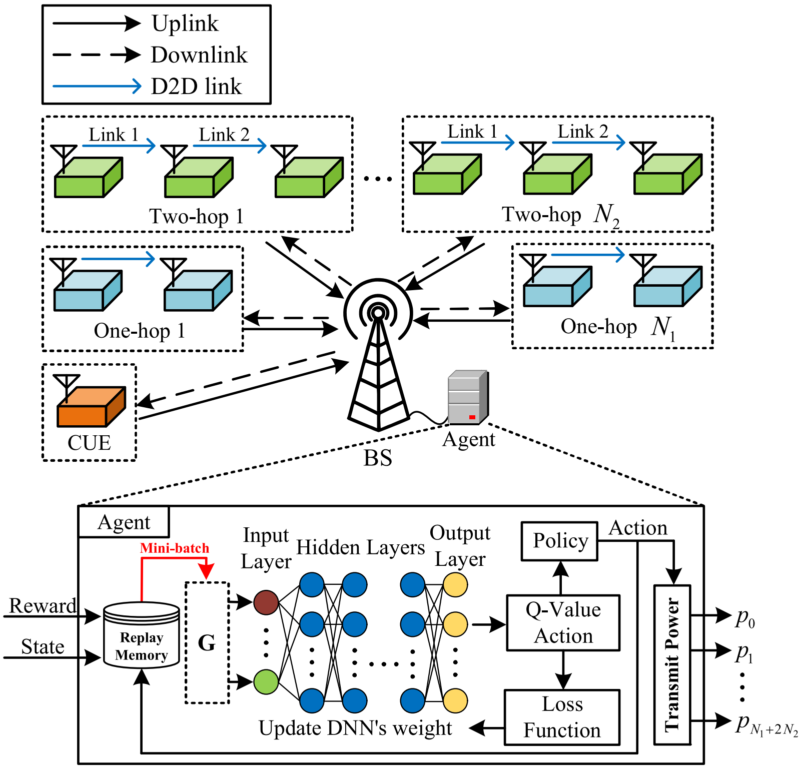


Fig. 3. Deep Q-learning scheme.

It is noted that the DNN training process is conducted offline, and the input datasets $\boldsymbol{G}$ is generated using the channel model. The testing or deployment process is conducted online, using actual channel gain values measured by channel state information (CSI).

4.2. Deep Q-learning

In this subsection, we propose a DQL algorithm to solve the optimization problem (10). The proposed DQN framework for the multi-hop D2D network system is illustrated in Fig. 3. We define agent, state, action, reward function, and policy as follows.

1. Agent: An agent is responsible for the execution of the DQL algorithm and is connected to the BS.

2. State: The state of DQL is defined as the set of channel gains

$$\mathcal{S} \in \{\boldsymbol{G}\}, \tag{22}$$

where $\boldsymbol{G}$ is the channel gain matrix defined in (9).

3. Action: The action of the proposed DQL algorithm is to adjust the transmit power level of transmitters, which is defined as

$$\mathcal{A} \in \left\{p_0, p_1, \dots, p_{N_1+2N_2}\right\}, \tag{23}$$

where $p_i = P_{\min} + k\frac{P_{\max}-P_{\min}}{M-1}$ for some $0 \le k \le M-1$ with M discrete levels. It is noted that the total number of actions in the DQL algorithm is $M^{N_1+2N_2+1}$.

4. Reward: After the agent performs an action, they will observe a new state and reward. The reward of the DQL scores how well the designated goal can be achieved. Because the proposed DQL algorithm aims to train the network so that it maximizes the reward value, the reward is the spectrum efficiency in (8), given by

$$\mathcal{R} = \text{SE}(\vec{p}). \tag{24}$$

5. Policy: To decide the action, we use ϵ-greedy policy, an exploration strategy in RL that takes an exploratory action with probability ϵ and a greedy action with probability $1-\epsilon$. The ϵ-greedy policy decides to select an action a', which is given by

$$a' = \begin{cases} \underset{A\in\mathcal{A}}{\text{argmax}} Q\left(s, A\right), & \text{with probability } 1-\epsilon \\ \text{random action}, & \text{with probability } \epsilon.. \end{cases} \tag{25}$$

