

Article

Resource Allocation for Network Slicing in RAN Using Case-Based Reasoning

Dandan Yan , Xu Yang and Laurie Cuthbert

Faculty of Applied Sciences, Macao Polytechnic University, Macao, China

* Correspondence: dandan.yan@mpu.edu.mo

Abstract: As a key technology of 5G, network slicing can meet the diverse needs of users. In this research, we study network slicing resource allocation in radio access networks (RAN) by case-based reasoning (CBR). We treat the user distribution scenario as a case and stored a massive number of cases in the library. CBR is used to match a new case with cases in the case library to find similar cases and determine the best slice bandwidth ratio of the new case based on these similar cases. In the matching process, the k-nearest neighbors (KNN) algorithm is used to retrieve similar cases, the nearest k neighbors being determined by considering sparsity reduction and locality-preserving projections. Although only an initial study, the results confirm that the proposed architecture is capable of allocating resources efficiently in terms of prediction error and computational cost.

Keywords: network slicing; radio access network; resource allocation; case-based reasoning; k-nearest neighbors

1. Introduction

Network slicing, in which the network infrastructure is separated into several virtual sub-networks, is a critical technology for 5G networks. Each slice can provide a specific slice service such as end-to-end low latency, high bandwidth, and massive device access [1]. The spectrum resource can be adjusted and optimized based on different service requirements by separating it into several slices in RAN. However, there are some challenges to allocate resource efficiently for limited spectrum resources. Among them, one important issue is the lack of real-time intelligence agents to effectively deal with unstable resource supply and demand and lack of real-time technologies to accurately track and analyze resource allocation and usage. Once slice users cannot obtain the minimum resources they need due to unreasonable resource allocation, they will not be able to realize the corresponding services. This is made even more difficult in fifth-generation (5G) cellular networks because end-users want larger data rates and lower end-to-end latency [2]. In this paper, we research the inter-slice bandwidth allocation in RAN to maximize the qualified user ratio (QUR), which obtained enough to meet its quality of service (QoS).

Luis Guijarro et al. [3] allocated a weight to each cell to enable operators to share resources. The weight denoted the proportion of resources that each network slice user can obtain in each cell to maximize the number of subscribers. It provided accurate approximation when the cells were heterogeneous and exact values for the Nash equilibrium. Liu et al. [4] proposed a two-level allocation to maximize a no close-form utility. They first learned how many resources were needed by users in each slice to fulfill their QoS requirements and then optimize the resource allocations. They take inter-slice resource allocation as the master optimal problem and suggested a convex optimization approach for it. The deep deterministic policy gradient method (DDPG) for learning the allocation policy intra-slice as the slave optimal problem. Deep reinforcement learning (DRL) was introduced by Wang et al. [5] to dynamically allocate resources between several slices. The goal is to maximize resource usage while maintaining a high quality of service. It outperforms



Citation: Yan, D.; Yang, X.; Cuthbert, L. Resource Allocation for Network Slicing in RAN Using Case-Based Reasoning. *Appl. Sci.* **2023**, *13*, 448. <https://doi.org/10.3390/app13010448>

Academic Editor: Christos Bouras

Received: 20 October 2022

Revised: 11 December 2022

Accepted: 16 December 2022

Published: 29 December 2022



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

heuristics, best efforts, and random methods. Sun et al. [6] reserve a portion of the radio resource in the BS, and then utilize DRL to change the resources in each slice autonomously in terms of resource consumption and satisfaction of the slices. Akgul et al. [7] proposed a negotiated spectrum resource sharing agreement between operators and infrastructure providers to improve spectral efficiency. To allocate resources across eMBB and URLLC slices, Feng et al. [8] proposed a two-timescale technique based on the Lyapunov optimization. It performed similarly to the exhaustive searching algorithm (the optimal solution). To achieve a trade-off between slice performances, Sun et al. [9] employed the Stackelberg game to distribute subchannels for local radio resource managers (LRRMs) within slices. They analyzed the service performance of power consumption and user satisfaction. The low-complexity methods perform similarly to exhaustive search and greatly outperform other benchmarks. Yang et al. [10] proposed using a game theoretic approach to allocate resource. Yang et al. [11] proposed using a genetic algorithm (GA) to determine the most suitable slice boundary to achieve the optimal inter-slice resource management. To deal with co-channel interference, Yang et al. [12] proposed a connection admission control (CAC) mechanism to achieve effective isolation for network slicing.

In conclusion, most optimization algorithms have not maximized the QUR in the case of a limited spectrum, and only obtained a performance similar to an exhaustive search. We research the slice bandwidth ratio prediction based on a case library, which stores a massive number of cases by an exhaustive search method that can find the optimal solution to maximize the QUR. It can realize very low prediction errors and run costs.

In summary, the key contributions of this paper are summarized as follows:

- To reduce the computational complexity and determine the optimal slice ratio (the proportion of bandwidth occupied by slices), we built a case library to store the user distribution scenario and the optimal slice ratio produced by exhaustive searching.
- We have considered the QUR, which is to optimize the slice service.
- The CBR framework is proposed to form a complete system. The predicted error is reduced by revising and retaining consistency.
- The KNN algorithm is proposed to determine the optimal slice bandwidth ratio for the new case based on the database. To reduce the prediction error and run cost, the sparsity reduction method, which joined the least square, sparse learning, and locality-preserving projection, is utilized to optimize the k value. We defined it as optimizing KNN (O-KNN).

The rest of the paper is structured as follows. Section 2 presents the system model and details the related components of the system architecture. Section 3 presents related intelligent techniques required to build the proposed approach. Section 4 presents and analyzes the results of the O-KNN in predicting the slice ratio for resource allocation. Section 5 presents the conclusion and scope for future work.

2. System Model

2.1. General Model

The virtualization of the cell from [13,14], is shown in Figure 1. The model used for the research here considers a single-cell self-organization allocation as a first step. In the single cell approach, the entire 360° plane is equally divided into a sectors with b bands, yielding $a \times b$ segments. By dividing the entire cell area into different segments, user locations can be mapped to virtual cells. Different cases are formed according to the distribution of users in each segment at each different snapshot. The interference from adjacent cells is considered in the mapping [13]. Initially, a two-slice scenario is considered, where each segment may contain slice1 users (S1) and slice2 users (S2) at the snapshot; $|S1|$ and $|S2|$ denote the number of S1 and S2 users, respectively. The slice ratio is defined as ω and $1 - \omega$ for S1 and S2, respectively, where ω is the fraction of total resource W , and at the snapshot, we assume that the ratio will not change [14].

