Probabilistic-Assured Resource Provisioning with Customizable Hybrid Isolation for Vertical Industrial Slicing

Qize Guo, Rentao Gu, *Senior Member, IEEE,* Hao Yu, Tarik Taleb, *Senior Member, IEEE,* and Yuefeng Ji, *Senior Member, IEEE*

***Abstract*—With the increasing demand of network slices in vertical industries, slice resource provisioning in transport networks has encountered two challenges, one is efficient slice resource provisioning in the presence of traffic uncertainty of slices, and another is flexible slice resource isolation for customizable isolation needs. In this paper, we propose an innovative flexible hybrid isolation model to support any customized resource isolation from complete isolation to full sharing, and solve the slice resource provisioning problem named Hybrid Slicing Minimum Bandwidth (HSMB) by considering traffic prediction error to mitigate the negative impact of traffic uncertainty in the proposed model. After analyzing the HSMB problem, 1) we first try to solve the problem in steps and decompose the HSMB problem into grouping sub-problem and adjusting sub-problem, 2) we then propose a low-complexity dynamic programming grouping algorithm and a fast iterative adjustment algorithm for the two sub-problems based on probabilistic feature-based analysis, 3) we combine the algorithms of the two sub-problems and further propose a linking algorithm for the potential insufficient resource dilemma and high computational complexity dilemma to improve the efficiency of the solution. The numerical results show that the proposed flexible hybrid isolation model with different factors can facilitate flexible slice isolation with customized isolation demands, while the proposed algorithm can realize efficient slice resource provisioning with a probabilistic guarantee. The comparison result shows the proposed algorithms outperform the other benchmark algorithms.**

***Index Terms*—Network slicing, resource provisioning, prediction error, vertical industrial.**

I. Introduction

NETWORK slicing has played an essential role in network resource mapping in various scenarios such as internet of things [1], vertical networks [2], virtual reality [3], etc. Recently, 5G has been developing rapidly to support vertical industries, and the vertical network slices in the 6G [4] are supposed to be more customized [5], intelligent [8], heterogeneous [9]. The slices are generated not only for different kinds of services but also for different tenants [10] running in parallel in the networks [11]. Compared with the conventional scenario, the demand for slicing in the vertical industry is more diversified and customized [6]. Taking smart grid [7] as an example, different tenant networks are featured with diverse service level agreements (SLAs), such as load control, distribution automation, sensor meter, inspection, operation, etc. For each tenant network, there will be multiple network slices (NS) belonging to different management roles to realize similar services, a large number of network resources will be consumed if these network slices are completely isolated. In fact, for some services such as inspection and operation, their traffic is elastic and the strict resource isolation between the slices is not necessary, the basic functions can be achieved with a certain amount of dedicated resources guaranteed, and the advanced functions can also be realized with more resources. In this case, customized slice isolation is necessary.

Therefore, two challenges are brought into the resource provisioning in vertical industrial slicing, the first one is the complex logical relation between slices makes simple isolation strategy complex in dealing with the flexible and customizable isolation requirements. The other one is, it is hard to make efficient dynamic network resource provisioning for massive slices along with traffic uncertainty with customized resource isolation.

There are limited researches that appropriately address resource isolation in the transport network [12], while the transport networks (e.g. backhaul, core network, etc.) realize isolation through different transmission channels, wavelengths, or time slots (e.g. slots of FlexE technology) which is referred as to physical isolation (hard isolation) [13], or inherits traditional virtual network scheme to achieve soft isolation based on logical channels (e.g. MPLS, VPN, VLAN or custom data fields) [14]. These two isolation schemes can guarantee different network isolation levels, but few researchers investigate flexible isolation based on these two isolation schemes.

Due to the dynamic traffic of network slices, slice resource provisioning in the transport networks needs to not only avoid over-provisioning but also ensure that the demands for slicing resources within one scheduling time period are met. It is feasible to realize some shared resources between slices to avoid under-provisioning. Some researchers have also proposed methods of resource sharing under the condition of resource isolation to realize the slice resource provisioning [15]. However, it is far from meeting the customizable re-

This research work is supported by the National Natural Science Foundation of China under Grant 61971070, China. It is also supported by the European Unions Horizon 2020 Research and Innovation Program through the aerOS project under Grant No. 101069732; the Academy of Finland 6Genesis project under Grant No. 318927 and the Academy of Finland IDEA-MILL project under Grant No. 352428.

Q. Guo, R. Gu and Y. Ji are with the State Key Laboratory of Information Photonics and Optical Communications, School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100086, China (e-mail: guoqize@bupt.edu.cn; rentaogu@bupt.edu.cn; jyf@bupt.edu.cn). Y. Ji is the corresponding author.

H. Yu and T. Taleb are with the Center for Wireless Communications, Oulu University, Oulu 90570, Finland (e-mail: hao.yu@oulu.fi; tarik.taleb@oulu.fi.)

quirements of tenants for flexible slice resource isolation. In this paper, we propose a novel flexible hybrid isolation model supporting the customized resource isolation level, which not only supports the general resource isolation strategies but also flexibly sets the isolation strategies from complete isolation to full sharing. Moreover, to avoid the negative impact of resource sharing that network slices may affect each other due to resource competition, a customizable grouping mechanism is introduced to reduce the impact caused by unexpected prediction deviation.

Meanwhile, researchers have widely adopted a prediction-based approach to dynamically adjusting network slicing resources to fit real-time changing traffic. Lots of predictive tools such as machine learning [16], deep neural network [17], naive benchmark, holt-winters, etc. [18] are proposed to realize the implementation of proactive dynamic network slicing based on prediction [19], [20].

However, prediction error always exists regardless of the prediction method, which leads to two potential consequences, over-provisioning or under-provisioning, which may cause resource waste or service degradation. In response to this problem, more accurate prediction methods are generally adopted. However, an accurate prediction method can only reduce the risk associated with uncertainty, but can not eliminate it. When adopting a proactive dynamic network slicing strategy, the existence of prediction errors should be taken into account while allocating network resources [11]. In this paper, we formulate a probabilistic-assured resource provisioning problem by considering the prediction error as the normal distribution based on our proposed model.

Furthermore, when dealing with the slicing resource provisioning problem to periodically allocate network resources with traffic uncertainty, one faces the potential resource dilemma, which means the actual allocatable resources can not meet the resource provisioning requirements of the slices. Moreover, when dealing with a large number of sliced resources, once probabilistic models are used, the multivariate distribution is necessarily introduced, which will greatly increase the computational complexity, and cause the potential computational dilemma. The handling of these dilemmas can greatly affect the feasibility of the slicing resource provisioning strategy. In this paper, based on our proposed model, corresponding strategies and algorithms are proposed for the existing dilemmas.

In this paper, we focus on the potential massive vertical industries slicing in the tenant network and propose a flexible hybrid isolation model (FHIM) for slice resources flexible and customizable isolation with hybrid slicing minimum bandwidth (HSMB) problem, to make an assured theoretical probability in resource provisioning based on the uncertainty of the predicted traffic. The main contributions of this paper are as follows.

1) Flexible Hybrid isolation model: We propose a novel flexible hybrid isolation model, that supports the customizable and flexible slice resource isolation for the tenants, and restricts the negative impact area between slices by grouping mechanism.

2) Efficient slice resource provisioning: We model the resource provisioning problem in FHIM named hybrid slicing minimum bandwidth (HSMB) with considering traffic prediction error to make assurance on the theoretical probability for resource provisioning with minimum allocated resource, in which the prediction error is modeled as the normal distribution.

3) Probabilistic-based Algorithm optimization: We divide the HSMB problem into two sub-problems: the grouping subproblem and the adjustment subproblem, and solve them efficiently based on probabilistic analysis. Moreover, we propose a linking algorithm (HSMB-L) to handle the potential insufficient resource dilemma and high computational complexity dilemma based on the combination of the two sub-algorithms.

The rest of the paper is structured as follows, Section II analyzes the related work. We proposed the hybrid isolation model and the HSMB problem in Section III. Section IV analyzes the HSMB problem and illustrates the solutions. Section V shows the simulation and the numerical results. Finally, Section VI concludes the paper.

II. Related Work

As mentioned in Section I, the optimization in network slicing is mainly oriented toward service uncertainty and resource isolation. Some related research on the uncertainty of traffic and resource isolation is analyzed as follows.

A. Uncertainty of the traffic in NS

Traffic uncertainty has been an impediment to the on-demand allocation of slicing resources. [18] compares different prediction methods and demonstrates that a better prediction result can reduce the uncertainty of the traffic. [21] is interested in improving prediction accuracy by predicting for each individual slice to facilitate slice resource allocation. In [22], further improvement of prediction accuracy is achieved by deep reinforcement learning to improve resource utility, [23] and [25] also propose a prediction method to guide the resource allocation. They all successfully reduce the uncertainty of prediction results in different scenarios and increase the profit of network resource allocation.

To better eliminate the impact of traffic uncertainty, appropriate over-provision is usually adopted. [19] divide the resources into multiple parts according to the fixed bandwidth and allocate one more part of the bandwidth to achieve redundancy according to the prediction results, this approach obviously lacks flexibility and does not handle enough fluctuations for some large bandwidth slices, while some resources are wasted for small bandwidth slices. Similar with [19], the authors in [24] also consider the over-provision for WDM optical network in RAN slicing based on prediction. In [11], the authors model the traffic as normal distribution and use a probabilistic model to set up network resources for potential service demand to maximize the operator's maximum revenue. [11] mainly uses weighted graphs to model the relation of the resource between different roles (user, mobile network operator, mobile network virtual operator). Moreover, the authors in [28] propose an optimal slice recovery mechanism for deterministic traffic demands and use it for evaluating the proposed robust network slicing algorithms, to ensure that it can provide adjustable tolerance of traffic uncertainty. The above papers all propose different algorithms to improve the ability of resource allocation to cope with traffic uncertainty. Furthermore, [27] adopt Wolverton–Wagner estimator [35] to model the service and use intra-slice resource pooling to support the unexpected traffic and maximize the isolation of the unpooled resources. It mainly focuses on RAN slicing while adopting statistical multiplexing in transport networks, and the optimized target is only to maximize the isolation, which is not suited for the case in transport networks. In [29], the authors proposed an algorithm to supply auxiliary resources based on prediction, and the methods for allocation of auxiliary resources. These researches all focus on the resource over-provision but ignore the resource isolation between slices. Nevertheless, these papers inspire us a lot in the design of resource over-provision.

B. Isolation of the resources in NS

In the state of the art, the most isolation strategies of NSs are completely isolated or fully shared in the transport network. In [36], the authors describe the various technologies for 5G transport network slicing and the methods for achieving resource isolation at different levels. In [30], the authors proposed two types of network slicing which are fixed allocation and fully dynamic sharing for spectrum intercell, this is a good attempt at isolation but limited to the radio resources. In [31], the authors mainly create a model to satisfy the end-to-end latency as well as isolation. However, [31] mainly focuses on the function isolation in RAN slicing and CN slicing, but merely mentions the isolation in the transport network. In [32], the authors mentioned the function isolation and transport isolation, and the slices are required for full or partial isolation from each other. [32] adopts separated wavelengths or physical links to achieve RAN slicing. All the researches mentioned above provide different understandings of resource isolation and apply them to RAN slicing, they have inspired us in designing of the hybrid isolation model in transport networks.

Compare to RAN, fewer isolation schemes have been proposed in transport networks. In [13], the authors describe the FlexE technology to realize the physical isolation with showcases for different Quality of Service (QoS) requirements, which enables the flexible hard isolation. The authors in [14] present an end-to-end architecture to provide transport network slices deployed over multi-layer IP over DWDM networks, and several degrees of isolation (from hard to soft) are required and implemented in the requested transport network slice. Based on this research, we have explored a more flexible customized hybrid isolation scheme.

In summary, the related works have driven progress in dealing with service uncertainty and resource isolation in network slicing, which gives us great inspiration. Based on the related research, we further study the problem of traffic uncertainty and resource isolation, and propose the flexible hybrid isolation model.

TABLE I
TABLE OF NOTATIONS

Symbol	*Description*
$\boldsymbol{G}$	Tenant network graph,$\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$
$\boldsymbol{V}$	Set of tenant network service endpoints, $\boldsymbol{V} = \{v_i, \forall i \in [1, M]\}$
$\boldsymbol{E}$	Set of tenant network links, $\boldsymbol{E} = \{e_{ij}, i, j \in \boldsymbol{V}\}$
$\boldsymbol{B}$	Set of available bandwidth of tenant network, $\boldsymbol{B} = \{\mathcal{B}_e, e \in \boldsymbol{E}\}$
B_δ	Minimum granularity unit of the bandwidth resource for leasing.
$\boldsymbol{S}$	Set of slices in tenant network, $\boldsymbol{S} = \{\boldsymbol{s_i}, \forall i \in [1, N]\}$
$\star(t)$	The data of $\star$ at time period Δt
$\mathcal{X}_i$	Predicted traffic of $\boldsymbol{s_i}$
$\mathcal{A}_i$	Actual traffic of $\boldsymbol{s_i}$
$\mathcal{D}_i$	Predicted error of $\boldsymbol{s_i}$
T	Historical data length of predicted error used for estimating μ_i and σ_i
τ	Defined hybrid isolation sensitivity of the NSs tenant network, $\tau = 1/T$
μ_i	Unbias estimate of the statistical mean of D_i^t with last T data
σ_i	Unbias estimate of the standard deviation of D_i^t with last T data
p_i	Normal distribution model of $\boldsymbol{s_i}$
$\boldsymbol{\mathcal{G}}$	Set of slice group in the tenant network, $\boldsymbol{\mathcal{G}} = \{\mathcal{G}_k, \forall k \in [1, K]\}$
B_d	Total required bandwidth on one link
B_d^k	The required bandwidth of $\mathcal{G}_k$ on one link
f_{ij}	The indicator variable which equals 1 while the i-th slice is assigned to the j-th slice group
b_i	Dedicated bandwidth for $\boldsymbol{s_i}$
b_s^k	Shared bandwidth for $\mathcal{G}_k$
l_i	Isolation level of $\boldsymbol{s_i}$, a parameter in hybrid isolation scheme
h_k	Hybrid degree of $\mathcal{G}_k$, a parameter in hybrid isolation scheme
$N_{\mathcal{G}_k}$	The actual number of slices in $\mathcal{G}_k$
$\mathcal{P}(\star)$	The probability of event $\star$ happens

III. HYBRID ISOLATION MODEL AND RESOURCE PROVISIONING PROBLEM

In this section, we first introduce our flexible hybrid isolation model (FHIM), and then propose the hybrid slicing minimum bandwidth (HSMB) problem. Here we mainly focus on the bandwidth resources, other slicing resources such as computing resources can be computed similarly by the algorithm proposed in this paper. Table I summarizes all the notations used in FHIM.

