Scholar Publishing Group
Distributed Processing System
https://doi.org/10.38007/DPS.2020.010205
ISSN 2790-0916 Vol. 1, Issue 2: 37-45

A Collaborative Design System of Distributed Computing Mode Based on Node Adaptive Algorithm

Muhama Hasan*

Indira Gandhi Delhi Technical University for Women, India

**corresponding author*

Keywords: A Node Adaptive Algorithm, Distributed Computing Mode, Collaborative Design System, Recommendation System

Abstract: An adaptive algorithm is relatively rare in the field of collaborative design systems for distributed computing models. Therefore, the existing adaptive algorithms may not be able to satisfy this systematic study in practical applications. However, as an important branch of adaptive filtering algorithms, adaptive algorithms in distributed systems have received more and more attention in recent years. The research purpose of this paper is a distributed computing mode collaborative design system under the node adaptive algorithm. In the experiment, the distributed framework Spark is constructed, and the collaborative filtering recommendation algorithm is used to investigate and analyze the user's movie tag preference and the operation time of distributed computing in the system. Combining the distributed computing model recommendation system with the collaborative design system has become an innovative point of the research.

1. Introduction

Due to its advantages of high reliability, scalability, and high profitability, distributed computing has gradually become the mainstream computing method for large-scale machine learning and big data analysis [1]. However, there are lagging nodes in distributed systems, which lead to an increase in the total time required for tasks, thus limiting the performance of distributed computing. Coding distributed computing is a new computing paradigm that uses coding methods to create storage or computational redundancy to mitigate the impact of unpredictable lagging nodes. However, most of the existing coding calculation schemes applied to the master-slave computing framework only use the calculation results of a certain number of fastest working nodes to restore the output, completely ignoring the work done by other nodes, resulting in low performance.

The rapid growth of the amount of information data in the Internet era has made it more difficult for users to find information.Giv H proposes an incremental adaptive algorithm capable of solving

distributed computations in Hamiltonian networks, where the measurements of a single node can be corrupted by severe excitation noise. In the proposed algorithm, each node receives an error offset in the update equation to reduce the adverse effects of spurious noise. In addition, the algorithm computes the optimal absolute error and unknown estimates simultaneously, thereby eliminating the requirement for prior knowledge of the statistical properties of the measurement noise. In addition to algorithm improvement, its steady-state performance and convergence analysis is also provided. Simulation results confirm the accuracy of the analysis and show that the proposed algorithm outperforms existing algorithms [2]. Dkerolu T studied the Steiner tree problem (STP) as a challenging problem. STP with revenue and budget is a segmentation of undirected graphs and proposes a BLS algorithm to solve STPRBH, which is a way to keep BLS supported opportunities at the number of master nodes and use the best performance opportunities sent by slave nodes to adjust/update they. Perform BLS searches from nodes and use a multi-start procedure to prevent them from getting stuck in local optima by restarting the search process. A master-slave communication topology is used to communicate with slave controllers [3].The adaptive algorithm provides basic theoretical support for solving the problem, and has been well applied in a variety of scenarios.

This paper studies the related research of adaptive algorithm, including the introduction of adaptive algorithm and adaptive related algorithm, an overview of distributed computing, the establishment of collaborative design system modules, and the combination of distributed computing model recommendation system and collaborative design system. In the experiment, a distributed framework Spark is constructed, text data and processing technology are obtained, and the collaborative filtering recommendation algorithm is used to investigate and analyze the user's movie tag preference and the operation time of distributed computing in the system. In formula computing, Spark platform has practical research significance in collaborative design system and recommendation system.

2. A Collaborative Design System of Distributed Computing Mode Based on Node Adaptive Algorithm

2.1. Research on Adaptive Algorithms

(1) Introduction of adaptive algorithm

Mining class association rules is a basic but time-complex task in association classification. A class association rule is simply a special association rule whose last item contains only one class label [4-5]. The traditional class association rule mining algorithm considers how to mine interesting class association rules more efficiently when the number of attributes of the rules does not change. Mining class association rules means mining the correspondence between frequently occurring attribute-value pairs and classes in antecedents in a dataset. But in reality, the properties of class association rules may not be static, but may be constantly changing. When the number of attributes describing class association rules increases, the traditional idea is to re-use the mining algorithm to mine the expanded data set again. Such an algorithm is complete, but the efficiency is very low. The reason for the low efficiency is that the results of the first mining are not fully utilized. After the attributes are expanded, the existing class association rules are repeatedly mined using the traditional algorithm [6-7]].