For a user u to be qualified, the bitrate of that slice user r_u must be greater or equal to its minimum requirement $r_{u,min}$. The rate of slice user u is defined as $r_u = \sum_{f=1}^F a_{f,u} r_{f,u} \geq$

$r_{u,\min}$, where $f \in \mathcal{F} = \{1, 2 \dots F\}$ is the set of resource block (RB), when the RB f is assigned to user u , $a_{f,u} = 1$, otherwise $a_{f,u} = 0$. From the well-known Shannon equation, the capacity contributed by RB f is

$$r_{f,u} = B \log_2 \left(1 + \frac{P_h^f g_{h,u}^f}{\sigma^2 + \sum_{l \in \mathcal{H}, l \neq h} P_l^f g_{l,u}^f} \right) \tag{1}$$

where $h \in \mathcal{H} = \{1, 2 \dots H\}$, \mathcal{H} is denoted a set of base station (BS). B is the bandwidth of RB. σ^2 is the noise power; $g_{h,u}^f$ denotes the channel gain between the BS h and user u by RB f , P_h^f is the transmit power of BS h by RB f , l is the surrounding cells. P_l^f is the transmit power of BS l by RB f , $g_{l,u}^f$ denotes the channel gain between the BS l and user u by RB f .

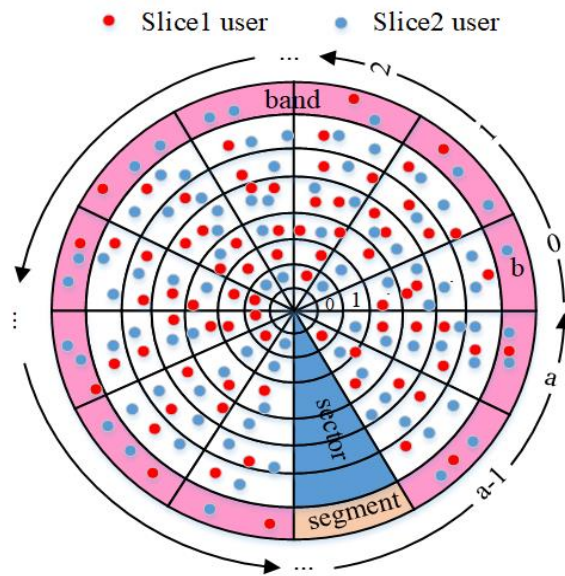


Figure 1. Schematic of slice user distribution.

A set of UEs is denoted as $\mathcal{N} = \{1, 2 \dots N\}$. The qualified user ratio over all slices for each cell is as follows:

$$QUR = \frac{\sum_{u=1}^n q_u}{|S1| + |S1|} \tag{2}$$

where q_u is a binary variable that represents the indicator of qualified user u . $n \in \mathcal{N}$. $q_u \in \{0, 1\}$, $q_u = 1$, if $r_u \geq \delta_s$, otherwise $q_u = 0$. The r_u is the data rate of user u . δ_s is the rate requirement of slice $s \in \{1, 2\}$.

The overall approach is to match the query case (a new case) to the values in the case library by CBR, as shown in Figure 2, by comparing the attributes of the query case to the cases in the library and determining their degree of match. The best matches are used to determine the slice ratio to be used for that query case. The advantage of CBR as a technique is that it can constantly enrich the case library to reduce the prediction error. The standard work process of CBR is introduced in [15]. The four main processes are *retrieve*, *reuse*, *revise*, and *retain*. The matching process described above is the retrieve process and if a suitable match is found and its corresponding case solution can be used for the reuse by resource allocation (RA). If a suitable solution for the new query cannot be found, a case revision needs to be carried out, which can be performed by exhaustive search. The revised case can be retained in the case library to enrich the samples.

The O-KNN is utilized in the retrieve stage. Distance measurement and k value selection are two key factors of KNN. Distance measurement is determined by Euclidean distance. The k value is determined by least square, sparse learning, and locality preserving projections. To speed up the retrieval, brute force was applied. Root mean square error

(RMSE) = $\sqrt{\frac{1}{n} \sum_{i=1}^n (w_i - \hat{w}_i)^2}$ was utilized to quantify the difference between the forecast and actual values. Where w_i and \hat{w}_i are the i -th test value and prediction value, respectively. In the reuse stage, the predicted slice ratio is utilized for two slices to measure the QUR by proportional fair scheduling. In case of the worse prediction, the solution of query case is revised by exhaustive search, and then retained in the case library.

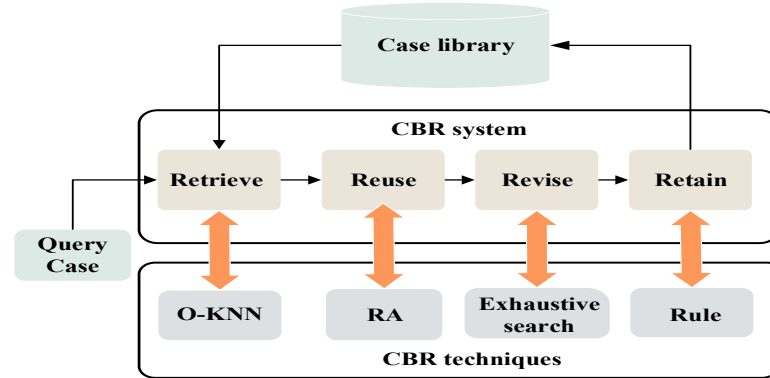


Figure 2. CBR framework.

2.2. Dataset

The experiments were performed using a dataset comprising of n cases in the case library plus a second set of test cases. To populate the case library we chose the n combinations to be uniformly distributed across the cell with a mix of different numbers of users in each slice. For each case, the value of the slice ratio was obtained by exhaustive search with a step length of 0.01 allocated within the slice to each user chosen using proportional fair scheduling. The best slice ratio was then linked to that user distribution to create a case that could be saved in the case library (Table 1).

The pseudocode of the case library build is given in Algorithm 1.

Algorithm 1 Pseudocode of case library build.

Input:

Bandwidth W ; $Searchstep = 0.01$;

Output:

Case library;

- 1: **for** each $i \in [1, n]$ **do**
 - 2: $\omega_1 = 0.01, \omega_2 = 0.99, QUR_{old} = 0$
 - 3: Insert user randomly,
 $U = \{U_{11}, U_{12} \dots U_{(a * b)1}, U_{(a * b)2}\}$;
 - 4: **for** $j = 0; j < J; j++$ **do**
 - 5: $B1 = W * \omega_1, B2 = W * \omega_2$;
 - 6: Resource allocation for U1 and U2 and caculate QUR by Equation (2).
 - 7: **if** $QUR > QUR_{old}$ **then**
 - 8: $QUR^* = QUR$;
 - 9: **else**
 - 10: $QUR^* = QUR_{old}$;
 - 11: **end if**
 - 12: $QUR_{old} = QUR$;
 - 13: $\omega_{1+} = Searchstep, \omega_{2-} = Searchstep$
 - 14: **end for**
 - 15: Case library add $\{i, U, [\omega_1, \omega_2], QUR^*\}$;
 - 16: **end for**
 - 17: **return** Case library;
-

We have d attributes (i.e., one attribute for each segment) and n cases in the library, where each attribute represents the number of users in each segment. Here, we denote the query case as a matrix $[q_1, q_2, \dots, q_m]$. Similarly, each case c in the library contains d attributes $A_{c1}, A_{c2}, \dots, A_{cd}$. Note that for simplicity in this feasibility stage, we restrict the attribute value per segment to one. The ratio of S1 and S2 users in each case remains unchanged.