A. Flexible hybrid isolation model

As shown in Fig. 1, suppose that the network operator owns the network infrastructure for vertical industries, and there are multiple tenants in the transport networks. Each tenant leases network resources from the network operator based on the requirements of the NSs it serves. As the resource scheduling, management, and internal traffic within a tenant network cannot affect other tenant networks, The network resources between different tenants must be hard isolated. Meanwhile, from the perspective of the tenants, there are multiple NSs in one tenant network with the same or different SLAs, the tenant provides the best-effort service or guaranteed service to the subscribed slice users, satisfies the isolation requirement of slices, and reduces the total rented resources as much as possible. Before each time period, the tenant requests the

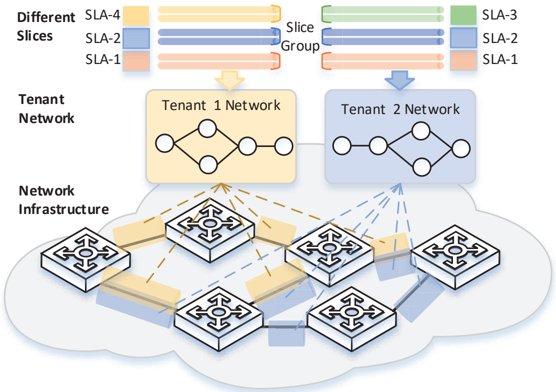


Fig. 1. The relation between NSs and network resource in hybrid isolation.

allocatable resources from the network operator, then makes traffic prediction of served slices, and requests the bandwidth resources based on the result of the prediction. Finally, the tenants lease the resources from the network operator to serve the slices.

The hybrid isolation policy is, that the slices of the tenant network are divided into groups, and the resources between the groups are hard isolated, while the resources between the slices in the same group can be either soft isolation or hard isolation to realize the customized isolation. Regarding the implementation of hybrid isolation policies, taking the IP over WDM network as an example, according to the characteristics of optical and IP networks, a hard isolation strategy can be implemented in optical networks, such as WDM, FlexE, etc. And soft isolation strategy can be implemented in the IP network, such as IP, MPLS, etc. Even if the hardware is capable enough, it is possible to achieve hybrid isolation using different layers of identification fields in the IP network, such as hard isolation through identification at the MAC layer and soft isolation through identification at the IP layer, thus achieving a hybrid isolation implementation in a pure IP network.

Assuming the predictable traffic of multiple types of slices, the proposed hybrid isolation model mainly copes with the tenant network, and the tenant can apply the hybrid isolation and set the different SLA-related parameters to different slices. For one tenant network, it can be described as a graph $\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E})$, where $\boldsymbol{V} : \{v_i, \forall i \in [1, M]\}$ and $\boldsymbol{E} : \{e_{ij}, \forall i, j \in \boldsymbol{V}\}$ represent the set of service endpoints and the set of links respectively. The bandwidth of the links is represented as $\boldsymbol{B} : \{\mathcal{B}_e, \forall e \in \boldsymbol{E}\}$. For the NSs, let $\boldsymbol{S} : \{\boldsymbol{s_i}, i \in [1, N]\}$ represent the set of NSs in the tenant network. The traffic of $\boldsymbol{s_i}$ is dynamic, and we assume an appropriate prediction method has been employed to predict the traffic of $\boldsymbol{s_i}$. Since network resources have a minimum allocation unit in a transport network, we let B_δ be the minimum granularity, which means the bandwidth of the tenant network is an integer multiple of B_δ.

In time period Δt, let $\boldsymbol{X}(t) : \{\mathcal{X}_i(t), \forall i \in [1, N]\}$ represent the set of predicted traffic of the NSs, $\boldsymbol{Y}(\boldsymbol{t}) : \{\mathcal{Y}_i(t), \forall i \in [1, N]\}$ represent the actual traffic of the NSs. $\boldsymbol{D}(t) : \{\mathcal{D}_i(t), \forall i \in [1, N]\}$ represents the absolute error of the NSs between the predicted traffic and actual traffic at Δt. The relation between them is shown in Eq.(1).

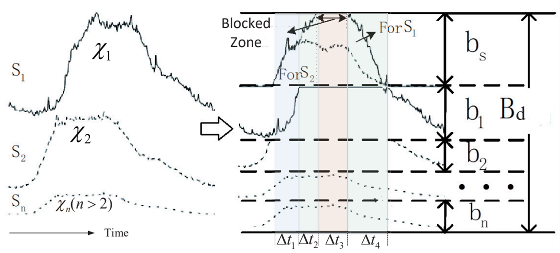


Fig. 2. The traffic and the bandwidth allocation of NSs in one slice group.

$$\boldsymbol{Y}(t) = \boldsymbol{D}(t) + \boldsymbol{X}(t) \tag{1}$$

Besides, as previously mentioned, to perform flexible and customizable network resource isolation, all the NSs in the same tenant network are divided into $\boldsymbol{\mathcal{G}}$ groups by SLA or service type, and the slices in the same group have the same SLA. Let $\mathcal{G}_k$, $\forall k \in [1, K]$ represent the k-th group. For better modeling, let $\boldsymbol{F}$, which element is f_{ij}, be the indicator variable matrix, which equals 1 while the i-th slice is assigned to the j-th group, otherwise is 0. The relation between slices and groups is shown in Eq. (2).

$$\boldsymbol{S} \cdot \boldsymbol{F} = \boldsymbol{\mathcal{G}} \tag{2}$$

Meanwhile, there is a constraint that the NSs in $\mathcal{G}_k$ have the same source endpoint and destination endpoint, and we assume they take the same transport path.

Hence, the bandwidth B_d^k will be allocated to the k-th slice group $\mathcal{G}_k$ in the tenant network. With considering the resource independence of network slices and the utilization of the network, we split the bandwidth (B_d^k) into two parts: dedicated bandwidth $\boldsymbol{b} : \{b_i, \forall i \in [1, N_{\mathcal{G}_k}]\}$ and sharing bandwidth b_s^k, in which $N_{\mathcal{G}_k}$ is the number of the NSs in $\mathcal{G}_k$, b_i means the bandwidth only used to support i-th slice, and b_s^k is used as the sharing resource for all NSs in $\mathcal{G}_k$. Then the total distributed bandwidth of group $\mathcal{G}_k$ can be calculated using Eq. (3).

$$B_d^k = \sum\nolimits_{\boldsymbol{s_i} \in \mathcal{G}_k} b_i + b_s^k \tag{3}$$

Fig. 2 shows an example of the relation of b_i and b_s^k in group $\mathcal{G}_k$ (to simplify the representation, the superscript k has been removed from the figure). The left part shows n slices $\boldsymbol{s}_i, \forall i \in [1, n]$ with actual traffic $\mathcal{X}_i$ in one group, while the right part shows their resource allocation and actual occupation based on our model. Each dedicated resource b_i is allocated for $\boldsymbol{s}_i$, b_s is the shared resources of the group. For each slice $\boldsymbol{s}_i$, the resource occupancy priority of b_i is higher than that of b_s. Before time period Δt_1, the traffic of each slice is lower than their dedicated resource and used their dedicated resource to transport the traffic. In time period Δt_1, $\mathcal{X}_2$ is greater than b_2, and begins to occupy the resources in b_s. In time period Δt_2, $\mathcal{X}_1$ turns to be greater than b_1, and begin to use b_s together with $\mathcal{X}_2$, and the resource is enough due to $\mathcal{X}_1 + \mathcal{X}_2 \leq b_1 + b_2 + b_s$. In time period Δt_3, the service degradation happens in both

$\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, because $\mathcal{X}_1+\mathcal{X}_2>b_1+b_2+b_s$, but $\boldsymbol{s}_i, \forall i \in [3,n]$ are not affected because $\mathcal{X}_i<b_i, \forall i \in [3,n]$ and they do not occupy b_s, this shows if the actual traffic of some slices is quite different from the prediction result, they may cause the slices in the same group degraded whose actual traffic larger than their dedicated bandwidth. Time period Δt_4 is similar to time period Δt_2, and the time periods after Δt_4 are similar to the time periods before Δt_1, no more analysis of service degrades here.

What we are mainly concerned with is whether the prediction errors obey normal distribution. Here the distribution of prediction errors can be either expected based on the prediction method and the traffic model, or adopt the normality test methods (Kolmogorov Smirnov test [33], Anderson-Darling test [34], etc.) to evaluate whether the forecast errors satisfy a normal distribution. Our model is applicable to the case where the prediction errors obey normal distribution. This is a necessary condition for our research.

As mentioned in section I, the prediction error exists no matter what prediction method is used. Thus the historical errors (last T time periods) of the predicted traffic can be seen as normally distributed, which is shown in Eq.(4).

$$\mathcal{D}_i\left(t\right) \sim \mathcal{N}\left(\mu_i\left(t\right), {\sigma_i}^2\left(t\right)\right) \tag{4}$$

in which $\mu_i\left(t\right)$ and $\sigma_i\left(t\right)$ represent the unbiased estimate of the mean and standard error of the predicted error of $\boldsymbol{s_i}$ respectively at the time period Δt, and they can be calculated by hypothesis testing based on the last T time periods. We use the data of the last T time periods instead of the whole historical data, it is because if all historical data are used, new data at time period $\Delta\left(t-1\right)$ will have little impact in calculating $\mu_i\left(t\right)$ and $\sigma_i\left(t\right)$, which cause the model insensitive to the feature of traffic changes.

As mentioned above, each slice s_i is assigned to a dedicated bandwidth b_i (similar to the guaranteed bit rate in IP networks). Meanwhile, the shared bandwidth b_s^k has been allocated in $\mathcal{G}_k$ for each slice in it, which means that s_i can use the bandwidth b_i first, if the actual traffic $\mathcal{X}_i$ exceed b_i, they can occupy b_s^k temporary to avoid the slices in $\mathcal{G}_k$ being degraded.

The hard isolation (resource between tenant networks and between different slice groups) and soft isolation (resource of the slice in the same slice group) are applied in the tenant network simultaneously, to support flexible resource provisioning for NSs.

For better configuration in the proposed FHIM, we define some parameters to support customizable resource isolation.

1) Hybrid isolation sensitivity τ: Let τ be the hybrid isolation sensitivity of the tenant network, which is related to T as $\tau=1/T$.

According to the definition of τ, τ affects the rate of change of the mean and variance of the prediction error statistics, as shown in Eq. (5) and Eq. (6). Meanwhile, the iterative equations of $\mu_i\left(t\right)$ and $\sigma_i\left(t\right)$ are shown in Eq. (7) and Eq. (8).

$$\mu_i\left(t\right) = \tau \sum\nolimits_{x=t-\frac{1}{\tau}}^{t} \mathcal{D}_i\left(x\right) \tag{5}$$

$${\sigma_i}^2\left(t\right) = \frac{\tau}{1-\tau}\sum\nolimits_{x=t-\frac{1}{\tau}}^{t}\left(\mathcal{D}_i\left(x\right)-\mu_i\left(t\right)\right)^2 \tag{6}$$

$$\mu_i\left(t\right)-\mu_i(t-1) = \tau\left(\mathcal{D}_i\left(t\right)-\mathcal{D}_i\left(t-1-\frac{1}{\tau}\right)\right) \tag{7}$$

$$\begin{aligned}{\sigma_i}^2\left(t\right)-{\sigma_i}^2\left(t-1\right) = \tau\left({\mathcal{D}_i}^2\left(t\right)-\mathcal{D}_i^2\left(t-1-\frac{1}{\tau}\right)\right)\\ -2\left(\mathcal{D}_i\left(t\right)-\mathcal{D}_i\left(t-1-\frac{1}{\tau}\right)\right)\hat{\mu}\end{aligned} \tag{8}$$

$$\hat{\mu} = \frac{\tau}{1-\tau}\sum_{x=t-\frac{1}{\tau}}^{t-1}\mathcal{D}_i\left(x\right) \tag{9}$$

where $\hat{\mu}$ is the mean value of $\mathcal{D}_i$ from time period $\Delta\left(t-1/\tau\right)$ to $\Delta\left(t-1\right)$, as shown in Eq. (9).

In general, the value of τ takes a relatively small value for enough historical data to be statistically significant, for example, if $\tau=0.02$, it means the number of historical data is $T=50$ for estimating μ_i^t and σ_i^t.

At the beginning of Δt, let $\boldsymbol{P}:\{p_i, i\in[1,N]\}$ represent the distribution of the traffic with different slices at Δt. Based on the assumptions, $p_i\left(t\right)$ belongs to normal distribution, and it is shown in Eq. (10).

$$p_i\left(t\right) \sim \mathcal{N}\left(\mathcal{X}_i\left(t-1\right)+\mu_i\left(t-1\right), \sigma_i^2\left(t-1\right)\right) \tag{10}$$

2) Slice resource provisioning level a_k: Let $a_k\in(0,1)$ be the slice resource provisioning level of $\mathcal{G}_k$, which indicates the theoretical probability that the provided resource of the slices in $\mathcal{G}_k$ can meet the requirement of traffic in them.

3) Isolation level l_i: Let $l_i\in[0\%,100\%]$ represent the isolation level of $\boldsymbol{s_i}$, which indicates the theoretical guaranteed availability of $\boldsymbol{s_i}$. It represents the assurance probability that the slices used the dedicated bandwidth b_i. It can be learned that it is the lower limit of a_k. In other words, l_i ensures the lower limit of allocated resources for supporting $\boldsymbol{s_i}$, the relation of l_i and b_i is shown in Eq. (11).

$$l_i \le \mathcal{P}\left(\mathcal{X}_i\left(t\right)\le b_i\right) \tag{11}$$

4) Hybrid degree h_k: Let an integer h_k be the hybrid degree of $\mathcal{G}_k$, it is the upper limit of the number of the network slices in $\mathcal{G}_k$, h_k limits the impact of the unexpected bursts of traffic, as network slices are potentially affected by other slices using the same shared bandwidth $b_s^{\mathcal{G}_k}$, especially the predicted traffic has a large error that is lower than actual traffic. h_k limits the scale of the negative impact of the abnormal prediction. So the actual number of slices $N_{\mathcal{G}_k}$ in $\mathcal{G}_k$ is constrained as Eq. (12).

$$N_{\mathcal{G}_k} = \sum\nolimits_{i\in[1,N]} f_{ik} \le h_k, \forall k\in[1,K] \tag{12}$$

According to the parameters, when $l_i, \forall \boldsymbol{s}_i\in\mathcal{G}_k$ is set to 100%, it means that the slices in $\mathcal{G}_k$ are hard isolated, and the resources between slices will not be shared. When $l_i, \forall \boldsymbol{s}_i\in\mathcal{G}_k$ is set to 0%, it means that the slices in $\mathcal{G}_k$ adopt statistical multiplexing, and the resources between slices are fully shared. If l_i is set to different values, customizable resource isolation in $\mathcal{G}_k$ is realized.

B. Hybrid slicing minimum bandwidth problem

In this section, we analyze the slice provisioning problem in time period Δt. To simplify the representation, we will omit the time period stamp on the symbol.

Suppose $\mathcal{A}$ is a subset of $\mathcal{G}_k$, we use $\mathcal{E}_{\mathcal{A}}^1$ to indicate the case that the traffic $\mathcal{X}_i$ of each slice $\boldsymbol{s_i} \in \mathcal{A}$ is lower than their dedicated bandwidth, as Eq. (13) shows. In slice set $\mathcal{A}$, Eq. (11) can be expressed as Eq. (14). $\mathcal{E}_{\mathcal{A}}^2$ represents the event that each traffic of the slices in $\mathcal{A}$ larger than their dedicated bandwidth, and the total traffic of all the slices in $\mathcal{A}$ is smaller than their total bandwidth which equals the sum of their dedicated bandwidth b_i and the total shared bandwidth b_s, the relation is shown in Eq. (15). Meanwhile, let $\bar{\mathcal{E}}_{\mathcal{A}}^1$ and $\bar{\mathcal{E}}_{\mathcal{A}}^2$ represent the complementary set of $\mathcal{E}_{\mathcal{A}}^1$ and $\mathcal{E}_{\mathcal{A}}^2$.

$$\mathcal{E}_{\mathcal{A}}^1 : \mathcal{X}_i \leq b_i, \forall \boldsymbol{s_i} \in \mathcal{A} \tag{13}$$

$$\mathcal{P}\left(\mathcal{E}_{\mathcal{A}}^1\right) \geq l_i, \forall \boldsymbol{s_i} \in \mathcal{A} \tag{14}$$

$$\mathcal{E}_{\mathcal{A}}^2 : \mathcal{X}_n > b_n, \sum_{n \in A} \mathcal{X}_n \leq \sum_{n \in \mathcal{A}} b_n + b_s, \forall n \in \mathcal{A} \tag{15}$$

With the assumption that the traffic of network slices is independent of each other, $\mathcal{E}_{\mathcal{A}}^1$ and $\mathcal{E}_{\mathcal{A}}^2$ can be derived from the univariate and multivariate normal distribution cumulative distribution function (CDF). Let $\bar{\mathcal{E}}_{\mathcal{A}}^b, \forall \mathcal{A} \subseteq \mathcal{G}_k$ represent the event that there is any slice in $\mathcal{A}$ has degraded due to the inadequate resource. In other words, when $\bar{\mathcal{E}}_{\mathcal{G}_k}^b$ happens, the slices in $\mathcal{A}, \forall \mathcal{A} \subseteq \mathcal{G}_k$, in which the traffic exceeds the dedicated bandwidth on the link will be blocked. Meanwhile, we let $\mathcal{E}_{\mathcal{A}}^b$ be the complement of $\bar{\mathcal{E}}_{\mathcal{A}}^b$ (Eq. (16)), which means the slices set $\mathcal{A}$ is normal and no slices are blocked. Based on our model, the probability that one group is not blocked can be calculated (shown in Eq. (17)), as well as the unblocked probability of the i-th slice $\mathcal{P}\left(\mathcal{E}_i^b\right)$ (shown in Eq. (18)).

$$\mathcal{P}\left(\mathcal{E}_{\mathcal{A}}^b\right) = 1 - \mathcal{P}\left(\bar{\mathcal{E}_{\mathcal{A}}^b}\right), \forall \mathcal{A} \subseteq \mathcal{G}_k \tag{16}$$

$$\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right) = \sum_{\mathcal{A} \subseteq \mathcal{G}_k} \mathcal{P}\left\{\mathcal{E}_{\mathcal{A}}^1\right\} \mathcal{P}\left\{\mathcal{E}_{\mathcal{G}_k - \mathcal{A}}^2\right\} \tag{17}$$

$$\mathcal{P}\left(\mathcal{E}_i^b\right) = \mathcal{P}\left(\mathcal{E}_i^1\right) + \sum_{\mathcal{A} \subset \mathcal{G}_k} \mathcal{P}\left(\mathcal{E}_{\mathcal{A}}^1\right) \mathcal{P}\left(\mathcal{E}_{(\mathcal{G}_k - \mathcal{A}) \cup i}^2\right) \tag{18}$$

Meanwhile, according to the definition, we can derive the relative of $\mathcal{P}\left(\mathcal{E}_i^b\right)$ and $\mathcal{P}\left(\mathcal{E}_{\boldsymbol{S}}^b\right)$, shown in Eq. (19):

$$\mathcal{P}\left(\mathcal{E}_i^b\right) - \mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right) = \mathcal{P}\left(\mathcal{E}_i^1\right) \mathcal{P}\left(\bar{\mathcal{E}}_{\mathcal{G}_k - i}^b\right) \geq 0 \tag{19}$$

in which, the right part of the equal sign in Eq. (19) means the probability of that $\boldsymbol{s_i}$ uses its dedicated bandwidth and the other slices in $\mathcal{G}_k$ have blocked. According to Eq. (19), $\mathcal{P}\left(\mathcal{E}_i^b\right), \forall \boldsymbol{s}_i \in \mathcal{G}_k$ is always not smaller than $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$.

Once we allocate resources based on a_k for each slice, while a portion of the resources of the slices is used for sharing in b_s, due to the benefits of statistical multiplexing, a_k will be improved, which means we can save bandwidth to reach the original a_k requirement. We set our object to minimize the allocated bandwidth, and the problem can be formulated as nonlinear programming named Hybrid Slicing Minimum Bandwidth (HSMB) problem, shown as follows:

$$\text{o.b.} \quad \min \sum\nolimits_{\mathcal{G}_k \in \boldsymbol{\mathcal{G}}} \lceil B_d^k / B_\delta \rceil * B_\delta \tag{OP1}$$

$$\text{s.t.} \quad \mathcal{P}\left(\mathcal{E}_i^b\right) \geq a_i \qquad , \forall i \in \mathcal{G}_k \tag{20}$$

$$B_d^k \leq \mathcal{B}_e \qquad , \forall k \in [1, K], \forall e \in \mathcal{R}_e \tag{21}$$

$$\sum\nolimits_{k \in [1, K]} f_{ik} \leq 1, \forall i \in [1, N] \tag{22}$$

$$b_i \geq 0, b_s^j \geq 0 \qquad , \forall i \in [1, N], \forall j \in [1, \mathcal{G}] \tag{23}$$

$$Eq.(3), (12), (14).$$

in which, Eq. (20) represents that all the blocking probability of the slices must meet the requirement of a_k. Eq. (21) indicates the total used bandwidth can not exceed the bandwidth of the link. Eq. (14) is denoted the isolation requirement of the slices. Eq. (12) meets the requirement of the number limitation of the slices in each group. Eq. (22) ensures each slice only can be assigned into one group.

The HSMB problem is a mixed integer nonlinear programming (MINLP) problem, it is NP-Hard, as it can be categorized as a special set partition problem. The problem can be described as: find a serials subsets $\boldsymbol{\mathcal{G}}$ of slices set $\boldsymbol{S}$, each subset $\mathcal{G}_k$ can derive the required minimum bandwidth B_d^k under the non-linear programming requirements of each slice $\boldsymbol{s_i} \in \mathcal{G}_k$, to minimize the total bandwidth.

Besides, Eq. (20) is nonlinear, and it is calculated by the cumulative distribution probability of multivariate normal distribution. Although it can be evaluated using the more mature algorithm [37], it still has high computational complexity. If we use a typical algorithm to solve the OP1 (such as heuristic algorithms, etc.), the computational complexity is very high, especially if the number of slices is large. Therefore, we do a deep analysis of the HSMB and propose a series of algorithms to solve OP1.

IV. HSMB-L Algorithm for HSMB problem

As HSMB can be seen as a set partition problem, we divide it into two steps, we first analyze the grouping condition that meets the requirements of minimum bandwidth, and group them with HSMB grouping (HSMB-G) algorithm based on the condition, and then proposed the HSMB adjustment algorithm (HSMB-A) to make bandwidth adjusting in each group based on probabilistic analysis. Moreover, we proposed HSMB linking (HSMB-L) algorithm to deal with the potential insufficient resource dilemma and high computational complexity dilemma, with the combination of HSMB-G and HSMB-A, to form the whole solution of HSMB.

A. Grouping

In this subsection, we mainly analyze the slices in one group and find the characteristics of the slices within the group in the best grouping scheme, in which the required resource is the least for the same degraded probability. For simplicity, the indication of the group in the symbol is omitted in this subsection, for example, we use B_d to represent B_d^k.

According to Eq. (19), the probability of $\boldsymbol{s}_i$ not be degraded is the sum of the CDF of multivariate normal distribution, and $\mathcal{P}\left(\mathcal{E}_i^b\right)$ is bigger than $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$. Here we are mainly concerned with the increasing $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$ for the minimum guarantee of resource provisioning. Based on the property of normal distribution, we can get the following conclusions.

Theorem 1: In HSMB, given a fixed B_d. $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_j}^b\right)$ is decreasing function of $b_k, \forall \boldsymbol{s}_k \in \mathcal{G}_j$, where b_k is the dedicated bandwidth of the k-th slice in $\mathcal{G}_j$, while b_i, $\forall \boldsymbol{s}_i \in \mathcal{G}_j - \boldsymbol{s}_k$ are fixed.

Proof 1: Suppose there are K slices in $\mathcal{G}_j$, express as $\mathcal{G}_j = \{s_k, k \in [1, K]\}$, in which the predicted traffic of them obey the normal distribution as $\mathcal{N}\left(\mathcal{X}_k + \mu_k, \sigma_k\right)$ respectively. So the blocking probability of $\mathcal{G}_j$ can be expressed by Eq. (17). Let $\Phi_{\mathcal{A}}^1\left(\boldsymbol{b}^{\star}\right) = \mathcal{P}\left(\mathcal{E}_{\mathcal{A}}^1\right)$, $\Phi_{\mathcal{A}}^2\left(\boldsymbol{b}^{\star}\right) = \mathcal{P}\left\{\mathcal{E}_{\mathcal{A}}^2\right\}$, in which $\boldsymbol{b}^{\star}$ is the standardized form of $[b_i, b_s]$, and their probability density are $\phi_{\mathcal{A}}^1\left(\boldsymbol{b}^{\star}\right)$ and $\phi_{\mathcal{A}}^2\left(\boldsymbol{b}^{\star}\right)$ respectively. Meanwhile, let $f_{\mathcal{A}}\left(\boldsymbol{b}\right) = \Phi_{\mathcal{A}}^1\left(\boldsymbol{b}^{\star}\right)\Phi_{\mathcal{G}_j-\mathcal{A}}^2\left(\boldsymbol{b}^{\star}\right) = \mathcal{P}\left(\mathcal{E}_{\mathcal{A}}^1\right)\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_j-\mathcal{A}}^2\right)$. As assumption, $b_i\left(i \in \mathcal{G}_j - i\right)$ and B_d are fixed, hence $d\left(b_s\right) = -d\left(b_k\right)$. It can be derived that $\frac{\partial f_{\mathcal{A}}(\boldsymbol{b})}{\partial(b_k)} = \begin{cases} \Gamma_{\mathcal{A}-k,k} + \Delta_{\mathcal{A},k}(\boldsymbol{b}^{\star}), k \in \mathcal{A} \\ -\Gamma_{\mathcal{A},k}, k \in \mathcal{G}_j - \mathcal{A} \end{cases}$, in which $\Gamma_{\mathcal{A},k} = \phi_k^1\left(b_k^{\star}\right)\Phi_{\mathcal{A}}^1\left(\boldsymbol{b}^{\star}\right)\Phi_{\mathcal{G}_j-k-\mathcal{A}}^2\left(\boldsymbol{b}^{\star}\right) + \Phi_{\mathcal{A}}\left(\boldsymbol{b}^{\star}\right)$ and $\Delta_{\mathcal{A},k}\left(\boldsymbol{b}^{\star}\right) = \Phi_{\mathcal{A}}^1\left(\boldsymbol{b}^{\star}\right)\partial\Phi_{\mathcal{G}_j-\mathcal{A}}^2\left(\boldsymbol{b}^{\star}\right)/\partial b_k < 0$. Meanwhile,$\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_j}^b\right) = \sum_{\mathcal{A}\subset\mathcal{G}_j} f_{\mathcal{A}}\left(\boldsymbol{b}\right)$. Observe that $\Gamma_{\mathcal{A}-k}, k \in \mathcal{A}$ and $\Gamma_{\mathcal{A}}, k \in \mathcal{G}_j - \mathcal{A}$ are same if go through $\mathcal{A} \subseteq \mathcal{G}_j$, and $\partial f\left(b\right)/\partial b_k = \Delta_{\mathcal{A}}\left(b^{\star}\right) < 0$, so $\partial\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_j}^b\right)/\partial b_k < 0$, proven.

