electronics

MDPI

Article

Optimization of Quality of AI Service in 6G Native AI Wireless Networks

Tianjiao Chen [1,2,*], Juan Deng [1,2], Qinqin Tang [3] and Guangyi Liu [1,2]

[1] China Mobile Research Institute, Beijing 100053, China; dengjuan@chinamobile.com (J.D.); liuguangyi@chinamobile.com (G.L.)
[2] ZGC Institute of Ubiquitous-X Innovation and Applications, Beijing 100088, China
[3] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China; qqtang@bupt.edu.cn
[*] Correspondence: chentianjiao@chinamobile.com

Abstract: To comply with the trend of ubiquitous intelligence in 6G, native AI wireless networks are proposed to orchestrate and control communication, computing, data, and AI model resources according to network status, and efficiently provide users with quality-guaranteed AI services. In addition to the quality of communication services, the quality of AI services (QoAISs) includes multiple dimensions, such as AI model accuracy, overhead, and data privacy. This paper proposes a QoAIS optimization method for AI training services in 6G native AI wireless networks. To improve the accuracy and reduce the delay of AI services, we formulate an integer programming problem to obtain proper task scheduling and resource allocation decisions. To quickly obtain decisions that meet the requirements of each dimension of QoAIS, we further transform the single-objective optimization problem into a multi-objective format to facilitate the QoAIS configuration of network protocols. Considering the computational complexity, we propose G-TSRA and NSG-TSRA heuristic algorithms to solve the proposed problems. Finally, the feasibility and performance of QoAIS optimization are verified by simulation.

Keywords: native AI wireless networks; quality of AI service; task scheduling; resource allocation

check for updates

Citation: Chen, T.; Deng, J.; Tang, Q.; Liu, G. Optimization of Quality of AI Service in 6G Native AI Wireless Networks. *Electronics* **2023**, *12*, 3306. https://doi.org/10.3390/electronics12153306

Academic Editor: Dimitris Kanellopoulos

Received: 27 June 2023
Revised: 20 July 2023
Accepted: 28 July 2023
Published: 1 August 2023

1. Introduction

After decades of research and development, communication networks have become critical information infrastructures for economic growth and social progress in today's world [1–3]. In recent years, along with the rapid advancement of artificial intelligence (AI) technology in new communication networks, intelligent applications of the Internet of Everything have been integrated into our lives and continue to drive and deepen a series of application scenarios, such as intelligent vehicle networking, smart industrial networking, smart cities, and smart healthcare [4–7]. The development of intelligent applications brings a great demand for network connection, computing, sharing data, and AI capability, and intelligence permeates every corner of the network, from the end user to the network edge and the remote cloud. However, computing, business data, and AI model resources in 5G are usually in mobile edge computing and cloud computing infrastructure [8–10]. It is difficult for the network to perceive and control the resources of the cloud AI platform in real-time to provide high-quality AI services with strict delay limitations according to changes in the wireless environment and user attributes. Therefore, the 6G network must consider deep integration with AI in the architecture design stage to natively provide AI capabilities. 为了保证高质量的AI服务，需要内生智能。

The native AI design of 6G needs to consider two aspects of requirements: (1) AI can support high-level autonomy of the network. AI can improve the efficiency of data measurements and decision optimization in the network, then realize fast automated operation, maintenance, detection, and network self-healing [11]. (2) AI can support intelligent applications in vertical industries. The 6G network should directly provide vertical industry users with quality-guaranteed AI services to create new market value. According to the above requirements, the 6G native AI wireless network is a unified architecture that deeply integrates communication and AI. It should have the ability to process the AI service logic, manage the full life cycle of the AI service, and provide AI services to the network itself as well as vertical industries [12]. In addition, native AI wireless networks should orchestrate and control the communication, computing, data, and AI model resources in the network, including the core network, radio access network (RAN), and terminals. In collaboration with edge and remote clouds, the native AI wireless networks can quickly adapt to the customization needs of diverse scenarios [13]. Hence, the 6G network will become the fundamental infrastructure for realizing ubiquitous intelligence to support various AI applications, such as real-time AI inference, distributed learning, and intelligent group collaboration.

One essential advantage of natively providing AI services in 6G networks is that resources can be controlled flexibly and on demand to ensure the quality of AI services (QoAISs) [14]. Current networks can already guarantee the quality of service (QoS) for communications. Moreover, 3GPP defines QoS-related standards and sets the communication index dimensions corresponding to the QoS, such as bandwidth, delay, jitter, and bit error rate. RAN protocols (such as service data adaptation protocol) will provide users with differentiated network quality assurance services according to preset QoS parameters. However, the 6G native AI wireless network introduces intelligent capabilities, so in addition to the communication performance, the AI service delay, model performance, data redundancy, overhead, privacy, and other aspects need to be considered [11].

Various studies have investigated how native AI wireless networks can optimize the network itself or provide AI services to third parties [15,16]. For these AI services, the accuracy of AI model training is a critical indicator of the QoAIS. Using high-quality data for training can significantly improve the accuracy of the AI model [11,17]. However, wireless and computing resources are limited, and more data will lead to more transmission and computing delays. Therefore, the QoAIS needs to include at least two indicators: the accuracy of the AI model and the delay of the AI service. To provide better QoAIS services, a reasonable task scheduling and resource allocation scheme should be designed to optimize the QoAIS. One way is to weigh the above two indicators and propose a single-objective optimization problem. However, when the network protocol configures the QoAIS, each of its indicators may have a threshold value. However, the weights in the single-objective optimization problem are fixed in advance, so it is challenging to select the optimization scheme precisely according to the QoAIS.

On the other hand, the AI models required by AI services are specific. For example, target recognition services for autonomous driving requires models such as the region-based convolutional neural network (R-CNN) and you only look once (YOLO). The operation of these models is based on the corresponding AI development framework (e.g., PyTorch, TensorFlow) and will be equipped with related dedicated AI acceleration hardware [18]. Before the AI service is provided, the corresponding environment and hardware need to be pre-configured and installed in the network. Limited by space and cost, it is difficult for a single network node to be equipped with the AI models required by all AI services. Therefore, when designing the task scheduling and resource allocation scheme for AI service, it is necessary to consider the collaboration between network nodes.

To this end, this paper considers a task scheduling and resource allocation scheme for AI training services in 6G native AI wireless networks to optimize the QoAIS, including the accuracy of training AI models and the delay of AI services. According to the wireless channel conditions of the network, the computing resources, and the type of AI model stored by each node, an effective mechanism is needed to select the appropriate data quality, bandwidth allocation, and node to complete the task of the AI service. Because of the conflict between the two indicators of the QoAIS, a single-objective integer programming problem is proposed to optimize the QoAIS. Further, considering the QoAIS configuration of network protocols, we transform this problem into a multi-objective optimization problem. Considering the computational complexity, we use the genetic task scheduling and resource allocation (G-TSRA) algorithm and the non-dominated sorting genetic task scheduling and resource allocation (NSG-TSRA) algorithm to solve the proposed problems. The main contributions of this paper are as follows.

- We propose a 6G native AI wireless network architecture for AI training services, which can reasonably utilize unevenly distributed wireless, computing, and AI model resources to provide AI services. Based on this architecture, the task scheduling and resource allocation schemes of AI training services are studied.
- We formulate the QoAIS optimization problem as a single objective integer programming optimization problem to jointly optimize accuracy and delay. Then a heuristic G-TSRA algorithm is proposed to solve the problem.
- We further propose a multi-objective QoAIS optimization problem to facilitate the QoAIS configuration of network protocols. The NSG-TSRA algorithm is designed to obtain the approximate Pareto-optimal set of AI task scheduling and resource allocation.
- The performance of our proposed G-TSRA and NSG-TSRA is evaluated through extensive simulations. Numerical results validate the effectiveness and superiority of our proposal compared with the benchmark schemes in terms of AI model accuracy and AI service delay.