Table 1. Case library.

Cases	Attributes	Solution
Case 1	$\{A_{11}, A_{12}, \dots, A_{1d}\}$	ω_1
Case 2	$\{A_{21}, A_{22}, \dots, A_{2d}\}$	ω_2
\vdots	\vdots	\vdots
Case n	$\{A_{n1}, A_{n2}, \dots, A_{nd}\}$	ω_n

3. Problem Formulation

O-KNN is used to predict the slice ratio. The similarity between the query case and the library cases is first calculated, and then a search is made of the nearest k neighbors. The k value is determined by the reconstruct process between the test sample and training sample. We use the training sample to reconstruct each test sample by least squares, sparse learning, and locality preserving projections. Finally, the solutions are predicted by weighting the distance of the k -nearest neighbors. The flowchart of the O-KNN method is shown in Figure 3.

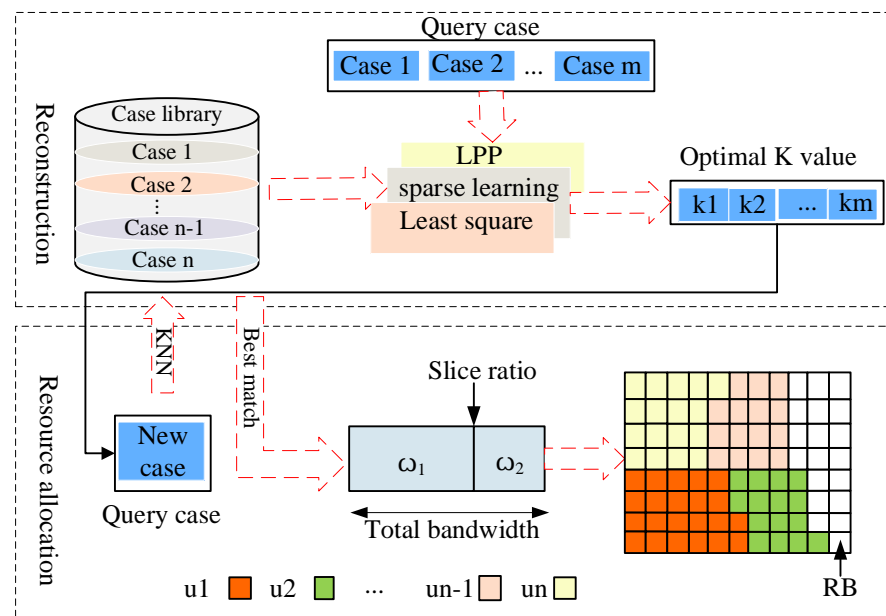


Figure 3. Flowchart of the O-KNN method.

To test the CBR approach, we used query cases from the test data where we had already used an exhaustive search to determine the optimum slice ratios. We compared the test case with the attributes in the case library using Euclidean distance. To speed up the retrieval, brute force [16] was applied. Investigating a faster search structure would form part of future work using the approach here as a benchmark.

Each case in the library is an d -dimensional space and the Euclidean distance between the query case (q) and each library case (c_i) represents the similarity and can be expressed as:

$$d_i = \sqrt{\sum_{j=1}^d (q_j - c_{i,j})^2} \tag{3}$$

By calculating this, we can find the set of k nearest neighbors, i.e., those with the smallest value of d_i . Each of those k neighbors has a value for the slice ratio and the KNN approach [17] takes these k values of distance and output (slice ratio) to predict the value of w for the query case, which we denote as w_q . This prediction is generally conducted by using a weighted combination [18] of the outputs with the weighting depending on distance. The $\alpha(\bullet)$ is the weight function and is shown in Table 2.

$$\omega_q = \frac{\sum_{i=1}^k \alpha(d_i) \times \omega_i}{\sum_{i=1}^k \alpha(d_i)} \tag{4}$$

Table 2. Weight functions.

Weight Functions	Formula
inverse distance	$\alpha(d) = 1/d$
quadratic kernel	$\alpha(d) = \begin{cases} (1 - d^2), & \text{if } d < 1 \\ 0, & \text{otherwise} \end{cases}$
tricube kernel	$\alpha(d) = \begin{cases} (1 - d ^3)^3, & \text{if } d < 1 \\ 0, & \text{otherwise} \end{cases}$
variant of the triangular kernel	$\alpha(d) = \begin{cases} \frac{1- d }{ d }, & \text{if } d < 1 \\ 0, & \text{otherwise} \end{cases}$

The k value metrics and distance metrics used in KNN algorithms have an impact on their performance. Using a set k value for all query cases is not feasible [19] as is using the time-consuming cross-validation approach [13] to obtain the k value for each query case. The k value should be different and should be learned from the data. The selection of an optimal k value for each query case as the nearest neighbor to conduct KNN by sparse learning, and locality preserving projections have shown to be highly effective [20,21].

The case library is denoted as a matrix $\mathbf{X} \in \mathbb{R}^{n \times d}$, where n and d , respectively, denote the quantity of practice samples and attributes. $\mathbf{X} = [x_{ij}]$, its i -th row and j -th column are denoted as x_i and x^j , respectively. $\mathbf{Y} \in \mathbb{R}^{d \times m}$ denotes the transpose matrix of the query cases, where m denotes the number of query case. In this paper, we propose to reconstructed query case \mathbf{Y} using case library \mathbf{X} , with the purpose of minimizing the distance between $\mathbf{X}^T \mathbf{w}_i$ and y_i . The reconstruction coefficient or correlation between the case library and query cases is denoted by $\mathbf{W} \in \mathbb{R}^{n \times m}$. To get the smallest reconstruction error, we use a least square loss function [22] as follows:

$$\min_{\mathbf{W}} \sum_{i=1}^m \|\mathbf{X}^T \mathbf{w}_i - y_i\|_2^2 = \min_{\mathbf{W}} \|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_F^2 \tag{5}$$

$\|\mathbf{X}\|_F^2 = \left(\sum_i \|x_i\|_2^2\right)^{1/2}$ is the Frobenius norms of \mathbf{X} . In this paper, we employ the following sparse objective function:

$$\min_{\mathbf{W}} \|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_F^2 + \rho_1 \|\mathbf{W}\|_1 \tag{6}$$

where $\|\mathbf{W}\|_1$ is an l_1 -norm regularization term [23] to generate the sparse reconstruction coefficient, and $\mathbf{W} \succeq 0$ means that each element of \mathbf{W} is nonnegative. The larger the value of ρ_1 , the more sparse is \mathbf{W} .