The Q-table is updated using the Bellman equation, which is given by

$$Q\left(s,a\right) \leftarrow Q\left(s,a\right) + \alpha[r + \gamma \max_{A\in\mathcal{A}} Q_k\left(s', A\right) - Q\left(s,a\right)], \tag{26}$$

where α is a learning rate and γ is a discount factor. Because Q-learning requires a huge amount of memory to store the table of Q-values, we employ the DQL algorithm to implement the Q-learning by using the supervised DNN or deep Q-network (DQN) as a fitting function $Q\left(s,a;\boldsymbol{\theta}\right)$ for approximation of the Q-value-actions. In the DQL algorithm, the DQN's weight $\boldsymbol{\theta}$ is trained in a way that minimizes the error function. In our work, we use the mean-square error function, which is given by

$$\text{Loss} = E\left[\left(y - Q\left(s,a;\boldsymbol{\theta}\right)\right)^2\right], \tag{27}$$

where y is the target value estimated by

$$y = r + \gamma \max_{a\in\mathcal{A}} Q_{\text{target}}\left(s', a\right). \tag{28}$$

In the proposed DQL algorithm, it is assumed that all devices (D2Ds and CUE) share their necessary information, such as data rate and channel gain power, to the BS via up-link transmission. With this information, BS determines the reward according to Eq. (24) and then forwards it to the agent. After receiving the reward, the agent performs an action based on the policy and sends it to the BS; it then broadcasts to all devices via down-link transmission. The agent is required to store the state, action, and reward in the replay memory D. When the replay memory is full, the agent separates the data in the replay memory into mini-batch samples to train the DQN. In the DQN scheme, we design the DQN input and output as the channel gain and the Q-value-actions, respectively. The pseudocode of the proposed DQL algorithm is given in Algorithm 2.

Algorithm 2 Proposed deep Q-learning algorithm

```
Initialize the replay memory $D$ with capacity $C$
Initialize $Q\left(s,a;\boldsymbol{\theta}\right)$ with random weights
**while** not convergence **do**
    Select an action $a'$ according to $\epsilon$-greedy policy
    Observe a new state $s'$
    Observe a reward $r$
    Store transition $\left(s,a,r,s'\right)$ in $D$
    **If** the replay memory $D$ is full **then**
        Sample a mini-batch from memory $D$
        Train DQN that minimize the loss function (27)
    **end if**
    Update the state $s = s'$
**end while**
```

4.3. Time complexity analysis

We consider a system model with a CUE, N_1 one-hop D2D networks, and N_2 two-hop D2D networks. Therefore, there are $\left(N_1 + 2N_2 + 1\right)$ transmitter devices in the system.

1. **Exhaustive search algorithm**: The time complexity of ES is $O\left(D^{\left(N_1+2N_2+1\right)}\right)$, where D is the power quantization level (Cormen & Leiserson, 1990).

Table 1
Time complexity analysis.

Algorithms	Time complexity	Time computation
ES	$O\left(D^{\left(N_1+2N_2+1\right)}\right)$	6.5 days
DQL	$O\left(TL_{\text{DQL}}M^{2\left(N_1+2N_2+1\right)}\right)$	0.051 s
Unsupervised DNN	$O\left(L_{\text{DNN}}\left(N_1+N_2\right)^4\right)$	0.011 s
GS	$O\left(\epsilon^{-2}\right)$	0.030 s