As described in Theorem. 1, $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$ increases as b_i decreases. As the universality of $\boldsymbol{s}_i$, $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$ can get the maximum value when all b_i get the minimum value. Meanwhile, if $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$ exceeds a_k, B_d can be reduced to make $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$ down close to a_k by a very small number δ. Following this strategy, the optimal B_d can be achieved when $b_i, \forall \boldsymbol{s}_i \in \mathcal{G}_j$ is as small as possible. Hence, the constraint shown in Eq. (14) can be reduced to the constraint shown in Eq. (24).

$$l_i = \mathcal{P}\left(\mathcal{E}_i^1\right), \forall i \in [1, \mathcal{G}] \tag{24}$$

Then we focus on the grouping strategy of the slices in $\mathcal{G}_j$. Suppose each $\boldsymbol{s}_i \in \mathcal{G}_k$ has a contribution on b_s, and we define the contribution degree b_{si} is shown as Eq. (25).

$$b_{si} = b_s \frac{\sigma_i}{\sum_{\boldsymbol{s}_j \in \mathcal{G}_k} \sigma_j} \tag{25}$$

Lemma 1: $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$ is decreasing function of the variance $\sigma_i, \forall \boldsymbol{s}_i \in \mathcal{G}_k$ while

$$b_i + b_{si} \geq \mu_i, \forall \boldsymbol{s}_i \in \mathcal{G}_k \tag{26}$$

Based on Lemma. 1, we take account into a special case, that $l_i = 0, \forall \boldsymbol{s}_i \in \mathcal{G}_k$. Then the slices in $\mathcal{G}_k$ can be regarded as complete statistical multiplexing, so the total traffic of $\boldsymbol{s}_i, \forall \boldsymbol{s}_i \in \mathcal{G}_k$ obeys the normal distribution $\mathcal{N}\left(\sum \boldsymbol{\mu}_i, \|\boldsymbol{\sigma}_j\|_2\right), \forall i \in \mathcal{G}_k$, in which $\|\boldsymbol{\sigma}_k\|_2$ means the 2-th norm of $\boldsymbol{\sigma}_k$. The conclusion can be extended to our hybrid isolation model because their trends are similar.

Then the unblocked probability $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_i}^b\right)$ of slices in $\mathcal{G}_k$ is $\Phi\left(\left(\sum b_i + b_s - \sum \mu_i\right)/\|\boldsymbol{\sigma}_k\|_2\right)$. Based on Theorem. 1, $\left(b_i - \mu_i\right)/\sigma_i, \forall i \in \mathcal{G}_i$ equals to l_i, which is a constant parameter here, $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$ can be derived as Eq. (27) shows.

$$\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right) = \Phi\left(\sum l_i * \sigma_i / \|\boldsymbol{\sigma}_k\|_2\right) \tag{27}$$

As Eq. (27) shows, $\mathcal{P}\left(\bar{\mathcal{E}}_{\mathcal{G}_j}^b\right)$ is a non-linear increasing function of $\sum \sigma_i / \|\boldsymbol{\sigma}_i\|_2$. We can find $\sum \sigma_i / \|\boldsymbol{\sigma}_k\|_2 \in \left[1, \sqrt{2}\right]$, and a_i always set to near 1 (Satisfy the condition of Theorem. (1) in actual application). The gradient here is already small, so we reduce $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$ to a linear mapping of $\sum \sigma_i / \|\boldsymbol{\sigma}_k\|_2$ for approximation.

Based on the approximation, some conclusions can be derived.

Theorem 2: In HSMB, to maximize the unblocked probability $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_k}^b\right)$ of $\mathcal{G}_k$, The condition that maximizes the $\sum \sigma_i / \|\boldsymbol{\sigma}_i\|_2, \forall \boldsymbol{s}_i \in \mathcal{G}_k$ should be met.

Corollary 1: In HSMB, with the same condition of the theorem. 2, the best grouping method is to divide the slices adjacent to the standard deviation into one group.

Proof 2: We focus on the monotonicity of $f = \sum \sigma_i / \|\boldsymbol{\sigma}_i\|_2$, and take the derivative of σ_i first, as Eq. (28) shows, it can be derived that f can get the maximum value while $\sigma_i = \sigma_j, \forall i, j \in \mathcal{G}_k$, and the value gradually decreases as the gap becomes larger. If the standard deviation of the slices in one group is not adjacent, f can be larger if the standard deviation of the slices adjacent to the standard deviation of the slices in the group is exchanged with the standard deviation of the slices at the edge in the group. Therefore, the best grouping scheme is to put the slices adjacent to the standard deviation into a group. proven.

$$\frac{\partial f}{\partial \sigma_i} = \frac{\sum_{j\in\mathcal{G}_j} \sigma_j\left(\sigma_j - \sigma_i\right)}{\|\boldsymbol{\sigma}_k\|_2^3}, \forall \boldsymbol{s}_i \in \mathcal{G}_k \tag{28}$$

Based on the theory of statistical multiplexing, we can get Lemma. 2.

Lemma 2: In HSMB, if $B_d^k = \sum_{\boldsymbol{s}_i\in\mathcal{G}_k}\left(b_i + b_{si}\right)$, then $\mathcal{P}\left(\mathcal{E}_{\mathcal{G}_i}^b\right) \leq \mathcal{P}\left(\mathcal{E}_{\mathcal{G}_j}^b\right)$ while $\mathcal{G}_i \subseteq \mathcal{G}_j$.

Theorem 3: In HSMB, given the fixed B_d^k, l_i and a_k, as the solution of HSMB, the number of groups $N_{\boldsymbol{S}}$, and the number of slices in k-th groups $N_{\mathcal{G}_k}$ should satify the equations shown in Eq. (29) and Eq. (30).

$$N_{\boldsymbol{S}} = \lceil N/h_k \rceil \tag{29}$$

$$N_{\mathcal{G}} \in \left[N_{\boldsymbol{S}} - \left(\left(h_k * N_S - N\right) \mod N_{\boldsymbol{S}}\right), h_k\right] \tag{30}$$

in which $\lceil\star\rceil$ means round $\star$ up to an integer.

Based on the above properties, we propose a dynamic programming algorithm to realize the grouping strategy, shown in Algorithm 1, in which some auxiliary variables are introduced. In Line 1, the algorithm is initialized, where I_{min} and I_{max} are arrays with length N, and their i-th element represents the upper and lower limits of the slice index when the first i slices complete grouping, they indicate the zone of next index which $\boldsymbol{s}_i$ need to participate in grouping calculation under the constraints of Theorem. 3. Line 2 calculates the $N_{\boldsymbol{S}}$ and $N_{\mathcal{G}_k}$, and Line 3 sorts the slices according to the standard deviation to ensure the effectiveness of Corollary. 1 when grouping. Line 4 to Line 10 are used to deal with the grouping directly when N is an integer multiple of h_k. We take it separately because the grouping is relatively simple in this case, which is just taking h_k values from $\hat{\boldsymbol{s}}$ in sequence as a group. Otherwise, the pseudo-code between Line 11 to Line 34 shows the dynamic programming algorithm of grouping. T_L is an array that stores the best grouping strategy. Each element of T_L is an NS set. The i-th element represents the best grouping strategy of the first i slices in $\hat{\boldsymbol{s}}$. In the dynamic programming algorithm, while $i \in [min(N_{\mathcal{G}_k}), h_k]$, $T_L[i]$ is the set of the first i slices in $\hat{\boldsymbol{s}}$, shown in Line 12 to Line 16, otherwise, let T_{Ltmp} be an array that store all possible grouping schemes for the first i slices, and T_1 used to store each grouping temporarily. We can get the recurrence formula as Eq. (31), shown from Line 18 to Line 24. Then the pseudo-code shown from Line 25 to Line 30 shows that we select the best grouping from T_{Ltmp} through the adjustment function (shown in Eq. (32)) based on Theorem. 2, so the last element of T_L is the best grouping scheme (shown in Line 35). Line 31 deals with the case that the first i slices cannot be grouped under the constraint of Theorem. 3.

$$T_1{=}T_L\left[j\right]\bigcup\left\{\bigcup_{k=j+1}^{i} s_k\right\}, \forall i \geq h_k, j \in [0, i-1] \tag{31}$$

$$F\text{unc}\left(\boldsymbol{T}\right)=\bigcup_{T_1\in\boldsymbol{T}}\left\{\sum_{Z\in T_1}\sum_{\mathcal{G}\in Z}\frac{\sum_{i\in\mathcal{G}}\sigma_i}{\|\boldsymbol{\sigma}_{\mathcal{G}}\|_2}\right\} \tag{32}$$

In order to better understand Algorithm 1, here is an example. Suppose that a set $\hat{\boldsymbol{S}} = \{\hat{\boldsymbol{s}}_0, \hat{\boldsymbol{s}}_1, \cdots, \hat{\boldsymbol{s}}_9\}$ with 10 sorted slices, grouped by $h_k = 4$, we can get $N_{\boldsymbol{S}} = 3$ and $N_{\mathcal{G}} = [2, 4]$. Then the value of T_L,I_{min},I_{max} in each iteration of the dynamic programming is shown in Table II, in which the $BEST\{\star\}$ means the optimal scheme of $\star$ calculated by Lines 26 and Line 27 in Algorithm 1, and the best grouping scheme is the last element of L_T (in this example is $L_T[9]$).

Algorithm 1: Dynamic Programming Algorithm for Grouping in HSMB (HSMB-G).

```
   Data: $\boldsymbol{s}_i \in \boldsymbol{S}, h_k$
   Result: The grouping list $A_{\mathcal{G}}$ of $\boldsymbol{S}$
 1 Initialization:$\boldsymbol{s}_i \in \boldsymbol{S}$,$\mathcal{B}_e$,$n^l$, temporary variable
     $I_{min} = \emptyset, I_{max} = \emptyset, T_L = [\emptyset] * N, T_{Ltmp}, L$, $T_1$ and
     $A_{\mathcal{G}} = \emptyset$ ;
 2 Calculate $N_{\boldsymbol{S}}$,$N_{\mathcal{G}}$ with Eq. (29) and Eq. (30) ;
 3 Sorted $\boldsymbol{s}_i$ into $\hat{\boldsymbol{s}}_i$ according to the value of $\sigma_i$ ;
 4 if $\min\left(N_{\mathcal{G}}\right)$ equals $h_k$ then
 5     for slice index $i$ = 0 to $N_{\boldsymbol{S}} - 1$ do
 6         Set $j \leftarrow i * h_k$ ;
 7         Set $A_{\mathcal{G}}.add\left(\{\hat{\boldsymbol{s}}_j, \hat{\boldsymbol{s}}_{j+1}, \cdots, \hat{\boldsymbol{s}}_{j+h_k-1}\}\right)$
 8     end
 9     return $A_{\mathcal{G}}$
10 end
11 for Index $i = 0$ to $N - 1$ do
12     if $\min\left(N_{\mathcal{G}_k}\right) \leq i + 1 \leq h_k$ then
13         $T_L\left[i\right].add\left(\{\hat{\boldsymbol{s}}_0, \hat{\boldsymbol{s}}_1, \cdots, \hat{\boldsymbol{s}}_i\}\right)$ ;
14         $I_{min}\left[i\right] \leftarrow h_k + \min\left(N_{\mathcal{G}}\right)$ ;
15         $I_{max}\left[i\right] \leftarrow h_k + i + 1$ ;
16     else
17         Set $\boldsymbol{T}_{Ltmp}$ = $\emptyset$ ;
18         for $j = 0$ to $i - 1$ do
19             if $I_{min}\left[j\right] \leq i + 1 \leq I_{max}\left[j\right]$ then
20                 $T_1 \leftarrow \left(T_L\left(j\right)\right)$ ;
21                 $T_1.add\left(\{\hat{\boldsymbol{s}}_{j+1}, \hat{\boldsymbol{s}}_{j+2}, \cdots, \hat{\boldsymbol{s}}_i\}\right)$ ;
22                 $\boldsymbol{T}_{Ltmp}.add\left(T_1\right)$ ;
23             end
24         end
25         if length($\boldsymbol{T}_{Ltmp}$) not equals $0$ then
26             $m \leftarrow Index\left(\max\left(Func\left(\boldsymbol{T}_{Ltmp}\right)\right)\right)$ ;
27             $T_L\left[i\right] = \boldsymbol{T}_{Ltmp}\left[m\right]$ ;
28             $I_{min}\left[i\right] \leftarrow \min\left(N_{\mathcal{G}}\right) + h_k * \lceil(i+1)/h_k\rceil$ ;
29             $I_{max}\left[i\right] \leftarrow h_k + i + 1$ ;
30         else
31             Set $T_L\left[i\right], I_{min}\left[i\right], I_{max}\left[i\right]$ to $None$
32         end
33     end
34 end
35 $A_{\mathcal{G}} = T_L\left[N-1\right]$ ;
36 return $A_{\mathcal{G}}$
```

B. Adjusting

After grouping, we can get different sets of slices. According to the hybrid isolation strategy mentioned in the last subsection, hard isolation is used between different groups. So we will discuss the adjusting method to optimize $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}\right)$ and $\mathcal{P}\left(\mathcal{E}^b_i\right), \forall \boldsymbol{s}_i \in \mathcal{G}_k$ in a single group.

In adjusting, we set b_s to be a dynamic variable and the main target is to find a better value of b_s. For the convenience of description, in this step, we describe $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}\right)$ as $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x_b\right)$ and treat it as a function of x_b, in which x_b represents the dynamically sharing bandwidth. The value of $\mathcal{P}\left(\mathcal{E}^b_{\boldsymbol{\mathcal{G}}}, x_b\right)$ represents the value of $\mathcal{P}\left(\mathcal{E}^b_{\boldsymbol{\mathcal{G}}}\right)$ in Eq. (17) when x_b is equal to b_s.

Based on the properties of normal distribution, $\mathcal{P}\left(\mathcal{E}^1_{\mathcal{G}_k-i}\right)$ and $\mathcal{P}\left(\mathcal{E}^2_{\mathcal{G}_k-i}\right)$ take the maximum value while $\sigma_i = \max(\boldsymbol{\sigma}_k), \boldsymbol{s}_i \in \mathcal{G}_k$. In our model, $\mathcal{P}\left(\mathcal{E}^1_i\right)$ is defined by l_i, which is given by the network tenant, and $\mathcal{P}\left(\mathcal{E}^b_i\right)$ is fixed. Based on Eq. (19), in order to make $\mathcal{P}\left(\mathcal{E}^b_i\right) \geq a_i$, the required value of $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x_b\right)$ can be got, and we mark it as $\mathcal{P}'\left(\mathcal{E}^b_{\mathcal{G}_k}, x_b\right)$, which should satisfy the equation shown in Eq. (33).

$$\begin{aligned}\mathcal{P}'\left(\mathcal{E}^b_{\mathcal{G}_k}, x_b\right) &= \min_{x_b}\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x_b\right)\\ &= \min_{x_b}\left\{\max_{s_i\in\mathcal{G}_k}\left\{\mathcal{P}\left(\mathcal{E}^b_i\right) - \mathcal{P}\left(\mathcal{E}^1_i\right)\mathcal{P}\left(\bar{\mathcal{E}}^b_{\mathcal{G}_k-i}, x_b\right)\right\}\right\}\\ &= \min_{x_b}\left\{\max_{s_i\in\mathcal{G}_k}\left\{a_k - l^l_i\mathcal{P}\left(\bar{\mathcal{E}}^b_{\mathcal{G}_k-i}, x_b\right)\right\}\right\}, \forall \boldsymbol{s}_i \in \mathcal{G}_k\end{aligned} \tag{33}$$

TABLE II
EXAMPLE OF RECURRENCE IN DYNAMIC PROGRAMMING ALGORITHM FOR GROUPING IN HSMB-G

i	$I_{min}[i]$	$I_{max}[i]$	$T_L[i]$
0	$\emptyset$	$\emptyset$	$\{\emptyset\}$
1	6	6	$\{\{\hat{\boldsymbol{s}}_0, \hat{\boldsymbol{s}}_1\}\}$
2	6	7	$\{\{\hat{\boldsymbol{s}}_0, \hat{\boldsymbol{s}}_1, \hat{\boldsymbol{s}}_2\}\}$
3	6	8	$\{\{\hat{\boldsymbol{s}}_0, \hat{\boldsymbol{s}}_1, \hat{\boldsymbol{s}}_2, \hat{\boldsymbol{s}}_3\}\}$
4	$\emptyset$	$\emptyset$	$\{\emptyset\}$
5	10	10	$BEST\left\{\begin{array}{l} \{T_L[1], \{\hat{\boldsymbol{s}}_2, \hat{\boldsymbol{s}}_3, \hat{\boldsymbol{s}}_4, \hat{\boldsymbol{s}}_5\}\} \\ \{T_L[2], \{\hat{\boldsymbol{s}}_3, \hat{\boldsymbol{s}}_4, \hat{\boldsymbol{s}}_5\}\} \\ \{T_L[3], \{\hat{\boldsymbol{s}}_4, \hat{\boldsymbol{s}}_5\}\} \end{array}\right\}$
6	10	11	$BEST\left\{\begin{array}{l} \{T_L[2], \{\hat{\boldsymbol{s}}_3, \hat{\boldsymbol{s}}_4, \hat{\boldsymbol{s}}_5, \hat{\boldsymbol{s}}_6\}\} \\ \{T_L[3], \{\hat{\boldsymbol{s}}_4, \hat{\boldsymbol{s}}_5, \hat{\boldsymbol{s}}_6\}\} \end{array}\right\}$
7	10	12	$\{L_T[3], \{\hat{\boldsymbol{s}}_4, \hat{\boldsymbol{s}}_5, \hat{\boldsymbol{s}}_6, \hat{\boldsymbol{s}}_7\}\}$
8	$\emptyset$	$\emptyset$	$\{\emptyset\}$
*9	14	14	$BEST\left\{\begin{array}{l} \{T_L[5], \{\hat{\boldsymbol{s}}_6, \hat{\boldsymbol{s}}_7, \hat{\boldsymbol{s}}_8, \hat{\boldsymbol{s}}_9\}\} \\ \{T_L[6], \{\hat{\boldsymbol{s}}_7, \hat{\boldsymbol{s}}_8, \hat{\boldsymbol{s}}_9\}\} \\ \{T_L[7], \{\hat{\boldsymbol{s}}_8, \hat{\boldsymbol{s}}_9\}\} \end{array}\right\}$

Meanwhile, the initial value of b_s of each group is the difference between the total resource allocated to each slice in the group according to a_k and the total resource allocated to each slice according to l_i, shown in Eq. (34).

$$b_s = \sum_{\boldsymbol{s}_i \in \mathcal{G}_k} \Phi^{-1}(a_k) * \sigma_i + \mu_i - \sum_{\boldsymbol{s}_i \in \mathcal{G}_k} \Phi^{-1}(l_i) * \sigma_i + \mu_i \quad (34)$$

According to Eq. (17) and Eq. (18), $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}\right)$ is a complex nonlinear concave function of b_s while $b_i + b_{si} \geq \mu_i$. So we design an iterative method to gradually optimize the value of $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}\right)$ and $\mathcal{P}\left(\mathcal{E}^b_i\right)$. Based on the properties of the concave function, we have the relationship shown in Eq. (35).

$$\frac{\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x^m_b\right) - \mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x'_b\right)}{\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x^n_b\right) - \mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x'_b\right)} \geq \frac{x^m_b}{x^n_b}, \quad (35)$$
$$\forall x'_b \geq \sum u_i + b_s, x^m_b \in [x_b, x^n_b], x^n_b > 0$$

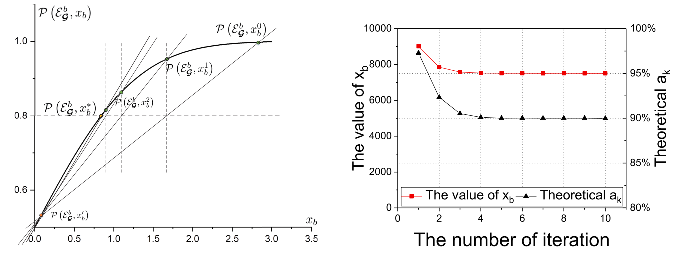
It is easy to conclude that when x_b is less than the optimal value x^*_b and as large as possible, the iteration can approach the optimal solution faster.

Let x'_b be the middle variable for iteration of x_b, our method is to adjust x'_b to get a sufficiently small $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}\right)$. Suppose $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x^m_b\right)$ is the result of $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x_b\right)$ after m-th iterations. At first, we initial the $x^0_b = b_s$ (b_s calculated by Eq. (34)), so $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x^0_b\right) = \mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, b_s\right)$, then we begin iteration according to Eq. (36), in which x^m_b means the value of x'_b after m times iteration.

$$x^m_b = \left(x^{m-1}_b - \hat{x_b}\right) * \frac{\mathcal{P}'\left(\mathcal{E}^b_{\mathcal{G}_k}, x^{m-1}_b\right) - \mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, \hat{x_b}\right)}{\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x^{m-1}_b\right) - \mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, \hat{x_b}\right)} + \hat{x_b} \quad (36)$$

Here is an example to illustrate the iteration. Assume that x^*_b is the ideal solution. And Fig. 3 shows a geometric illustration of an example iterative process. The value of each iteration is moving closer to the optimal value.

Considering that the CDF of multivariate normal variables needs to be calculated for each loop, in order to further reduce the number of loops, we, therefore, combined the above iterative algorithm with the Bisection algorithm [11]. Let $\hat{x_b}$ be a middle variable for the upper boundary of x_b in the Bisection algorithm, it can be calculated from the condition shown in Eq. (37).



(a) Iterative process (b) Iterative result

Fig. 3. An example of the iteration.

$$\hat{x_b} = max\left(\Phi^{-1}(a_k) * \|\boldsymbol{\sigma}_k\|_2 + \sum \mu_i - \sum b_i, 0\right) \quad (37)$$

Algorithm 2 shows the steps of the proposed iterative algorithm for Adjusting. Line 3 to Line 27 shows the pseudo-code of the iterative method . For more details, the termination condition of iteration is that either the number of iterations exceeds the preset value (I_n) or the probability error is less than the predetermined value (E_m). We define several temporary variables, which $\bar{x_b}$ stores the value of x_b in the Bisection algorithm, $\dot{x_b}$ indicates the value of x_b in the last loop, x'_b here is used to store the iterative value calculated by Eq. (36), C_s is used to indicates whether the iterative gap is less than B_δ, and C_s equals 1 indicates that the condition is satisfied (shown in Line. 12 to Line. 14), and the algorithm will set the iterative gap as B_δ while C_s equals 1 (shown in Line. 10). In each loop, while C_s equals 0, x_b will get the smaller one between $\bar{x_b}$ and x'_b (shown in Line .8). As analyzed above, if x_b gets the value of x'_b, $\mathcal{P}\left(\mathcal{E}^b_i, x_b\right)$ will always be higher than a_k. if $\mathcal{P}\left(\mathcal{E}^b_i, x_b\right)$ is smaller than a_k, there are 2 possible reasons: the first one is x_b gets the value of $\bar{x_b}$, and in this case, we handle the current value of x_b to be the start point of the optimization area with updating $\hat{x_b}$, to make our optimization area narrower (shown in Line. 19). The second one is the fluctuations in calculating $\mathcal{P}\left(\mathcal{E}^b_i, x_b\right)$. Since the value of $\mathcal{P}\left(\mathcal{E}^b_i, x_b\right)$ is not strictly exact, we define a very small bias ϵ, and add it to $\mathcal{P}\left(\mathcal{E}^b_i, x_b\right)$ (shown in Line. 17). After each iteration, let $\dot{x_b}$ save the value of x_b and then start the next loop. Finally, the result is saved in the list L_{b_s}.

Moreover, Eq. (38) ensures the total required resources of each group is the integer multiple of B_δ.

$$x_b = \lceil\left(\sum\nolimits_{\boldsymbol{s}_i \in \mathcal{G}} b^i_d + \hat{x_b}\right) / B_\delta\rceil * B_\delta \quad (38)$$

As mentioned, the final required minimum value of b_s of each group can be calculated. Together with b_i of each slice, we can get the total minimum resources B^k_d which need to allocate to $\mathcal{G}_k$.

C. Linking

According to the previous analysis, after acquiring the prediction result of the slices, the tenant first groups them

Algorithm 2: Cyclic Iterative Algorithm for Adjusting in HSMB (HSMB-A).

```
Data: The grouping list $A_{\mathcal{G}}$ of $\boldsymbol{S}$ and $\boldsymbol{s}_i \in \boldsymbol{S}$
Result: The list $L_{b_s}$ of value of $b_s$ for each $\mathcal{G} \in \boldsymbol{S}$ and $P_{\mathcal{G}_k}$
foreach $\mathcal{G} \in \boldsymbol{\mathcal{G}}$ do
    Initialization: $i = 0$,$L_{b_s} = \{0\} * K$, $x_b = b_s$, $\hat{x_b}$, $\dot{x_b} = b_s$,$C_s = 0$, $x'_b = 0$, $\bar{x_b} = 0$, $\mathcal{P}'\left(\mathcal{E}^b_{\mathcal{G}}, x_b\right)$ with Eq. (33), $\mathcal{P}\left(\mathcal{E}^b_i, x_b\right)$; Input $I_n$, $E_m$ ;
    while $i \leq I_n$ or $\min \mathcal{P}\left(\mathcal{E}^b_i, x_b\right) - a_k \geq E_m$ do
        $i \leftarrow i + 1$ ;
        if $C_s$ equals 0 then
            Update $x'_b$ using Eq. (36);
            Update $\bar{x_b} = 1/2 * (x_b + \hat{x_b})$ ;
            $x_b \leftarrow \min\left(x'_b, \bar{x_b}\right)$
        else
            $x_b = x_b - B_\delta$
        end
        if $\dot{x_b} - x_b < B_\delta$ then
            $C_s \leftarrow 1$ ;
            Update $x_b$ using Eq. (38) ;
        end
        Calculate $\mathcal{P}\left(\mathcal{E}^b_i, x_b\right)$ using Eq. (19);
        if $\min \mathcal{P}\left(\mathcal{E}^b_i, x_b\right) < a_k - \epsilon$ then
            if $x_b == \bar{x_b}$ then
                Set $\hat{x_b} \leftarrow x_b$, $x_b \leftarrow \dot{x_b}$
            end
            if $C_s == 1$ then
                Break;
            end
        else
            $\dot{x_b} \leftarrow x_b$
        end
    end
    if $\min \mathcal{P}\left(\mathcal{E}^b_i, x_b\right) < a_k$ then
        $x_b \leftarrow x_b + B_\delta$
    end
    Update $x_b$ using Eq. (38) ;
    $L_{b_s}.add\left(x_b\right)$ ;
end
return $P_{\mathcal{G}_i}$, $L_{b_s}$
```

according to Algorithm 1. The proposed grouping algorithm makes the constraints of Eq. (3), Eq. (12), and (22) hold, and then calculates b_s by Algorithm 2 to realize the constraints of Eq. (14), (20), and (23). However, there are two problems still on top of solving OP1, The first problem is that Algorithm 2 still has more integral operations and requires more computation time, which we call the computational dilemma. And the second problem which we call the resource dilemma, which caused by the constraint in Eq. (21). We address these two dilemmas in this subsection and proposed Algorithm 3 to link the grouping algorithm and adjustment algorithm together to form the final solution of HSMB.