To make sure that the k -closest neighbors of the original data are kept in the new space after dimensional reduction, we impose the locality preserving projections (LPP) regularization term between the case library and themselves [24]. The case library $\mathbf{X} \in \mathbb{R}^{n \times d}$ is mapped to $\mathbf{Y} \in \mathbb{R}^{d \times m}$, a projection matrix is $\mathbf{W} \in \mathbb{R}^{n \times m}$, and $y_j = \mathbf{X}^T \mathbf{w}_j$. As a result, LPP's objective function is as follows:

$$\min_{\mathbf{Y}=\mathbf{X}^T\mathbf{W}} \frac{1}{2} \sum_{i,j} s_{ij} \|y_i - y_j\|_2^2 = \min_{\mathbf{W}} \text{tr}(\mathbf{W}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{W}) \tag{7}$$

Let s_{ij} denote the attribute similarity matrix. $\mathbf{S} = [s_{ij}] \in \mathbb{R}^{d \times d}$, which encodes the relationships between attributes. Let G denote a graph with d attributes. If nodes i and j are connected, we put $s_{ij} = e^{-\frac{\|x^i - x^j\|^2}{2\sigma}}$, (σ is a tuning parameter constant). We use an adjacency matrix \mathbf{S} to denote the attribute correlations graph. \mathbf{D} is a diagonal matrix whose entries are column (or row, since \mathbf{S} is symmetric) sums of \mathbf{S} , $\mathbf{D}_{ii} = \sum_{j=1}^d s_{ij}$. $\mathbf{L} \in \mathbb{R}^{d \times d}$, $\mathbf{L} = \mathbf{D} - \mathbf{S}$ is the Laplacian matrix [25]. The proof is shown in Appendix A.

Finally, the objective function for the reconstruction process can be defined as follows:

$$f(\mathbf{W}) = \min_{\mathbf{W}} (\|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_F^2 + \rho_1 \|\mathbf{W}\|_1 + \rho_2 \text{tr}(\mathbf{W}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{W}), \mathbf{W} \succeq 0 \tag{8}$$

$\text{tr}(\bullet)$ is the trace operator. We let Equation (8) take the first derivative with respect to \mathbf{W}^k , and then make it equal to 0 using the accelerated proximal gradient approach. The $\text{min} f(\mathbf{W}) = s(\mathbf{W}) + r(\mathbf{W})$. $s(\mathbf{W})$ is convex and differentiable as follows:

$$s(\mathbf{W}) = \min_{\mathbf{W}} (\|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_F^2 + \rho_2 \text{tr}(\mathbf{W}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{W})) \tag{9}$$

$r(\mathbf{W}) = \rho_1 \|\mathbf{W}\|_1$ is an l_1 -norm regularization term [26]. Further let $S(\mathbf{W}) = s(\mathbf{W})_1 + \rho_2 s(\mathbf{W})_2$.

$$\begin{aligned} s(\mathbf{W})_1 &= \|\mathbf{X}^T \mathbf{W} - \mathbf{Y}\|_F^2 \\ &= (\mathbf{X}^T \mathbf{W} - \mathbf{Y})^T (\mathbf{X}^T \mathbf{W} - \mathbf{Y}) \\ &= \mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W} - \mathbf{W}^T \mathbf{X} \mathbf{Y} - \mathbf{Y}^T \mathbf{X}^T \mathbf{W} + \mathbf{Y}^T \mathbf{Y} \end{aligned} \tag{10}$$

According to the Derivative properties of matrices $\frac{\partial(\mathbf{X}^T \mathbf{A} \mathbf{X})}{\partial \mathbf{X}} = (\mathbf{A} + \mathbf{A}^T) \mathbf{X}$, $\frac{\partial \mathbf{A}^T \mathbf{X}}{\partial \mathbf{X}} = \mathbf{A}$, we set $\mathbf{A} = \mathbf{X} \mathbf{X}^T$, and $\mathbf{X} = \mathbf{W}$. So we can obtained the following:

$$\frac{\partial(\mathbf{W}^T \mathbf{X} \mathbf{X}^T \mathbf{W})}{\partial \mathbf{W}} = (\mathbf{X} \mathbf{X}^T + \mathbf{X} \mathbf{X}^T) \mathbf{W} = 2 \mathbf{X} \mathbf{X}^T \mathbf{W} \tag{11}$$

set $\mathbf{A} = \mathbf{X} \mathbf{Y}$ and $\mathbf{X} = \mathbf{W}$

$$\frac{\partial(\mathbf{W}^T \mathbf{X} \mathbf{Y})}{\partial \mathbf{W}} = \frac{\partial(\mathbf{Y}^T \mathbf{X}^T \mathbf{W})}{\partial \mathbf{W}} = \mathbf{X} \mathbf{Y} \tag{12}$$

So

$$\frac{\partial s(\mathbf{W})_1}{\partial \mathbf{W}} = 2 \mathbf{X} \mathbf{X}^T \mathbf{W} - 2 \mathbf{X} \mathbf{Y} \tag{13}$$

As $s(\mathbf{W})_2 = \min_{\mathbf{W}} \text{tr}(\mathbf{W}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{W})$, according to the derivative formula of matrix trace $\frac{\partial \text{tr}(\mathbf{W}^T \mathbf{A} \mathbf{W})}{\partial \mathbf{W}} = (\mathbf{A} + \mathbf{A}^T) \mathbf{W}$ let $\mathbf{A} = \mathbf{X} \mathbf{L} \mathbf{X}^T$. Therefore,

$$\frac{\partial s(\mathbf{W})_2}{\partial \mathbf{W}} = (\mathbf{X} \mathbf{L} \mathbf{X}^T + \mathbf{X} \mathbf{L} \mathbf{X}^T) \mathbf{W} = 2 \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{W} \tag{14}$$

wherein, \mathbf{L} is a symmetric matrix, so $\mathbf{L} = \mathbf{L}^T$.

$$\frac{\partial s(\mathbf{W})}{\partial \mathbf{W}} = 2\mathbf{X}\mathbf{X}^T\mathbf{W} - 2\mathbf{X}\mathbf{Y} + 2\rho_2\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{W} = 2(\mathbf{X}\mathbf{X}^T + \rho_2\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{W} - 2\mathbf{X}\mathbf{Y} \quad (15)$$

Gradient descent is performed on the smooth term to obtain an intermediate result.

$$\mathbf{W}^{k+\frac{1}{2}} = \mathbf{W}^k - \alpha * \nabla s(\mathbf{W}^k) \quad (16)$$

where α is the learning rate. So

$$\mathbf{W}^{k+\frac{1}{2}} = \mathbf{W}^k - 2\alpha(\mathbf{X}^T\mathbf{X} + p_2\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{W}^k + 2\alpha\mathbf{X}\mathbf{Y} \quad (17)$$

Then the intermediate result is substituted into the non-smooth term to obtain the projection of its adjacent points, that is, to complete an iteration.

$$\mathbf{W}^{k+1} = \underset{\mathbf{W}}{\operatorname{argmin}} \{ \rho_1 \|\mathbf{W}\|_1 + \frac{1}{2\alpha} \|\mathbf{W} - \mathbf{W}^{k+\frac{1}{2}}\|^2 \} \quad (18)$$

Let

$$v(\mathbf{W}) = \left\{ p_1 \|\mathbf{W}\|_1 + \frac{1}{2\alpha} \|\mathbf{W} - \mathbf{W}^{k+\frac{1}{2}}\|^2 \right\} \quad (19)$$

$$\frac{\partial v(\mathbf{W})}{\partial (\mathbf{W})} = \frac{1}{\alpha} (\mathbf{W} - \mathbf{W}^{k+\frac{1}{2}}) + p_1 \operatorname{sgn}(\mathbf{W}) \quad (20)$$

For the derivatives of the $\rho_1 \|\mathbf{W}\|_1$, soft thresholding [27] is used. Then, we update the \mathbf{W} as follows:

$$\mathbf{W}^{k+1} = \operatorname{soft}(\mathbf{W}^{k+\frac{1}{2}}, \alpha p_1) = \operatorname{sign}(\mathbf{W}^{k+\frac{1}{2}}) \max\{|\mathbf{W}^{k+\frac{1}{2}}| - \alpha p_1, 0\} \quad (21)$$

$$\mathbf{W}^{k+1} = \begin{cases} \mathbf{W}^{k+\frac{1}{2}} + \alpha p_1, & \mathbf{W}^{k+\frac{1}{2}} < -\alpha p_1 \\ 0, & |\mathbf{W}^{k+\frac{1}{2}}| \leq \alpha p_1 \\ \mathbf{W}^{k+\frac{1}{2}} - \alpha p_1, & \mathbf{W}^{k+\frac{1}{2}} > \alpha p_1 \end{cases} \quad (22)$$

By the convergence theorem [28],

$$f(\mathbf{W}(t_k)) - f(\mathbf{W}^*) \leq \frac{2\gamma L \|\mathbf{W}(1) - \mathbf{W}^*\|_F^2}{(t+1)^2} \quad (23)$$

where γ is a positive predefined constant, L is a gradual Lipschitz constant. In combination with Equation (22), the \mathbf{W}^{k+1} is as Equation (24)

$$\mathbf{W}^{k+1} = \begin{cases} \mathbf{W}^k - 2\alpha(\mathbf{X}\mathbf{X}^T + p_2\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{W}^k + 2\alpha\mathbf{X}\mathbf{Y} + \alpha p_1, & \mathbf{W}^{k+\frac{1}{2}} < -\alpha p_1 \\ 0, & |\mathbf{W}^{k+\frac{1}{2}}| < \alpha p_1 \\ \mathbf{W}^k - 2\alpha(\mathbf{X}\mathbf{X}^T + p_2\mathbf{X}\mathbf{L}\mathbf{X}^T)\mathbf{W}^k + 2\alpha\mathbf{X}\mathbf{Y} - \alpha p_1, & \mathbf{W}^{k+\frac{1}{2}} > \alpha p_1 \end{cases} \quad (24)$$

The pseudocode of O-KNN is given in Algorithm 2.

Algorithm 2 Pseudocode of O-KNN.**Input:** $\mathbf{X}, \mathbf{Y} (\mathbf{X} \in \mathbb{R}^{n \times d}, \mathbf{Y} \in \mathbb{R}^{d \times m});$ **Output:** predict solution \hat{y} ;

```

1: Initialize  $y_0 = \mathbf{W}^0 \in \mathbb{R}^{n \times m}, t_0 = 1.$ 
2: Calculate the Laplace matrix  $\mathbf{L} \in \mathbb{R}^{d \times d};$ 
3:  $k = 0$ 
4: while (1) do
5:   Update  $\mathbf{W}^{k+\frac{1}{2}} = y_k - \alpha \times \nabla s(y_k)$  by Equation (17)
6:   Update  $\mathbf{W}^{k+1}$  by Equation (24)
7:   Update step length  $\alpha$ 
8:   Update  $t_{k+1} = \frac{1}{2} + \frac{1}{2} \sqrt{1 + 4t_k^2}$ 
9:   Update  $y_{k+1} = \mathbf{W}^k + \left(\frac{t_k-1}{t_{k+1}}\right) (\mathbf{W}^{k+1} - \mathbf{W}^k)$ 
10:   $k+ = 1$ 
11:  If condition Equation (23) is satisfied Break;
12: end while
13: print  $\mathbf{W}^* \in \mathbb{R}^{n \times m}$ 
14: for each  $i \in m$  do
15:    $k_i = 0$ 
16:   for each  $j \in n$  do
17:     if  $\mathbf{W}_{ij}^* \neq 0$  then
18:        $k_i+ = 1;$ 
19:     end if
20:   end for
21: end for
22: Print  $k^* = [k_1, k_2, \dots, k_m]$ 
23: Similar calculate by Euclidean distance Equation (3)
24: The most similar  $k$  cases with its corresponding solution is searched by brute-force.
25: Predicted the slice ratio by Equation (4)

```

4. Numerical Results*4.1. Experimental Settings*

The purpose of this research is to propose a reliable scheme for estimating the slice ratio for a query case. The scenario parameter settings are shown in Table 3. We consider an RAN with seven cell wrap-arounds (matching the user distribution of the center cell only, other cells acting as interference sources). The hardware used to carry out this experiment is a processor Core Intel(R) Core(TM) i5-8250U.

Table 3. Simulation configuration.