2. **Gradient search algorithm**: The time complexity of GS with barrier function is $O\left(\epsilon^{-2}\right)$, where ϵ is the error tolerance (Cartis et al., 2010).
3. **Unsupervised DNN algorithm**: The proposed unsupervised DNN algorithm includes an input layer, an output layer, and L_{DNN} hidden layers. Let $n_{\text{input}}^{\text{DNN}}$, $n_{\text{hidden}}^{\text{DNN}}$ and $n_{\text{output}}^{\text{DNN}}$ be the numbers of neurons in the input, hidden, and output layers, respectively. According to the matrix computation (Sedgewick & Wayne, 2016), the feed-forward calculations of input layer with hidden layer, hidden layer with hidden layer, and hidden layer with output layer require $n_{\text{input}} \times n_{\text{hidden}}$, n_{hidden}^2, and $n_{\text{hidden}} \times n_{\text{output}}$ computation times, respectively. Therefore, the time complexity of the unsupervised DNN becomes

$$O\left(n_{\text{input}} n_{\text{hidden}} + \left(L_{\text{DNN}} - 1\right) n_{\text{hidden}}^2 + n_{\text{hidden}} n_{\text{output}}\right). \tag{29}$$

In our simulation experiment, we consider the neurons in the hidden layer equal to the neurons in the input layer. Since $n_{\text{input}} \gg n_{\text{output}}$, we can formulate the time complexity of the proposed unsupervised DNN as

$$\begin{aligned} O\left(L_{\text{DNN}} n_{\text{input}}^2\right) &= O\left(L_{\text{DNN}}\left(N_1 + 2N_2 + 1\right)^4\right) \\ &= O\left(L_{\text{DNN}}\left(N_1 + N_2\right)^4\right). \end{aligned} \tag{30}$$

4. **DQL algorithm**: In the proposed DQL simulation experiment, we consider a hidden layer whose number of neurons is equal to that in the output layer. The total number of neurons in the output layer is equal to the number of Q-value-actions, which is $M^{\left(N_1+2N_2+1\right)}$, where M is the number of discrete power levels. We assume that the proposed algorithm converges after T iterations; therefore, the time complexity of the DQL algorithm becomes

$$O\left(TL_{\text{DQL}}\left(M^{\left(N_1+2N_2+1\right)}\right)^2\right) = O\left(TL_{\text{DQL}}M^{2\left(N_1+2N_2+1\right)}\right). \tag{31}$$

For simulation runs, we used MATLAB 2020b on a PC equipped with an AMD Ryzen 5 3600 6-core processor, an NVIDIA GeForce RTX 2080 Ti GPU, and 16 GB of RAM. Table 1 summarizes the comparison of the computational complexity and the running time of the ES, DQL, unsupervised DNN, and GS algorithms when $N_1 = N_2 = 2$, $D = 25$, $M = 4$, $L_{\text{DNN}} = 5$, and $\epsilon = 10^{-5}$. The execution time is determined by using a single test channel. Because the ES requires a very long time to find the optimal transmit power, we use multiple computers to obtain the result.

5. Simulation results and discussion

For performance evaluation, we consider a single cell with a radius of 500 m. CUE, and one- and two-hop D2D users are randomly deployed over the cell coverage, as shown in Fig. 4. The distance between the D2D transmitter and receiver is set following a uniform distribution between 10 m and 20 m. Table 2 shows the simulation parameters used in this work. To obtain the ES result, we set the numbers of equi-spaced divisions of transmit power, $\vec{p}$, to 10^2. The GS-with-barrier algorithm is set by the learning rate of $\beta = 10^{-3}$, penalty parameter $t = 10^2$, and