1) Resource Dilemma: Since the relation between B^k_d and input parameters is nonlinear, it is difficult to decide how many NSs should be refused when the total required bandwidth is more than what can be provided. Assume that the excess required bandwidth is ΔB. First, we choose the lowest priority grouping $\hat{\mathcal{G}}$ to check if the total resource $\hat{B_d}$ is larger than the ΔB, and adjust the $\boldsymbol{\mathcal{G}}$ as Eq. (39) shows.

$$\boldsymbol{\mathcal{G}} = \begin{cases} \boldsymbol{\mathcal{G}} & \hat{B_d} > \Delta B, \\ \boldsymbol{\mathcal{G}} - \hat{\mathcal{G}} & \text{else.} \end{cases} \tag{39}$$

The iteration runs through Eq. (39) until the number of groups stops decreasing. And then we check the slices in the new $\hat{\mathcal{G}}$. The strategy within the group is more flexible, it can be determined by the SLA of the slices, either by traditional methods (best-effort, DiffServ, etc.), or simply rejecting some of the slices. If the latter is used, Eq. (40) can be used to get the set of refused slices, in which $\mathcal{G}_f$ represent the subset of refused slices in $\mathcal{G}$.

$$\sum\nolimits_{\boldsymbol{s}_i \in \mathcal{G}_f} \left(b_i + b_{si}\right) \geq \Delta B \tag{40}$$

The pseudo-code in Line 27 to Line 33 of Algorithm 3 shows the operation of dealing with resource dilemma.

2) Computational Dilemma: Although we reduce the computational complexity greatly by HSMB-G and HSMB-A, the computational complexity of Algorithm 2 is still relatively high if the number of slices within one group is large. Here we explore further methods to reduce the computational complexity.

In the proposed FHIM, it can be found that the value of b_i is related to the l_i, while the value of b_s is only related to the prediction error $\boldsymbol{\sigma}_k$, and the computational complexity is mainly contributed by calculating b_s. As mentioned in Section III, the value of τ is generally taken to be relatively small, which also means that the mean and variance of the predicted NSs change more slowly, this also means that the results of the grouping rarely change in a short period. Meanwhile, the presence of B_δ allows us to have some bandwidth margin to accept small changes in the predicted error on the traffic of the slice. Therefore, at time period Δt, we can calculate the required bandwidth based on the calculated result in time period $\Delta(t-1)$. Therefore, we first calculate the change of dedicated bandwidth $\Delta {B_d}^t$, shown in Eq. (41). Then we calculated the $\mathcal{P}\left(\mathcal{E}^b_{\mathcal{G}_k}, x_b\right)$ by Eq. (19) with $x_b = \left\lceil \left(B^k_d\left(t-1\right) + \Delta {B_d}^t\right) / B_\delta \right\rceil * B_\delta$. Finally, we iteratively adjust x_b in step B_δ to calculate the bandwidth $B^k_d\left(t\right)$.

Algorithm. 3 shows the final linking algorithm (HSMB-L). We use I_d to indicate whether the resource dilemma happens, which I_d equals 1 means the resource dilemma does not happen and otherwise 0. So we set I_d to 0 to calculate the required resources first. After grouping, if one group exists in the last period, we start our algorithm for dealing with computational dilemma (from Line. 6 to Line. 25), that we adjust $b^k_s\left(t\right)$ based on $b^k_s\left(t-1\right)$, and adjust them with the gap B_δ (form Line. 11 to Line. 19). If a new group exists, then startup our proposed Algorithm. 2 to calculate the new b_s (Line. 5). After obtaining the whole resource requirement of resources, Line. 27 to Line. 33 deal with resource dilemma if resource dilemma happens, otherwise set I_d to 1 and break the while-loop. And return the final resource configuration.

Algorithm 3: Linking Algorithm for Linking in HSMB (HSMB-L).

```
Data: The grouping list $A_{\mathcal{G}}(t-1)$, $\boldsymbol{s}_i \in \boldsymbol{S}$, $\mathcal{G}$, $E_m$,
      $P_{\mathcal{G}_k}(t-1)$, $L_{b_s}(t-1)$, $P_t$, $\boldsymbol{D}$, $a_k$, $h_k$, $B_\delta$ and
      $I_d = 0$
Result: Slice set $\boldsymbol{\mathcal{G}}$ and list of $b_s^k, \mathcal{G}_k \in \boldsymbol{\mathcal{G}}$
while $I_d$ equals 0 do
    Grouping with Algorithm 1 ;
    foreach $\mathcal{G}_k \in A_{\mathcal{G}}(t)$ do
        if $\mathcal{G}_k \notin A_{\mathcal{G}}(t-1)$ then
            Calculate $P_{\mathcal{G}_k}$ and $L_{b_s}$ with Algorithm 2;
        else
            Calculate $\Delta B_d(t)$ with Eq. (41) ;
            $x_b \leftarrow b_s^k(t-1)$ ;
            Calculate $\mathcal{P}\left(\mathcal{E}_i^b, x_b\right)$ using Eq. (19);
            $P_t \leftarrow \min \mathcal{P}\left(\mathcal{E}_i^b, x_b\right)$ ;
            while Both $P_t$ and $\min \mathcal{P}\left(\mathcal{E}_i^b, x_b\right)$ larger
             or smaller than $a_k + E_m$ do
                if $P_t - a_k \leq E_m$ then
                    $x_b \leftarrow x_b + B_\delta$
                end
                if $P_t - a_k \geq E_m$ then
                    $x_b \leftarrow x_b - B_\delta$
                end
                Calculate $\mathcal{P}\left(\mathcal{E}_i^b, x_b\right)$ using Eq. (19);
            end
            if
             $P_t - a_k \leq E_m \& \min \mathcal{P}\left(\mathcal{E}_i^b, x_b\right) - a_k \geq E_m$
             then
                $L_{b_s}.add(x_b)$ ;
            else
                $L_{b_s}.add(x_b + B_\delta)$ ;
            end
        end
    end
    if $\sum_{\mathcal{G}_k \in \boldsymbol{\mathcal{G}}} B_d^k > \mathcal{B}_e$ then
        Iterate Eq. (39) until $|\boldsymbol{\mathcal{G}}|$ stop decreasing ;
        Solve the $\mathcal{G}_f$ with the constrain Eq. (40) ;
        $\boldsymbol{S} = \bigcup \boldsymbol{\mathcal{G}} - \mathcal{G}_f$ ;
    else
        $I_d \leftarrow 1$
    end
end
return $\boldsymbol{\mathcal{G}}, L_{b_s}$
```

$$\Delta B_d(t) = \sum\nolimits_{\boldsymbol{s}_i \in \mathcal{G}_k} \left(b_i(t) - b_i(t-1)\right) \tag{41}$$

V. Numerical results

In this section, we conduct numerical simulations to estimate the performance of the proposed algorithms. We first evaluate the performance of the algorithm under different influencing factors and then compare the performance with benchmark algorithms. Table III summarizes the parameters we defined in the evaluation, and Table IV shows the value of the parameters of FHIM we used in each evaluation. The traffic data and prediction model for evaluation adopt our previous traffic prediction research [19]. In addition, we use the *Scipy* module in *Python*3.10 to calculate the CDF of normal distribution in Algorithm 2, which does not affect the performance of our proposed algorithms.

TABLE III
Definition of metrics in evaluation

Symbol	*Description*
e^r	Historical prediction relative absolute error of the slices, in which $e^r(t)$ means the e^r on time Δt.
η	Network resource utilization, as the ratio of actual traffic $\mathcal{Y}$ to allocated bandwidth B_d
η_b	Network resource utilization of b_s.
γ_k	The reduce ratio of the total distributed bandwidth with different l_i to the case $l_i = 100\%$.
r_d	The ratio that the same group in adjacent time-separated groupings and the total number of groupings

TABLE IV
Table of Parameters in Simulation

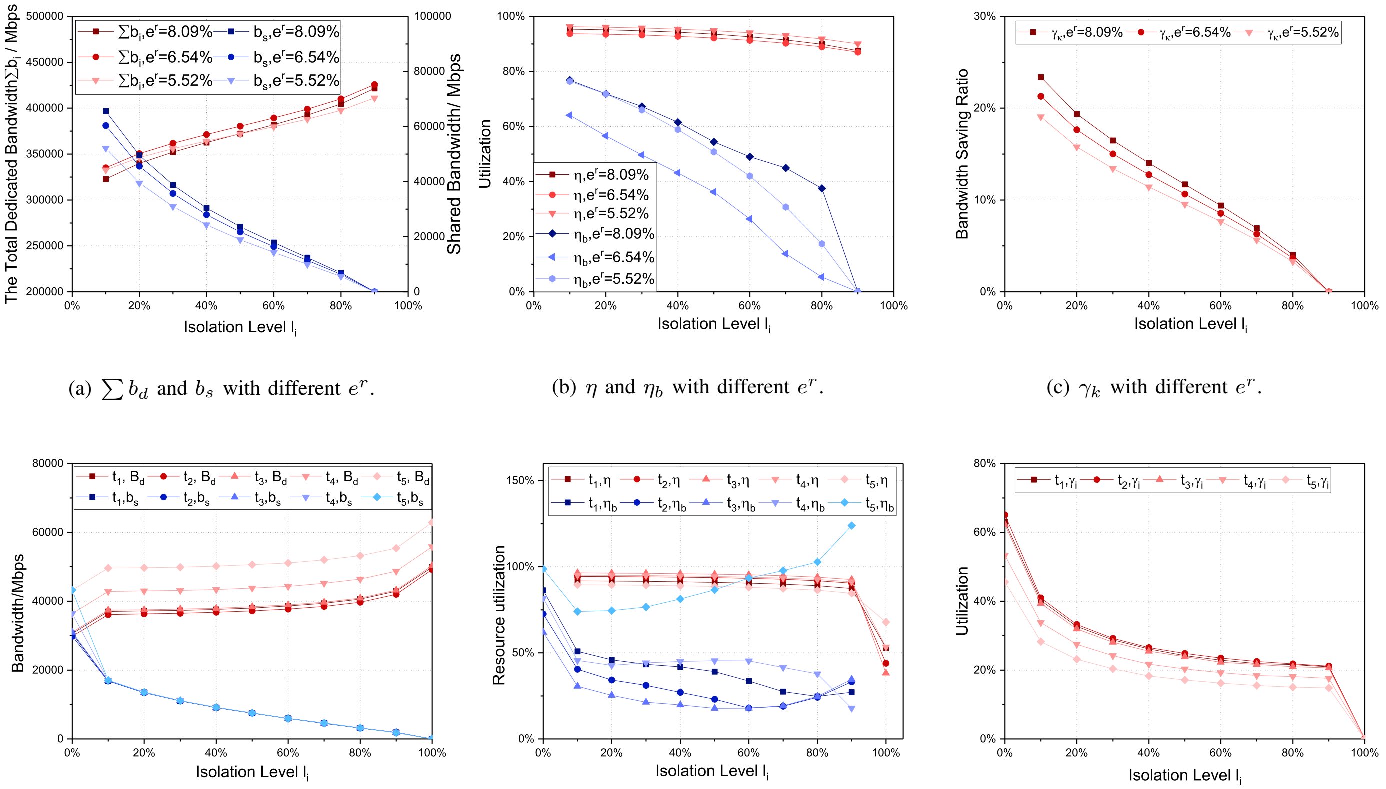
Evaluation types	Parameters	Value
Generals	a_k	99%
	B_δ	100Mbps
	I_n	20
	E_m	0.001
Prediction Error e^r	τ	0.02
	l_i	10% - 90% step by 10%
	h_k	15
	s_k	15
Isolation level l_i	τ	0.02
	l_i	10% - 90% step by 10%
	h_k	10
	s_k	10
Hybrid degree h_k	τ	0.02
	l_i	60%
	h_k	5-10 step by 1
	s_k	50
Hybrid isolation sensitivity τ	τ	$\frac{1}{T}$,T:10-90 step by 10
	l_i	10% - 90% step by 10%
	h_k	10
	s_k	10
Comparative analysis	τ	0.02
	l_i	50%
	h_k	7
	s_k	50
Comprehensive analysis	τ	0.02
	l_i	20% to 80% step by 20%
	h_k	4,6,8,10
	s_k	50

A. Algorithm evaluation

We analyze our algorithm in five aspects, which are: prediction error e^r, isolation level l_i, hybrid degree h_k, hybrid isolation sensitivity τ and a comprehensive analysis based on l_i and h_k.

As defined, η indicates the matching between allocated resources and the actually used resources, if η exceeds 1, it indicates that there are some slices that have been degraded. Meanwhile, η_b represents the resource utilization of b_s.

1) Prediction error e^r: To understand the performance of the algorithm with different e^r, we adopt our proposed neural network prediction method [19] with different parameters to do prediction to general different prediction errors on the same slices. Let e_i^r represent the predicted error of $\boldsymbol{s}_i$, it can be easily



(a) $\sum b_d$ and b_s with different e^r.

(b) η and η_b with different e^r.

(c) γ_k with different e^r.

(d) B_d and b_s with different t.

(e) η and η_b with different t.

(f) γ_k with different t.

Fig. 4. The indicators in different times and isolation levels (l_i).

found that the larger e_i^r, the larger σ_i, so e_i^rr and σ_i in our model are positively correlated. With different e^r, we use the algorithm proposed in this paper to compare the performance. The simulation results are got by averaging the results for 5 consecutive periods.

Fig. 4(a) illustrates the sum of dedicated bandwidth $\sum_{\boldsymbol{s}_i \in \mathcal{G}_k} b_i$ and the shared bandwidth b_s with $e^r = 8.09\%$, 6.54% and 5.52% respectively. According to Fig. 4(a), the sum of dedicated bandwidth has no obvious trends with different e^r, while b_s is influenced by the change of e^r, which is more e^r, more b_s. This emphasizes our previous analysis that b_s is related to e^r, which also illustrates the fact that the accuracy of the prediction is still very important, and the higher the accuracy of the prediction, the better the results can be derived under our proposed algorithm.