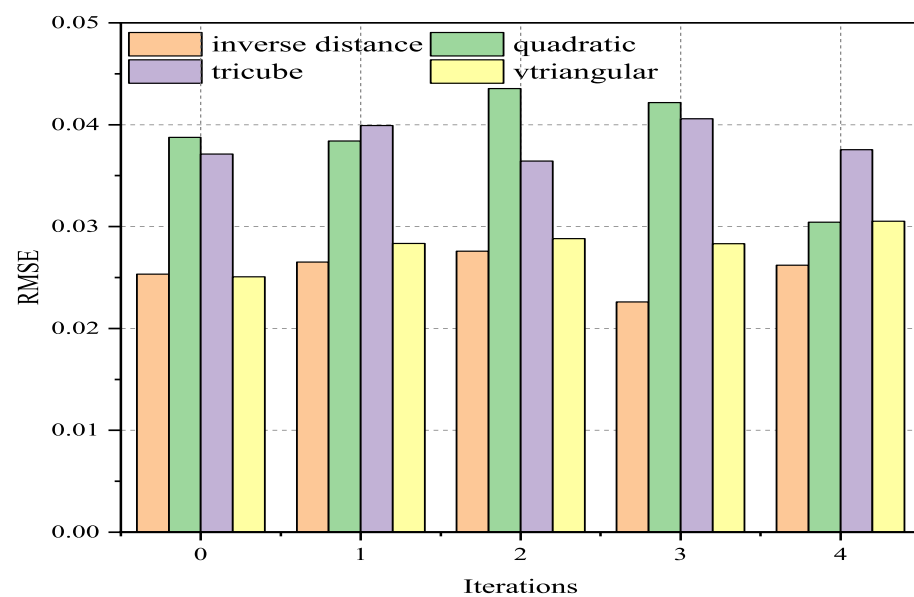
Parameter	Value
Cell radius	1 km
Cell number	7
Modulation format	OFDM
Number of slices	2
Cell bandwidth	10 MHz
Number of users	200
Antenna height	75 m
Carrier frequency	2 GHz

Table 3. *Cont.*

Parameter	Value
Subcarrier spacing	15 kHz
Transmit power	23 dBm
Shadow fading	log-normal
Path loss	COST231-Hata
Required bitrate of S1 users	1 Mbps
Required bitrate of S2 users	2 Mbps
User distribution	Uniform
No. of cases in library	Up to 10,000
No. of test (query) cases	300

4.2. Choice of Weight Function

We tested the distance with different weight functions from Table 2, and the k value were determined by cross-validation with 300 query cases. The result is shown in Figure 4 and shows that the inverse distance has the minimum RMSE of the slice ratio. Hence, the inverse distance is used for all subsequent tests.

**Figure 4.** Prediction error with different weight functions.

4.3. Determination of k

For choosing the best value of k , we compared three different k value determination methods, fixed value k (1,3,5,7), the K value determined by cross-validation as CV-KNN [13], and the proposed O-KNN. The prediction error for these different approaches over different numbers of query cases is shown in Figure 5. Obviously, the O-KNN and CV-KNN outperform the fixed k value with the O-KNN being slightly better.

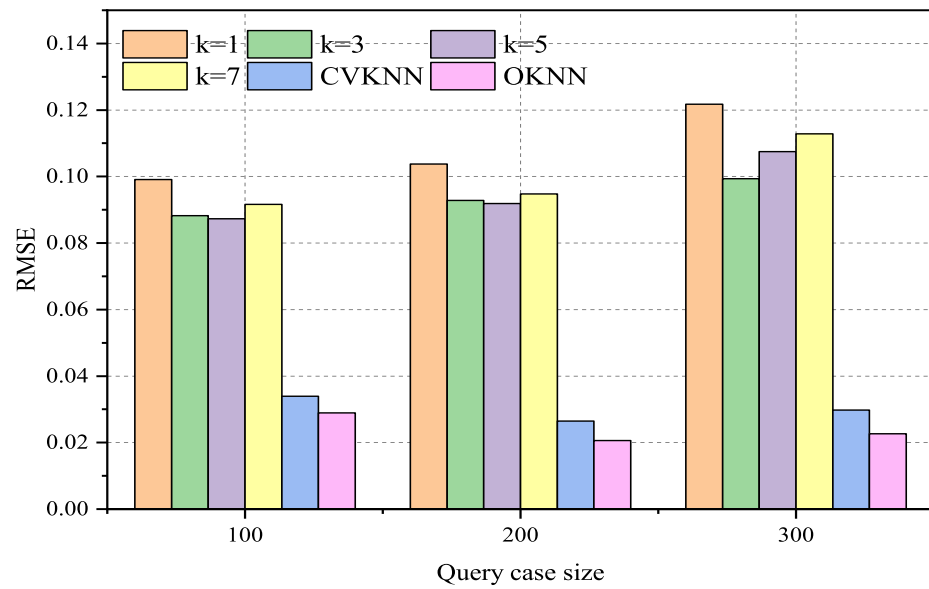


Figure 5. Prediction error for different methods.

The other factor to consider is the search time to find the best k value. This is illustrated in Figure 6. The search time is averaged over 300 query cases. The result show that O-KNN has a search time that is approximately one-tenth of that of CV-KNN because the search space is reduced by spare learning.

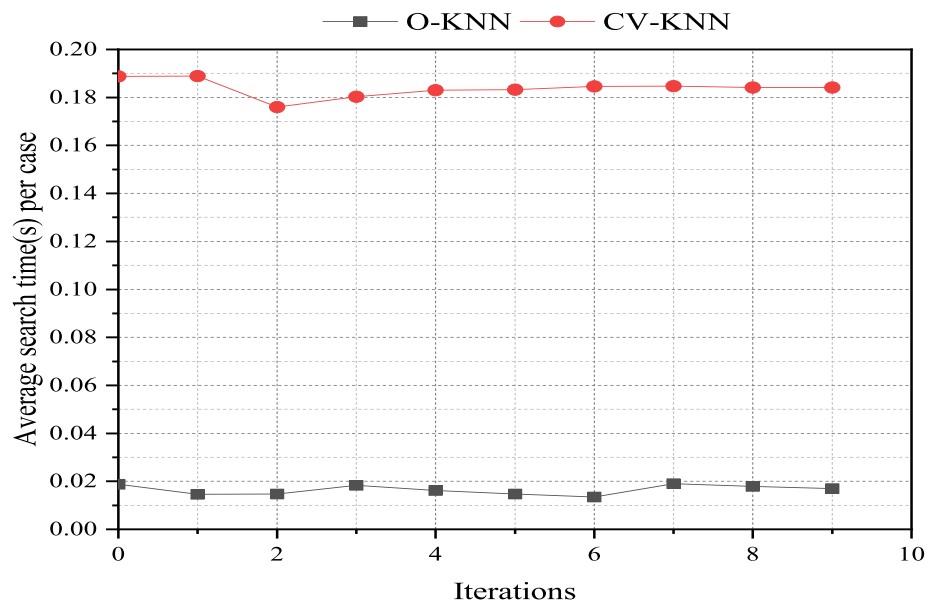


Figure 6. K-value search time.

4.4. Slice Bandwidth Ratio Prediction

The test result above shows that the O-KNN is suitable for slice ratio prediction in terms of running cost and RMSE. Therefore, the O-KNN can be used to test the slice ratio prediction. The slice ratio distribution of the test and prediction is shown in Figure 7. The test the prediction of two slices represented on the horizontal axis, the slice ratio is represented on the vertical axis. From Figure 7, we see that the slice ratio of slice1 median is around 0.55, the slice ratio of slice2 median is around 0.45, and their sum is 1. However, both the slice ratio distribution of the test and prediction have almost the same distribution; thus, the prediction is good. The RMSE of all the 300 query cases is 0.01986 by O-KNN.

