BALTICA
2025 38(3)
ISSN: 0067-3064

Deep Neural Network Assisted Monte Carlo Tree Search Algorithm to Solve Bandwidth Slicing Placement Problem

Liang-Chun Chen

Department of Information Management, National Taipei University of Nursing and Health Sciences, Taiwan, R.O.C.
e-mail: jimychen@go.edu.tw

Abstract

To solve the network slicing placement problem, the methods based on CNN/RNN were inadequate in handling the randomness of fluctuating channel quality and bandwidth needs for each network slice. While the Monte Carlo Tree Search (MCTS) methodology effectively deals with the unpredictability of each slice's channel quality and bandwidth request to optimize throughput, it remains time-consuming in finding an optimal solution. The cause is that MCTS relies on a uniform distribution to randomly sample one possible solution, which leads to subpar sampling efficiency. Our objective is to integrate a deep neural network (DNN) to assist MCTS. Specifically, the DNN first analyzes the current allocation situation to predict probability distributions for achieving optimizing throughput. MCTS then leverages this DNN-produced probability distribution to pinpoint the best allocation scenario. Experimental results indicate that the performance of DNN-based MCTS with only 50 search iterations surpasses that of the original MCTS with 4,000 search iterations.

Keywords: Network Slicing, Deep Neural Network, Monte Carlo Tree Search, Channel Quality, Bandwidth Request, Optimize Throughput

1. Introduction

Network slicing placement problem means how to intelligently allocate sufficient bandwidth resource to each slice according to the channel diversity and varying requests of the slices. This problem is generally regarded as a maximum-matching problem to allocate N channels to M slices, and it is also a proved nondeterministic polynomial-time completeness problem (NP-completeness) [1].

Various AI-based methodologies have been suggested to address the network slicing placement problem [2-6], encompassing Q-learning based methods such as Deep Q-Network (DQN) [3], time-sequential based methods like Long Short-Term Memory (LSTM) [4], and actor-critic policy gradient methods [5-6], such as Deep Deterministic Policy Gradient (DDPG), Asynchronous Advantage Actor-Critic (A3c), or Proximal Policy Optimization (PPO). However, these methodologies, rooted in convolutional or recurrent operations, face inherent difficulties when they usually capture localized and proximate features. Regrettably, this inherently character precludes the parallelization and distribution of information across training samples, failing to discern dependencies between distantly positioned elements [7].

Some studies apply sampling methodology, *i.e.*, the Monte Carlo Tree Search (MCTS) methodology [8], to tackle the unpredictable nature of various parameters. Specifically, it demonstrates commendable efficiency in addressing the uncertainties associated with each slice's channel quality and bandwidth requests. This attribute of MCTS makes scenarios possible to understand and manage variable channel qualities. However, delving deeper into the intricacies of the traditional MCTS, the conventional MCTS employs a uniform distribution to determine its sampling points. This random selection method, while straightforward, often demands lots number of samplings to inch closer to an optimal solution. The rationale behind this is that when points are chosen without any guided strategy, the algorithm might spend an inordinate amount of time exploring regions of the search space that are far from optimal. This unguided exploration implies that MCTS requires significantly more sampling iterations to potentially stumble upon or converge to an optimal solution.

Based on the above discussion, MCTS can address the issues with DNN: it struggles to handle unpredictable channel quality and bandwidth requirements. However, MCTS cannot incorporate the advantages of DNN. Conversely, DNN can solve the uniform distribution problem of MCTS, but it also cannot incorporate the advantages of MCTS. Since MCTS and DNN are complementary methods, this study therefore to integrate the capabilities of deep neural networks (DNN) and the Monte Carlo Tree Search (MCTS) methodology, called D-MCTS. Deep neural networks, with their multi-layered architectures and capability to discern complex patterns, offer a promising avenue to enhance the efficiency of traditional MCTS. In the proposed model, the DNN serves as an analytical engine, meticulously examining the current allocation state. By processing vast amounts of data and leveraging its trained weights, the DNN is equipped to project and generate probability distributions that represent potential allocation scenarios for the future. These projections are not mere random predictions; they are informed, calculated estimates based on the neural network's understanding of the data it has been trained on. Once these probability distributions are formulated, MCTS takes the baton. Instead of operating on a simple uniform distribution as it traditionally does, MCTS, in this enhanced setup, utilizes the DNN-derived probability distribution. This collaboration ensures a more guided search, allowing MCTS to focus its efforts more efficiently and effectively pinpoint the most optimal allocation scenario.

This study was the extended versions of our previous work in CoDIT 2023 [9]. This study will first explain the Problem Definition of Bandwidth Slicing Placement Problem. Section 3 will describe the D-MCTS method we proposed. Section 4 will cover the simulation experiments, and Section 5 will present the conclusion.

2. Literature Review

The 3rd Generation Partnership Project (3GPP) defines the architecture for Beyond 5G (B5G) network slicing [10-11], enabling telecommunication service providers to create more customized and flexible network deployments, thereby facilitating the potential for various enterprise-specific networks. The network slicing architecture defined by 3GPP [10-11] includes the Service Instance (SI) Layer, the Network Slice Instance (NSI) Layer, and the Resource Layer. The SI serves as the network service provided by operators for various usage demands, while the NSI aggregates the network resource services required for each SI. Operators can establish three network slices based on three main network attributes (latency, transmission speed, and the number of connected devices): a low-latency network service (URLLC SI), a high-bandwidth mobile network service (eMBB SI), and a massive machine-type communication service (mMTC SI) [12], each allocated with different proportions of Access Network (AN) bandwidth resources.

In the Beyond 5G (B5G) core network, the allocation of power/frequency/time bandwidth resources across three axes is distributed among the three types of network slices. Each Service Instance (SI) compiles the required core network (CN) and access network (AN) resources through the NSI Layer. The eMBB SI can request the establishment of an NSI A network slice from the NSI layer, which then aggregates resources from Core Network Slicing Subset Instance 1 (CN NSSI 1), Access Network Slicing Subset Instance 1 (AN NSSI 1), and Access Network Slicing Subset Instance 3 (AN NSSI 3) as shown in Fig. 1.

ISSN: 0067-3064

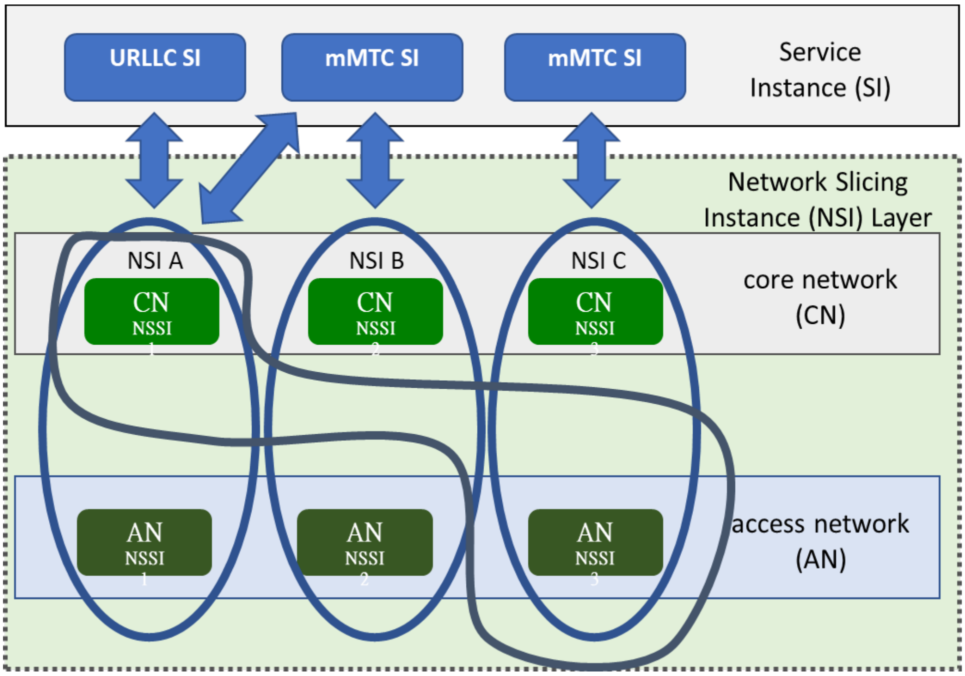

Fig. 1. 3GPP Network Slice Structure

In contrast to the traditional 4G network's "Best Effort" one-size-fits-all approach, the Resource layer of network slicing aims to segment various network functions to provide a corresponding end-to-end dedicated network typology, meeting diverse industry and customer business requirements [13]. According to the 5G white papers published by relevant American technical and regulatory bodies [13-15], from a commercial operation to a network technology perspective, five technical aspects of network resource segmentation functions can be outlined. As shown on the left y-axis of Fig. 2, from top to bottom, these are Traffic Quality of Service (QoS), core network, protocol stack, bandwidth, and antenna. User demands range from low customization public networks to highly isolated dedicated networks, which can be roughly divided into three categories with a total of six network slicing modes [13-15], as illustrated on the x-axis at the bottom of Fig. 2:

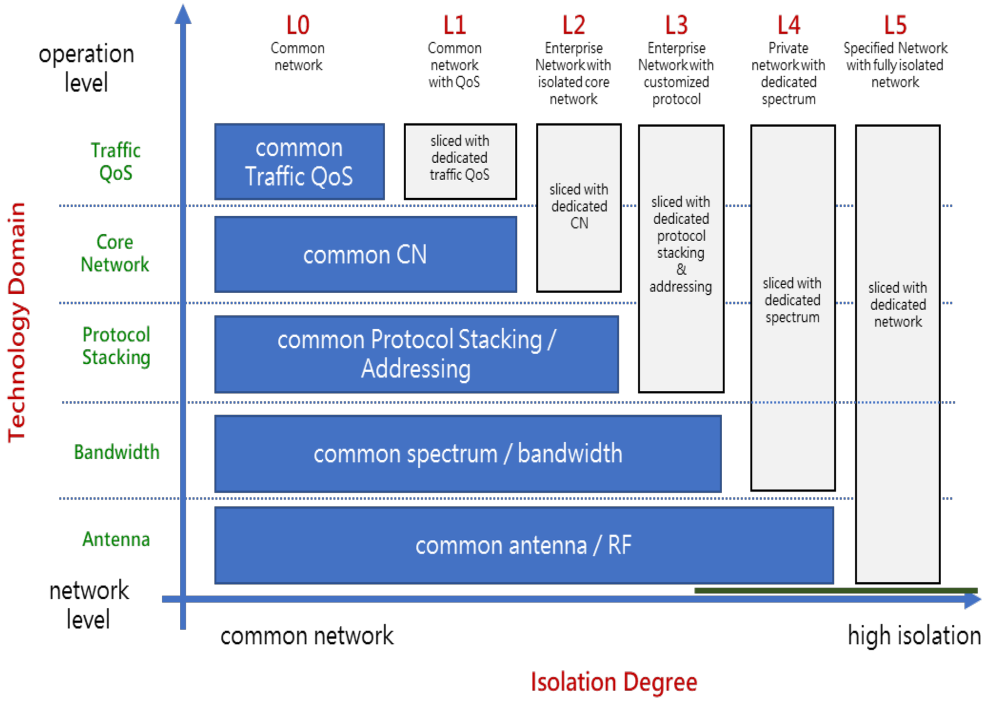

Fig. 2. Network Slice Types and Classification

(1) Public Network Slices (Level 0~1): These slices inherit the telecommunication network services provided to individuals during the 4G era, ensuring consistent Traffic Quality of Service (QoS) or offering exclusive network traffic QoS for VIP users, yet still utilizing the same core network.

(2) Enterprise Dedicated Network Slices (Level 2~4): For general enterprise network users, these slices provide customized core network functionalities (such as data storage and core network computational capabilities), customized communication protocols and network addressing functions, or further exclusive network bandwidth resources based on user demands.

(3) Licensed Industry-Specific Network Slices (Level 5): Aimed at users with special requirements for high isolation or high network transmission quality guarantees, such as satellite special networks, government-licensed exclusive networks for public interest (e.g., railways and aviation), electric power networks, military networks, etc., these slices offer full network slicing capabilities within the existing 5G network infrastructure.

This research project will focus on the network slicing bandwidth allocation issues related to the "bandwidth resource" on the y-axis, referred to as Bandwidth Resource Slicing. Bandwidth resource slicing denotes the allocation of bandwidth resources to each network slice within the B5G core network, based on the user demands of each slice and the pricing strategies of telecommunications providers.

The lifecycle and bandwidth allocation process of bandwidth slicing, as depicted in Fig. 3, can be broadly divided into five steps [16-17]:

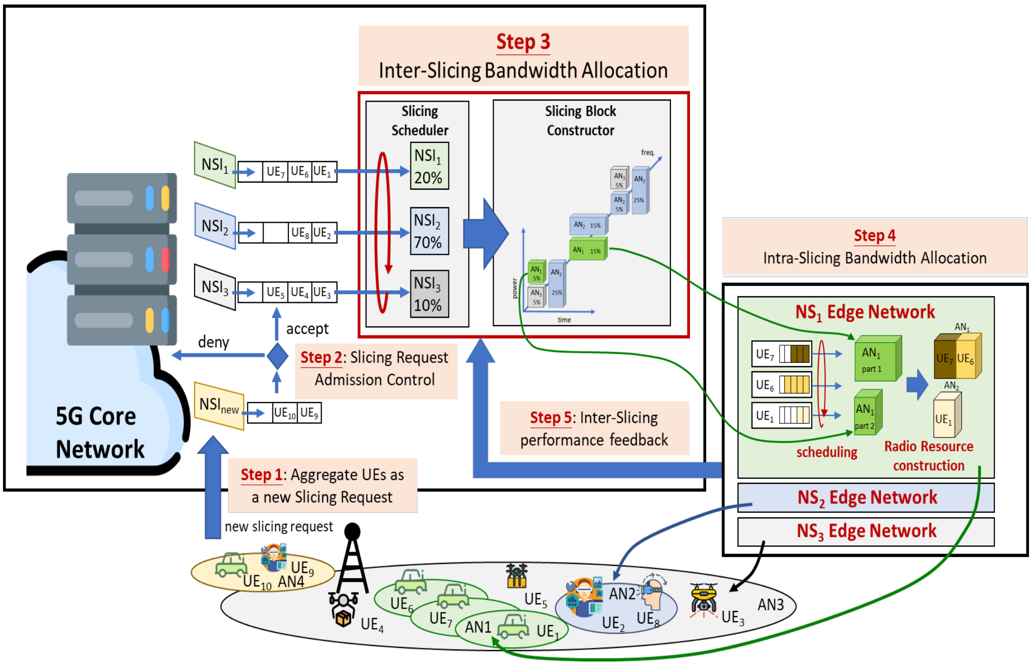

Fig. 3. Network Slicing Life-cycle

Step 1: Aggregate UEs Requirement as a new Slicing Request

At this stage, the service requirements of various User Equipment (UE) are aggregated into a network slice service request, serving as the basis for subsequent segmentation of network resources by the B5G core network. This stage primarily focuses on research related to the architectural design of the 5G core network [16-17].

Step 2: Slicing Request Admission Control

Upon the transmission of a new network slice request to the B5G core network, an admission control mechanism for network slicing is utilized to confirm whether the current 5G network bandwidth resources can support its requirements. This mechanism extends traditional admission control methods [9], analyzing the queuing situation of network slice demands based on different objectives (such as maximizing profit, maximizing transmission speed, minimizing latency) using established algorithms (e.g., Greedy, First-in-first-out, Priority-based algorithms).

Step 3: Inter-Slicing Bandwidth Allocation

This phase involves the B5G core network allocating bandwidth resources to each network slice according to their service requirements.

Step 4: Intra-Slicing Bandwidth Allocation

In this stage, base stations of the edge network allocate the bandwidth resources, assigned by the B5G core network, to individual User Equipment (UE). The algorithms for bandwidth allocation in this stage continue the practices of 4G network bandwidth allocation algorithms [9], such as the traditional Fair Queue algorithm for distributing bandwidth resources.

Step 5: Intra-Slicing Performance Evaluation

This final stage evaluates the efficiency of bandwidth usage within slices and provides feedback to the third stage of inter-slicing bandwidth allocation. This research will focus on the third step, the design of algorithms for inter-slicing bandwidth resource allocation, as highlighted in red in Fig. 4.

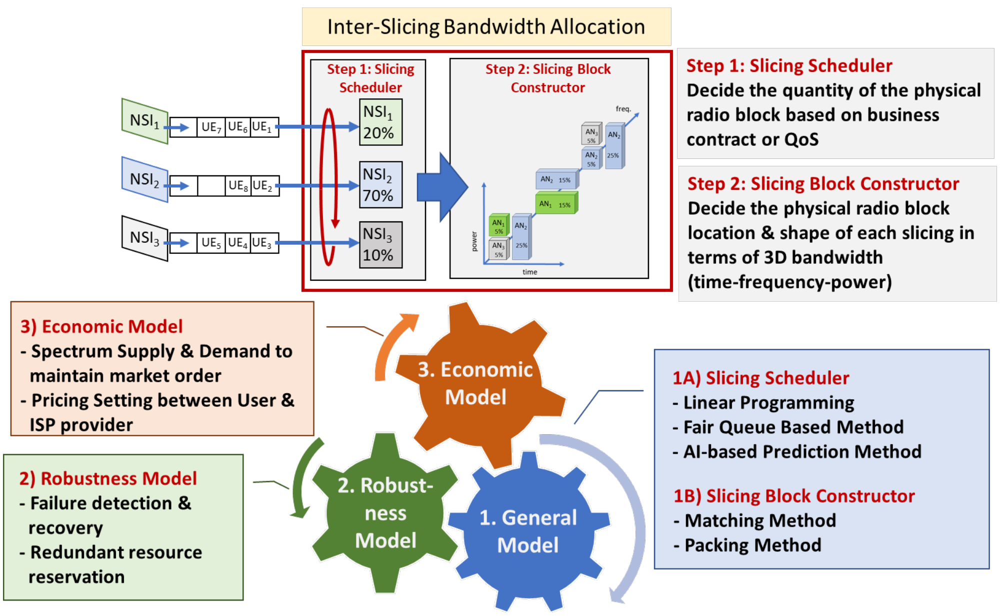

Fig. 4. Two-Stage Bandwidth Allocation Problem for Cross-Network Slice Bandwidth Allocation

In recent years, numerous scholars have proposed various algorithms for network slice bandwidth allocation, each with different solution objectives leading to different designs. These algorithms can be broadly classified into three categories based on their solution objectives: General Model (network technical aspects), Robustness Model (information security aspects), and Economic Model (economic policy aspects), as shown in the lower half of Fig. 4.

- The General Model aims to maximize network utility, such as maximizing throughput, maximizing spectrum utilization, and minimizing delay time, while considering bandwidth slicing constraints [1-8].
- The Robustness Model, building on the General Model, addresses issues of bandwidth robustness derived from B5G bandwidth slicing: the allocation algorithms not only aim to maximize bandwidth efficiency but also handle various unpredictable network events to ensure the robustness of the B5G network. Unpredictable network events include failures of network spectrum hardware/software or network congestion [18-22].
- The Economic Model considers the economic activities of users, such as app usage habits, user mobility patterns, demographic density, and age distribution in a region, to estimate the spectrum demand for various network slices in that area, thereby maintaining order in the B5G telecommunications market [23-27].

This study focus on the General Model, and the designed of our proposed algorthm is shown in Section 3.

3. Problem Definition of Bandwidth Slicing Placement Problem

The problem definition and notation (shown in Table 1) for the proposed method is provided as follows:

Given: A bipartite graph $G = (S, C; E, R)$ with I slice vertices (S), J channel vertices (C), edge $e_{j,i} \in E$ has a nonnegative bandwidth capacity $w_{j,i}$ where $i \in S$ and $j \in C$, and each slice i has its own requested bandwidth $r_i \in R$

Target: To find the set $M = \{m_1, m_2, \dots, m_i, \dots, m_I\}$ where $m_i = \{e_{j,i}, \dots)\}$ is the matching that assigns channels j to slice i in order to maximize the value function, *i.e.*, overall throughput cal_th(M) is maximized.

Constraint:

(1) The maximum throughput of each slice i would not exceed its bandwidth requirements $r_i \in R$ and its assigned channel capacity, i.e., $\text{cal\_th}(m_i) = \min\left(r_i \ , \ \sum_{e_{j,i} \in m_i} w_{j,i} \right)$.

(2) The number of channels assigned to slice i, i.e., $a_i = cal\_size(m_i)$, would not exceed the overall number of channels, i.e., $\sum_{i \in I} a_i = \leq J$

(3) One channel can only be assigned to one slice, one slice may have multiple channels.

4. Methodology

The proposed D-MCTS integrates the deep neural network techniques and our previous work [8] MCTS-RA to approximate the optimal solution more accurately. Figure 1 depicts the three essential stages in constructing the decision tree. Each phase is detailed below:

Phase 1 - Sampling: In this phase, MCTS randomly designates an unoccupied channel j to a slice i. This designation aids in expanding the decision tree through the addition of a child node. Following this, a simulation runs that allocates the remaining unoccupied channels until each one is designated.

Table 1. Notation used in this study

Notation	Definition
S	The set of all slices.
I	The number of slices in this network.
C	The set of all channels.
J	The number of channels.
E	The set of all possible channel-slice edges.
R	The set of required bandwidth resource for each slice. The unit is bytes.
r_i	The bandwidth requirement of slice i
$e_{j,i}$	The edge that assigns channel j to slice i.
$w_{j,i}$	The bandwidth capacity for $e_{j,i}$
$\text{cal\_th}(m_i)$	Throughput of the matching m_i
$\text{cal\_size}(m_i)$	The number of channels assigned to slice i

Although D-MCTS randomly assigned one channel to one slice, in this phase, instead of relying solely on the conventional uniform distribution, D-MCTS employs a probability distribution informed by DNN. This synergy promotes a targeted search, allowing MCTS to concentrate its resources and identify the most advantageous allocation setup with greater precision.

Based on above mentioned, the detailed procedure comprises 3 fundamental steps. As described in problem definition and shown in the bottom part of Fig. 5, the input consists of a $J{\times}I$ matrix $MCS = \begin{bmatrix} w_{1,1} & \cdots & w_{1,I} \\ \vdots & \ddots & \vdots \\ w_{J,1} & \cdots & w_{J,I} \end{bmatrix}$ and $R = [r_1, \dots, r_j, \dots , r_I]$, where MCS is all possible bandwidth resource can be assigned to each slice, and R is the requested bandwidth of each slice. Initially, the 2-dimensional matrix MCS is flattened into a one-dimensional matrix $[w_{1,1}, \dots, w_{J,I}]$. Subsequently, this flatten matrix merges with the request matrix $R = [r_1, \dots, r_j, \dots , r_I]$, yielding P, that is s $P = [w_{1,1}, \dots, w_{J,I}, r_1, \dots , r_I]$,, as the $(J+1){\times}I$ elements. Then, D-MCTS performs the convolution and pooling procedure to fetch the features. Finally, a fully connected operation morphs features to 2-dimentional P^e as an $l{\times}e$ weight matrix. The P^e is the probability distribution to provide MCTS to accurately locate the optimal sample point.

Phase 2 - Evaluating: Here, MCTS refreshes the simulation outcome, representing the total throughput of the simulated match, from the newest leaf to the root node. Throughout this stage, every node retains the mean throughput from earlier simulation outcomes.

Phase 3 - Selecting: In this concluding phase, MCTS picks a child node that boasts the highest mean throughput, determining the channel allocation to each slice. Subsequently, MCTS circles back to Phase 1, continuing the cycle until every channel finds its designation.

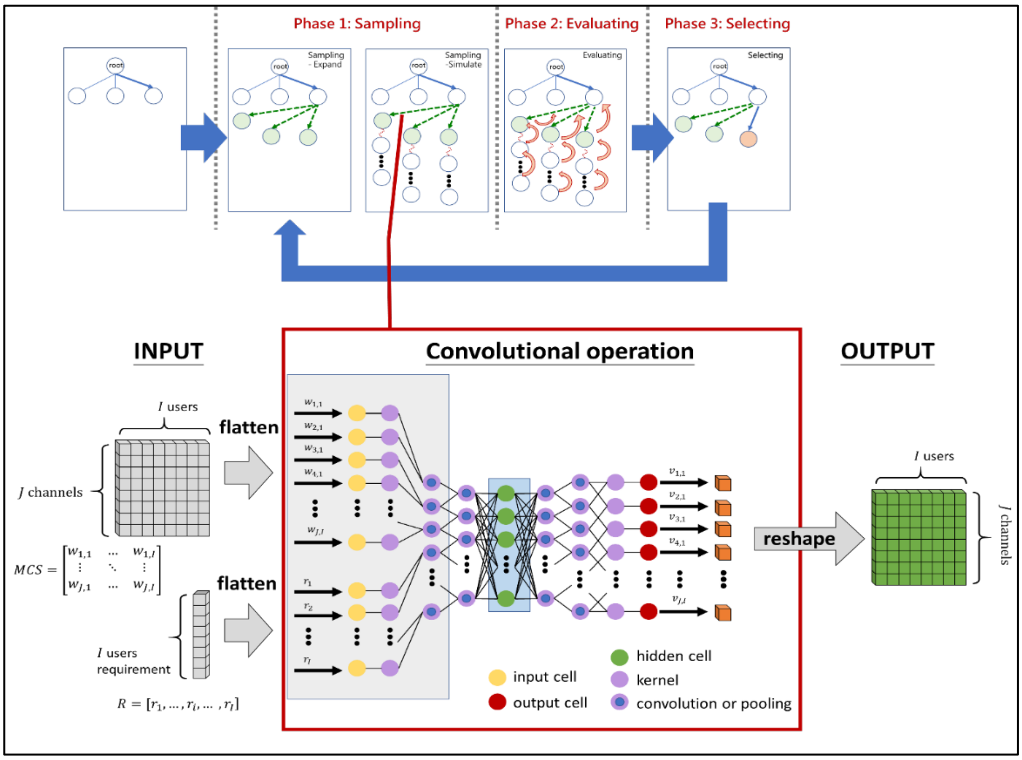

Fig. 5. Conceptual Framework

5. Performance Evaluation

Simulations were conducted to compare the performance of the proposed D-MCTS algorithm (50 search iterations) with three types of conventional AI based methods: DQN, PPO, and MCTS-RA (4000) [8] that 4000 means MCTS-RA adopts 4,000 search iterations.

The simulation environment was based on a 10MHz LTE system with three types of services (eMBB, URLLC, mMTC), 300 user equipments (UEs). Various Modulation and Coding Schemes (MCSs) were used based on the received Signal-to-Noise Ratio (SNR), including QAM1/2, 16QAM1/2, 16QAM3/4, 256QAM1/2, and 256QAM3/4.

To investigate the effect of average requested bandwidth of each slice on total throughput, the aggregated data arrival rate of each slice was varied from 4 Mbps to 8 Mbps, and the number of slices are 6 in this scenario.

Fig. 6 illustrates the training performance of all the algorithms. D-MCTS achieved an approximate 27Mbps, outperforming DQN and PPO, which hovered around 26Mbps. This advantage stems from D-MCTS's capacity to discern features from various positions within the input simultaneously, while the others primarily detect features in adjacent matrix positions. Conversely, D-MCTS demands about 2 million training epochs to stabilize its performance.

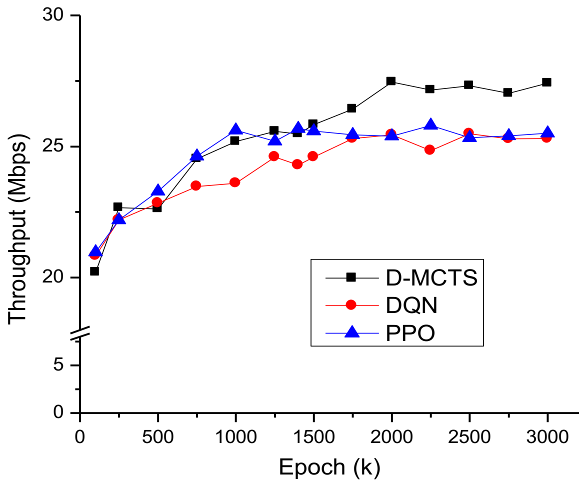

Fig. 6. The System Throughput during the Training Process

The Entropy and Loss status supports this argument that become stable after 2 million training epochs as shown in Fig. 7 and Fig. 8 In contrast, PPO needs roughly 1 million epochs, and DQN calls for 1.75 million. The extended training period for D-MCTS arises from its need to explore dependencies between distantly placed elements to avoid overfitting problem, whereas the other algorithms might overfit the training data, potentially compromising performance on unseen dependencies between distantly positioned elements within datasets.

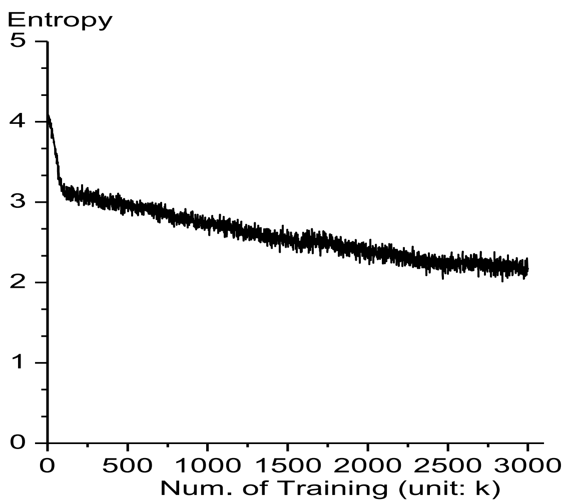

Fig. 7. Entropy status during D-MCTS training process

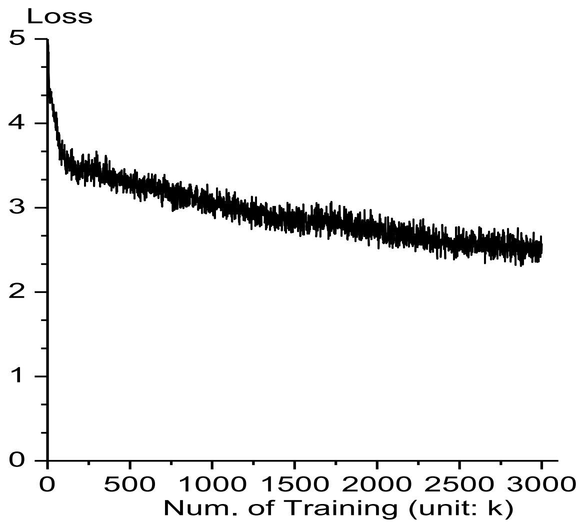

Fig. 8. Loss status during D-MCTS training process

This observation is further corroborated in other simulation settings, as depicted in Fig. 9 and Fig. 10.

Fig. 9 dissects the effects of fluctuating traffic loads on system throughput, examining average traffic spans between 1Mbps and 10Mbps for each slice. It's evident that when traffic is restrained below 3Mbps, the algorithms' performances align, given the adequate bandwidth which sustains diverse placement strategies. However, when traffic swells beyond 4Mbps, especially noticeable at the 6Mbps threshold, D-MCTS leads the pack, outstripping DQN, PPO, and MCTS-RA (4000) by margins of 11.4%, 8.1%, and 7.9% respectively.

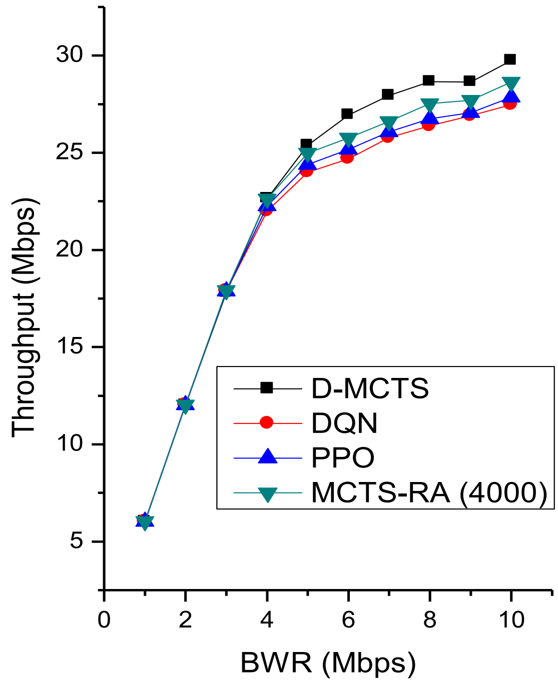

Fig. 9. System Throughput in Varying System Loading Environments

Shifting focus to Fig. 10, we observe the influence of channel quality variations on total throughput. Here, while the average Signal-to-Noise Ratio remains steady at 18 dB, the standard deviation oscillates between 0 and 10 dB. With a standard deviation at 0 dB, all algorithms converge at an optimal performance, approximately 18 Mbps, due to uniform channel quality. But as this deviation widens, D-MCTS continues its trend of superiority, surpassing DQN, PPO, and MCTS-RA (4000) by approximately 6.8%, 4.3%, and 3.4%, respectively.

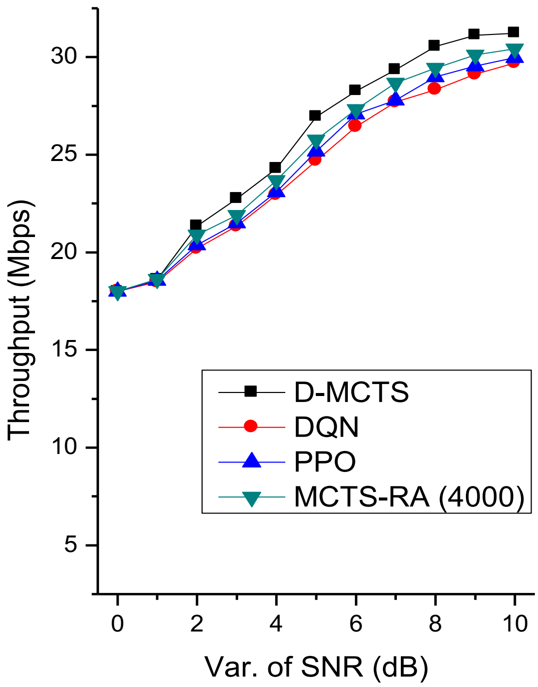

Fig. 10. System Throughput in Varying Signal Noise Environments

6. Conclusion

While conventional methods based on CNN or MCTS have shown inadequacies in addressing the complexities of network slicing placement problems, the integration of Deep Neural Network (DNN) with Monte Carlo Tree Search (MCTS) presents a promising solution. The proposed method D-MCTS optimally utilizes the predictive capabilities of DNN to enhance the efficiency of MCTS. Simulations reveal that D-MCTS, even with a mere 50 search iterations, outperforms traditional MCTS that uses 4,000 iterationsm. Additionally, when compared to other established AI techniques such as DQN and PPO, D-MCTS demonstrates superior performance, especially under challenging conditions such as fluctuating traffic loads and varying channel quality. Specifically, as traffic surpasses the 6Mbps threshold and channel quality deviations increase, D-MCTS consistently outperforms its counterparts by significant margins. This underscores the potential of D-MCTS as a robust and efficient tool for network slicing placement in complex communication environments.

References

[1] Y.-H. Chen, "Self-Competence Learning for Bandwidth Slicing and Confidential Computing Requirement," IEEE Access, vol. 9, pp. 169156-169160, Dec. 2021.
[2] Y.-H. Xu, J.-H. Li, W. Zhou, and C. Chen, "Learning-Empowered Resource Allocation for Air Slicing in UAV-Assisted Cellular V2X Communications," in IEEE Systems Journal, vol. 17, no. 1, pp. 1008–1011, Mar. 2023.
[3] G. Wu and G. Tan, "An Adaptive Network Slice Combination Algorithm Based on Multistep Temporal-Difference Learning," in IEEE Wireless Commun. Lett., vol. 11, no. 6, pp. 1128–1132, Jun. 2022.
[4] Y. Yuan, L. Lei, T. X. Vu, Z. Chang, S. Chatzinotas, and S. Sun, "Adapting to Dynamic LEO-B5G Systems: Meta-Critic Learning Based Efficient Resource Scheduling," in IEEE Trans. Wireless Commun., vol. 21, no. 11, pp. 9582–9595, Nov. 2022.
[5] T. Mai, H. Yao, N. Zhang, W. He, D. Guo, and M. Guizani, "Transfer Reinforcement Learning Aided Distributed Network Slicing Optimization in Industrial IoT," in IEEE Trans. Ind. Inf., vol. 18, no. 6, pp. 4308–4316, Jun. 2022.
[6] J. J. Alves Esteves, A. Boubendir, F. Guillemin, and P. Sens, "A Heuristically Assisted Deep Reinforcement Learning Approach for Network Slice Placement,"in IEEE Trans. Netw. Serv. Manage., vol. 19, no. 4, pp. 4794–4806, Dec. 2022.
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need." arXiv, Dec. 05, 2017. Accessed on: May 10, 2023.
[8] S.C. Chang, "The Monte Carlo tree search based bandwidth slicing allocation algorithm," in Wireless Network, Oct. 2023.
[9] Y.-H Chen, "Federated Learning Architecture to Integrate AI Models from Different Internet Service Providers: Using Bandwidth Slicing Resource Management as Case Study," in Proc. of 2023 9th International Conference on Control, Decision and Information Technologies (CoDIT), Rome, Italy, pp. 2639-2642, 2023.
[10]N. Seifi, J. Zhang, R. W. Heath, T. Svensson and M. Coldrey, "Coordinated 3D Beamforming for Interference Management in Cellular Networks," in IEEE Transactions on Wireless Communications, vol. 13, no. 10, pp. 5396-5410, Oct. 2014.
[11]3GPP R15, TR 28.801 Telecommunication management ; Study on management and orchestration of network slicing for next generation network, 2018.
[12]Z. Ding, X. Lei, G. K. Karagiannidis, R. Schober, J. Yuan and V. K. Bhargava, "A Survey on Non-Orthogonal Multiple Access for 5G Networks: Research Challenges and Future Trends," in IEEE Journal on Selected Areas in Communications, vol. 35, no. 10, pp. 2181-2195, Oct. 2017.
[13]5G Americas, "Network Slicing for 5G and Beyond," November 2016.
[14]GSMA, "Network Slicing Use Case Requirements," April 2018.
[15]FCC Technological Advisory Council, "5G Network Slicing Whitepaper," 2018.
[16]M. Jiang, M. Condoluci and T. Mahmoodi, "Network slicing management & prioritization in 5G mobile systems," in Proc. of European Wireless 2016; 22th European Wireless Conference, Oulu, Finland, pp. 1-6, 2016.
[17]R. Ferrus, O. Sallent, J. Perez-Romero and R. Agusti, "On 5G Radio Access Network Slicing: Radio Interface Protocol Features and Configuration," in IEEE Communications Magazine, vol. 56, no. 5, pp. 184-192, May 2018.
[18]S. S. Mohammed et al., "A New Machine Learning-based Collaborative DDoS Mitigation Mechanism in Software-Defined Network," in Proc. of 2018 14th

International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 1-8, Oct. 2018.
[19]Y. Liu, M. Dong, K. Ota, J. Li, and J. Wu, "Deep Reinforcement Learning based Smart Mitigation of DDoS Flooding in Software-Defined Networks," in Proc. of 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), pp. 1-6, Sep. 2018.
[20]Y.-H. Chen, Pi-Tzong Jan, Ching-Neng Lai, ChunWei Huang, Chih-Han Chang and Yo-Cih Huang, "Detecting Linking Flooding Attacks using Deep Convolution Network," in Proc. of 2020 3rd International Conference on Computers in Management and Business (ICCMB2020), Tokyo, pp. 70-74, Jan. 2020.
[21]Y. Liu et al., "A Survey on Neural Trojans," in Proc. of 2020 21st International Symposium on Quality Electronic Design (ISQED), Santa Clara, CA, USA, pp. 33-39, 2020.
[22]R. Wen, G. Feng, W. Tan, R. Ni, S. Qin and G. Wang, "Protocol Function Block Mapping of Software Defined Protocol for 5G Mobile Networks," in IEEE Transactions on Mobile Computing, vol. 17, no. 7, pp. 1651-1665, 01 July 2018.
[23]D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis and X. Costa-Perez, "Optimising 5G infrastructure markets: The business of network slicing," in Proc. of IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta, GA, pp. 1-9, 2017.
[24]P. Caballero, A. Banchs, G. de Veciana and X. Costa-Pérez, "Network slicing games: Enabling customization in multi-tenant networks," in Proc. of IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, Atlanta, GA, pp. 1-9, 2017.
[25]M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini and H. Flinck, "Coalitional Game for the Creation of Efficient Virtual Core Network Slices in 5G Mobile Systems," in IEEE Journal on Selected Areas in Communications, vol. 36, no. 3, pp. 469-484, March 2018.
[26]Y.-H. Chen, Pi-Tzong Jan, "Impossible trinity: A guideline to shape telecommunication policy by mediating bandwidth supply," Computer Standards & Interfaces (ISSN: 0920-5489), vol. 65, pp. 167-179, July 2019.
[27]Y.-H. Chen, "Drafting spectrum policy in an access-price targeting perspective and exploring its embedded biological nature," Computer Standards & Interfaces (ISSN: 0920-5489), vol. 62, pp. 128-139, Feb. 2019.