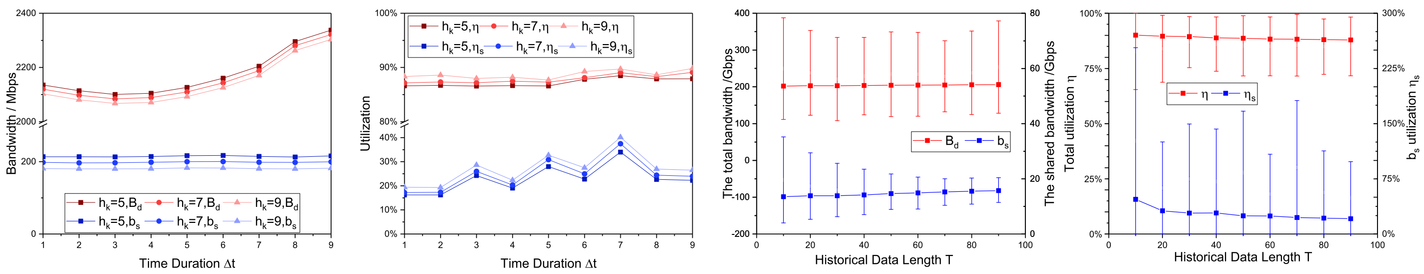
Fig. 4(b) illustrates the η and η_b with different l_i, it shows that η is similar to l_i, while η_b also does not have obvious regularity, this is because η_b is influenced by predicting traffic and actual traffic together, and e^r is the absolute error, the actual traffic of slices may be higher or lower than the predicted traffic.

In addition, let γ_k represent the ratio of saving bandwidth in $\mathcal{G}_k$ while adopt different l_i compared to the case $l_i = 100\%$. Fig. 4(c) illustrates γ_k with the 3 corresponding prediction errors, it shows that our algorithm can perform a better auxiliary reference when the prediction error is not well. This also shows that our algorithm cannot exist independently of the prediction algorithm, and it can well correct the mismatch in resource provisioning due to prediction errors.

2) Isolation level l_i: According to Table IV, all 10 slices are in the same group because the isolation level only affects the resource provisioning within the group. We randomly select a time interval for traffic prediction and bandwidth allocation and set l_i from 0% to 100%. The simulation results are shown in Fig. 4(d) Fig. 4(e) and Fig. 4(f).

Based on the results shown in Fig. 4(d), we can conclude that, as l_i increases, the dedicated bandwidth for slices gradually increases, b_s gradually decreases, and the total allocated bandwidth is gradually increasing. One thing to note is that the predicted and actual bandwidth of slices keep changing with l_i changes, but b_s changes minimally, and it is only when l_i approaches 0% that the shared bandwidth at different time periods changes significantly. This can also justify the feasible of the policy for the computational dilemma in Algorithm 3, at adjacent time duration Δt, there only needs to be reallocated bandwidth for the corresponding dedicated bandwidth with little iterative adjustment of b_s in most times.

Fig. 4(e) illustrates the η and η_b. It should be noted that if η_b reaches 100%, it means that there are already some slices being blocked in the group. Overall, as l_i increases, the η gradually decreases. It is important to note that in practice the case of utilization greater than 100% does not exist, but in Fig. 4(e) we use the ratio of actual bandwidth to allocated bandwidth to show utilization, such that when utilization is



(a) B_d and b_s with different h_k.

(b) η and η_b with different h_k.

(c) B_d and b_s with different T

(d) η and η_b with different T

Fig. 5. The indicators in different times Δ_t or historical data length T.

greater than 100%, higher utilization indicates the level of network congestion. At time periods $t_1 - t_4$, no congestion occurs within the group, and at period t_5, congestion event occurs. η_b is none while l_i is 100% because b_s is 0 while $l_i = 100\%$. If no congestion occurs, η gradually decreases as l_i increases, which is because $\sum b_i$ turns large and more probability that resources will waste, and B_d increase as the l_i increases, causing η gradually decrease. And when congestion occurs, the actual traffic of some slices is much larger than the distributed bandwidth. with the increasing l_i, the reduction of b_s reduces the capacity of the overflowing traffic of these slices and increases congestion. However, it is important to note that a lower l_i can lead to congestion in other slices within the group whose actual traffic is greater than the allocated dedicated bandwidth. For example, at moments t_5, the overall congestion gradually improves as the l_i increases, but the probability that other slices will be affected is reduced. Therefore, when congestion does not occur, a smaller l_i can effectively improve the resource utilization, and when congestion occurs, a smaller l_i can alleviate the congestion, but it will bring the risk of congestion spreading (slice degraded in one slice make other slices in the group degraded).

As shown in Fig. 4(f), γ_k decreases as l_i increases. The γ_k can even reach 40.19% (t_1) and 28.24% (t_5) while $l_i = 10\%$. Therefore, γ_k can take values between 0% and these values at different l_is, so a right l_i can ensure isolation while effectively reducing allocated bandwidth.

Hence, in the efficiency-first slice type, l_i can be set lower, and in the availability-first slice type, a lower l_i will bring greater instability to the availability, so it is necessary to make careful trade-offs and set the appropriate l_i according to the slice requirements when setting this indicator.

3) Hybrid degree h_k: In this part. we generate 50 NSs. and set h_k from 5 to 10 respectively. According to the previous analysis, we can derive that when $h_k = 1$, it is also equivalent to no shared bandwidth b_s between the NSs, and a hard isolation policy is completely used between them. As h_k increases, more and more slices in the group use shared bandwidth, which is similar to reducing l_i. In this part, we used a fixed value of isolation level (set $l_i = 60\%$), and mainly focused on the allocated bandwidth and bandwidth utilization with different h_k and the continuous Δ_t.

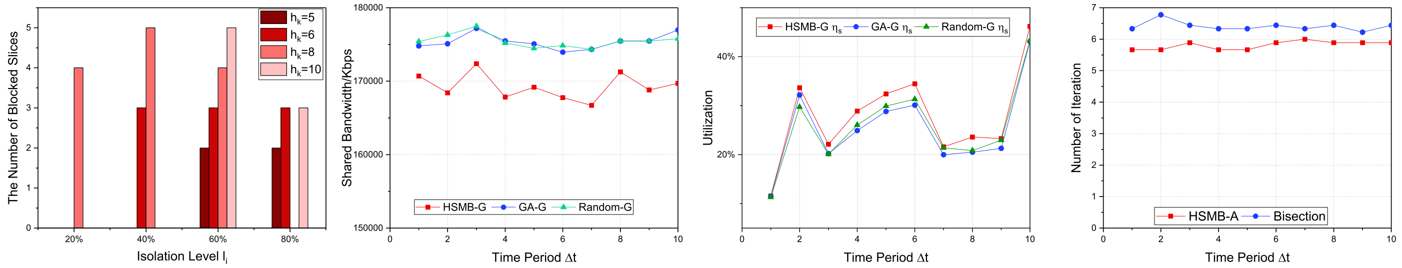
For better illustration, we have drawn only the values while h_k = 5, 7 and 9, and the results of simulation are shown in Fig. 5(a) and Fig. 5(b). As mentioned above, along with the increase of h_k, the total resource B_d, sharing bandwidth $\sum b_s$ are decreasing, and η,η_b are increasing.

Furthermore, according to Fig. 5(a), it can be seen that the total b_s is relatively stable, which means that the change of b_s values in adjacent time periods is not significant, and our proposed Algorithm. 3 is more meaningful in the computational dilemma.

4) Hybrid isolation sensitivity τ: As mentioned above, τ determines the rate of change of the statistical value of the prediction error and therefore affects the grouping in adjacent time periods. Since $\tau = \frac{1}{T}$, we get different τ by changing T, and set T from 10 to 90 step by 10. We track 20 consecutive time periods, the statistical results are shown in Fig. 5(c) and Fig. 5(d), in which the lines show the mean values of the indicators, while the error distribution lines (vertical lines) show the maximum and minimum values of the indicators at different T.

In Fig. 5(c), the changes of both the mean of B_d and mean of b_s are stable, because we use the same slices with different T, which cause a small difference between the overall error of the slices. However, it can be seen by the error range that there is no obvious pattern in the fluctuation range of B_d, but the fluctuation range of b_s decreases as T increases, this is because the scope of the changes on σ_s of the NSs used to participate in the resource calculation is slowly decreasing as T increases. In Fig. 5(d), the changes in the mean of η are stable, while the changes in the mean of η_b decrease as T increases. In terms of the fluctuation range, η_b has a larger fluctuation range than η. Both Fig. 5(c) and Fig. 5(d) show that at the very beginning when T increases, all indicators become better, but when T reaches a threshold, the trend of indicators getting better slows down or starts to fluctuate. It can be concluded that as T increases, the historical data works well as a guide at first, but when T becomes larger, too much historical data reduces the information sensitivity to the new data, which leads to the indicators starting to fluctuate. Therefore, the value of T needs to be calculated with comprehensive consideration.

Meanwhile, let r_d denote the ratio that the number of the same group in the adjacent time-separated grouping sets and the total number of grouping sets. It can be imagined that with the increases in T, r_d will show an increasing trend. Table.



(a) Number of blocked slices with different l_i and h_k.

(b) b_s with different time period Δt.

(c) η_s with different time period Δt.

(d) Mean Iterative times with different time period Δt

Fig. 6. The indicators with different parameters.

TABLE V
r_d WITH DIFFERENT T

T	0	10	20	30	40
r_d	-	0.53%	1.58%	3.68%	3.16%
T	50	60	70	80	90
r_d	3.68%	5.79%	5.79%	7.37%	4.74%

V shows the values of r_d with different T in our simulation. Despite some fluctuations of r_d, the overall trend of r_d is increasing with T.

5) Comprehensive analysis: We focus on how our proposed model inflects the number of degraded slices, with the parameters in Table IV. The results are illustrated in Fig. 6(a).

As analyzed above, a higher l_i means a larger b_i and B_d, but smaller b_s. Along with the increase in l_i, the number of blocked slices increases first and then decreases, and the turning point occurs at different locations with different h_k, which represents different traffic prediction accuracy of slices. While l_i gets a smaller value, there are a larger b_s and suit for slices with small prediction error, while l_i is larger, a smaller b_s is suited for unstable prediction error, which limits the error slice to occupy much more b_s to lead to degrading on other slices.

Meanwhile, different h_k causes different numbers of degraded slices, that a higher h_k means an abnormal slice can affect more slices, but also can be allocated larger b_s. Although a larger h_k can use a smaller B_d, it is important to make confirmation that the accuracy of the prediction will not output large deviations.

Based on the different prediction results, the best value of l_i and h_k is related to the slices, So they can be set for better B_d as well as the number of blocked slices based on experience.

Based on the analysis above, our proposed algorithm can be adapted to different scenarios by different parameter settings among the slice resource provisioning problems, and make our proposed FHIM model more flexible and customizable.

In the applications of vertical industrial, different network slices can be assigned different hybrid model parameters based on the SLA of the services in the slices, thus achieving a balance of resource utilization and sliced QoS. Taking the smart grid mentioned in Section I as an example again, higher l_i and smaller h_k can be selected when dealing with important small traffic services such as load control and distribution, while lower l_i and larger h_k can be selected when dealing with the services such as inspection and operation.

B. Comparative analysis

In this section, we compare our proposed algorithm with other benchmarks. Since we first proposed the solution to the HSMB problem for hybrid isolation based on our best survey, we refer to some of the previous studies that have solved the same sub-problem of HSMB. Therefore, we evaluate our proposed algorithm based on the solutions to the sub-problems.

1) Grouping: For the grouping problem in HSMB, in order to verify the performance of our grouping algorithm, we choose two benchmarks: randomly generated grouping (Random-G) and genetic algorithm-based grouping (GA-G) (the initialized populations we set is 30, and the number of iterations is 30). We use the proposed adjusting algorithm and linking algorithm for all benchmarks to calculate the final provisioned resources. To better evaluate the differences in the algorithms, we selected the same slice samples and the same time period Δt. According to Theorem. 3, we have obtained the optimal grouping size (the number of slices in one slicing group) $N_{\mathcal{G}_k}$, the benchmarks use the same group size as HSMB-G, as well as both HSMB-A and HSMB-L for subsequent processing. The numerical simulation is used to prove that our proposed HBSM-G algorithm is the optimal solution for grouping. We simulate 10 consecutive time periods, and the results are illustrated in Fig. 6(b) and Fig. 6(c).

Fig. 6(b) illustrates the total b_s of the algorithms, we only focus on b_s because the same slices and parameters used in these algorithms, and the result of total dedicated bandwidth are the same. We can find that the HSMB-G algorithm can get the lowest b_s, while GA-G and Random-G are not very different, It shows that for the overall HSMB solution the grouped results have more local optimization solutions, thus making it difficult for the general heuristic algorithm to obtain the optimal grouping results. Also, it can be seen that our proposed HSMB-G algorithm can achieve a better grouping result. Fig. 6(c) illustrates the utility of the tenant network in different grouping algorithms, the η_s of HSMB-G outperforms the other algorithms for most of the time periods, while the two benchmarks, GA-G and Random-G, mutual highs and lows at different time periods. To clarify indicating the results, Table VI shows the η_s obtained by the three algorithms at time periods 1,7, and 9 (the results are overlaid in these periods).

TABLE VI
η_s WITH DIFFERENT ALGORITHMS AT DIFFERENT TIME PERIODS

Algorithms	HSMB-G	GA-G	Random-G
$\Delta t = 1$	11.59%	11.48%	11.31%
$\Delta t = 7$	21.60%	20.00%	21.37%
$\Delta t = 9$	23.28%	21.29%	22.95%

Overall, using different algorithms, η_s has the same trend and they have similar results, but HSMB-G always outperforms the comparison algorithm. The numerical simulation results shown in Fig. 6(c) prove that our proposed HSMB-G algorithm provides the optimal grouping results.