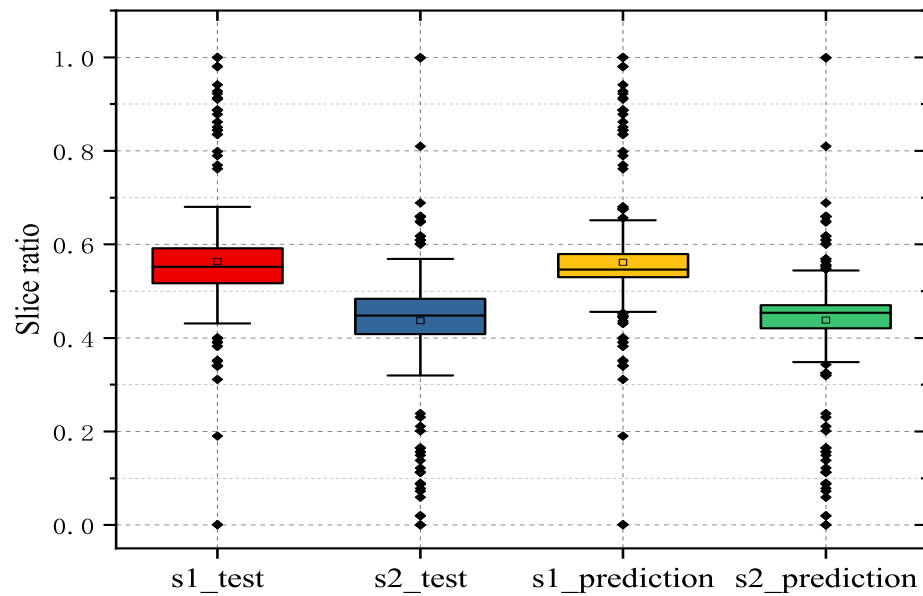


Figure 7. The slice ratio distribution of test and prediction.

4.5. Qualified User Ratio

QUR is the ultimate measure of performance. We tested the QUR of an additional 20 query cases with a different number of users and the results are shown in Figure 8. PQUR is the predicted QUR using the O-KNN. TQUR is the QUR from exhaustive search for those cases. HQUR denotes the QUR by the hard slicing. Hard slicing means that each service is always allocated with $\frac{1}{2}$ of the whole bandwidth (because there are two types of services in total).

Obviously, the PQUR by the O-KNN approach has a very low prediction error with the TQUR, and is outperformed by HQUR. Therefore, the O-KNN approach is effective.

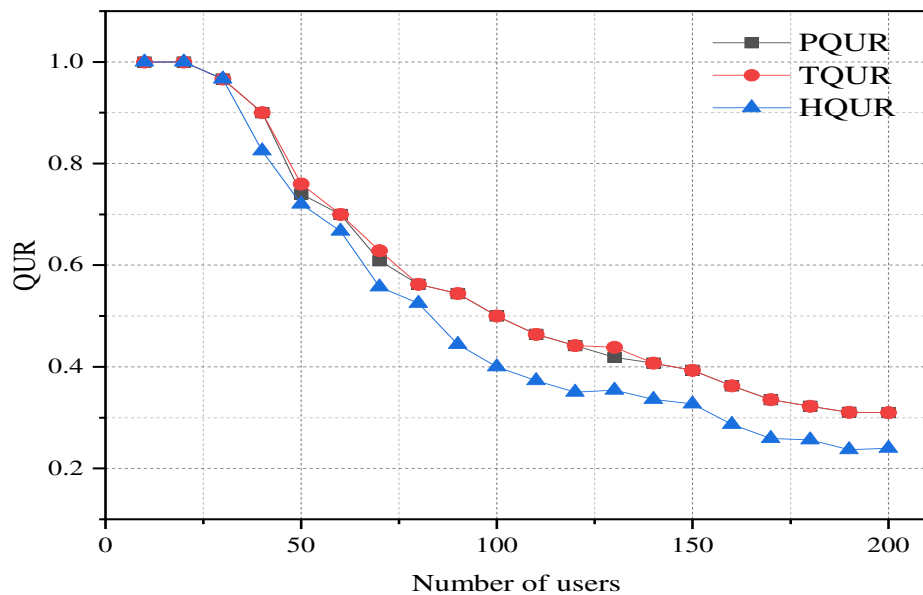


Figure 8. QUR test.

5. Conclusions

In this paper, we investigated the feasibility of using the sparse learning and locality-preserving projections approach in the CBR framework for the problem of finding the best match to determine the slice ratio of network slicing in RAN. We first tested the error

in determining the ratio using different weighting functions. The best weight function was found to be the inverse distance function. When comparing the KNN methods, we showed that O-KNN outperforms other approaches to determine the k value, although only marginally better than CV-KNN. However, the k value search time of O-KNN was much better than that of CV-KNN because of the reduction in the search space. The outcome was that the proposed algorithm can perform effective resource allocation for S1 and S2 hybrid services and is worth pursuing in a more complex scenario.

Overall, therefore, O-KNN is very effective for resource allocation in network slicing. In the future, we will simulate a more practical environment with multiple attributes to increase the matching with fewer cases and to allow for different values for S1 and S2. We will investigate a more general optimization framework that can cope with more than two slices.

Author Contributions: Methodology, L.C.; Software, D.Y.; Formal analysis, X.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

As shown in Equation (A1), where, $\mathbf{D}_{ii} = \sum_j s_{ij}$. \mathbf{L} and \mathbf{D} are real symmetric matrices.

$$\begin{aligned}
 \frac{1}{2} \sum_{i,j} s_{ij} \|y_i - y_j\|_2^2 &= \frac{1}{2} \sum_{i=1}^d \sum_{j=1}^d s_{ij} (y_i^T y_i - 2y_i^T y_j + y_j^T y_j) \\
 &= \frac{1}{2} \left(\sum_{i=1}^d \left(\sum_{j=1}^d s_{ij} \right) y_i^T y_i + \sum_{j=1}^d \left(\sum_{i=1}^d s_{ij} \right) y_j^T y_j - 2 \sum_{i=1}^d \sum_{j=1}^d y_i^T y_j s_{ij} \right) \\
 &= \sum_{i=1}^d D_{ii} y_i^T y_i - \sum_{i=1}^d \sum_{j=1}^d y_i^T y_j s_{ij} \\
 &= \sum_{i=1}^d \left(\sqrt{D_{ii}} y_i \right)^T \left(\sqrt{D_{ii}} y_i \right) - \sum_{i=1}^d y_i^T \left(\sum_{j=1}^d y_j s_{ij} \right) \\
 &= \text{tr}(\mathbf{Y}^T \mathbf{D} \mathbf{Y}) - \text{tr}(\mathbf{Y}^T \mathbf{S} \mathbf{Y}) \\
 &= \text{tr}[\mathbf{Y}^T (\mathbf{D} - \mathbf{S}) \mathbf{Y}] \\
 &= \text{tr}(\mathbf{Y}^T \mathbf{L} \mathbf{Y}) \\
 &= \text{tr}(\mathbf{W}^T \mathbf{X} \mathbf{L} \mathbf{X}^T \mathbf{W})
 \end{aligned} \tag{A1}$$