(2) Adaptive correlation algorithm

The adaptive algorithm is also one of the main factors affecting various performance indicators of the adaptive filter, which corresponds to the adaptive update part of the adaptive filter [8-9]. The

function of the adaptive algorithm is to adjust the parameters of the adaptive filter through the input and output values of the filter, so that the output result of the adaptive filter is closest to the theoretical value. Therefore, the pros and cons of the adaptive algorithm will also affect the performance of the adaptive filter. Adaptive algorithms are mainly classified according to different derivation methods, and the two main categories are stochastic gradient algorithms and least squares algorithms.

The stochastic gradient algorithm is based on the statistical characteristics of the input data, and the mean square error of the estimated algorithm is used as the derivation criterion of the algorithm. One of the most classic algorithms is the least mean square algorithm [10-11]. Due to its low computational complexity, easy-to-understand algorithm and simple structure, the least-mean-square algorithm is often used as the basic algorithm for adaptive filter research, and many derivative algorithms have been improved on the basis of this algorithm.

The least squares algorithm is also known as the exact algorithm. It uses the second-order statistic, that is, the minimum sum of squares of errors as the derivation criterion of the algorithm. The correlation function of the algorithm contains the collected real data. The algorithm that meets this kind of quasi-measure is generally called Recursive least squares algorithm.

2.2. Overview of Distributed Computing

As more and more computing tasks are faced with the challenges of large data volume, rapid growth, and diverse information, the distributed computing model is beginning to show its extraordinary value for solving the above challenges and is improving rapidly. Distributed computing is a scientific computing method that divides engineering data that requires a large amount of calculation into small pieces, and is calculated separately by multiple computers. At present, Hadoop and Spark are the most widely used. Hadoop is the first distributed computing framework used in recommender systems, which significantly improves the recommendation speed. However, due to the storage structure and computing mode of Hadoop, the I/O operations in the computing process are quite frequent, which in turn causes its computing speed to slow down significantly when a large amount of data is exchanged. After researching and analyzing the defects of Hadoop in the laboratory, researchers designed Spark with better stability and performance. Compared with Hadoop, Spark's biggest improvement is to use Resilient Distributed Dataset (RDD) to solve the problem of excessive I/O, and it is also compatible with the Map Reduce model in the Hadoop framework [14-15]. At present, Spark has publicly disclosed the interface information and the complete deployment method. The friendly open source model has made Spark widely used in various engineering application fields, and has completely replaced Hadoop in most fields. However, the application of Spark in the field of recommendation system is still very small, and the degree of integration is also very low. Therefore, the real-time recommendation system under the distributed computing framework still has high research value [16].

2.3. Co-design System Modules

The network-based collaborative design system of functional modules provides users with a collaborative platform portal. The business logic of the entire system is divided into modules and designed to modularize functions. Each module is further subdivided into finer-grained system functions, and modules work cooperatively based on interface communication to provide users with collaborative information [17]. The system module architecture diagram is shown in Figure 1:

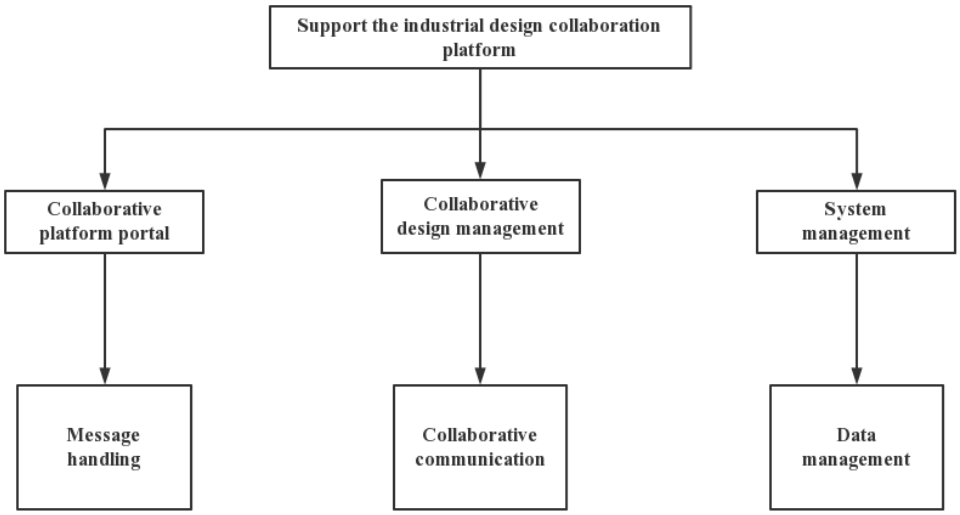

Figure 1. Co-design of the system module

2.4. The Combination of Distributed Computing Model Recommendation System and Collaborative design System

Combined with the recommender system to better study the collaborative design system of distributed computing mode based on adaptive algorithm. Since we are in the era of big data, the explosion of information volume has brought troubles to people's lives. In this environment, both providers and consumers of information face great challenges. Because they do not know the specific needs of consumers, information providers may provide consumers with information that they do not need, which not only does not help consumers, but also brings information harassment to consumers, and consumers also because of It is troubled by the inability to efficiently filter out the information that is useful to you. In this case, it is very valuable for consumers to infer their preference information based on the historical behavior of consumers, and then send the information in a targeted manner, which is also the origin of the recommendation system. The combination of recommendation system and collaborative design system makes the research more perfect and reasonable.

3. Investigation and Research on a Distributed Computing Mode Collaborative Design System under a Node Adaptive Algorithm

3.1. Building a Distributed Framework Spark

Spark is both a big data processing engine and a distributed big data processing framework. Recommender systems usually actively collect user data, generate recommendation lists by predicting user preferences, and recommend them to users. The massiveness and diversity of big data have spawned many big data application technologies. Among them, distributed frameworks such as Hadoop and Spark and distributed cluster technologies have great advantages in solving the storage, analysis and processing problems of massive data.

3.2. Acquisition and Processing Technology of Text Data

In order to ensure the real availability of data analysis results, it is necessary to improve the quality of data and the level of data processing. Since the quality of the dataset will greatly affect the training results of the recommendation model, the source of data acquisition is very important. This paper obtains movie-related data from the largest domestic movie review website, and performs cleaning, screening, format conversion and quantification operations on the obtained data, so that it can finally be analyzed and processed by a mathematical model.

3.3. Collaborative Filtering Recommendation

Collaborative filtering recommendation is a recommendation based on the user's behavior, and its targeted behavior can be browsing, purchasing and scoring of products in the past. Whether it is a user-based recommendation or an item-based recommendation, the similarity is calculated. There are several ways to calculate similarity, as follows:

(1) Jaccard similarity

In the formula, there are two sets a and b, and the Jaccard similarity is defined as the ratio of the size of the intersection of a and b to the size of the union of a and b. The larger the Jaccard value, the higher the similarity.

$$J(a,b)=\frac{|a\cap b|}{|a\cup b|}=\frac{|a\cap b|}{|a|+|b|-|a\cap b|} \tag{1}$$

(2) Cosine similarity

$$cos\theta=\frac{a\bullet b}{\|a\|\times\|b\|} \tag{2}$$

Among them, the known vectors a and b, and the cosine similarity is to calculate the cosine value between the vectors. The larger the cosine value, the more similar it is.

4.Analysis and Research on the Combination of Distributed Computing Mode Collaborative Design System and Recommendation System

4.1 User's Movie Tag Preference

Optimizing the user's preference for tags If a user's rating behavior is less, it will lead to errors in the prediction results. The details are shown in Table 1 and Figure 2:

Table 1. Tag data table

Label	How much the user u likes the tag	Optimized how much the user u likes the tags
Tags 1	4.45	4.24
Tags 2	4.21	4.18
Tags 3	4.05	4.12

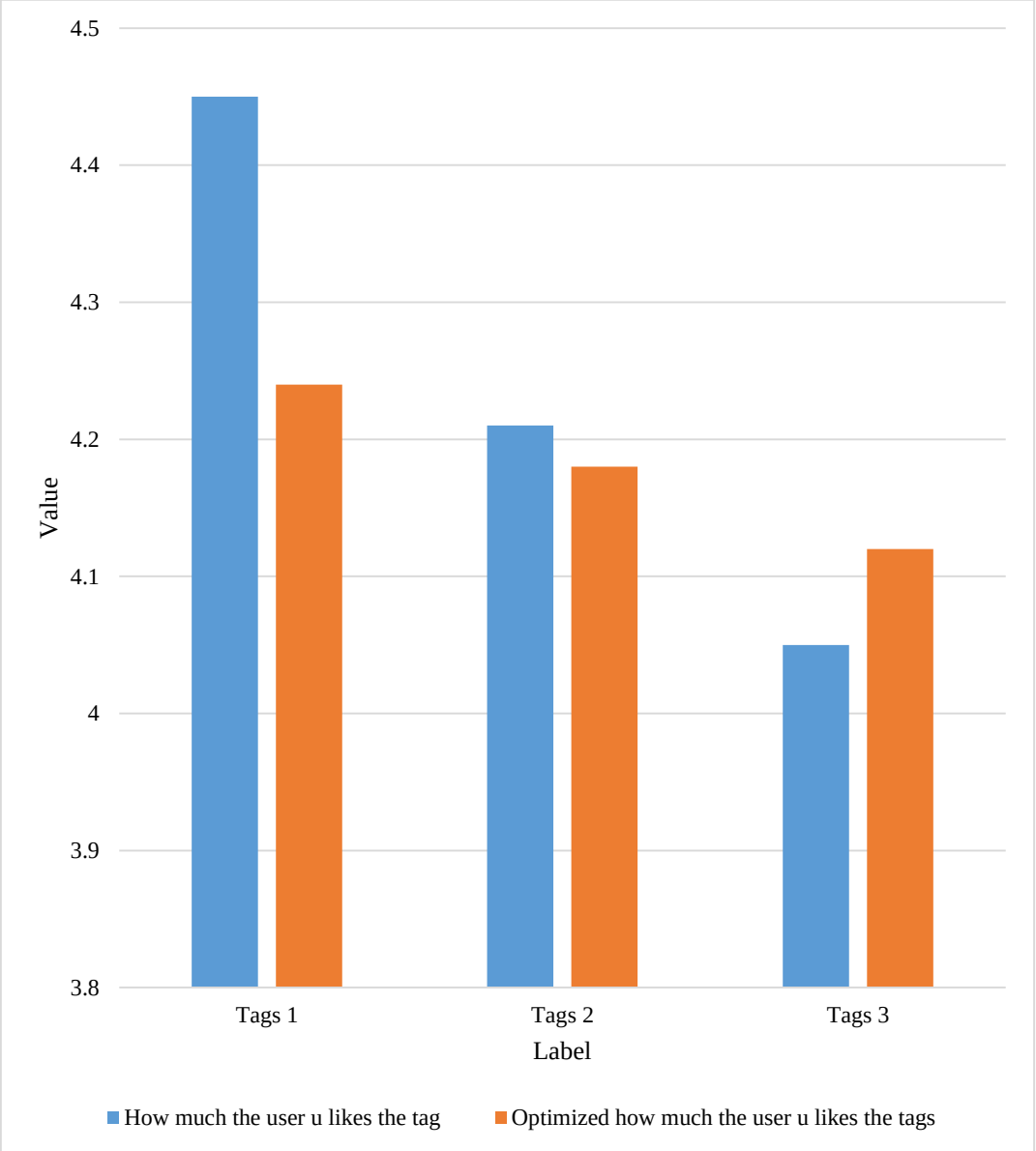

Figure 2. Comparison of the data before and after the addition of the influence factors

Analysis of the experimental results, comparison of the operation of each algorithm As the predecessor of Spark, Hadoop provides the storage capacity of massive data, and Map Reduce gives Hadoop the computing power to process these data. The RDD in Spark abstractly defines the use of distributed memory, so that Spark allows data sets to be cached in memory during data calculation, avoiding the time-consuming operation of Hadoop frequently repeatedly reading the disk. Therefore, the design and comparison of the efficiency of Hadoop and Spark in computing and processing data in this paper has far-reaching significance on which distributed technology to choose to deal with big data problems.

4.2. Operating Time of Distributed Computing in the System

The comparison of the running time of each part of the algorithm model in this paper for

processing datasets on Hadoop and Spark platforms is shown in Table 2 and Figure 3:

Table 2. Data sheets of different algorithms

Algorithms	Hadoop	Spark
TF-IDF	1785	236
Offline recommendation	684	86
Real-time recommendation	1856	257
Statistical recommendation	1452	185

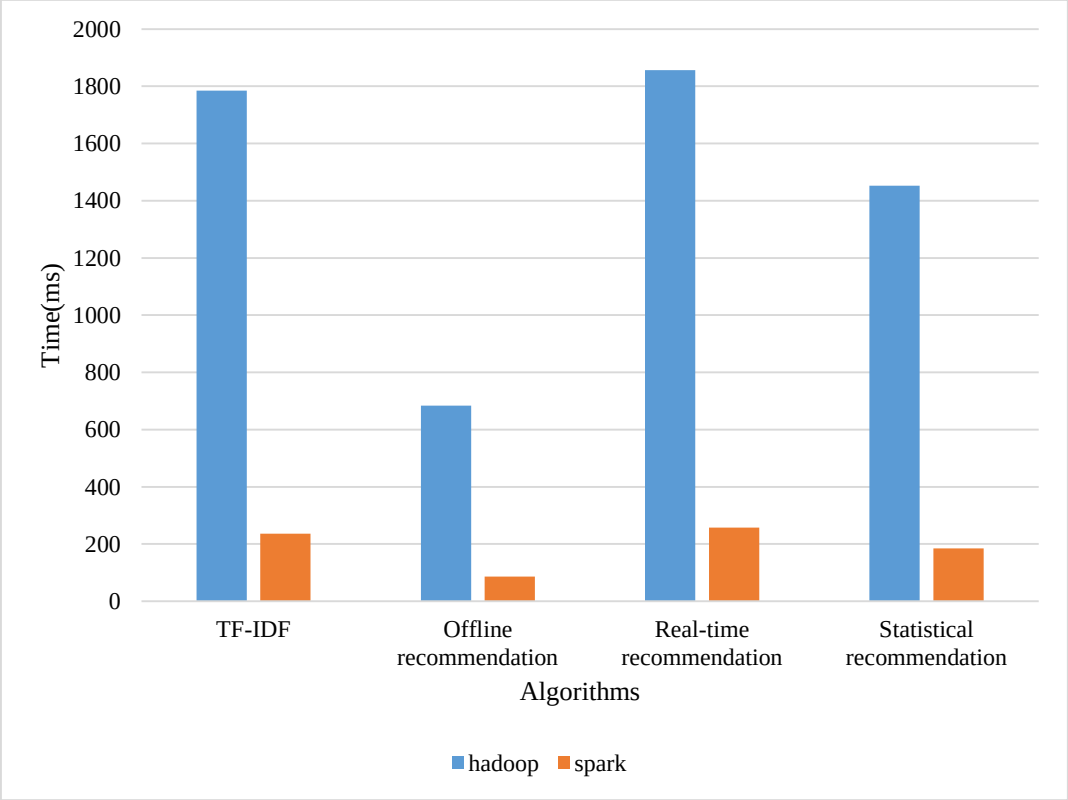

Figure 3. Running speed comparison diagram of different algorithms

The purpose of this experiment is to unify the size of the data set to verify the advantages of Spark's computing speed over Hadoop. From the above figure, it is obvious that Spark's running time is far less than Hadoop's when training data sets of various adaptive algorithms in this paper, especially when the real-time recommendation algorithm requires a large number of data access operations, Spark It still runs faster than Hadoop. This also proves that Spark platform in distributed computing has practical research significance in collaborative design system and recommendation system.

5. Conclusion

As the research on adaptive algorithms in distributed systems has received extensive attention, in the field of system identification, adaptive algorithms in distributed systems are often used to simulate some biological behaviors in nature, such as the disordered movement of bacteria and the migration of birds flight and ocean current fluctuations. This paper studies the combination of distributed computing model collaborative design system and recommendation system, analyzes the user's movie tag preference, and performs data processing and comparison on the running time of processing data sets on Hadoop and Spark platforms, etc. There are still many deficiencies in this study, and we hope to contribute to this research.

Funding

This article is not supported by any foundation.

Data Availability

Data sharing is not applicable to this article as no new data were created or analysed in this study.

Conflict of Interest

The author states that this article has no conflict of interest.

References

[1] Manoojkumaar S B, Poongodi C. A Novel Method for Node Connectivity With Adaptive Dragonfly Algorithm And Graph-Based m- Connection Establishment in MANET. Computers, Materials and Continua, 2020, 65(2):1649-1670. https://doi.org/10.32604/cmc.2020.010781

[2] Giv H, Khalili A, Rastegarnia A, et al. A Robust Adaptive Estimation Algorithm for Hamiltonian Multi-Agent Networks. IEEE Control Systems Letters, 2020, 5(4):1243-1248. https://doi.org/10.1109/LCSYS.2020.3029332

[3] Dkerolu T. A self-adaptive and stagnation-aware breakout local search algorithm on the grid for the Steiner tree problem with revenue, budget and hop constraints. Soft Computing, 2018, 22(1):1-19. https://doi.org/10.1007/s00500-017-2630-7

[4] Manoojkumaar S B, Poongodi C. A Novel Method for Node Connectivity With Adaptive Dragonfly Algorithm And Graph-Based m- Connection Establishment in MANET. Computers, Materials and Continua, 2020, 65(2):1649-1670.

[5] Giv H, Khalili A, Rastegarnia A, et al. A Robust Adaptive Estimation Algorithm for Hamiltonian Multi-Agent Networks. IEEE Control Systems Letters, 2020, 5(4):1243-1248.

[6] Sharma V, Pandey K, Patel R, Et Al. Robust Channel Encoding For Mmtc Node In 5G (NR). International Journal of Advanced Research, 2020, 8(10):234-239.

[7] Ansere J A, Han G, Wang H. A Novel Reliable Adaptive Beacon Time Synchronization Algorithm for Large-Scale Vehicular Ad Hoc Networks. IEEE Transactions on Vehicular Technology, 2019, 68(12):11565-11576. https://doi.org/10.21474/IJAR01/11848

[8] Jain N K, Verma A. Relay Node Selection in Wireless Sensor Network Using Fuzzy Inference System. Journal of Communications, 2019, 14(6):423-431.

https://doi.org/10.1109/TVT.2019.2946225

[9] Martin R, Rajasekaran S. Information centric approach to analysing security threats and node behaviour in underwater senor networks. International Journal of Security and Networks, 2018, 13(3):139-152.

[10] Kumar P S. Adaptive Power Control and Load Balancing Algorithm for Cluster Based Wireless Sensor Networks. Sensor Letters, 2019, 17(4):296-305.

[11] Fratta G D, T F ̈uhrer, Gantner G, et al. Adaptive Uzawa algorithm for the Stokes equation. ESAIM Mathematical Modelling and Numerical Analysis, 2019, 53(6):1841-1870.

[12] Ali G, Nithya A. Modelling an Adaptive Cluster Head Positioning Based Map Reducing Strategy for Data Transmission in Medical IoT. International Journal of Engineering and Advanced Technology, 2019, 9(1):78-89. https://doi.org/10.1051/m2an/2019039

[13] Ray N K, Ashok K, Turuk. techniques to enhance the lifetime of mobile ad hoc networks techniques to enhance the lifetime of mobile ad hoc networks doctor of philosophy. EURASIP Journal on Wireless Communications and Networking, 2019, 2012(1):1-17.

[14] Gurgul P, Jopek K, Pingali K, et al. Applications of A Hyper–Graph Grammar System in Adaptive Finite–Element Computations. International Journal of Applied Mathematics and Computer Science, 2018, 28(3):569-582.

[15] Abadi M, Shafiee M S. Distributed Estimation Over an Adaptive Diffusion Network Based on the Family of Affine Projection Algorithms. Signal and Information Processing over Networks, IEEE Transactions on, 2019, 5(2):234-247.

[16] Bressan M, Leucci S, Panconesi A. Motivo: fast motif counting via succinct color coding and adaptive sampling. Proceedings of the VLDB Endowment, 2019, 12(11):1651-1663. https://doi.org/10.1109/TSIPN.2018.2866296

[17] Hanyu A, Kawamoto Y, Nishiyama H, et al. Adaptive Frequency Band and Channel Selection for Simultaneous Receiving and Sending in Multiband Communication. Wireless Communications Letters IEEE, 2019, 8(2):460-463. https://doi.org/10.14778/3342263.3342640