2) Adjustment: The same sub-problem was encountered in the [11] in solving a similar problem and they proposed to use the bisection search method for optimization, therefore, here we use the Bisection algorithm as a benchmark algorithm for evaluation. And we evaluate the two algorithms in terms of their convergence speed. We use the HSMB-G algorithm for grouping, and compare the HSMB-A algorithm with the Bisection algorithm, the results are shown in Fig. 6(d).

The mean number of iterations of our proposed HSMB-A algorithm is 5.81 while the Bisection algorithm is 6.41, and the iteration times of HSMB-A are lower than Bisection in every time period. Fig. 6(d) illustrates the superiority of our proposed HSMB-A to the Bisection algorithm.

VI. Conclusion

This paper solves the probabilistic assured resource provisioning problem with customized resource isolation for vertical industrial slices. The proposed FHIM model realizes flexible and customized resource isolation through a series of network parameters. Meanwhile, we formulate the HSMB as a nonlinear programming problem for efficient resource provisioning in FHIM.

We divide the HSMB problem into two sub-problems: the HSMB-G algorithm to perform grouping by probabilistic-based analysis, the HSMB-A algorithm for a faster iteration than the general method. Furthermore, facing the potential resource dilemma and computational dilemma, the solution (HSMB-L) of the HSMB is proposed to get the minimum resources with ensuring the resource provisioning in a theoretical probabilistic way.

The simulation results prove the flexibility and customizability in resource isolation of FHIM under probabilistic-assured resource provisioning. Since the resource provisioning problem with FHIM is first proposed, we compare sub-algorithms by genetic algorithms grouping, random grouping, and Bisection algorithm, respectively. The results show that our proposed algorithm derives the minimum resource consumption.

References

[1] S. Wijethilaka and M. Liyanage, "Survey on Network Slicing for Internet of Things Realization in 5G Networks," in IEEE Communications Surveys & Tutorials, vol. 23, no. 2, pp. 957-994, Secondquarter 2021, doi: 10.1109/COMST.2021.3067807.

[2] X. Li et al., "5Growth: An End-to-End Service Platform for Automated Deployment and Management of Vertical Services over 5G Networks," in IEEE Communications Magazine, vol. 59, no. 3, pp. 84-90, March 2021, doi: 10.1109/MCOM.001.2000730.

[3] J. Feng, Q. Pei, F. R. Yu, X. Chu, J. Du and L. Zhu, "Dynamic Network Slicing and Resource Allocation in Mobile Edge Computing Systems," in IEEE Transactions on Vehicular Technology, vol. 69, no. 7, pp. 7863-7878, July 2020, doi: 10.1109/TVT.2020.2992607.

[4] X. Shen, J. Gao, W. Wu, M. Li, C. Zhou and W. Zhuang, "Holistic Network Virtualization and Pervasive Network Intelligence for 6G," in IEEE Communications Surveys & Tutorials, vol. 24, no. 1, pp. 1-30, Firstquarter 2022, doi: 10.1109/COMST.2021.3135829.

[5] W. Guan, H. Zhang and V. C. M. Leung, "Customized Slicing for 6G: Enforcing Artificial Intelligence on Resource Management," in IEEE Network, vol. 35, no. 5, pp. 264-271, September/October 2021, doi: 10.1109/MNET.011.2000644.

[6] P. Rost et al., "Customized Industrial Networks: Network Slicing Trial at Hamburg Seaport," in IEEE Wireless Communications, vol. 25, no. 5, pp. 48-55, October 2018, doi: 10.1109/MWC.2018.1800045.

[7] R. Liu, X. Hai, S. Du, L. Zeng, J. Bai and J. Liu, "Application of 5G network slicing technology in smart grid," 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), 2021, pp. 740-743, doi: 10.1109/ICBAIE52039.2021.9389979.

[8] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq and H. Abou-Zeid, "Intelligent Radio Access Network Slicing for Service Provisioning in 6G: A Hierarchical Deep Reinforcement Learning Approach," in IEEE Transactions on Communications, vol. 69, no. 9, pp. 6063-6078, Sept. 2021, doi: 10.1109/TCOMM.2021.3090423.

[9] H. Chergui et al., "Zero-Touch AI-Driven Distributed Management for Energy-Efficient 6G Massive Network Slicing," in IEEE Network, vol. 35, no. 6, pp. 43-49, November/December 2021, doi: 10.1109/MNET.111.2100322.

[10] M. R. Raza, M. Fiorani, A. Rostami, P. Öhlen, L. Wosinska and P. Monti, "Dynamic slicing approach for multi-tenant 5G transport networks [invited]," in Journal of Optical Communications and Networking, vol. 10, no. 1, pp. A77-A90, Jan. 2018, doi: 10.1364/JOCN.10.000A77.

[11] Q. -T. Luu, S. Kerboeuf and M. Kieffer, "Uncertainty-Aware Resource Provisioning for Network Slicing," in IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 79-93, March 2021, doi: 10.1109/TNSM.2021.3058139.

[12] S. Kukliński, L. Tomaszewski, R. Kołakowski and P. Chemouil, "6G-LEGO: A framework for 6G network slices," in Journal of Communications and Networks, vol. 23, no. 6, pp. 442-453, Dec. 2021, doi: 10.23919/JCN.2021.000025.

[13] N. Huin et al., "Hard-isolation for Network Slicing," IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2019, pp. 955-956, doi: 10.1109/INFCOMW.2019.8845282.

[14] A. Alcalá et al., "Multi-layer Transport Network Slicing with Hard and Soft Isolation," 2021 Optical Fiber Communications Conference and Exhibition (OFC), 2021, pp. 1-3.

[15] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini and H. Flinck, "Network Slicing and Softwarization: A Survey on Principles, Enabling Technologies, and Solutions," in IEEE Communications Surveys & Tutorials, vol. 20, no. 3, pp. 2429-2453, thirdquarter 2018, doi: 10.1109/COMST.2018.2815638.

[16] Y. Ji, et al. "Artificial intelligence-driven autonomous optical networks: 3S architecture and key technologies." Science China Information Sciences, 63.6 (2020): 1-24.

[17] R. Gu, Z. Yang, Y. Ji, "Machine learning for intelligent optical networks: A comprehensive survey," Journal of Network and Computer Applications, 2020, 157: 102576.

[18] P. Cortez, M. Rio, M.Rocha, P.Sousa, "Multi-scale Internet traffic forecasting using neural networks and time series methods." Expert Syst. J. Knowl. Eng. 29 (2012): 143-155.

[19] Q. Guo et al., "Proactive Dynamic Network Slicing with Deep Learning Based Short-Term Traffic Prediction for 5G Transport Network," 2019 Optical Fiber Communications Conference and Exhibition (OFC), 2019, pp. 1-3.

[20] Z. Li, R. Gu, L. Wang and Y. Ji, ”Computing-aware Proactive Network Reconfiguration for Optical Networks Interconnected Edge Computing System,” 2021 Optical Fiber Communications Conference and Exhibition (OFC), 2021, pp. 1-3.

[21] D. Ferreira, A. Braga Reis, C. Senna and S. Sargento, ”A Forecasting Approach to Improve Control and Management for 5G Networks,” in IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1817-1831, June 2021, doi: 10.1109/TNSM.2021.3056222.

[22] F. Wei, G. Feng, Y. Sun, Y. Wang, S. Qin and Y. -C. Liang, ”Network Slice Reconfiguration by Exploiting Deep Reinforcement Learning With Large Action Space,” in IEEE Transactions on Network and Service Management, vol. 17, no. 4, pp. 2197-2211, Dec. 2020, doi: 10.1109/TNSM.2020.3019248.

[23] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia and A. Banchs, ”Mobile traffic forecasting for maximizing 5G network slicing resource utilization,” IEEE INFOCOM 2017 - IEEE Conference on Computer Communications, 2017, pp. 1-9, doi: 10.1109/INFOCOM.2017.8057230.

[24] H. Yu, F. Musumeci, J. Zhang, M. Tornatore, L. Bai and Y. Ji, ”Dynamic 5G RAN slice adjustment and migration based on traffic prediction in WDM metro-aggregation networks,” in Journal of Optical Communications and Networking, vol. 12, no. 12, pp. 403-413, December 2020, doi: 10.1364/JOCN.403829.

[25] D. Bega, M. Gramaglia, M. Fiore, A. Banchs and X. Costa-Pérez, ”DeepCog: Optimizing Resource Provisioning in Network Slicing With AI-Based Capacity Forecasting,” in IEEE Journal on Selected Areas in Communications, vol. 38, no. 2, pp. 361-376, Feb. 2020, doi: 10.1109/JSAC.2019.2959245.

[26] N. Reyhanian and B. Maham, ”Statistical Slice Selection in Multi-Tenant Networks with Maximum Isolation of Reserved Resources,” 2020 54th Asilomar Conference on Signals, Systems, and Computers, 2020, pp. 1002-1006, doi: 10.1109/IEEECONF51394.2020.9443543.

[27] F. Wei, G. Feng, Y. Sun, Y. Wang and S. Qin, ”Proactive Network Slice Reconfiguration by Exploiting Prediction Interval and Robust Optimization,” GLOBECOM 2020 - 2020 IEEE Global Communications Conference, 2020, pp. 1-6, doi: 10.1109/GLOBECOM42002.2020.9322440.

[28] R. Wen et al., ”On Robustness of Network Slicing for Next-Generation Mobile Networks,” in IEEE Transactions on Communications, vol. 67, no. 1, pp. 430-444, Jan. 2019, doi: 10.1109/TCOMM.2018.2868652.

[29] C. Sexton, N. Marchetti and L. A. DaSilva, ”On Provisioning Slices and Overbooking Resources in Service Tailored Networks of the Future,” in IEEE/ACM Transactions on Networking, vol. 28, no. 5, pp. 2106-2119, Oct. 2020, doi: 10.1109/TNET.2020.3004443.

[30] X. Yang, Y. Liu, I. C. Wong, Y. Wang and L. Cuthbert, ”Effective isolation in dynamic network slicing,” 2019 IEEE Wireless Communications and Networking Conference (WCNC), 2019, pp. 1-6, doi: 10.1109/WCNC.2019.8885563.

[31] D. Sattar and A. Matrawy, ”Optimal Slice Allocation in 5G Core Networks,” in IEEE Networking Letters, vol. 1, no. 2, pp. 48-51, June 2019, doi: 10.1109/LNET.2019.2908351.

[32] H. Yu, F. Musumeci, J. Zhang, M. Tornatore and Y. Ji, ”Isolation-Aware 5G RAN Slice Mapping Over WDM Metro-Aggregation Networks,” in Journal of Lightwave Technology, vol. 38, no. 6, pp. 1125-1137, 15 March15, 2020, doi: 10.1109/JLT.2020.2973311.

[33] Lilliefors H W. On the Kolmogorov-Smirnov test for normality with mean and variance unknown[J]. Journal of the American statistical Association, 1967, 62(318): 399-402.

[34] Razali N M, Wah Y B. Power comparisons of shapiro-wilk, kolmogorov-smirnov, lilliefors and anderson-darling tests[J]. Journal of statistical modeling and analytics, 2011, 2(1): 21-33.

[35] F. Comte, and N. Marie. ”Bandwidth selection for the Wolverton–Wagner estimator.” Journal of Statistical Planning and Inference 207 (2020): 198-214.

[36] S. Huang, B. Guo and Y. Liu, ”5G-Oriented Optical Underlay Network Slicing Technology and Challenges,” in IEEE Communications Magazine, vol. 58, no. 2, pp. 13-19, February 2020, doi: 10.1109/MCOM.001.1900583.

[37] A.Genz, ”Numerical Computation of Multivariate Normal Probabilities.” Journal of Computational & Graphical Statistics 1.2(1992):141-149.

Qize Guo received his B.E. degree in communication engineering from Information Engineering University, China in 2009. Since 2016 he is working towards his Ph.D. degree in Beijing University of Posts and Telecommunications, China. His research interests are mainly in the field of optical transport networks, mobile networks, and network slicing optimization.

Rentao Gu received his Ph.D. degree from Beijing University of Posts and Telecommunications (BUPT), China. He is a Professor of Beijing University of Posts and Telecommunications (BUPT), China. His current research interests include optical network and intelligent information processing. He is a senior member of China Institute of Communications.

Hao Yu received the B.E. and Ph.D. degrees in communication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2015 and 2020, respectively. He was also a joint-supervised Ph.D. student with the Politecnico di Milano, Milan, Italy. He is currently a Postdoctoral Researcher with the Center of Wireless Communications, Oulu University, Oulu, Finland. His research interests include network automation, time sensitive networks, and deterministic networks

Tarik Taleb is currently a Professor at the Center of Wireless Communications, The University of Oulu, Finland. He is the founder and director of the MOSA!C Lab (www.mosaic-lab.org). Between Oct. 2014 and Dec. 2021, he was a Professor at the School of Electrical Engineering, Aalto University, Finland. Prior to that, he was working as Senior Researcher and 3GPP Standards Expert at NEC Europe Ltd, Heidelberg, Germany. Before joining NEC and till Mar. 2009, he worked as assistant professor at the Graduate School of Information Sciences, Tohoku University, Japan. From Oct. 2005 till Mar. 2006, he worked as research fellow at the Intelligent Cosmos Research Institute, Sendai, Japan. He received his B. E degree in Information Engineering with distinction, M.Sc. and Ph.D. degrees in Information Sciences from Tohoku Univ., in 2001, 2003, and 2005, respectively. Prof. Taleb's research interests lie in the field of telco cloud, network softwarization & network slicing, AI-based software defined security, immersive communications, mobile multimedia streaming, and next generation mobile networking.

Yuefeng Ji (Senior Member, IEEE) received the Ph.D. degree from BUPT. He is currently a Professor and the Deputy Director of the State Key Lab of Information Photonics and Optical Communications. His research interests include the area of broadband communication networks and optical communications, with emphasis on key theory, realization of technology, and applications. He is a fellow of CIC, a fellow of CIE, and a fellow of IET.