References

1. Zhang, H.; Liu, N.; Chu, X.; Long, K.; Aghvami, A.H.; Leung, V.C. Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges. *IEEE Commun. Mag.* **2017**, *8*, 138–145. [[CrossRef](#)]
2. Afolabi, I.; Taleb, T.; Samdanis, K.; Ksentini, A.; Flinck, H. Network slicing and softwarization: A survey on principles, enabling technologies, and solution. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 2429–2453. [[CrossRef](#)]
3. Guijarro, L.; Vidal, J.R.; Pla, V. Competition Between Service Providers With Strategic Resource Allocation: Application to Network Slicing. *IEEE Access* **2018**, *9*, 6503–76517.
4. Liu, Q.; Han, T.; Zhang, N.; Wang, Y.D. DeepSlicing: Deep Reinforcement Learning Assisted Resource Allocation for Network Slicing. In Proceedings of the 2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.

5. Wang, H.; Wu, Y.; Min, G.; Xu, J.; Tang, P.D. Data-driven dynamic resource scheduling for network slicing: A Deep reinforcement learning approach. *Inf. Sci.* **2019**, *498*, 106–116. [[CrossRef](#)]
6. Sun, G.; Gebrekidan, Z.T.; Boateng, G.O.; Ayepah-Mensah, D.; Jiang, W. Dynamic Reservation and Deep Reinforcement Learning Based Autonomous Resource Slicing for Virtualized Radio Access Networks. *IEEE Access* **2019**, *7*, 45758–45772. [[CrossRef](#)]
7. Akgul, O.U.; Malanchini, I.; Capone, A. Anticipatory Resource Allocation and Trading in a Sliced Network. In Proceedings of the ICC 2019–2019 IEEE International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–7.
8. Feng, L.; Zi, Y.; Li, W.; Zhou, F.; Yu, P.; Kadoch, M.D. Dynamic resource allocation with RAN slicing and scheduling for uRLLC and eMBB hybrid service. *IEEE Access* **2020**, *8*, 34538–34551. [[CrossRef](#)]
9. Sun, Y.; Peng, M.; Mao, S.; Yan, S. Hierarchical Radio Resource Allocation for Network Slicing in Fog Radio Access Networks. *IEEE Trans. Veh. Technol.* **2020**, *64*, 3866–3881. [[CrossRef](#)]
10. Yang, X.; Liu, Y.; Chou, K.; Cuthbert, L. A game theoretic approach to network slicing. In Proceedings of the 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, Australia, 22–24 November 2017; pp. 1–4.
11. Yang, X.; Wang, Y.; Wong, I.C.; Liu, Y.; Cuthbert, L. Genetic Algorithm in Resource Allocation of RAN Slicing with QoS Isolation and Fairness. In Proceedings of the Proceedings-2020 IEEE Latin-American Conference on Communications, LATINCOM, Santo Domingo, Dominican Republic, 18–20 November 2020.
12. Yang, X.; Liu, Y.; Wong, I.C.; Wang, Y.; Cuthbert, L. Effective isolation in dynamic network slicing. In Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC, Marrakesh, Morocco, 15–18 April 2019.
13. Yan, D.; Yang, X.; Cuthbert, L. Regression-based K Nearest Neighbours for Resource Allocation in Network Slicing. In Proceedings of the 2022 Wireless Telecommunications Symposium (WTS), Pomona, CA, USA, 6–8 April 2022; pp. 1–6.
14. Yao, N. *A CBR Approach for Radiation Pattern Control in WCDMA Networks*; Queen Mary University of London: London, UK, 2007.
15. Chantaraskul, S. *An Intelligent-Agent Approach for Managing Congestion in W-CDMA Networks*; Queen Mary University of London: London, UK, 2005.
16. Morton, A.B.; Mareels, I.M. An efficient brute-force solution to the network reconfiguration problem. *IEEE Trans. Power Deliv.* **2000**, *15*, 996–1000. [[CrossRef](#)]
17. Yao, Z.; Ruzzo, W.L. A regression-based K nearest neighbor algorithm for gene function prediction from heterogeneous data. *BMC Bioinform.* **2006**, *7*, 1–11. [[CrossRef](#)] [[PubMed](#)]
18. Kang, S. k-Nearest Neighbor Learning with Graph Neural Networks. *Water Resour. Res.* **2021**, *9*, 830. [[CrossRef](#)]
19. Lall, U.; Sharma, A. A nearest neighbor bootstrap for resampling hydrologic time series. *Water Resour. Res.* **1996**, *32*, 679–693. [[CrossRef](#)]
20. Zhang, S.; Li, X.; Zong, M.; Zhu, X.; Wang, R. A Efficient kNN classification with different numbers of nearest neighbor. *IEEE Trans. Neural Netw. Learn. Syst.* **1996**, *29*, 1774–1785. [[CrossRef](#)] [[PubMed](#)]
21. Zhang, S.; Zong, M.; Sun, K.; Liu, Y.; Cheng, D. Efficient kNN algorithm based on graph sparse reconstruction. In *International Conference on Advanced Data Mining and Applications*; Springer: Cham, Switzerland, 2014; pp. 153–160.
22. Zhu, X.; Suk, H.I.; Shen, D. Matrix-similarity based loss function and feature selection for alzheimer’s disease diagnosis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 3089–3096.
23. Li, X.; Pang, Y.; Yuan, Y. L1-norm-based 2DPCA. *IEEE Trans. Syst.* **1996**, *40*, 1170–1175.
24. Zhu, X.; Suk, H.I.; Shen, D. A novel multi-relation regularization method for regression and classification in AD diagnosis. In *International Conference on Medical Image Computing and Computer-Assisted Intervention. (2014, September)*; Springer: Cham, Switzerland, 2014; pp. 401–408.
25. He, X.; Niyogi, P. Locality preserving projections. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2003; Volume 16.
26. Wright, S.J.; Nowak, R.D.; Figueiredo, M.A. Sparse reconstruction by separable approximation. *IEEE Trans. Signal Process.* **2009**, *57*, 2479–2493 [[CrossRef](#)]
27. Beck, A.; Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.* **2009**, *2*, 183–202. [[CrossRef](#)]
28. Gong, Y.; Dend, Z.; Sun, K.; Liu, Y. kNN Regression Algorithm Based on LPP and Lasso. *J. Chin. Comput. Syst.* **2015**, *36*, 2604–2608.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.