The remainder of this paper is organized as follows. We first present the related work in Section 2. Then, we describe the model of the native AI wireless network for AI training services in Section 3. The single objective QoAIS optimization problem and G-TSRA are proposed to solve it in Section 4. Further, we present the multi-objective optimization problem and develop the NSG-TSRA in Section 5. Finally, we demonstrate the numerical results in Section 6 and conclude this paper in Section 7.

2. Related Work

Building native AI capabilities in a 6G network can improve operation efficiency, reduce maintenance costs, and enhance user experience. On the other hand, 6G networks can utilize native AI to provide ubiquitous and easily accessible AI services for various industries and users. Driven by such benefits, native AI has recently attracted significant attention from the industry and academia. In [15], Wu et al. proposed the AI-native network slicing architecture, through the synergy of artificial intelligence and network slicing, to promote intelligent network management and support AI services in 6G networks. In [19], Hoydis et al. presented a 6G AI-native air interface designed in part by AI to enable optimized communication schemes for any hardware, radio environments, and applications. In [20], Soldati et al. identified two critical factors for the effective integration and systematization of AI in the future RAN system: the design of AI algorithms must aim to promote the entire RAN environment, and the RAN system must be equipped with an advanced and scalable learning architecture. Due to the current network slicing architecture not being native AI, the heterogeneity of the slicing arrangement is difficult to adapt to the machine learning paradigm. Therefore, Moreira et al. in [21] proposed and evaluated a distributed AI-native slice orchestration architecture that can provide machine learning capabilities in all life cycles of network slices. In [12], the 6G Alliance of Network AI (6GANA) offers the essential technical features needed for the native AI architecture of the 6G network, including the self-generation of use cases, QoAIS, task-oriented scheme, etc. A unified architecture is expected to provide quality-guaranteed AI services for the network and third-party users.

Compared with cloud AI providers, AI services provided by 6G native AI wireless networks have the advantage of guaranteed service quality. In 5G networks, 5QI (5G QoS identifier) is a parameter used to identify different service quality requirements [22]. The value range of 5QI is 1–255, and each value corresponds to a set of preset performance values, including default priority level, packet delay budget, packet error rate, etc. Network operators configure QoS according to user requirements and network resource conditions.

According to the combination of different performance values represented by 5QI, the wireless network protocol provides communication services of different qualities, such as low-latency services, high-reliability services, and high-speed broadband services. For AI services provided by 6G networks, the service quality dimensions will be further expanded, such as the delay of AI services, the accuracy of AI models, communication overhead, computing overhead, data privacy, etc. Therefore, studying the available QoAIS optimization methods for wireless network protocols is necessary.

There have been some studies focusing on the training accuracy of AI models. In [23], Liu et al. proposed an improved particle swarm optimization algorithm (LK-PSO), aimed at the scheduling problem of AI data-intensive computing tasks in the Internet of Things, to effectively improve the scheduling performance of AI data-intensive computing tasks in the edge environment From the perspective of edge intelligence systems, Wang et al. in [24] proposed a deep neural network (DNN) layer-partitioning-based fine-grained cloud–edge collaborative dynamic task scheduling mechanism to greatly reduce the average task response time and deploy more complex DNN models in cloud–edge systems with limited resources.

Based on the above discussion, although there are currently some studies on AI task scheduling, most focus on optimizing model training in resource-constrained networks and only consider AI task delay. Therefore, this work, driven by native AI, proposes an efficient task scheduling and resource allocation scheme for AI training services. Considering the dynamic changes of wireless networks, heterogeneous resources, and data distribution, this work optimizes the accuracy and completion delay of AI training simultaneously and provides QoAIS multi-dimensional indicators that 6G wireless network protocols can use.

3. System Description

The architecture of native AI includes the import of user requirements, the analysis of requirements to QoAISs, the full life cycle management and scheduling of the multiple resources of AI tasks, and the final delivery results. In this paper, we focus on scheduling the multiple resources of AI. In this section, we describe the native AI wireless network model for AI training services, including the communication model, the computation model, and the AI training model.

3.1. System Model

As shown in Figure 1, we consider a 6G native AI wireless network to provide various AI training services. A set of APs (access points) is distributed in an interested area. The APs are connected through wireless channels. Each AP can cover multiple areas, such as roads, parks, factories, etc. Users in these areas will have different service requirements for AI model training, such as pedestrian detection and fire monitoring. Due to limited local resources, users expect the AP to provide AI training services.

Each AP is equipped with hardware to provide communication, computing, data, and AI model resources for AI training tasks, including antennas, computing servers, and AI model caches. After the AP receives the task request, it can obtain the data required for AI training from the user. For each type of AI training, multiple data quality levels can be selected. As the performances of AI training results, such as DNNs, are closely related to the quality of data used for training, APs can request the highest quality data possible according to the remaining resources to obtain better training results. On the other hand, the type of AI training task corresponds to the type of AI model. Due to the resource capacity limitations, the AI model required for an AI task may only be stored in a few APs. When the AP stores the required AI model resources, the AI model training will be completed locally. When an AP does not store the AI model that matches the task, the AP will transmit the data through its antenna to another qualified AP for processing.

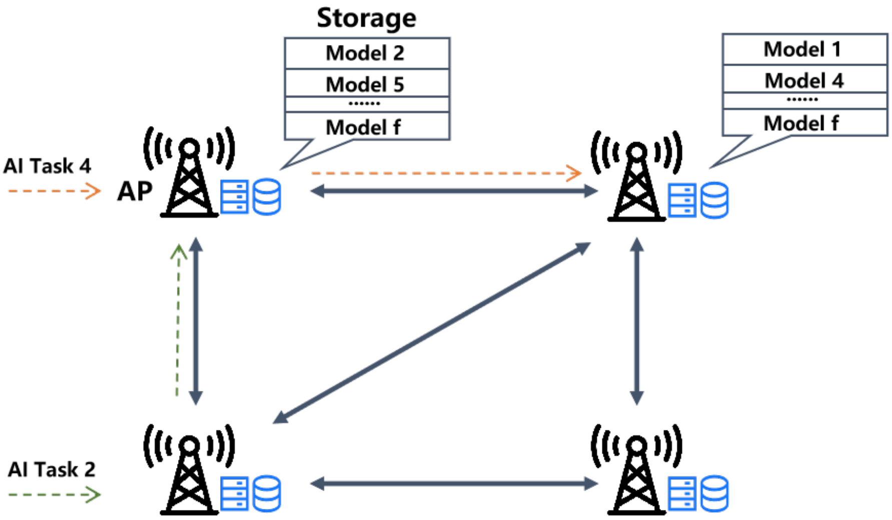

Figure 1. Native AI wireless network architecture for AI training services.

In order to obtain global information and output better decisions, a software-defined network (SDN)-enabled controller is installed at a base station (BS) to centrally make the task scheduling and resource allocation decisions over its coverage area. At each time slot, the AP receives AI service requests containing type information and reports them to the BS along with channel conditions. The BS makes task scheduling and resource allocation decisions based on the collected information, and sends the decisions to the corresponding AP. The AP selects the data quality of users' AI tasks and sends them to the corresponding AP for execution based on the decision.

