Softwarized Resource Allocation in Digital Twins-Empowered Networks for Future Quantum-Enabled Consumer Applications

Haotong Cao, *Member, IEEE*, Sahil Garg, *Member, IEEE*, Shahid Mumtaz, *Senior Member, IEEE*, Mubarak Alrashoud, *Member, IEEE*, Longxiang Yang, Georges Kaddoum, Member, *IEEE*

***Abstract*—Network softwarization (NetSoft), recognized as crucial attribute of 6G networks, promises to provide enhanced and advanced services, including future quantum-enabled consumer applications. Softwarized resource allocation is the core issue in NetSoft concept. Digital twins (DT) guarantees to generate the corresponding digital world that reflects and interacts with the original physical world seamlessly. With DT empowering, the digital replica of softwarized networks can be generated to predict, simulate, analyze the softwarized resource allocation in more economical, convenient and scalable methods.In this paper, we research the softwarized resource allocation of requested services, usually, called as slices, in DT-empowered networks for future quantum-enabled consumer applications. We focus on developing efficient softwarized resource allocation algorithm. At first, we present models of the DT-empowered networks and service requests by using graph theory and hypergraph theory. Then, we design one softwarized resource management framework, labeled as *DT-Slice-Soft-6G*. This framework has the functions of managing softwarized resources, calculating resource allocation solution in digital replica and sending the calculated solution back to softwarized 6G networks. Thereafter, one efficient and fine-grained softwarized resource allocation algorithm, inserted in *DT-Slice-Soft-6G*, is detailed. This algorithm is labeled as *Heu-DT-Slice-6G* and is proposed based on efficient heuristic methods. To validate the highlights of *DT-Slice-Soft-6G* and *Heu-DT-Slice-6G*, we conduct the simulation work in our self-developed simulator.**

***Index Terms*—6G networks, quantum-enabled applications, digital twins, network softwarization.**

The work was partly supported by Natural Science Research Start up Foundation of Recruiting Talents of Nanjing University of Posts and Telecommunications (NJUPT) under Grant No. NY223029 and supported by Open Research Fund of Jiangsu Engineering Research Center of Communication and Network Technology, NJUPT. This work was also supported by the Researchers Supporting Project Number (RSPD2024R800), King Saud University, Riyadh, Saudi Arabia. This work was also a part of the 6G-SENSES project from the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101139282.*(Corresponding author: Longxiang Yang.)*

Haotong Cao and Longxiang Yang are with College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications (NJUPT), Nanjing 210003, China, and Jiangsu Engineering Research Center of Communication and Network Technology, NJUPT, Nanjing 210003, China. (e-mails: haotong.cao1992@gmail.com, yanglx@njupt.edu.cn)

Sahil Garg and Georges Kaddoum are with Electrical Engineering Department, École de technologie supérieure, Montréal, QC H3C 1K3, Canada. (e-mail: sahil.garg@ieee.org and georges.kaddoum@etsmtl.ca)

Shahid Mumtaz is with Department of Applied Informatics Silesian University of Technology Akademicka 16 44-100 Gliwice, Poland and Nottingham Trent University, Department of Computer Sciences. (e-mail: dr.shahid.mumtaz@ieee.org)

Mubarak Alrashoud is with Department of Software Engineering (SWE), College of Computer and Information Sciences (CCIS), King Saud University, Riyadh 11543, Saudi Arabia. (e-mail: malrashoud@ksu.edu.sa)

I. Introduction

CURRENTLY, 6G are investigated for providing enhanced and advanced applications and services, including quantum related applications and consumer electronics [1][2] (Fig. 1). Though official standards and consensuses of 6G have not launched, massive literature [1-3] and (experiment or simulation) data have been released. Learning from released publications and our gained knowledge, 6G networks are designed to have softwarization and virtualization attributes. Known to all, network softwarization and virtualization (Fig. 1) are mainly realized via SDN and NFV technologies [3]. Consequently, traditional dedicated network hardware and physical infrastructure (e.g., base stations, switches, routers, servers) will be decoupled into software blocks (resource and function) and general-purpose hardware (e.g., high volume servers) in 6G era. Services, also called as slices, can be executed by chaining these (resource and function) software blocks in predefined order on top of the general-purpose hardware. This chaining scheme contributes to constituting customized services. Initiated slices usually vary in terms of performance metrics, including emerging quantum-enabled applications and consumer applications.

Hence, it is crucial to investigate how to deploying a variety of requested slices optimally in resource-finite softwarized 6G networks. This technical issue is collectively called as resource allocation (RA) of slices [2] or service function chain (SFC)[4] in the literature. Till now, abundant research results (e.g., [4-10][35-37]) have been published. For instance, the authors of re. [4] and ref. [9] conducted comprehensive overviews on SFC and resource allocation in NFV-enabled networks. Representative works, such as linear programming based approaches, heuristic approach methods and meta-heuristic approach based methods, were discussed. In ref. [5], Li *et al.* proposed one intelligent algorithm, based on deep values, to address the end-to-end slicing resource allocation. Though having interest, the network scale, crucial attributes and resources were not considered. Cao *et al.* investigated the SFC placement and resource allocation in 6G networks. However, only crucial attributes and general services could be adopted by the proposed resource algorithm. This limited the contribution of this paper. In ref. [7], Chen *et al.* investigated the slicing resource allocation in vehicular networks. However,

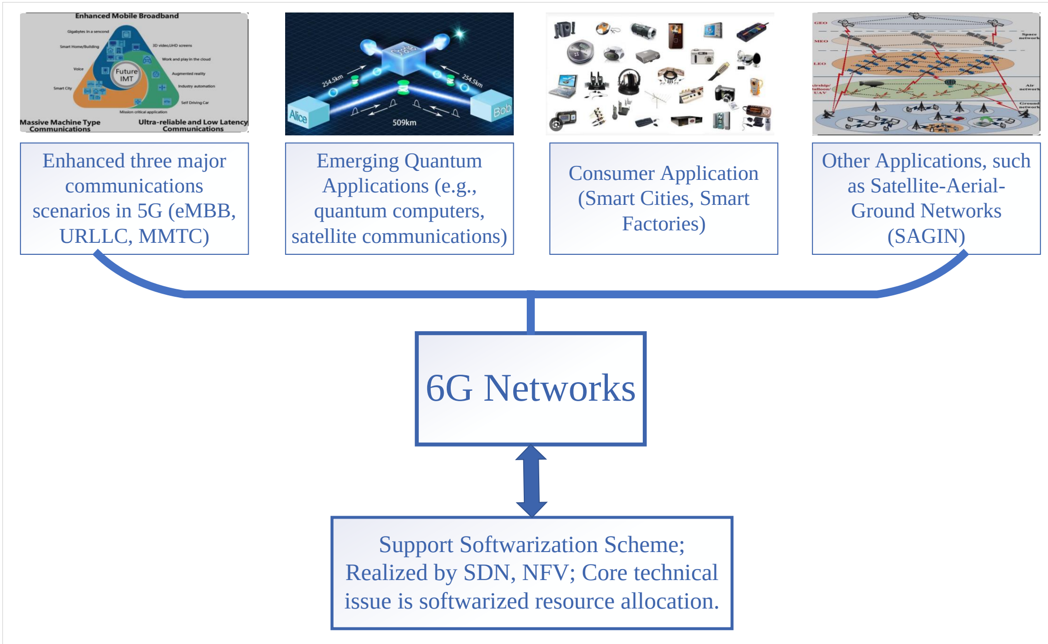

Fig. 1: Major Application Scenarios and Emerging Scenarios in 6G Networks and Softwarization Attribute

the service boundary is the RAN part and cannot be applied to broad coverage. In ref. [8], Cao *et al.* further proposed one service classification and developed the corresponding efficient algorithm to do softwarized resource allocation. The major limitation is the scale of underlying 6G networks and cannot be applied to real application. Zhang *et al.* adopted the policy gradient method and greedy approach to do resource allocation in virtualization environment. Though achieving efficient performance, the training and testing sets are all pre-defined and constructed. This key flaw limits the contribution. Apparently, most of existing researches are simply applied to small-scale or medium-scale networks. In addition, it is wasteful and inefficient to construct the large-scale softwarized 6G networks specially for exploring the long-term performance of proposed algorithms. Thus, existing researches are short of scalability. Last but not least, existing researches [4-10] are conducted without considering the dynamic and changed resource attributes of underlying networks [1][4]. That is to say, these researches are not evaluated in continuous time and are not suitable for dynamic environment. Therefore, it is crucial to find another alternative approach. The goal of this alternative approach is exploring the slice resource allocation in more economical, convenient and scalable manners before deploying for real large-scale commercial application.

Digital twins (DT)[11] technology, originated from the manufacturing, emerges in recent years [12-20]. With DT empowering, 6G digital replica can be adopted to predict, simulate, and analyze the dynamic management and resource allocation. The DT-empowering method is efficient and economical without constructing the physical objects specifically. This does benefits to simulating and evaluating emerging applications and scenarios, including quantum-related and consumer application. Hence, we incorporate DT into softwarized 6G networks and investigate the resource allocation of requested slices in DT-empowered softwarized 6G networks. We design one softwarized resource management and allocation framework, abbreviated as *DT-Slice-Soft-6G*, so as to realize the bidirectional connection between the softwarized 6G networks and their digital replica and provide efficient resource allocation solution per slice service. Before doing slice resource allocation, the digital replica of softwarized 6G networks will be generated, including all attributes (e.g., resource, topology, function) and states. When receiving one service request, *DT-Slice-Soft-6G* will turn to calculating and providing the efficient resource allocation solution of this slice. According to different optimization goals of requested slices, we propose one efficient and fine-grained softwarized resource allocation algorithm in *DT-Slice-Soft-6G* framework. This efficient algorithm is heuristic [22] and is proposed based on the Markov random model [21] and the extracted crucial resource and topology attributes. The heuristic algorithm is labeled as *Heu-DT-Slice-6G* and guarantees to find the efficient slice resource allocation solution within limited algorithm execution time. After calculating the resource allocation solution, the digital replica of 6G networks will be updated, including all resource and function attributes. At the same time, the softwarized 6G networks (physical object) will receive softwarized resource allocation solution of this slice from their digital replicas and be updated. All results are sent back by the bidirectional communication links. To validate the merits of our proposed algorithm in *DT-Slice-Soft-6G*, we make the comprehensive simulation work. Typical and classic algorithms [23] are selected for doing performance comparison. In continuous simulation scheme, our *Heu-DT-Slice-6G* achieved the best long-term performance among all evaluated algorithms.

Main contributions of this paper are selected and presented below:

1) We incorporate DT into the softwarized 6G networks so as to investigate the softwarized resource allocation in more efficient and economical manners. This scheme also does benefits to developing emerging applications (e.g., quantum-enabled consumer applications) in future 6G networks.

2) We propose one softwarized resource management framework, abbreviated as *DT-Slice-Soft-6G*, in order to send tailored resource allocation solution and information to physical 6G networks in time. We also design one efficient and fine-grained softwarized resource allocation algorithm *Heu-DT-Slice-6G*. This *Heu-DT-Slice-6G* is guaranteed to provide slice resource allocation solution within polynomial time.

3) We conduct the evaluation work so as to validate the merits of *DT-Slice-Soft-6G* framework and *Heu-DT-Slice-6G* algorithm. Evaluation results are carefully discussed. Typical works are selected to constitute the whole evaluation work.

The rest of this paper are organized as follows: We present the problem models in Section II. In Section III, we present the resource management framework for DT-empowered softwarized 6G networks. The efficient and fine-grained algorithm is detailed in Section IV. We present the whole evaluation work in Section V. In Section VI, we conclude this paper.

II. Problem Models and Dominant Performance Metrics

A. Problem Models of DT-Empowered 6G Networks and Slice Services

We present problem models of DT-empowered 6G networks and slice services in this subsection.

With respect to the problem model of DT-empowered 6G networks, it consists of two sub-models and are located in two isolated layers. Two layers are physical layer, digital (virtual) layer, respectively. Two sub-models are inter-connected by two directed communication links in order to achieve the seamless information exchange, states updations, decision making timely. The first sub-model is the physical 6G networks. Three major parts constitute 6G networks: radio access network (RAN) part, transmission network (TN) part, and core network (CN) part. Within RAN part, a large amount of edge nodes, access nodes and base stations are involved. Within TN, multiple physical nodes (e.g., switches) are involved. Within the CN, multiple core nodes are involved, such as the hosts, servers. These physical nodes are fully connected by (inter- or intra-) optical links. Physical 6G networks are enabled to be softwarized and virtualized. That is to say, the physical elements in 6G networks and their equipped resources can be softwarized and sliced into software resource blocks. The second sub-model is the digital (virtual) replica of 6G networks, located on top of softwarized 6G networks. The digital replica is the abstraction of softwarized 6G networks and consists of multiple nodes and links. Take note that all resource and functional attributes of 6G networks are digitalized and stored in the digital replica. The digital replica of 6G networks can be modeled by undirected graphs [24] and hypergraph: $6GNetworks = (6GNodes, 6GLinks)$. $6GNodes$ represents the node set of 6G networks and $6GLinks$ indicates the link set of 6G networks. We select one digital node M as the example. In this paper, digital node M has three major resource attributes: computing resource $Comp(M)$, storage resource $Stor(M)$, and capacity resource $Capa(M)$. Available resource sum (computing, storage and capacity) is selected as the hyperedge attribute. This attribute will be used in the calculation of node value (Section IV). In addition, M has two major functional attributes: network functions and data processing delay. With respect to the network function attributes, they include firewall, NAT and so on [2]. Firewall function can represented by $Func1(M)$. In this paper, four different functions in total are considered. With respect to the data processing delay attribute of M, it is abbreviated as $Delay(M)$. We also select one digital link MN as the example. In this paper, the bandwidth of MN is selected as the resource attribute and is labeled as $Band(MN)$.

To assist understanding the DT-empowered 6G networks and the corresponding digital replica, we further plot Fig. 2. In the bottom of Fig. 2, the softwarized 6G networks are plotted. The corresponding digital replica are plotted on top of the softwarized 6G networks. Within the digital replica, resource and functional attributes are all plotted.

With respect to the problem model of slice service, we model it by using undirected weighted graphs and locate it in the slice request layer. In softwarization research, service requests, initiated by end-users, are abstracted into multiple nodes and links and named as slices. Each slice service has arbitrary topology and tailored resource requests. We select one slice i, abbreviated as $Slice(i)$, as the example. $Slice(i) = (Nodes(i), Links(i))$. $Nodes(i)$ indicates the node set of $Slice(i)$ while $Links(i)$ is the link set of $Slice(i)$. One node m of $Node(i)$ is selected as the example. With respect to m, it has three dominant resource requests: computing resource request $Comp(m)$, storage resource demand $Stor(m)$, and capacity resource request $Capa(m)$. m has two functional attribute requests: network function demand $Func(m)$ and required data processing delay $Delay(m)$. With respect to certain one selected link mn, it has the bandwidth resource request, labeled as $Band(mn)$. In the top part of Fig. 2, two initiated slices, having different topology archituetures, are plotted. Their resource and functional demands are different and highlighted. Deployment and allocation results of two slices are highlighted in the digital replica of softwarized 6G networks.

Learning from slice resource allocation results in digital 6G networks, we can get three dominant resource allocation and deployment principles. The first principle is that different nodes from same slice are not allowed to be deployed in the same 6G node, aiming at getting rid of possible vulnerability of sharing same 6G node. The second principle is that digital 6G element (node or link) must reserve abundant resources and function attributes to fulfill the demands of its accommodated slice element (node or link). The last principle is that new graphs (e.g., digital 6G networks) are designed to be generated and updated periodically. When certain services

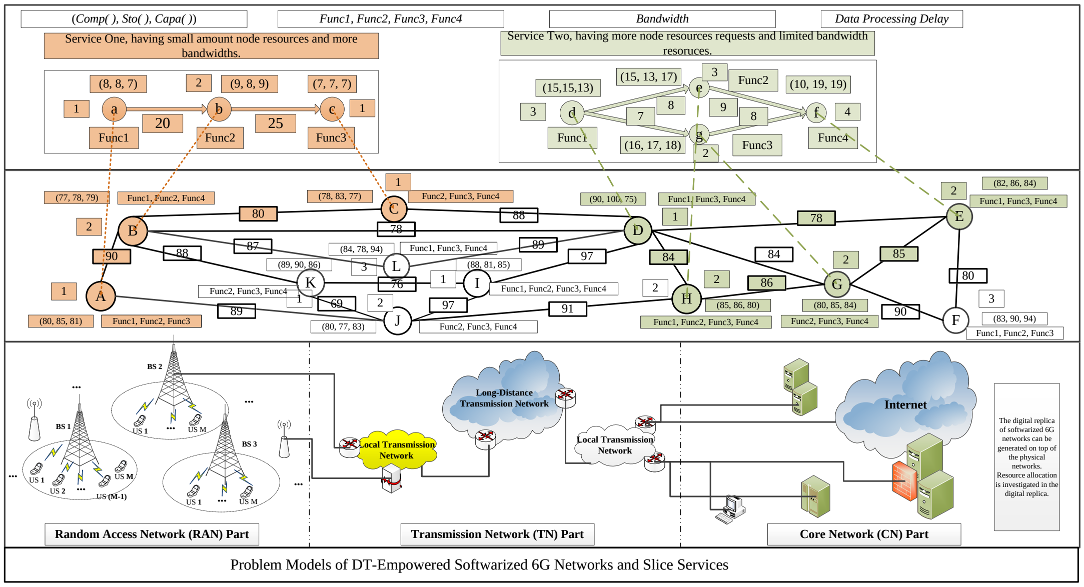

Fig. 2: DT-Empowered Softwarized 6G Networks and Two Requested Slice Services

are successfully deployed in the physical 6G networks, the corresponding digital replica will be updated, including its resource related information and attributes. Related information, such as resource changed information, are exchanged via communications of digital and physical parts.

B. Dominant Performance Metrics

This subsection focuses on formulating dominant performance metrics that can be adopted to evaluate slice resource allocation approaches and results in DT-empowered 6G networks. The first dominant performance metric is named as slice acceptance ratio. We present the formulation in Expression (1).

$$SliceRatio(\ ) = \frac{SuccessSlices(\ )}{TotalSlices(\ )} \tag{1}$$

where $SuccessSlices(\ )$ is a variable recording the number of successfully allocated slices. $TotalSlices(\ )$ is a variable having the function of recording the number of total initiated slice requests. Within the bracket, the name of selected allocation algorithm is included. In addition, two evaluation schemes exists in the slicing research: continuous time scheme (CTS) and discrete time scheme (DTS). The CTS aims at evaluating the long-term performance and ability of the selected algorithm while DTS aims at evaluating the batching processing ability of the selected algorithm. We consider CTS scheme in this paper.

The second dominant performance metric is named as resource utilization. Their concrete formulations are presented, ranging from Expression (2) to Expression (4).

$$CompUtili(\ ) = \frac{ConsumComp(\ )}{TotalComp(\ )} \tag{2}$$

where the name of selected slice allocation algorithm is included in the bracket. The variable $CompUtili(\ )$ reveals the computing resource consumption ability of the selected slice algorithm. If more slices are allocated and deployed, more computing resources will be consumed. Thus, the value of $CompUtili(\ )$ will be high.

$$StorUtili(\ ) = \frac{ConsumStor(\ )}{TotalStor(\ )} \tag{3}$$

$$CapaUtili(\ ) = \frac{ConsumCapa(\ )}{TotalCapa(\ )} \tag{4}$$

where the name of selected slice allocation algorithm is included in the brackets, revealing the storage and capacity resources utilization. Three expressions ((2), (3), (4)) belong to the node resource type in softwarization study.

$$BandUtili(\ ) = \frac{ConsumBand(\ )}{TotalBand(\ )} \tag{5}$$

where the name of selected slice allocation algorithm is included in the bracket.

The third dominant performance metric is named as slice algorithm execution time, abbreviated as $SliceExec(\ )$. The name of selected algorithm is included in the bracket. The $SliceExec(\ )$ has the function of recording the consumed time of calculating slice resource allocation solution. In DT and softwarization research, $SliceExec(\ )$ is crucial. Since slice services are initiated dynamically in networking environment (e.g., continuous time), consumed time should be minimized.

III. Softwarized Resource Management Framework Design *DT-Slice-Soft-6G*

In this section, we present the resource management design, abbreviated as *DT-Slice-Soft-6G*. To assist understanding *DT-Slice-Soft-6G*, we plot flow diagram (Fig. 3). