Computers, Materials & Continua
DOI:10.32604/cmc.2022.021941
Article

Tech Science Press

Deep Q-Learning Based Optimal Query Routing Approach for Unstructured P2P Network

Mohammad Shoab and Abdullah Shawan Alotaibi*

Department of Computer Science, Faculty of Science at Al Dawadmi, Shaqra University, Shaqra, Saudi Arabia
*Corresponding Author: Abdullah Shawan Alotaibi. Email: a.shawan@su.edu.sa
Received: 21 July 2021; Accepted: 22 August 2021

Abstract: Deep Reinforcement Learning (DRL) is a class of Machine Learning (ML) that combines Deep Learning with Reinforcement Learning and provides a framework by which a system can learn from its previous actions in an environment to select its efforts in the future efficiently. DRL has been used in many application fields, including games, robots, networks, etc. for creating autonomous systems that improve themselves with experience. It is well acknowledged that DRL is well suited to solve optimization problems in distributed systems in general and network routing especially. Therefore, a novel query routing approach called Deep Reinforcement Learning based Route Selection (DRLRS) is proposed for unstructured P2P networks based on a Deep Q-Learning algorithm. The main objective of this approach is to achieve better retrieval effectiveness with reduced searching cost by less number of connected peers, exchanged messages, and reduced time. The simulation results shows a significantly improve searching a resource with compression to k-Random Walker and Directed BFS. Here, retrieval effectiveness, search cost in terms of connected peers, and average overhead are 1.28, 106, 149, respectively.

Keywords: Reinforcement learning; deep q-learning; unstructured p2p network; query routing

1 Introduction

Machine Learning (ML) is widely used to analyze the data and to create or assisting in the formulation of predictions with the help of some algorithms and methods in the discipline of computer science and statistics [1–3]. In today's computerized societies, machine learning plays a key role, and ML-based components will undoubtedly be included in almost every gadget and machines to better control the operations and accustom to their environment. Machine learning is an excellent technique for resolving complex issues and has proven its efficacy in image and speech recognition, robot guidance, autonomous automobile guidance, telecommunications, and various other fields. ML is recognized to give (often) better outcomes than humans for tasks such as categorization and optimization [4].

Based on how learning is carried out, machine learning approaches are divided into four categories [1,2]: supervised, unsupervised, semi-supervised, and reinforcement. In supervised learning, input and output variables are used to learn the mapping function from input to output; the goal is to approximate the mapping function to the point where an output (also known as the label) can be accurately predicted from its associated input. Unsupervised learning, also known as learning without teacher, only input is used; the goal is to model the structure or distribution of data (e.g., data clustering) in order to understand specific features about data. Semi-supervised learning is similar to supervised learning, with the exception that not all observations are labeled (outputs). Finally, reinforcement learning is a behavioral psychology-inspired technique that enables system modeling based on agents interacting with their environments [5,6]. In the sequel, the paper only focuses on deep reinforcement learning (DRL) application to route the query in unstructured P2P networks.

The two types of Reinforcement Learning algorithms are model-free and model-based RL algorithms. Model-free RL algorithms predict future states and rewards without learning a model of their environment's transition function. The best examples of model-free learning are Policy Gradient, Deep Q-Networks, and Q learning, as they don't create a model of the environments transition function [6]. The existence of two things distinguishes reinforcement learning from supervised or unsupervised learning:

- An environment: This may be a maze, a video game, the financial market, or something else.
- An agent: This is the AI that learns how to operate and succeed in a given environment.

An iterative feedback loop is used to teach the agent how to operate in the environment. The state will alter as a result of the action taken by the agent, depending on the rewards received either won or lost. Here Fig. 1 shows a visual representation of this iterative feedback loop of actions, states, and rewards. The agent can learn which actions are beneficial in a given state by taking actions and obtaining rewards from the environment.

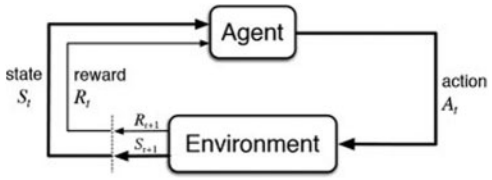

Figure 1: Iterative feedback loop of Deep Q Learning (Source: Sutton and Barto (2018))

DRL (Deep Reinforcement Learning) is a rapidly developing field that combines Reinforcement Learning and Deep Learning. It's also the most popular sort of Machine Learning since it can handle a wide range of complicated decision-making tasks previously unsolvable by a machine with human-like intelligence. The "deep" part of reinforcement learning refers to artificial neural networks with several (deep) layers that mimic the structure of the human brain [7]. Deep learning necessitates a lot of training data and a lot of computer power. The Deep Q-Learning algorithm is one of the most important principles in DRL. Instead of explicitly computing Q-values in Q learning through value iterations, a function approximator could be employed to estimate the effective Q-function, with neural networks currently being the method of choice [8]. Deep Q-Learning incorporates artificial neural networks into the Q-learning process, and a network that uses neural networks has been used to approximate Q-functions [9].

The complexity and variability of modern P2P networks, as well as end-user quality of service (QoS) and security requirements, service provider economics, and social inter-networking, have all expanded dramatically since the early P2P networks. P2P network technology has progressed from wired and manually configured networks to highly dynamic and autonomous networks, particularly unstructured P2P networks [4]. The majority of today's networks have evolved beyond human administration and configuration. As a result, machine learning approaches have been used to handle concerns and challenges in the networking area, such as traffic classification and prediction, fault management, configuration management, congestion control, QoS monitoring, energy efficiency, and security management [10–12]. The goal of ML applications to networks is to automatically learn the dynamics of P2P networks, including new flow arrivals, congestion points, topology changes, link quality, and energy consumption, in order to improve the service quality provided to end-users while optimizing network resources and providers' revenues.

In P2P networks, query routing is the problem of selecting paths to search the requested resource, while meeting QoS requirements, if any. Searching in P2P systems has been achieved by two techniques Blind search and Informed search. These search techniques relied on various query mechanisms in unstructured P2P systems, such as gossiping [13–15], random walk [16,17], k-walker [16,18], controlled flooding, and pure flooding [19]. Among these, gossiping is an attractive and widely adopted mechanism for modern query routing approaches [15,20]. Rather than selecting k neighbors at random, as k-walker, random walk, and controlled flooding do, gossiping routes the query to the k neighbors with the highest probability of holding the requested resource [14,15,20,21]. The current gossiping mechanism introduces the K-Neighbor Selection (K-NS) problem, which uses a particular scoring function to select the k relevant neighbors with the maximum score. To overcome this problem, two query routing approaches have been proposed: content-oriented and query-oriented routing methods [20]. An index about neighbor's collection has been created on each peer by previously collected data. This index with a specific scoring function assigns the neighbors' weights according to their query content and shared resources. As a result, the query has been routed to the first k high-scoring neighbors. The Query-oriented routing approach selects the suitable k anticipated neighbors for future queries based on data acquired from previously sent queries. In order to define or develop scoring functions, several machine learning and data mining techniques have been applied [20,21].

Query-oriented methods are more efficient than content-oriented methods as they only use data collected from previous queries, so no additional network communication is required to create and maintain the index [14,15,20,21]. In this respect, a novel query routing algorithm, 'Deep Reinforcement Learning Route Selection (DRLRS)' has been introduced for efficient neighbor selection in unstructured P2P networks using the Deep Q-Learning algorithm. Deep Q-Learning is a part of reinforcement learning where intelligent agents take actions in an uncertain and potentially complex environment to maximize cumulative reward. The primary objective of the DRLRS algorithm is to reduce the number of connected peers and overhead in order to achieve high retrieval effectiveness with lower communication costs. This study's three key contributions are listed below, along with their novelty.

(1) This study introduced a fully distributed query routing algorithm DRLRS for unstructured P2P network that depends on the locally gathered data for learning on every peer.
(2) The K-NS problem has been provided a new formulation, and deep reinforcement learning has been considered to formalized for natural model for this issue in which a peer learns through performing actions that result in a reward after completion of every search. As a result, a peer's aim is to discover a neighbor selection policy that maximizes the total

reward. To achieve this, a Deep Q-Learning algorithm is used at every forwarder peer to select the k neighbors for each search query. Every selected neighbor generates a binomial reward that indicates its capacity to locate the query's relevant resource. To optimize the cumulative reward, a peer must learn a K-NS strategy to generate greater user satisfaction.

(3) The cold start problem, which is a significant flaw in query routing methods, is also addressed. This issue arises when a new peer enters the network, and traditional query-oriented approaches presume that the relevant peer has already submitted a specific number of queries and replies, which are saved in the sender log file as training data. However, for the recently connected peer, this assumption is incorrect. As a result, existing approaches for generating training data required the deployment of the k-random walker strategy for a given number of queries, which surely resulted in poor performance at first. To address this issue, a Deep Q-Learning-based route selection algorithm was developed, which learns about neighbors and chooses the neighbor with the largest estimated reward based on prior queries.

2 Existing Systems and Algorithms

In this section, a discussion has been made about some unstructured P2P systems' routing methods. The main objective of these systems and algorithms is to find those peers sharing related resources for the queries with decreasing number of connected peers and network traffic. Therefore, these systems and algorithms have been divided in to the following three categories:

- **Basic Searching Algorithms:** The most common and foremost algorithms to search relevant resources in unstructured P2P systems are Breadth-First Search (BFS) or flooding [22,23]. In this algorithm, the forwarder peer which initiates or receives a query first searches the related resource in the local collection of resources. When a resource is found in the local collection, it will reply to the requesting peer with a message containing a list of related resources that have been retrieved. The message is forwarded through the reverse request path to the requesting peer. This is where the query propagation process stops until the Time To Live (TTL) reaches a specific predefined value [22]. BFS tries to find the maximum number of results. However, this generates a large number of messages and increases the number of connected peers with heavy network traffic compared to other approaches [23]. Napster and Gnutella are the suitable examples who implemented this routing method [24,25].

  Although, this method is robust, it is prolonged and consumes network resources excessively. Hence, several other methods such as Depth First Search (DFS), controlled flooding, k-walker, and gossiping were introduced as an improvement of BFS. In contrast to BFS, which sends queries to all neighbors, DFS allows each peer to choose a candidate neighbor to whom the query should be sent. If the query forwarder doesn't get a response within a specific amount of time (TTL), the peer chooses another neighbor to deliver the query [23]. This is an iterative process that is performed until the query is answered or until all of the neighbors have been queried. FreeNet is an example of a P2P system that uses the DFS approach [26,27]. Instead of sending the query to all neighboring peers as BFS does, the controlled flooding forwards the query to the arbitrarily selected k neighbors. When DFS and controlled flooding are combined, another method called K-Walker is created. The peer sends the query to k randomly chosen neighbors, who then send the request to the next random neighbor, and so on, until the relevant peer is discovered or the TTL value is reached [22]. Another exciting technique widely used in intelligent query routing methods

is Directed BFS. The primary purpose of this searching technique is to direct the query to k neighbors that may have appropriate query resources [14,15,21], rather than arbitrarily selecting neighbors. Therefore, a specific scoring algorithm is employed to identify the k highest scoring relevant neighbors. The particular scoring function utilizes the previously collected meta-data about neighbors to rank neighbors concerning the query content and shared resource. Therefore, the query is directed to the top k high-scoring neighbors [28,29].

- **Content-Oriented Algorithms:** These algorithms utilize information extracted from each peers' shared content and create a local index with global knowledge. Further, this index provides an approximate view of the entire network shared content with peers' profiles. Therefore, a query forwarder peer will be able to route the query with better retrieval effectiveness efficiently. An algorithm based of notation of semantic communities is Improved Niche Genetic Algorithm (INGA) [30]. Each peer is expected to play a specific role in the network, such as recommender, content provider, and so on. INGA can determine the most suitable peer to redirect the query by the function associated with it. Moreover, each peer collects and manages information and facts locally that creates a topical knowledge of the peer. Scalable Query Routing (SQR) is another algorithm that aim to achieve low bandwidth [31]. This algorithm keeps a routing table on each peer that, based on previous experience, advises the position of items in the network. Furthermore, the Exponentially Decaying Bloom Filter (EDBF) data structure compresses probabilistic routing tables and enables for efficient query propagation [23].
  Another algorithm similar to directed BFS and intelligent search is the routing index-based search algorithm. The entire search process has been guided by three approaches using neighbor information, but the information collection and utilization is differ. In directed BFS, neighbors' information has been utilized by only the query issuing peer; the rest of the peers uses BFS to forward the query if they don't find the requested resource in their local collection. Whereas, Intelligent search makes use of information from earlier queries that have been replied by neighbors. However, the routing-index based search method stores information about the number of documents and subjects of documents available at neighbor peers. This information helps to select the best neighbor peer to forward the query [32]. For query routing, specific P2P systems use a classification problem. The classifier in this issue tries to classify an item based on particular features. The Semantic Overlay Model adepts the classification problem to locate appropriate peers to answer a specific query. This model routes the query to the semantically similar peers instead of broadcasting the query. To classify into categories, semantic vectors have been used that represents the peers' semantic similarity and uses meta-information to classify peers by interests. As a result, it increases the recall rate and, at the same time, reduce the hops and messages [33].

- **Query-Oriented Algorithms:** The Query-oriented routing methods relies on previously collected data in previously sent queries in order to select k neighbors for the upcoming queries. To attain this, different statistical and intelligent methods have been utilized to define or learn the scoring function. In this respect, Alanazi and Yeferny, (2019) proposed a reinforcement learning (RL) based query routing approach, which is based on a classical RL problem called Multi-Armed Bandit (MAB) for the K-NS problem. Initially, to select one relevant neighbor for each search query, three MAB-based learning algorithms are used. These three algorithms are Epsilon-Greedy (EG), Upper Confidence Bound (UCB), and Thompson Sampling (TS) and called as 1-neighbour-Selection (1-NS) algorithms. Further, one of the three algorithms run for K times in the K-NS algorithm to select k relevant

neighbors. Shamshirband and Soleimani (2021) introduced a novel and efficient query routing algorithm using reinforcement learning with learning automata (LA) called "learning automata adaptive probabilistic search (LAAPS) algorithm" and performs a keyword-based search on the routing tables with scores. In this method, the LAAPS estimation of the current state of the P2P system is related to the Markov process of the discrete-parameters of the discrete-state, and depends on the revising action and the selection of the next update stage of each action. Therefore, the goal is to update the action based on the information obtained by interacting with the environment and finding the correct peer for the route. Each peer requesting for routing has an LA with certain operations, and each adjacent peer is considered to be the neighborhood of that node [34]. Kalogeraki et al. (2002) introduced a query-oriented method called Intelligent Search (IS). This method manages a routing table in which neighbor configuration files are stored and provides a simple vector representation of each neighbor configuration file, including the recently processed queries by the neighbor and the number of queries received. Every time, a peer initiates or receives a query, it conducts an online evaluation of its neighbors about its configuration and the content of the query. Then send the query to the first k relevant neighbors. Formal Concept Analysis (FCA) theory [35] has been used by Arour et al. [20]. for efficient query routing in P2P information retrieval systems . Typically, It is a method that extracts interested relational data sets, implicitly extracts user interests from previously submitted queries and associated query-hits, and stores the user interests in a local per peer knowledge base. To route the query, the learning query routing algorithm first calculates the similarity between the user's interest and the keywords in the query, and then identifies the peers that are closest to the query contained in the user's interest as k related peers, and these peers are selected as forwarding peer for the query Alanazi et al. [22]. (2009) presented a supervised machine learning algorithm, called "Route Learning", to solve the K-NS problem. In the learning phase of this routing algorithm, the data about sent queries and related queries is accumulated. In addition, adjacent peers and forthcoming queries are regarded as classes and new objects to be classified in the supervised multi-classification problem. Therefore, neighbors (i.e., classes) to which the query is stipulated are considered as the k relevant neighbors [21,22].

Every peer have the same responsibility in pure P2P unstructured and decentralized network. However, some query methods in this type of P2P network uses a notation of super-peer like in Backpressure algorithm [36]. In this algorithm, super-peer resolves the query or forward the query to other super-peers for their underling peers. Whenever super-peers received a query first they check the desired resource they have, as well as check their underling group of peers. This algorithm is a query-oriented algorithm that utilize the previously collected information to decide the route of a query. Another algorithm called Self Learning Query Routing Algorithm that aims to improve it's knowledge by learning the peer's interest based on the peer's previous search history [37]. Moreover, the rank of friendship between two peers is determined by the number of shared files. So, only friend peers are involved in query routing initially. If the friend peers doesn't have the desired resource, a broadcast search is performed. Previous results of searching allows peers to learn gradually about other peers in the network that share the same interest [23].

3 Problem Formulation

According to the literature, it has been observed that most of the searching techniques in the unstructured P2P network are based on the gossiping method. In this method, the query forwarder

peer selects k neighbor from the set of N neighbors according to the relevance for the query. Moreover, there are some other approaches relies on various specific scoring functions to select k neighbors with the highest score. The formulation to select k neighbors in this study is different from existing methods. Thus, a natural model called "Deep Reinforcement Learning" for this issue has been considered, in which an agent learns from its previous actions that produce rewards. The agent's aim is to find a selection criteria that maximize the collective reward. In this way, the forwarder peer selects k neighbors using the deep q learning algorithm for each search query. Every selected neighbor provides a reward value that explicit its ability to produce a relevant resource for the query. In fact, for higher user satisfaction, the agent should learn neighbor selection criteria to maximize the collective reward. However, most existing RL algorithms are designed to select only one neighbor (action) from a set of various neighbors (actions), whereas the k neighbor selection algorithm must select several neighbors for each query. To address this issue, a generic neighbor selection algorithm has been introduced that uses a Deep Q-Learning algorithm to select k neighbors.

4 Deep Q-Learning

Learning Q-values–the value of taking specific action in a given condition–is the foundation of Q-Learning. Deep Q-Networks (DQNs) are similar to tabular Q-learning in principle, but instead of keeping all of Q-values in a look-up table, these have been represented as a neural network in Deep Q-Learning. This allows for more generalization and a more diverse representation [38,39]. DQN is a combination of reinforcement learning and deep learning algorithm. This is motivated by the fact that the storage space of a Q-table in the traditional reinforcement learning algorithm Q-Learning is limited, whereas the state in the real world or even the virtual world is nearly infinite, making it impossible to build a Q-table that can store an ample state space. However, in machine learning, there is a method that is particularly good at this, namely, the neural network, which can take the state and action as inputs and then obtain the Q-value of the action after neural network analysis, eliminating the need to record the Q-value in the table and instead of using the neural network to predict the Q-value directly [40].

In Deep Q-Learning, the user stores all past experiences in memory, as well as the future action determined by the network's Q-output. Thus, Q-network gains the Q-value at state S_t, and at the same time target network (Neural Network) calculates the Q-value for the state S_{t+1}(next state) to make the training stabilized and blocks the abruptly increments in Q-value count by copying it as training data on each iterated Q-value of the Q-network. It has already been demonstrated that using a random batch of previous data boosts the stability of neural network training [39]. So, to increase agents' performance, Deep Q-Learning employs yet another concept: experience replay, which is nothing more than the stocking of previous experiences. The target network uses experience replay for training and by the Q-network for calculating the Q-value.

4.1 Target Network and Experience Replay

Switching to a DQN representation for Q-learning brings a few challenges that the tabular versions do not have to deal with. This is induced by the function approximator's nonlinear deep neural network and data sequence-dependent correlations and frequent updates to Q approximation. The correlations arise as a result of taking action a at each time step t while looking at an episode. The reward received at the time $t(R_t)$ is strongly correlated to the state and action at the time $t-1$, which is linked to the state and action at the time $t-2$ and so on. Because of the path dependence, training a DQN is difficult [9].

This is handled with experience replay, which is a memory bank comprising different states, actions, and rewards that have been randomly sampled. When the learning has been updated by network smoothing, the random sampling breaks any sequence dependency in the data. There's also a neuroscience component here, as experience replay is thought to occur in the brain to encode long-term memories [38].

Estimation of the next state is the second source of instability. The following equation could be used to update values in tabular Q-learning.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_t + \gamma max_a Q(s_{t+1}) - Q(s_t, a_t)] \quad (1)$$

With the tabular method, only $Q(s_t, a_t)$ is updating, however, using the DQN technique, this update is slightly different, and the entire network is updated at each step. When the whole network is updated, the optimum action is estimated at the next state ($max_a Q(s_{t+1})$). Because the target is constantly moving, the backpropagating error for the identical states and actions varies from one update to the next, making it more difficult for the network to learn [9,38].

Here, α is the learning rate, γ is the discount factor and max_α is the maximum reward attainable in the state. The values of both α and γ are generally set between 0 and 1. Setting α to 0 means that the Q-values are never updated, and nothing is learned. However, setting α to a high value such as 0.9 means that learning can occur quickly. Furthermore, if γ is closer to 0, the agent will tend to consider only immediate rewards. If γ is closer to 1, the agent will consider future rewards with greater weight and be willing to delay the reward. Considering previous studies, the values of α and γ are set to 0.5 and 0.9 in the current study.

To deal with this, a target network is built that is a clone of the training neural network but only copies it every N time steps. This means that the error will stay steady for a while, enabling the network to learn, before the target network is updated to a better approximation, allowing the network to learn all over again.

4.2 Loss Function

Because a deep neural network represents the Q-function, an alteration to the rule is required to make it relevant to backpropagation. This loss function must also be differentiable in terms of the network's (θ) parameters. The DQN loss function works in the same way as the tabular Q-learning update rule: an action is performed using the Q-function, and the reward received is compared to the best action estimate in the new state [38,39].

$$L = (R_t + \gamma max_a(Q(s_{t+1}; \theta_t)) - Q(s_t, a_t))^2 \quad (2)$$

Still having rewards and Q-value estimations for states and actions, but now there are two networks designated as θ and θ_t, which are current network and target network, respectively. Errors are squared here to penalize massive errors far more severely than minor ones. To update the network, the backpropagation algorithm, which takes the derivative of all the values in the layers and modifies the related layers, might be utilized [38,39].

Algorithm 1: Deep Q-Learning Algorithm with experience replay (Source: DataHubs (2019); Mnih et al. (2015); Fan et al. (2020))

Initialize replay memory D to capacity N
Initialize action-value function Q with random weight θ

(Continued)

Initialize target action-value function $\hat{Q}$ with weights $\theta^{-} = \theta$
For episode = 1, M do
Initialize sequence $s_1 = x_1$ and preprocessed sequence $\emptyset_1 = \emptyset(s_1)$
For $t = 1$, T do
With probability ε select a random action a_t
otherwise select $a_t = arg\, max_a Q(\emptyset(s_t), a; \theta)$
Execute action a_t in emulator and observe reward r_t and image x_{t+1}
Set $s_{t+1} = s_t, a_t, x_{t+1}$ and processes $\emptyset_{t+1} = \emptyset(s_{t+1})$
Store transition $(\emptyset_t, a_t, r_t, \emptyset_{t+1})$ in D
Sample random minibatch of transitions $(\emptyset_j, a_j, r_j, \emptyset_{j+1})$ from D
Set $y_j = \begin{cases} r_j, & \text{if episode terminates at step } j+1 \\ r_j + \gamma max_a \hat{Q}(\emptyset_{j+1}, a_j, \theta^{-}), & \text{otherwise} \end{cases}$
Perform a gradient decent step on $(y_j - Q(\emptyset_j, a_j; \theta))^2$ with respect to the network parameters θ
Every C steps reset $\hat{Q} = Q$
End For
End For

5 Proposed Algorithm

A generic Deep Reinforcement Learning Route Selection (DRLRS) algorithm has been proposed to select the k neighbors that rely on the Deep Q-Learning algorithm to forward the query. Thus, every peer can locally determine other peer's connections. Initially, every peer selects it's neighbor arbitrarily. After each query is given within the network, the choice of k neighbors has been made by the Q-value for the forthcoming query. In detail, each peer holds a Deep Q-Learning algorithm for every state and action related to its neighbor.

In the proposed DRLRS algorithm, the inputs are the set of all neighbors $N = n_1, n_2, \ldots, n_m$, the search query q_t at time t, and the number of neighbours to be selected k. The result is the set of S which is relevant neighbors to the forward query q_t. On every iteration $i = 1, \ldots, k$, DRLRS depends on the Deep Q-Learning algorithm to select from the set $N = n_1, n_2, \ldots, n_m$ the neighbor n_j with the highest expected probability of success according to Deep Q-Learning algorithm. The selected neighbor n_j at iteration i is than added to the set of relevant neighbors S and removed from the set of neighbors N. Algorithm 2 illustrates the proposed DRLRS algorithm.

Algorithm 2: Deep reinforcement learning based route selection algorithm
Input:
q_t: search query at time t
$N = n_1, n_2, \ldots, n_m$: set of all neighbours
k: number of neighbours to be selected for q_t
Output:
S: set of neighbours to be selected
$S = \emptyset$:

(Continued)

While $i = 1 \ldots\ldots k$ do
  select neighbours n_j using Deep Q-Learning algorithm
  $S = S \cup n_j$
  $N = N \backslash \{n_j\}$
End

6 Performance Evaluation

To evaluate the performance of the proposed algorithm, first, a network simulator has been developed in python language using pyvis library, which is meant for quick generation of visual network graphs and Fast Network Simulation Setup (FNSS) toolchain to setup a network experiment scenario. Moreover, Apache STORM has been configured and used by system libraries for distributed real-time computations. Fig. 2 illustrates the developed unstructured P2P network for simulation. An Apple MAC book pro with Intel Core i7 Quad-core 2.9 GHz CPU and 16.0 GB RAM was used to perform all experiments. A well-known document collections acquired from TREC has been used as a dataset that contains around 25,000 documents. In the experimental topology, 5000 queries were distributed among 1000 peers. The TTL of the query is set to 10, and the number of neighbors to be selected by the forwarder peer is set to 4. In this simulation, the system is considered in fix mode, where significant improvements are observed, and the issue of 'churn' is not explicitly considered [41].

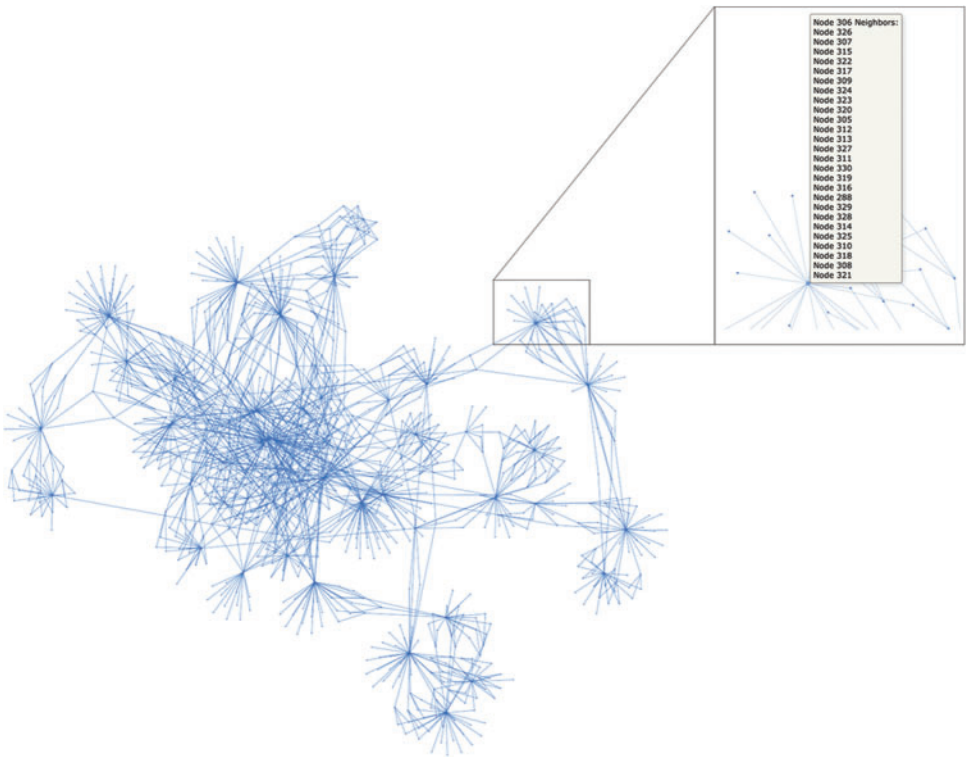

Figure 2: Topological representation of developed unstructured P2P network

In the developed network simulator, to experiment with Deep Q-Learning, it is necessary to create a custom environment. An environment in reinforcement learning must have four traits in general: (i) a state representation, (ii) an action representation, (iii) a reward function, and (iv) a time step function. The developed custom environment contains all the above four traits to select the k relevant neighbors for the current query. An environment-class called "EnvDrlrs" was created with the help of the Open AI Gym toolkit, and the different functions were implemented from the base gym.Env class.

6.1 Complexity Analysis

The time complexity of the DRLRS algorithm was the same as that of Apache STORM, which is $O(kT|E)$, where k represents the number of training epochs and T represents the time. For space complexity, the DRLRS algorithm does not store any Q-table, so the space complexity could be calculated as $|S||A|$, where S was the explored network state, and $A = 4$, a set of action selections for this experiment. The DRLRS algorithm does not occupy too much space, as it stores only variables.

6.2 Efficiency Measures

Alanazi and Yeferny (2019) proposed efficiency measures as retrieval effectiveness and the search cost of their study about query routing based on reinforcement learning in the P2P system. In this study, the same evaluation measures have been considered. The retrieval effectiveness of DRLRS and its competitors has been evaluated using a Recall metric. The recall $R(q_i)$ for the query q_i is defined as follows:

$$R(q_i) = \frac{RRR}{RLR'} \tag{3}$$

where, RRR is number of relevant resources retrieved and RLR is the number of relevant resources. The cumulative average recall up to n queries CAR_n is defined as follows:

$$CAR_n = \frac{\sum_{i=1}^{n} R\,(q_i)}{n} \tag{4}$$

For search cost evaluation, two metrics have been used:

I. Connected Peers (CP): for query q_i, $CP(q_i)$ is the number of connected peers. The average number of connected peers for n sent queries ($ACPn$) is defined as:

$$ACP_n = \frac{\sum_{i=1}^{n} CP(q_i)}{n} \tag{5}$$

II. Average Overhead: This is measured by the number of exchanged messages for the query q_i. The average overhead up to n sent queries (AO_n) is defined as:

$$AO_n = \frac{\sum_{i=1}^{n} Overhead(q_i)}{n} \tag{6}$$

7 Results and Discussion

This section discussed the results based on the retrieval effectiveness and search cost compression for a single peer executing 5000 queries. Furthermore, ten peers were selected randomly that

executed 5000 different queries per peer, and the efficiency was measured in terms of number of average connected peers and the average time taken.

7.1 Retrieval Effectiveness Compression

Fig. 3 represents the average recall of k-Random Walker, Directed BFS, and DRLRS. As shown in figure k-Random Walker has a low but stable recall of around 0.45. Whereas, initially, the recall of Directed BFS is low as k-Random Walker than it shows significant improvement after each calculation of approximation. Moreover, it has been observed that the recall of the proposed DRLRS algorithm rapidly increased after sending few queries. The average recall of k-Random Walker, Directed BFS, and DRLRS for all sent queries is 0.4, 0.85, and 1.28, respectively. These outcome shows that the balance between exploration and exploitation has been efficiently achieved by DRLRS, which relies on the Deep Q learning algorithm. Moreover, it has been observed that the routing performance of DRLRS improves continuously and, therefore, finishes the cold start phase in significant time.

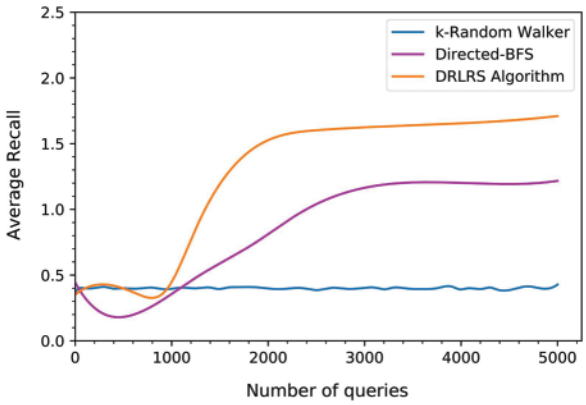

Figure 3: Evolution of average recall

7.2 Search Cost Compression

The search cost of the proposed routing algorithm DRLRS has been illustrated in Figs. 4 and 5 with the comparison of k-Random Walker and Directed BFS. During the simulation period, the number of connected peers and average overhead per query of k-Random Walker were found to be high but constant at around 135 and 240, respectively. In addition, the number of connected peers and average overhead per peer query of Directed BFS initially increased, but after around half number of queries, it decreased. Indeed, the Directed BFS starts with Zero knowledge and relies on approximation function, which leads to achieving lower performance initially but increases the performance after some time. However, Directed BFS improved after each peer's value approximation and shows the number of connected peers and average overhead around 123 and 224, respectively. Moreover, the search cost of the proposed DRLRS algorithm is going down rapidly and outperform k-Random Walker and Directed BFS. The connected peers and average overhead per query are 106 and 149, respectively, that shows the DRLRS algorithm locates more peers holding relevant resources for the query.

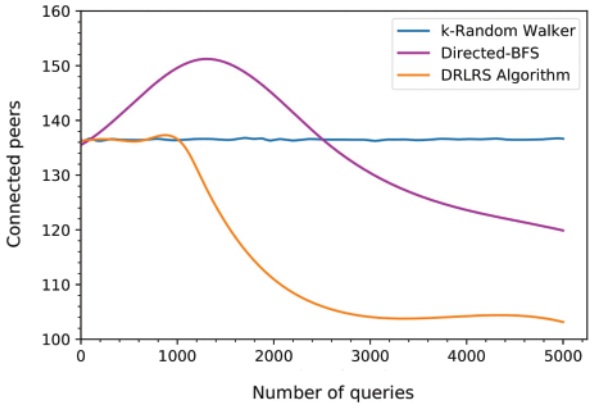

Figure 4: Evolution of number of connected peers

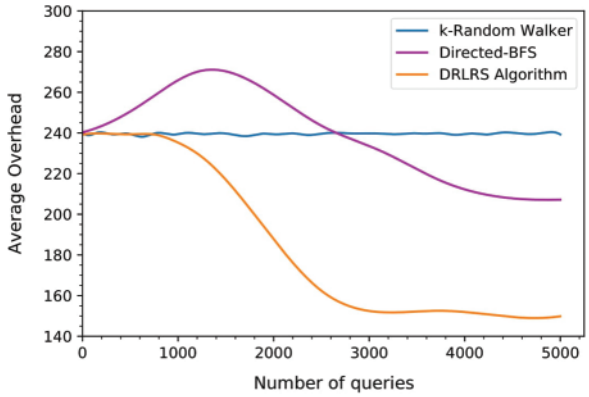

Figure 5: Evolution of average overhead

Previous results have been obtained by one peer executing 5000 queries and selecting suitable neighbors for the desired resource. Further, ten peers have been selected randomly, and each peer executed 5000 different queries to evaluate the performance of DRLRS and it's competitors. Fig. 6 shows the average connected peers on each selected peer. Moreover, the average time taken in executing queries by each selected peer has been shown in Fig. 7. The results shows that DRLRS outperforms it's competitors and set fewer peers to find the requested resource in less time.

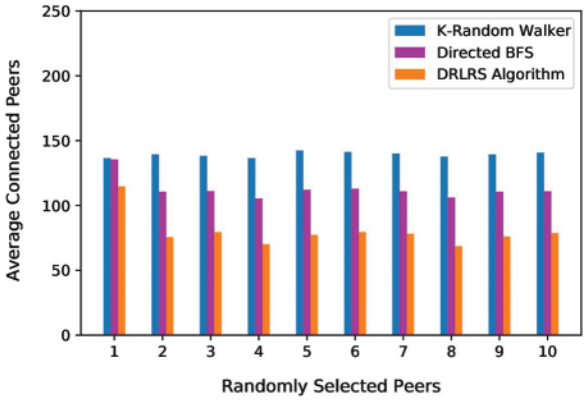

Figure 6: Evolution of average connected peers for randomly selected peers

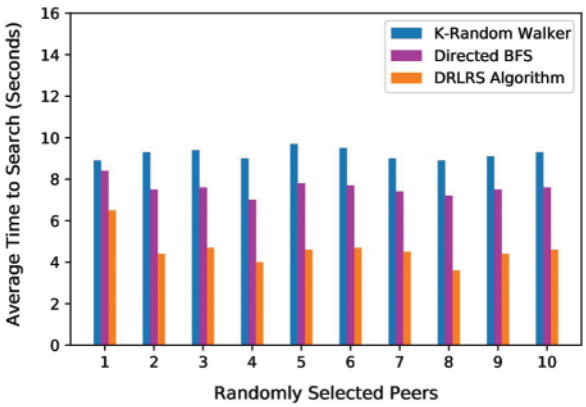

Figure 7: Evolution of average time taken to search by randomly selected peers

8 Conclusion

In this paper, a novel approach has been proposed to route the query efficiently to find the pertinent resource. In this regard, the query routing problem has been described as a deep reinforcement learning problem, and a fully distributed strategy to solving it has been developed. Therefore, the Deep Q-Learning based query routing algorithm DRLRS is introduced to select the neighbors intelligently in order to improve the performance of the P2P network. Results shows that the DRLRS systematically learns from the previously sent queries and efficiently selects the best neighbors holding the relevant resource for the current query. The proposed algorithm improves the retrieval effectiveness and search cost continuously and outperforms the k-Random Walker and Directed BFS. However, in the future, the churn problem can be included in the study by taking into the account that at any time peers could join and leave the network that leads the frequent link changes between the peers, which may affect the performance of neighbors selection.

Funding Statement: Authors would like to thank the Deanship of Scientific Research at Shaqra University for supporting this work under Project No. g01/n04.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] T. M. Mitchell, *Machine Learning*, 1st ed., vol. 1, New York: McGraw-Hill Education, pp. 1–432, 1997.
[2] K. P. Murphy, "Machine learning: A probabilistic perspective," *The MIT Press*, vol. 1, pp. 1–1104, 2012.
[3] G. James, D. Witten, T. Hastie and R. Tibshirani, "An introduction to statistical learning: With applications in R," *Springer Texts in Statistics*, vol. 1, pp. 1–426, 2013.
[4] Z. Mammeri, "Reinforcement learning based routing in networks: Review and classification of approaches," *IEEE Access*, vol. 7, pp. 55916–55950, 2019.
[5] R. S. Sutton and A. G. Barto, "Reinforcement learning: An introduction," in *Adaptive Computation and Machine Learning Series*, 2nd ed., Cambridge, MA, USA: Bradford Books, The MIT Press, 2018.
[6] K. Ali, C. Y. Wang and Y. S. Chen, "A novel nested q-learning method to tackle time-constrained competitive influence maximization," *IEEE Access*, vol. 7, pp. 6337–6352, 2019.
[7] M. Kim, M. Jaseemuddin and A. Anpalagan, "Deep reinforcement learning based active queue management for IoT networks," *Journal of Network and Systems Management*, vol. 29, no. 3, pp. 34, 2021.
[8] J. Torres, "D RL 01: A Gentle Introduction to Deep Reinforcement Learning," [Online]. Available: https://towardsdatascience.com/drl-01-a-gentle-introduction-to-deep-reinforcement-learning-405b79866bf4 [Accessed on July 14, 2021].
[9] DataHubs, "Deep Q-learning 101," [Online]. Available: https://www.datahubbs.com/deep-q-learning-101/ [Accessed on July 14, 2021].
[10] S. Ayoubi, N. Limam, M. A. Salahuddin, N. Shahriar, R. Boutaba *et al.*, "Machine learning for cognitive network management," *IEEE Communications Magazine*, vol. 56, no. 1, pp. 158–165, 2018.
[11] R. V. Kulkarni, A. Förster and G. K. Venayagamoorthy, "Computational intelligence in wireless sensor networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 13, no. 1, pp. 68–96, 2011.
[12] M. Wang, Y. Cui, X. Wang, S. Xiao and J. Jiang, "Machine learning for networking: Workflow, advances and opportunities," *IEEE Network*, vol. 32, no. 2, pp. 92–99, 2018.
[13] M. Dietzfelbinger, "Gossiping and broadcasting versus computing functions in networks," *Discrete Applied Mathematics*, vol. 137, no. 2, pp. 127–153, 2004.
[14] D. N. da Hora, D. F. Macedo, L. B. Oliveira, I. G. Siqueira, A. A. F. Loureiro *et al.*, "Enhancing peer-to-peer content discovery techniques over mobile ad hoc networks," *Computer Communications*, vol. 32, no. 13, pp. 1445–1459, 2009.
[15] T. Yeferny, S. Hamad and S. B. Yahia, "Query learning-based scheme for pertinent resource lookup in mobile P2P networks," *IEEE Access*, vol. 7, pp. 49059–49068, 2019.
[16] Z. Jia, J. You, R. Rao and M. Li, "Random walk search in unstructured P2P," *Journal of Systems Engineering and Electronics*, vol. 17, no. 3, pp. 648–653, 2006.
[17] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proc. of the 16th Int. Conf. on Supercomputing*, New York, USA, pp. 84–95, 2002.
[18] V. Kalogeraki, D. Gunopulos and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in *Proc. of the 11th Int. Conf. on Information and Knowledge Management*, New York, USA, pp. 300–307, 2002.
[19] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham and S. Shenker, "Making gnutella-like P2P systems scalable," in *Proc. of the 2003 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communications*, New York, USA, pp. 407–418, 2003.

[20] K. Arour and T. Yeferny, "Learning model for efficient query routing in P2P information retrieval systems," *Peer-to-Peer Networking and Applications*, vol. 8, no. 5, pp. 741–757, 2015.
[21] S. Ciraci, İ KörpeogÎu and Ö. Ulusoy, "Reducing query overhead through route learning in unstructured peer-to-peer network," *Journal of Network and Computer Applications*, vol. 32, no. 3, pp. 550–567, 2009.
[22] F. Alanazi and T. Yeferny, "Reinforcement learning based query routing approach for P2P systems," *Future Internet*, vol. 11, no. 12, p. 253, 2019.
[23] A. L. Nicolini, C. M. Lorenzetti, A. G. Maguitman and C. I. Chesñevar, "Intelligent query processing in P2P networks: Semantic issues and routing algorithms," *Computer Science and Information Systems*, vol. 16, no. 2, pp. 409–442, 2019.
[24] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network," in *Proc. First Int. Conf. on Peer-to-Peer Computing*, Linköping, Sweden, pp. 99–100, 2001.
[25] Wikipedia, "Napster," [Online]. Available: https://en.wikipedia.org/w/index.php?title=Napster&oldid=1001516831 [Accessed on February 07, 2021].
[26] I. Clarke, O. Sandberg, B. Wiley and T. W. Hong, "Freenet: A distributed anonymous information storage and retrieval system," in *Proc. Designing Privacy Enhancing Technologies: Int. Workshop on Design Issues in Anonymity and Unobservability Berkeley*, CA, USA, H. Federrath, Ed. Berlin, Heidelberg, Springer, pp. 46–66, 2001.
[27] Freenet, "Freenet," [Online]. Available: https://freenetproject.org/index.html [Accessed on February 10, 2021].
[28] J. Lu and J. Callan, "Content-based retrieval in hybrid peer-to-peer networks," in *Proc. of the 12th Int. Conf. on Information and Knowledge Management*, New Orleans, Louisiana, USA, pp. 199–206, 2003.
[29] W. Shen, S. Su, K. Shuang and F. Yang, "SKIP: An efficient search mechanism in unstructured P2P networks," *The Journal of China Universities of Posts and Telecommunications*, vol. 17, no. 5, pp. 64–71, 2010.
[30] A. Löser, S. Staab and C. Tempich, "Semantic methods for P2P query routing," in *Proc. of Multiagent System Technologies*, Berlin, Heidelberg, pp. 15–26, 2005.
[31] R. Kumar, P. Jawale and S. Tandon, "Economic impact of climate change on mumbai, India," *Regional Health Forum*, vol. 12, no. 1, pp. 5, 2008.
[32] A. Forestiero and G. Papuzzo, "Agents-based algorithm for a distributed information system in internet of things," *IEEE Internet of Things Journal*, Early Access, pp. 1–1, 2021.
[33] A. A. Alshdadi, "Query-based learning method for query routing in unstructured P2P systems," in *Proc. on the Int. Conf. on Artificial Intelligence (ICAI)*, Athens, Greece, pp. 373–378, 2019.
[34] S. Shamshirband and H. Soleimani, "LAAPS: An efficient file-based search in unstructured peer-to-peer networks using reinforcement algorithm," *International Journal of Computers and Applications*, vol. 43, no. 1, pp. 62–69, 2021.
[35] B. Ganter and R. Wille, "Formal concept analysis: Mathematical foundations," *Springer Science & Business Media*, vol. 1, pp. 1–284, 1997.
[36] V. Shah, G. de Veciana and G. Kesidis, "A stable approach for routing queries in unstructured P2P networks," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 3136–3147, 2016.
[37] H. Chen, Z. Gong and Z. Huang, "Self-learning routing in unstructured P2P network," *International Journal of Information Technology*, vol. 11, no. 12, pp. 59–67, 2005.
[38] J. Fan, Z. Wang, Y. Xie and Z. Yang, "A theoretical analysis of deep q-learning," *In Learning for Dynamics and Control*, vol. 120, pp. 486–489, 2020. [Online]. Available: http://proceedings.mlr.press/v120/yang20a.html [Accessed on July 14, 2021].
[39] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[40] H. Y. Ong, K. Chavez and A. Hong, "Distributed deep q-learning," arXiv:1508.04186 [cs], 2015. [Online]. Available: http://arxiv.org/abs/1508.04186, [Accessed on July 14, 2021].

[41] M. Shojafar, J. H. Abawajy, Z. Delkhah, A. Ahmadi, Z. Pooranian *et al.*, "An efficient and distributed file search in unstructured peer-to-peer networks," *Peer-to-Peer Networking and Applications*, vol. 8, no. 1, pp. 120–136, 2015.