Denote the set of APs as $\mathcal{N} = \{1, 2, \dots, N\}$. To clearly describe the connection relationship between different APs, we choose i and j as the indices of different APs, $i, j \in \mathcal{N}$. In time slot t, AP i obtains M kinds of AI training tasks. According to the storage of AI model resources, tasks can be processed locally or in the corresponding AP j. For each AP, the task is denoted by $\mathcal{M} = \{1, 2, \dots, M\}$. The set of AI models stored by AP j is denoted by π_j. To ensure the successful execution of the task, AP j must store the AI model required by task m, which is $f_{i,m} \in \pi_j$, where $f_{i,m}$ is the type of task m.

$x_{i,m,j} \in \{0, 1\}$ is a binary decision variable that denotes whether the task m of AP i is transmitted to AP j. Each task can only choose one AP to be processed at time slot t, given by $\sum_{j \in \mathcal{N}} x_{i,m,j}(t) = 1$.

3.2. Communication Model

The task scheduling between AP i and j is facilitated through wireless communications. According to [25], the data transmission rate between AP i and j can be calculated as

$$R_{i,m,j}(t) = W_{i,m} \log_2 \big(1 + \mathrm{SNR}_{i,j}(t)\big) \tag{1}$$

where $W_{i,m}$ is the bandwidth of AP i allocated to task m. $\mathrm{SNR}_{i,j}(t)$ is the signal-to-noise ratio (SNR) between the two APs.

$$\mathrm{SNR}_{i,j}(t) = \frac{p_i h_{i,j}(t)}{\sigma^2} \tag{2}$$

where p_i is the transmission power of each link, $h_{i,j}(t)$is the channel gain, and σ^2 is the noise power.

Each type of task has different qualities b, denoted by $\mathcal{B} = \{1, 2, \dots, B\}$. The data size of task m in AP i is $Z_{i,m}(t) = a_{f_{i,m}} b_{i,m}(t)$, where $a_{f_{i,m}}$ is the amount of data per unit level related to the task type. Hence, the transmission time is given by

$$T^{c}_{i,m,j}(t) = \frac{Z_{i,m}(t)}{R_{i,m,j}(t)} \tag{3}$$

3.3. Computation Model

After receiving tasks from other APs, AP j will allocate computing resources to each task according to their requested CPU cycle $C_{i,m}(t)$:

$$C_{i,m}(t) = c_{f_{i,m}} Z_{i,m}(t)\tau_{i,m}(t) \tag{4}$$

where $c_{f_{i,m}}$ is the CPU cycle for each bit of the data, and $\tau_{i,m}(t)$ denotes the number of training iterations determined by AP i at time slot t.

The total computing resource of each AP j is Φ_j. Hence, the computing resource $\Phi_{j,i,m}(t)$ allocated to task m of AP i is given by

$$\Phi_{j,i,m}(t) = \frac{\Phi_j C_{i,m}(t)}{\sum_{i,m} C_{i,m}(t)x_{n,f,j}(t)} \tag{5}$$

Consequently, the computing delay is calculated as

$$T^{p}_{i,m,j}(t) = \frac{C_{i,m}(t)}{\Phi_{j,i,m}(t)} = \frac{\sum_{i,m} C_{i,m}(t)x_{n,f,j}(t)}{\Phi_j} \tag{6}$$

3.4. AI Training Model

After allocating computing resources, the AP will use task data for AI training, and output the trained AI model. The training of an AI model is the training of a large number of data samples. In typical AI training, for a data sample $\{\boldsymbol{x}_n, y_n\}$ with a multi-dimensional input feature $\boldsymbol{x}_n$, the goal is to find a model parameter vector $\boldsymbol{\omega}$ that represents the labeled output y_n with a loss function $loss_n(\boldsymbol{\omega})$. The loss function of a local dataset with a number of D data samples can be defined as

$$Loss(\boldsymbol{\omega}) = \frac{1}{D}\sum_{n\in D} loss_n(\boldsymbol{\omega}) + \xi g(\boldsymbol{\omega}) \tag{7}$$

where $g(\cdot)$ is a regularizer function and can be given as $g(\cdot) = \frac{1}{2}\|\cdot\|^2; \forall \xi \in [0,1]$.

Denote $\boldsymbol{\omega}^*_i$ as the optimal model parameter for AP i. AP i trains its local AI model in an iterative manner [26]:

$$\boldsymbol{\omega}^*_i = \arg\min_{\omega} Loss_i(\boldsymbol{\omega}|\boldsymbol{\omega}_i, \nabla Loss_i(\boldsymbol{\omega})) \tag{8}$$

The performance of an AI model can be evaluated using the accuracy of the model, denoted by $\varphi \in [0,1]$. The accuracy of AI models is related to the allocated computing resources, the quality/size of data, the number of training iterations, the learning rate, the algorithm used for training, and so on. Similar to [11], the accuracy of AI model m of AP i processed by AP j satisfies

$$\varphi_{i,m,j}(t) = 1 - \exp\left(-\varsigma^{\mathrm{lc}}\Phi_{j,i,m}(t)(Z_{i,m}(t)\tau_{i,m}(t)^{\alpha})^{v}\right) \tag{9}$$

where ς^{lc} and v are weight factors. $\Phi_{j,i,m}(t)$ is the allocated computing resource to each task. α is the learning factor that reflects the marginal revenue of iterations and depends on the selected learning algorithm.

To assess the quality of the solution, the local φ accuracy satisfies

$$\|Loss(\boldsymbol{\omega}^*)\| \leq (1-\varphi)\|Loss(\mathbf{0})\|. \tag{10}$$

Here, the implementation of $\varphi = 1$ needs to find the exact maximum, while $\varphi = 0$ means that no improvement is achieved in AP.

4. Single Objective QoAIS Optimization

4.1. Problem Formulation

We formulate the AI task scheduling and wireless resource allocation problem in the 6G native AI wireless network. The objective is to optimize the quality of AI training services, including delay and accuracy. The BS maintains resource information for all APs, including the available bandwidth, computing power, and the types of AI models. At each time slot, the BS makes task scheduling and resource allocation decisions and sends the decisions to APs.

To maximize the total accuracy of the trained AI models and minimize the total delay of AI training tasks simultaneously, the optimization problem can be transformed into a single-objective problem by assigning different weights to each objective.

$$\text{Minimize:} \sum_{i\in\mathcal{N}} \sum_{m\in\mathcal{M}} \sum_{j\in\mathcal{N}} (\alpha * (T^{c}_{i,m,j} + T^{p}_{i,m,j}) + 1 - \varphi_{i,m,j}) \tag{11}$$

$$\text{Subject to:} \quad x_{i,m,j} \in \{0,1\} \tag{12}$$

$$\sum_{j\in\mathcal{N}} x_{i,m,j}(t) = 1, \forall i \in \mathcal{N}, m \in \mathcal{M} \tag{13}$$

$$\sum_{m\in\mathcal{M}} W_{i,m} \leqslant W_i, \forall i \in \mathcal{N} \tag{14}$$

$$1 \leqslant b_{i,m} \leqslant B \tag{15}$$

$$f_{i,m} x_{i,m,j} \in \pi_j \tag{16}$$

where α is a weight parameter to balance the trade-off between the delay and accuracy. Constraint (13) indicates that one task can only be transmitted to one AP; (14) is the bandwidth constraint, and (15) is the data quality constraint. Moreover, (16) indicates that the AP must cache the corresponding AI model when processing tasks.