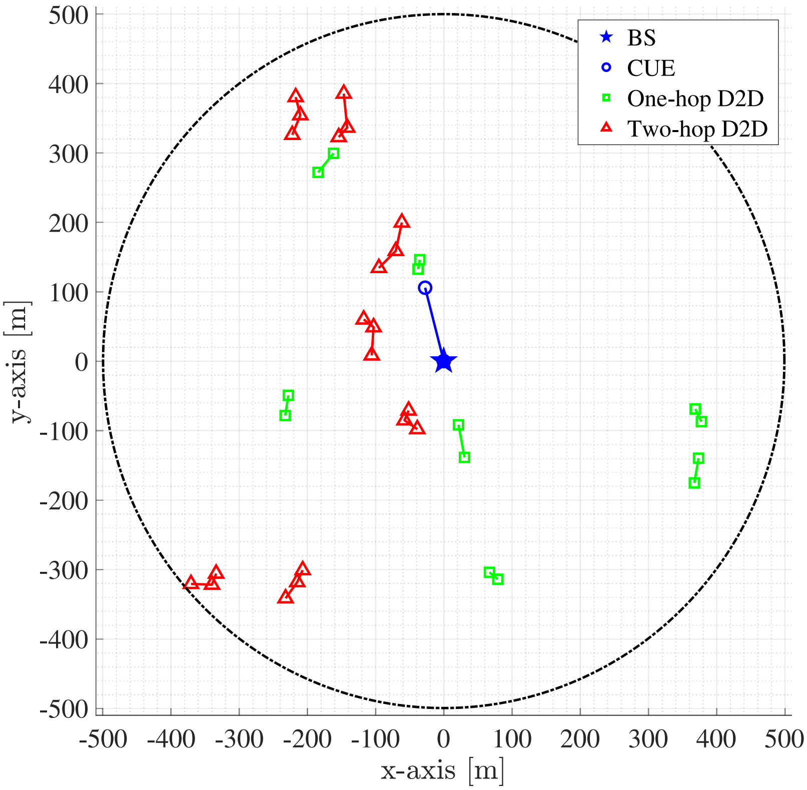


Fig. 4. The network generated model.

tolerance error $\epsilon = 10^{-5}$. For the unsupervised DNN and the DQL, we use a fully connected neural network; the parameters related to the unsupervised DNN and DQL models are summarized in Tables 3 and 4, respectively. The simulation results of the unsupervised DNN, DQL, and GS algorithms are obtained by Monte-Carlo simulation runs, averaging over 10^4 independent channel gains; for the ES performance evaluation, 10^3 independent channels are used considering the long computation time.

Fig. 5 shows the spectrum efficiencies of the ES, DQL, unsupervised DNN, and GS algorithms versus the maximum transmit power $p_{\max}$ when the numbers of one- and two-hop D2D pairs are set as $N_1 = N_2 = 2$. In Fig. 5, we simulate the performance of DQL with various numbers of discrete transmit power levels given by M values of $\{4,5,6,7,8\}$. The result shows that the performance of DQL increases as the transmit power level increases. Moreover, the simulation results for DQL and unsupervised DNN achieve a near global optimum and are much better than the GS results. In addition, DQL outperforms the unsupervised DNN as the maximum transmit power increases, but at the cost of higher computational complexity than unsupervised DNN.

Fig. 6 shows the spectrum efficiencies of ES, DQL, unsupervised DNN, and GS algorithms versus the number of one- and two-hop D2D pairs of $N_1 = N_2 \in \{2,3,\ldots,7\}$, where the number of actions in DQL is set to $M = 8$ and the maximum transmit power is $p_{\max} = 32$ dBm. The simulation graph shows that the DQL and the unsupervised DNN achieve near-global optimal solutions compared to GS. Moreover, the DQL shows higher performance than the unsupervised DNN as the number of one- and two-hop D2Ds increases. On the other hand, according to the time complexity analysis in Table 1, the DQL requires more computation time compared to the unsupervised DNN and GS.

In addition, it is noted that the optimality of the proposed ML-based approach against the global optimal solution cannot be directly compared with another method because there is no work dealing with solving optimization model for spectrum efficiency maximization in multi-hop D2D communication. However, we can indirectly verify it that the proposed DQL achieves 98.5 percent of the global optimal solution while conventional interference avoidance method such as dynamic source–destination pair-subchannel matching algorithm (for multi-hop but not D2D networks) achieves 90 percent of the global optimal solution.

It is noted that the SINR models given in (1), (3), (5), and (6) in the main manuscript do not consider intercell interference assuming a single cell network environment (Dai et al., 2017; Gui & Deng, 2018; Liu et al., 2017; Mohamed et al., 2020). However, if we consider intercell interferences from neighboring cells, the performance of our proposed ML schemes is expected to be slightly decreased.

Table 2
Network parameters.

Parameter	Value
Radius of a cell R	500 m
Distance between D2D link	10 m→20 m
Minimum transmit power $P_{\min}$	10 dBm
Rician small scale fading gain	5 dBm
Noise power spectrum density σ_A	−174 dBm/Hz
Path-loss exponent	3.6
Path-loss model for cellular link	$128.1 + 37.6 \log(\text{distance})$
Path-loss model for D2D links	$130 + 40 \log(\text{distance})$

Table 3
Unsupervised DNN's parameters.

Parameter	Value
Learning rate	0.01
Number of hidden layers L_{DNN}	10 layers
Optimizer	SGD
Batch size	1000

6. Conclusion

In this work, we studied the optimization model to maximize spectrum efficiency considering a multi-hop environment in the context

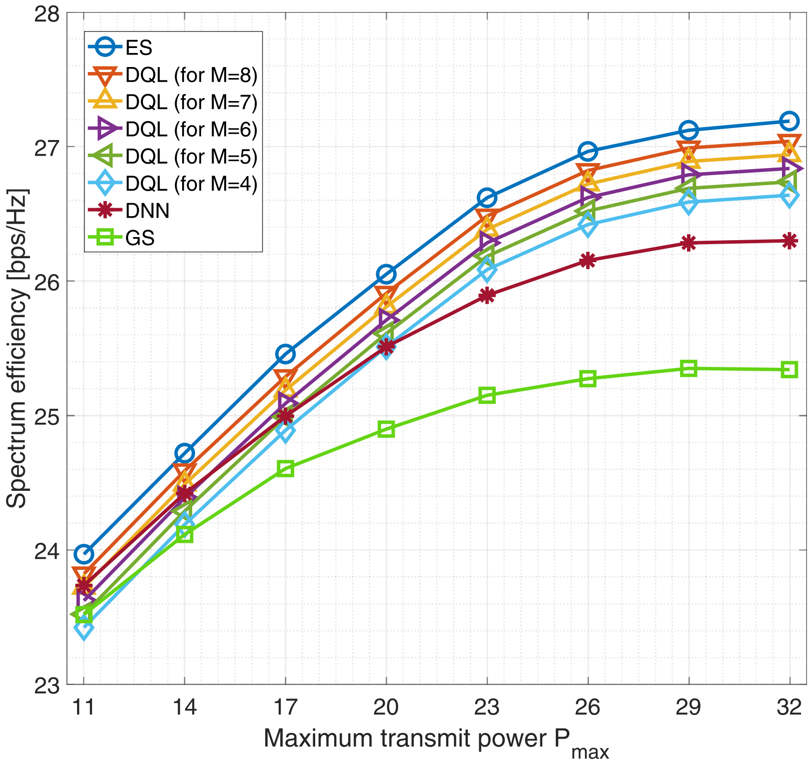


Fig. 5. SE vs. $p_{\max}$ when $N_1 = N_2 = 2$.

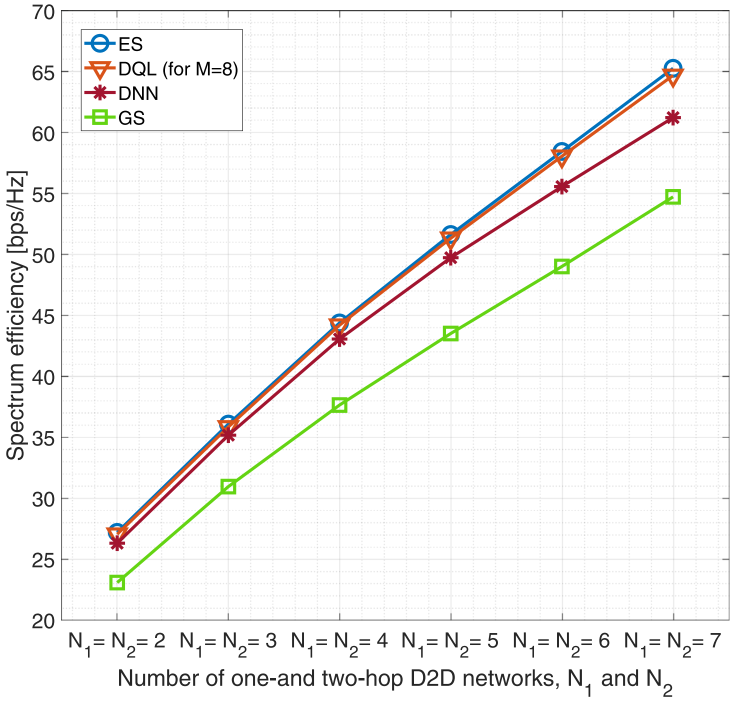


Fig. 6. SE vs. number of one- and two-hop D2D when $P_{\max} = 32$.

of a D2D underlay cellular network system. To get the solutions of the optimization model, we proposed unsupervised DNN and DQL algorithms and compared the performance of the proposed algorithms to optimization-based iterative methods together with time-complexity analysis. Based on the simulation results, we found that both the DQL and unsupervised DNN algorithms achieve a near-global solutions. Our result show that the DQL and the unsupervised DNN achieve near-global solutions. In addition, and the DQL outperforms the unsupervised DNN in terms of optimality, while the time complexity analysis shows that the unsupervised DNN requires the least computation time among the algorithms considered.

As stated in the bottom of Section 5, we assumed a single cell environment for performance evaluation in this work. Although the effect of intercell interference on the system performance may be marginal,

Table 4
DQL parameters.

Parameter	Value
Learning rate α	0.01
Number of hidden layers L_{DQL}	1 layer
ϵ-greedy ϵ	0.1
Discount factor γ	0.99
Replay memory size D	1000
Mini batch size	50
Optimizer	SGD
Activation function	ReLu

consideration of multicell environment can results in more practical and exact results. Therefore, in a future work related to resource management in D2D communication underlaid cellular network, we will consider intercell interference for performance evaluation assuming multicell environment.

CRediT authorship contribution statement

Sengly Muy: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Writing – original draft. **Jung-Ryun Lee:** Supervision, Writing – review & editing, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

Ali, K., Nguyen, H. X., Shah, P., Vien, Q.-T., & Ever, E. (2017). D2D multi-hop relaying services towards disaster communication system. In *2017 24th international conference on telecommunications (ICT)* (pp. 1–5). IEEE.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, *34*(6), 26–38. http://dx.doi.org/10.1109/MSP.2017.2743240.

Ban, T.-W. (2020). An autonomous transmission scheme using dueling DQN for D2D communication networks. *IEEE Transactions on Vehicular Technology*, *69*(12), 16348–16352. http://dx.doi.org/10.1109/TVT.2020.3041458.

Cartis, C., Gould, N. I., & Toint, P. L. (2010). On the complexity of steepest descent, Newton's and regularized Newton's methods for nonconvex unconstrained optimization problems. *SIAM Journal on Optimization*, *20*(6), 2833–2852.

Cormen, T. H., & Leiserson, C. E. (1990). *RRL, introduction to algorithms*. Cambridge: MIT Press.

Dai, J., Liu, J., Shi, Y., Zhang, S., & Ma, J. (2017). Analytical modeling of resource allocation in D2D overlaying multihop multichannel uplink cellular networks. *IEEE Transactions on Vehicular Technology*, *66*(8), 6633–6644.

Gui, J., & Deng, J. (2018). Multi-hop relay-aided underlay D2D communications for improving cellular coverage quality. *IEEE Access*, *6*, 14318–14338.

Jameel, F., Hamid, Z., Jabeen, F., Zeadally, S., & Javed, M. A. (2018). A survey of device-to-device communications: Research issues and challenges. *IEEE Communications Surveys & Tutorials*, *20*(3), 2133–2168. http://dx.doi.org/10.1109/COMST.2018.2828120.

Klaiqi, B., Chu, X., & Zhang, J. (2018). Energy-and spectral-efficient adaptive forwarding strategy for multi-hop device-to-device communications overlaying cellular networks. *IEEE Transactions on Wireless Communication*, *17*(9), 5684–5699.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lee, W., Kim, M., & Cho, D.-H. (2018). Deep power control: Transmit power control scheme based on convolutional neural network. *IEEE Communications Letters*, *22*(6), 1276–1279. http://dx.doi.org/10.1109/LCOMM.2018.2825444.

Lee, W., Kim, M., & Cho, D.-H. (2019). Transmit power control using deep neural network for underlay device-to-device communication. *IEEE Wireless Communications Letters*, *8*(1), 141–144. http://dx.doi.org/10.1109/LWC.2018.2864099.

Lee, K., Lee, J.-R., & Choi, H.-H. (2020). Learning-based joint optimization of transmit power and harvesting time in wireless-powered networks with co-channel interference. *IEEE Transactions on Vehicular Technology*, *69*(3), 3500–3504. http://dx.doi.org/10.1109/TVT.2020.2972596.

Levin, G., & Loyka, S. (2012). Amplify-and-forward versus decode-and-forward relaying: Which is better? In *22th international Zurich seminar on communications (IZS)*. Eidgenössische Technische Hochschule Zürich.

Li, Y. (2017). Deep reinforcement learning: An overview. arXiv preprint arXiv:1701.07274.

Liang, L., Ye, H., Yu, G., & Li, G. Y. (2019). Deep-learning-based wireless resource allocation with application to vehicular networks. *Proceedings of the IEEE*, *108*(2), 341–356.

Liu, C., He, C., & Meng, W. (2017). A tractable multi-rats offloading scheme on d2d communications. *IEEE Access*, *5*, 20841–20851.

Melki, L., Najeh, S., & Besbes, H. (2016). Interference management scheme for network-assisted multi-hop D2D communications. In *2016 IEEE 27th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)* (pp. 1–5). IEEE.

Mohamed, E. M., Elhalawany, B. M., Khallaf, H. S., Zareei, M., Zeb, A., & Abdelghany, M. A. (2020). Relay probing for millimeter wave multi-hop D2D networks. *IEEE Access*, *8*, 30560–30574.

Sedgewick, R., & Wayne, K. (2016). *Computer science: An interdisciplinary approach*. Addison-Wesley Professional.

Shaikh, F. S., & Wismüller, R. (2018). Routing in multi-hop cellular device-to-device (D2D) networks: A survey. *IEEE Communications Surveys & Tutorials*, *20*(4), 2622–2657. http://dx.doi.org/10.1109/COMST.2018.2848108.

Sun, H., Chen, X., Shi, Q., Hong, M., Fu, X., & Sidiropoulos, N. D. (2017). Learning to optimize: Training deep neural networks for wireless resource management. In *2017 IEEE 18th international workshop on signal processing advances in wireless communications (SPAWC)* (pp. 1–6). http://dx.doi.org/10.1109/SPAWC.2017.8227766.

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT Press.

Tehrani, M. N., Uysal, M., & Yanikomeroglu, H. (2014). Device-to-device communication in 5G cellular networks: challenges, solutions, and future directions. *IEEE Communications Magazine*, *52*(5), 86–92. http://dx.doi.org/10.1109/MCOM.2014.6815897.

Wei, L., Hu R. Q., Qian, Y., & Wu, G. (2016). Energy efficiency and spectrum efficiency of multihop device-to-device communications underlaying cellular networks. *IEEE Transactions on Vehicular Technology*, *65*(1), 367–380. http://dx.doi.org/10.1109/TVT.2015.2389823.

Zhao, Y., & Pottie, G. J. (2013). Optimal spectrum management in multiuser interference channels. *IEEE Transactions on Information Theory*, *59*(8), 4961–4976.

Zou, Z., Yin, R., Chen, X., & Wu, C. (2019). Deep reinforcement learning for D2D transmission in unlicensed bands. In *2019 IEEE/CIC international conference on communications workshops in China (ICCC workshops)* (pp. 42–47). http://dx.doi.org/10.1109/ICCChinaW.2019.8849971.