4.2. Genetic Task Scheduling and Resource Allocation Algorithm

A genetic algorithm [27] is a heuristic algorithm based on natural selection and natural genetics that can find an optimal solution in a limited time. Therefore, we propose the genetic task scheduling and resource allocation algorithm (G-TSRA) to solve the single objective QoAIS optimization problem.

In the proposed problem, each solution is encoded as an individual, including AP index, data quality, and bandwidth allocation for each task. The individual is given by

$$o = (\boldsymbol{x}_{i,m,j}, \boldsymbol{b}_{i,m}, \boldsymbol{W}_{i,m}), \forall i, j \in \mathcal{N}, m \in \mathcal{M} \tag{17}$$

The quality of each individual is evaluated by the function (11), which represents the degree of fitness to the environment. Multiple individuals form a population and evolve according to the principle of "survival of the fittest".

During evolution, the size of the population remains constant. Individuals are selected according to their quality, then crossover and mutation operations are performed to form a new population. Through continuous iterations, the optimized solution is finally obtained.

5. Multi-Objective QoAIS Optimization

5.1. Problem Formulation

In the 6G native AI wireless network, to guarantee the QoAIS, each type of indicator needs to meet its requirements. However, in the single objective QoAIS optimization problem, the weight value is pre-configured, and it can only be judged whether each indicator meets the requirements after the algorithm is executed. Multiple tunings will be required, and the above operations must be repeated when the optimization requirements change.

To solve this challenge, using a multi-objective evolutionary algorithm to obtain the Pareto-optimal solution set is an effective method. When receiving QoAIS requirements, it can directly select a solution that meets the requirements according to the Pareto-optimal solution set. Even when QoAIS requirements change, the system does not need to re-run the algorithm.

In the following, we formulate the task scheduling and resource allocation problem as a multi-objective integer programming problem.

$$\text{Maximize: } \sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}}\sum_{j\in\mathcal{N}}\varphi_{i,m,j} \tag{18}$$

$$\text{Minimize: } \sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}}\sum_{j\in\mathcal{N}}(T^{c}_{i,m,j}+T^{p}_{i,m,j}) \tag{19}$$

$$\text{Subject to:}(12)-(16) \tag{20}$$

5.2. Non-Dominated Sorting Genetic Task Scheduling and Resource Allocation Algorithm

To solve the proposed multi-objective optimization problem of the 6G native AI wireless network with relatively low computational complexity, we designed an NSG-TSRA algorithm based on the idea of the non-dominated sorting genetic algorithm II (NSGA-II) [28]. Before presenting the details of the NSG-TSRA algorithm, we first introduce two key approaches: a fast non-dominated sorting approach and a crowding-comparison approach.

5.2.1. Pareto-Optimal Solution

Suppose a multi-objective optimization problem is as follows: $\max_x(f_1(x), f_2(x), \cdots, f_K(x))$ where K is the number of objective functions.

Definition 1. *(Pareto dominance): Solution x_1. The Pareto dominates solution x_2, i.e., $x_1 \preceq x_2$, if and only if $f_k(x_1) \leq f_k(x_2)$ for $\forall k \in \{1,\ldots,K\}$ and $\exists q \in \{1,\ldots,K\}$, satisfying $f_q(x_1) < f_q(x_2)$.*

Definition 2. *(Pareto-optimal Set): The Pareto-optimal set can be defined as $P^* = \{x^* \in \Omega \mid \exists x \in \Omega, x \prec x^*\}$, where x^* is Pareto optimality.*

Multiple optimization goals are often in conflict with each other. Therefore, a multi-objective optimization algorithm will involve a collection of optimal solutions. Hence, without additional conditions, there is no significant difference between the solutions in the set.

5.2.2. Fast Non-Dominated Sorting Approach

In the NSG-TSRA algorithm, fast non-dominated sorting involves dividing the population $O = \{1, 2, \ldots, o\}$ into several layers according to the dominance relationship. The function of this approach is to guide the search toward the Pareto-optimal solution set. Each individual o is a solution, which consists of an AP index, data quality, and bandwidth allocation for each task.

$$o = (\boldsymbol{x}_{i,m,j}, \boldsymbol{b}_{i,m}, \boldsymbol{W}_{i,m}), \forall i,j \in \mathcal{N}, m \in \mathcal{M} \tag{21}$$

To facilitate uniform optimization, the objective of maximizing model accuracy needs to be translated into minimizing the relative model performance. Since $0 \leqslant \varphi_{i,m,j} \leqslant 1$, we have

$$Obj_1 = \sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}}\sum_{j\in\mathcal{N}}(1-\varphi_{i,m,j}) \tag{22}$$

$$Obj_2 = \sum_{i\in\mathcal{N}}\sum_{m\in\mathcal{M}}\sum_{j\in\mathcal{N}}(T^{c}_{i,m,j}+T^{p}_{i,m,j}) \tag{23}$$

When an individual does not satisfy the constraints, a penalty value can be added to the objective function. The first layer is the set of non-dominated individuals in the population; the second layer is the set of non-dominated individuals obtained after removing individuals in the first layer, and so on. As shown in Figure 2, 12 individuals will be assigned to the corresponding fronts and individuals with the same color are at the same front. The details are shown in Algorithm 1. Each individual has two kinds of parameters: domination count κ_o and domination set S_o. κ_o represents the number of individuals that dominate individual o, and S_o represents the individual set dominated by individual o. First, the algorithm calculates κ_o and S_o for each individual according to the dominance relationship and obtains the first non-dominated front F_1. Specifically, if o dominates l, l will be added to the set of solutions dominated by o. Otherwise, the domination counter of o increases and o belongs to the first front. Then, for each individual in F_1, the domination counters of its dominant solutions are subtracted by one. If the domination counter is 0, this domination will be added to F_2. After multiple iterations, the individuals in S_o are iteratively divided into different layers according to their rank. Eventually, sorted layers $(F_1, F_2, ...)$ are obtained. The total complexity of finding all members of the different non-dominated levels in the population is $\mathcal{O}(K(2O)^2)$, where K is the number of objectives. Hence, the worst-case complexity of fast non-dominated sorting is $\mathcal{O}(K(2O)^2)$.

Algorithm 1 Fast non-dominated sorting algorithm.

```
Input: Population $O$
for each $o$ in $O$ do
    Set $S_o = \phi$
    Set $\kappa_o = 0$
    for each $l$ in $O$ do
        if $o$ dominates $l$ then
            $S_o = S_o \cup \{l\}$
        else if $l$ dominates $o$ then
            $\kappa_o = \kappa_o + 1$
        end if
    end for
    if $\kappa_o = 0$ then
        $rank_o = 1$
        $F_1 = F_1 \cup \{o\}$
    end if
end for
Set $s = 1$
while $F_s \neq \emptyset$ do
    $Q = \phi$
    for each $o$ in $F_s$ do
        for each $l$ in $S_o$ do
            $\kappa_l = \kappa_l - 1$
            if $\kappa_l = 0$ then
                $rank_l = s + 1$
                $Q = Q \cup \{l\}$
            end if
        end for
    end for
    $s = s + 1, F_s = Q$
end while
Output: Sorted layers $(F_1, F_2, ...)$
```

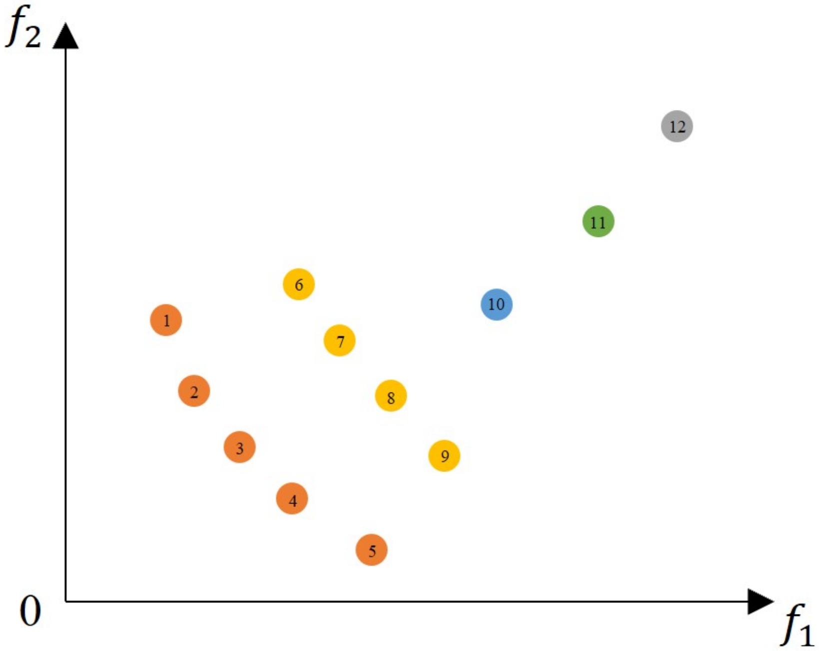

Figure 2. Fast non-dominated sorting approach.

5.2.3. Crowding-Comparison Approach

In the NSG-TSRA algorithm, the crowding-comparison approach is adopted to maintain the diversity of the population, which can reduce the time complexity compared to the sharing function approach in NSGA. This approach consists of the density estimation and the crowding-comparison operator.

Density estimation: The crowding distance estimates the density of an individual surrounded by others in the population. First, the non-dominated individuals of each layer are arranged according to the value of each objective function in ascending order. Then, the crowding degree of individual o can be quantified as the difference value of two adjacent individuals, Obj (i.e., $o+1$ and $o-1$), in the same layer. The crowding distance $\mathcal{I}$ of each individual is the sum of the crowding degrees under each objective function. Moreover, the distance values need to be normalized before summing. $Obj_k^{\max}$ and $Obj_k^{\min}$ are the maximum and minimum function values in this population.

$$\mathcal{I}_k(o) = \frac{Obj_k(o+1) - Obj_k(o-1)}{Obj_k^{\max} - Obj_k^{\min}}, k = 1, 2. \tag{24}$$

Crowded-comparison operator: After completing the fast non-dominated sorting and density estimation, we obtain the non-dominated rank $rank_o$ and the crowding distance $\mathcal{I}(o)$. To achieve a wider distribution of the Pareto-optimal set, the crowding-comparison operator is used to select individuals according to two conditions, as follows. Let $\prec^*$ denote an order of comparison. The details of the crowding-comparison approach are shown in Algorithm 2.

Condition 1: A smaller rank means that the individual is closer to the Pareto front. Therefore, individuals with lower ranks will be selected.

$$o \prec^* l, \text{ if } (rank_o < rank_l). \tag{25}$$

Condition 2: When two individuals have the same rank, a larger crowding distance means that the individual is more dispersed from other individuals. Therefore, individuals with larger crowding distances will be selected.

$$o \prec^* l, \text{ if } ((rank_o = rank_l) \text{ and } (\mathcal{I}(o) > \mathcal{I}(l))). \tag{26}$$

The worst-case complexity of the crowding-distance assignment is $\mathcal{O}(K(2O)\log(2O))$.

Algorithm 2 Non-dominated sorting genetic task scheduling and resource allocation algorithm.

```
Input: Task $m$ of each AP, model storage $\pi_j$, channel gain $h_{i,j}$, computing resource $\Phi_j$
Initialize the parameters
Generate an initial population by random means
Obtain $R(t) = O(t) \cup Q(t)$
Rank the population $R(t)$ by the fast non-dominated sorting approach
$O(t+1) = \emptyset$ and $s = 1$
while $|O(t+1)| + |F_s| < O$ do
    *******Crowding Distance Assignment******
    Set $Num = |F_s|$
    for each $o \in F_s$ do
        Set $\mathcal{I}(o) = 0$
    end for
    for each $k$ do
        Sort $F_s = sort(F_s, k)$
        Set $\mathcal{I}(1) = \mathcal{I}(Num) = \infty$
        for $i$ = 2 to $Num - 1$ do
            $\mathcal{I}(i) = \mathcal{I}(i) + \mathcal{I}_k(i)$
        end for
    end for
    $O(t+1) = O(t+1) \cup F_s$
    $s = s + 1$
end while
Sort($F_s \prec^*$)
Set $O(t+1) = O(t+1) \cup F_s[1 : (O - |O(t+1)|)]$
Obtain $Q(t+1)$ = make new pop($O(t+1)$)
Set $t = t + 1$
Output: Pareto-optimal set
```

5.2.4. Algorithm Design

The NSG-TSRA algorithm adopts the ideas of elitism and tournament selection. The detailed procedure of the NSG-TSRA algorithm is shown in Algorithm 2.

When the algorithm is executed, the initial population $O(t)_{t=0}$ is randomly generated and sorted by the fast non-dominated sorting approach. The elitism strategy involves retaining the best individuals in the current population to the next generation population without additional genetic operations. To implement this strategy, an offspring population $Q(t)_{t=0}$ is created by selection, crossover, mutation, and other operations. Then, $O(t)$ and $Q(t)$ are combined to generate the expanded population, $R(t)$. $R(t)$ is sorted by the fast non-dominated sorting approach shown in Algorithm 1 and the divided layers $(F_1, F_2, ...)$ are obtained.

After the division, individuals will be sequentially selected, starting from the first layer F_1 until the entire population $O(t+1)$ is filled. However, the size of $R(t)$ is 2 times that of $O(t+1)$. Assume that it is not possible to put all the individuals of the vth layer F_v into $O(t+1)$ during the filling process. The crowding-comparison approach is used to sort F_v. Individuals will be sequentially added to the next population $O(t+1)$ according to the crowding distance until the number of individuals in the population reaches O. The remaining solutions are deleted. The worst-case complexity of sorting on $\prec^*$ is $\mathcal{O}(2O\log(2O))$.

Finally, for $O(t+1)$, the tournament selection and crossover and mutation operations are used to create the new population, $Q(t+1)$. Here, the tournament selection operation is according to the crowded-comparison operator. Through continuous iteration, the algorithm finally outputs the approximate solution of the Pareto-optimal set. Considering the time complexity of fast non-dominated sorting, crowding comparison, and sorting on $\prec^*$, the overall complexity of the NSG-TSRA algorithm is $\mathcal{O}(E \cdot K(2O)^2)$, where E is the number of iterations. After the approximate set is obtained, the solution can be flexibly selected according to the need.

6. Numerical Results and Discussion

In this section, we simulate the performance of the proposed single and multi-objective QoAIS optimization scheme for AI training services in the 6G native AI wireless network. Specifically, the number of APs is 5, and each AP can accept 2 different types of AI tasks. There are three quality data levels, and the corresponding data size is $[1,2,3]$ Gbit. The computing capacity of each AP is randomly distributed in $[0.5,3]$ Gcycle/s. Each AP cache has two or three types of AI models, and the set of AI models in all AP caches meets the needs of all tasks. The SNR between two APs is randomly distributed in $[20,40]$ dB. The bandwidth of each AP is 200 MHz and the number of sub-channels is 4.

For the parameter settings of the G-TSRA and NSG-TSRA algorithms, the population size is 100. The number of iterations is 1000 for G-TSRA and 200 for NSG-TSRA. The number of genes in each individual and the value range of each gene are set according to the above network parameters. The crossover probability parameter is 2. The probability of mutation is 0.1 for G-TSRA and 0.08 for NSG-TSRA.

Figure 3 shows the performance of G-TSRA as the α weight changes. α is randomly distributed in $[0.01,1]$. The algorithm converges around 250 iterations, and the delay and accuracy performances are obtained. The delay gradually increases as the weights decrease while the relative model performance decreases. This is because the weight belongs to the delay, so the weight reduction means that the delay's importance is gradually reduced compared to the relative model performance. The algorithm will tend to optimize the relative model performance. The figure shows that the performance changes drastically when the weight value drops from 0.08 to 0.07, while the change between 1 and 0.6 is relatively stable. Therefore, the performance does not change continuously and smoothly with the weight value, so it is impossible to determine the required QoAIS by presetting the weight value before the algorithm is executed.

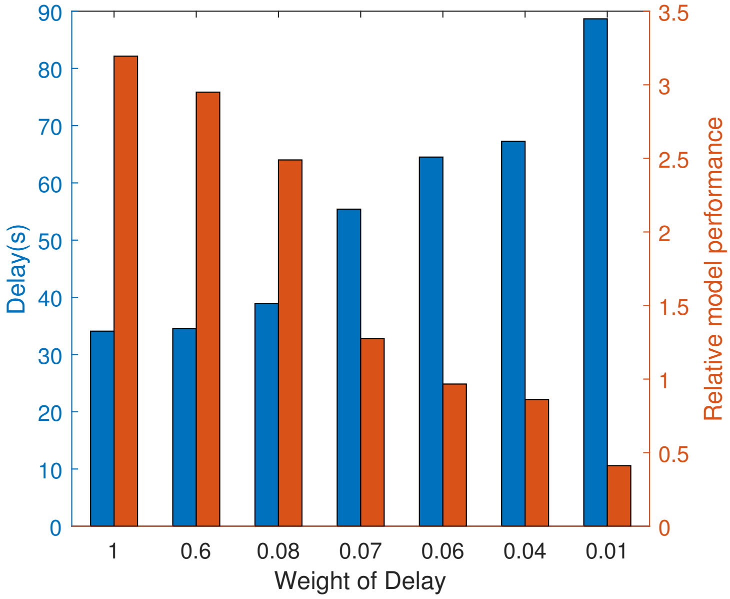

Figure 3. Impact of the weight on the G-TSRA performance.

Figure 4 shows the convergence of the proposed NSG-TSRA algorithm. Since the proposed optimization problem is multi-objective, the output of each iteration is a set of solutions. Thus, the convergence trend is shown by calculating the average delay and the average relative model performance of the population, but each value is the sum of the delay and relative model performance of the 10 tasks. At the initial population, both latency and relative model performance are high. As the number of iterations increases, the performances of the two optimization objectives gradually decrease in the fluctuations. Due to exploration, the latency is minimized at 110 iterations at the cost of training accuracy. Finally, the performance reaches convergence at 190 iterations.

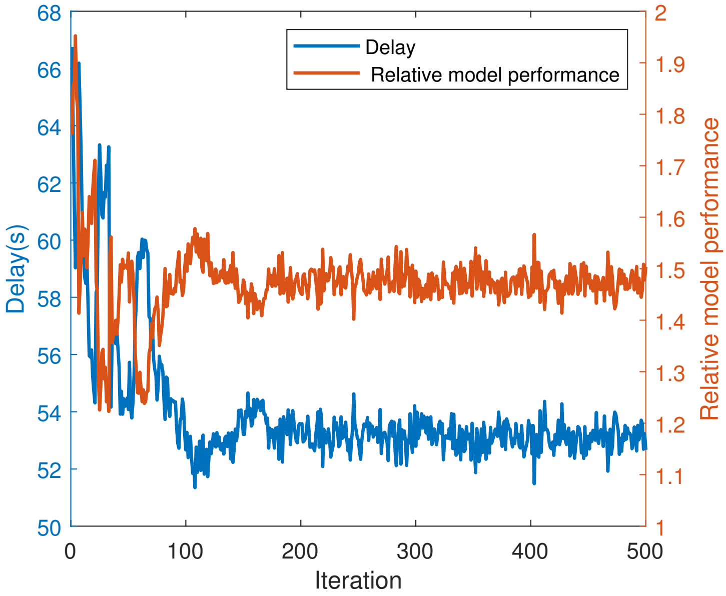

Figure 4. The convergence performances of different objectives.

In Figure 5, we investigate the performances of G-TSRA, NSG-TSRA, the multi-objective evolutionary algorithm based on decomposition (MOEA/D) [29], and the greedy-based scheme. The greedy-based scheme selects the training node for the AI task based on the product of the computing performance of the node and the channel conditions from this node to all other nodes. Moreover, the maximum selection numbers are set for each training node, which can prevent a decline in performance due to the large number of tasks selected for the same node. Figure 5 shows that the performance of the Pareto solution of the NSG-TSRA-based scheme is better than that of the greedy-based scheme. The performances of G-TSRA and MOEA/D are slightly worse compared to NSG-TSRA.

In Figure 6, we investigate the performances under different data sizes, which are $[0.5, 2.5]$, $[1, 3]$, and $[1.5, 3.5]$ Gbit. The delay in task completion increases with the data size, but the accuracy of the trained model also increases. Since the value ranges of the data sizes partially overlap, the data sizes are similar under some specific data quality selections. Therefore, the different curves will partially overlap.

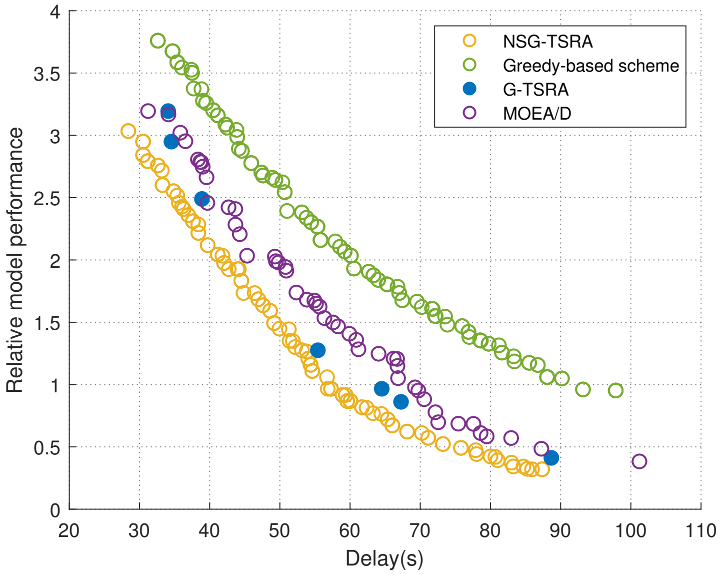

Figure 5. The performances of different algorithms.

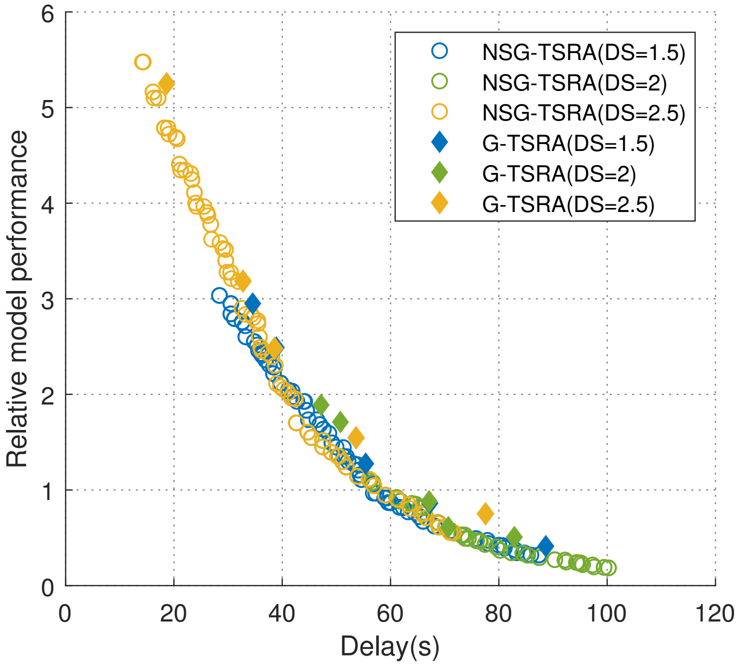

Figure 6. The performances under different data sizes.

In Figure 7, we investigate the performances under different bandwidths, which are 150, 200, and 300 MHz. With the same relative model performance, the task completion time decreases as the bandwidth of the AP increases. Since the bandwidth mainly affects the transmission rate, even if the transmission delay has an impact on the selection of APs, the impact on the performance of the final relative model is still limited. Under the parameter settings of different bandwidths, there is no overlapping part of the Pareto solution sets.

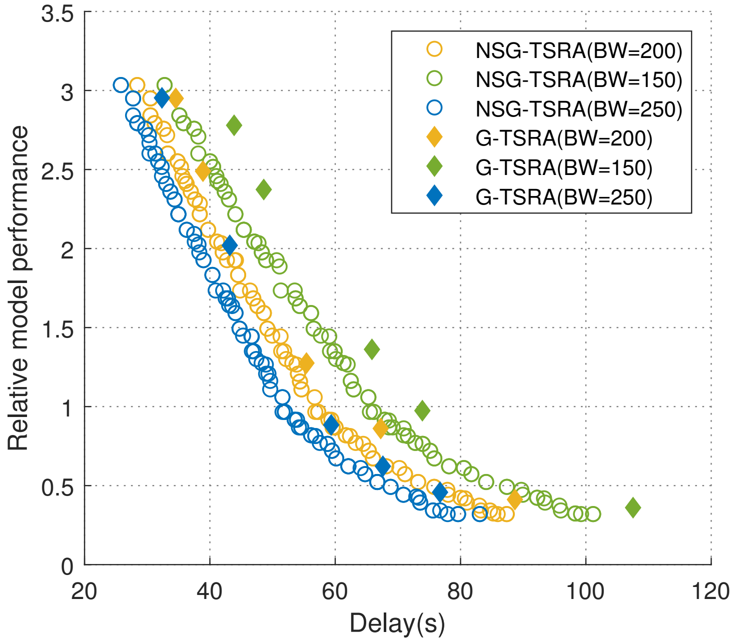

Figure 7. The performances under different bandwidths.

In Figure 8, we investigate the performances under different numbers of APs, which are 4, 5, and 6. The computing resources and channel conditions of APs are generated randomly, with the average AP computing resources and channel gain gradually decreasing. With the same relative model performance, the task completion time increases with the number of APs. The analysis is as follows: as the number of APs increases, the resources in the network increase. However, the number of requests also increases, and the decline in average resources caused by random generation in the simulation leads to a decrease in overall performance.

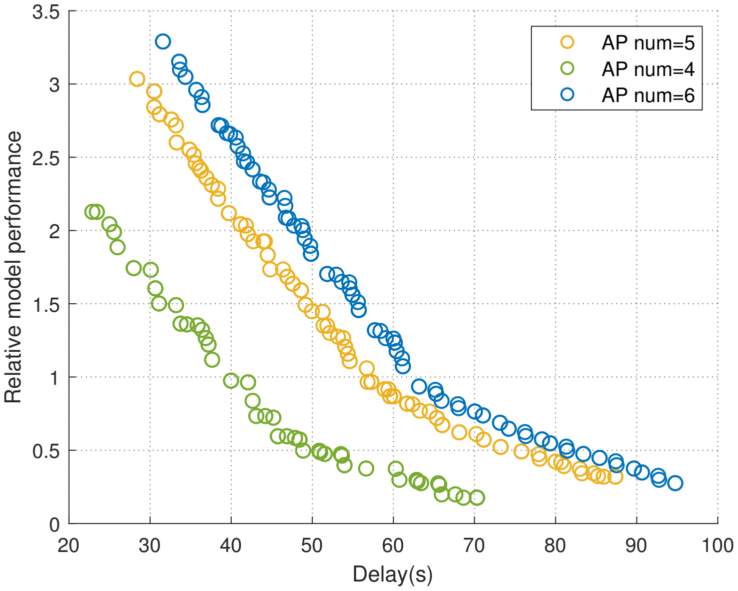

Figure 8. The performances under different AP numbers.

Based on the above analysis, the multi-objective QoAIS optimization scheme performs better than the single-objective optimization scheme. It can output an unbiased solution set that is more suitable for the QoAIS guarantee in 6G native AI wireless networks.

7. Conclusions

This paper proposes a QoAIS optimization method for AI training services in 6G native AI wireless networks. To improve the accuracy of AI models and reduce task latency, we formulated a single-objective integer programming problem to obtain reasonable task scheduling and resource allocation decisions. Further, to better meet the requirements of various indicators of the QoAIS, we transformed the problem into a multi-objective format, facilitating the configuration of network protocols. We proposed G-TSRA and NSG-TSRA heuristic algorithms to solve the above problems, and performed simulations to demonstrate the feasibility of multi-dimensional QoAIS optimization.

In the future, we will consider more QoAIS dimensions, such as the privacy and security of AI services, to achieve comprehensive QoAIS optimization. In addition, the overhead of AI model scheduling and decision-making are key factors affecting system performance and will be considered in future work.

Author Contributions: Conceptualization, T.C. and J.D.; data curation, T.C.; formal analysis, T.C.; methodology, T.C.; software, T.C.; supervision, G.L.; validation, T.C. and Q.T.; visualization, T.C.; writing—original draft, T.C.; writing—review and editing, J.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Key R&D Program of China (Project Number: 2022YFB2902100).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: Part of this work has been accepted at IEEE ICC 2023 in the paper "Efficient Task Scheduling and Resource Allocation for AI Training Services in Native AI Wireless Networks".

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

References

1. Jiang, W.; Han, B.; Habibi, M.A.; Schotten, H.D. The road towards 6G: A comprehensive survey. *IEEE Open J. Commun. Soc.* **2021**, *2*, 334–366. [CrossRef]
2. Nguyen, D.C.; Ding, M.; Pathirana, P.N.; Seneviratne, A.; Li, J.; Niyato, D.; Dobre, O.; Poor, H.V. 6G Internet of Things: A comprehensive survey. *IEEE Internet Things J.* **2021**, *9*, 359–383. [CrossRef]
3. Guo, F.; Yu, F.R.; Zhang, H.; Li, X.; Ji, H.; Leung, V.C. Enabling massive IoT toward 6G: A comprehensive survey. *IEEE Internet Things J.* **2021**, *8*, 11891–11915. [CrossRef]
4. Zhang, C.; Lu, Y. Study on artificial intelligence: The state of the art and future prospects. *J. Ind. Inf. Integr.* **2021**, *23*, 100224. [CrossRef]
5. Peres, R.S.; Jia, X.; Lee, J.; Sun, K.; Colombo, A.W.; Barata, J. Industrial artificial intelligence in industry 4.0-systematic review, challenges and outlook. *IEEE Access* **2020**, *8*, 220121–220139. [CrossRef]
6. Tong, W.; Hussain, A.; Bo, W.X.; Maharjan, S. Artificial intelligence for vehicle-to-everything: A survey. *IEEE Access* **2019**, *7*, 10823–10843. [CrossRef]
7. Secinaro, S.; Calandra, D.; Secinaro, A.; Muthurangu, V.; Biancone, P. The role of artificial intelligence in healthcare: A structured literature review. *BMC Med. Inform. Decis. Mak.* **2021**, *21*, 125. [CrossRef]
8. Ham, D.; Kwak, J. Survey on 6G System for AI-Native Services. In Proceedings of the 2022 13th International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Republic of Korea, 19–21 October 2022; pp. 1520–1522. [CrossRef]
9. Tang, Q.; Xie, R.; Yu, F.R.; Chen, T.; Zhang, R.; Huang, T.; Liu, Y. Distributed Task Scheduling in Serverless Edge Computing Networks for the Internet of Things: A Learning Approach. *IEEE Internet Things J.* **2022**, *9*, 19634–19648. [CrossRef]
10. Tang, Q.; Xie, R.; Yu, F.R.; Huang, T.; Liu, Y. Decentralized computation offloading in IoT fog computing system with energy harvesting: A dec-POMDP approach. *IEEE Internet Things J.* **2020**, *7*, 4898–4911. [CrossRef]
11. Tang, Q.; Xie, R.; Yu, F.R.; Chen, T.; Zhang, R.; Huang, T.; Liu, Y. Collective Deep Reinforcement Learning for Intelligence Sharing in the Internet of Intelligence-Empowered Edge Computing . *IEEE Trans. Mob. Comput.* **2022**, 1–16. [CrossRef]
12. 6GANA. Ten Questions of 6G Native AI Network Architecture. *6GANA White Paper*, 24 March 2023.
13. Letaief, K.B.; Shi, Y.; Lu, J.; Lu, J. Edge Artificial Intelligence for 6G: Vision, Enabling Technologies, and Applications. *IEEE J. Sel. Areas Commun.* **2022**, *40*, 5–36. [CrossRef]

14. Liu, G.; Deng, J.; Zheng, Q.; Li, G.; Sun, X.; Huang, Y. Native intelligence for 6G mobile network: Technical challenges, architecture and key features. *J. China Univ. Posts Telecommun.* **2022**, *29*, 27–40.
15. Wu, W.; Zhou, C.; Li, M.; Wu, H.; Zhou, H.; Zhang, N.; Shen, X.S.; Zhuang, W. AI-Native Network Slicing for 6G Networks. *IEEE Wirel. Commun.* **2022**, *29*, 96–103. [CrossRef]
16. Tang, Q.; Yu, F.R.; Xie, R.; Boukerche, A.; Huang, T.; Liu, Y. Internet of Intelligence: A Survey on the Enabling Technologies, Applications, and Challenges. *IEEE Commun. Surv. Tutor.* **2022**, *24*, 1394–1434. [CrossRef]
17. Rudol, P.; Doherty, P. *Evaluation of Human Body Detection Using Deep Neural Networks with Highly Compressed Videos for UAV Search and Rescue Missions*; Springer: Cham, Switzerland, 2019.
18. Capra, M.; Bussolino, B.; Marchisio, A.; Masera, G.; Martina, M.; Shafique, M. Hardware and Software Optimizations for Accelerating Deep Neural Networks: Survey of Current Trends, Challenges, and the Road Ahead. *IEEE Access* **2020**, *8*, 225134–225180. [CrossRef]
19. Hoydis, J.; Aoudia, F.A.; Valcarce, A.; Viswanathan, H. Toward a 6G AI-Native Air Interface. *IEEE Commun. Mag.* **2021**, *59*, 76–81. [CrossRef]
20. Soldati, P.; Ghadimi, E.; Demirel, B.; Wang, Y.; Sintorn, M.; Gaigalas, R. Approaching AI-native RANs through generalization and scalability of learning. *Ericsson Technol. Rev.* **2023**, *2023*, 2–12. [CrossRef]
21. Moreira, R.; Martins, J.S.; Carvalho, T.C.; Silva, F.D.O. On Enhancing Network Slicing Life-Cycle Through an AI-Native Orchestration Architecture. In Proceedings of the 37th International Conference on Advanced Information Networking and Applications (AINA-2023), Juiz de Fora, Brazil, 29–31 March 2023; Springer: Cham, Switzerland, 2023; Volume 2, pp. 124–136.
22. Joda, R.; Elsayed, M.; Abou-Zeid, H.; Atawia, R.; Sediq, A.B.; Boudreau, G.; Erol-Kantarci, M. QoS-Aware Joint Component Carrier Selection and Resource Allocation for Carrier Aggregation in 5G. In Proceedings of the ICC 2021—IEEE International Conference on Communications, Xiamen, China, 28–30 July 2021; pp. 1–6. [CrossRef]
23. Liu, L.; Wang, H.; Liu, Y.; Zhang, M. Task Scheduling Model of Edge Computing for AI Flow Computing in Internet of Things. In Proceedings of the 2022 Global Conference on Robotics, Artificial Intelligence and Information Technology (GCRAIT), Chicago, IL, USA, 30–31 July 2022; pp. 256–260. [CrossRef]
24. Wang, X.; Li, X.; Wang, N.; Qin, X. Fine-grained Cloud Edge Collaborative Dynamic Task Scheduling Based on DNN Layer-Partitioning. In Proceedings of the 2022 18th International Conference on Mobility, Sensing and Networking (MSN), Guangzhou, China, 14–16 December 2022; pp. 155–162. [CrossRef]
25. Liu, Y.; Yu, H.; Xie, S.; Zhang, Y. Deep Reinforcement Learning for Offloading and Resource Allocation in Vehicle Edge Computing and Networks. *IEEE Trans. Veh. Technol.* **2019**, *68*, 11158–11168. [CrossRef]
26. Yang, Z.; Chen, M.; Saad, W.; Hong, C.S.; Shikh-Bahaei, M. Energy Efficient Federated Learning Over Wireless Communication Networks. *IEEE Trans. Wirel. Commun.* **2021**, *20*, 1935–1949. [CrossRef]
27. Zhou, R.; Liu, Y.; Zhang, K.; Yang, O. Genetic Algorithm-Based Challenging Scenarios Generation for Autonomous Vehicle Testing. *IEEE J. Radio Freq. Identif.* **2022**, *6*, 928–933. [CrossRef]
28. Zhu, J.; Wang, X.; Huang, H.; Cheng, S.; Wu, M. A NSGA-II Algorithm for Task Scheduling in UAV-Enabled MEC System. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 9414–9429. [CrossRef]
29. Ma, G.; Li, J.; Zhang, X.P. Energy Storage Capacity Optimization for Improving the Autonomy of Grid-Connected Microgrid. *IEEE Trans. Smart Grid* **2023**, *14*, 2921–2933. [CrossRef]

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.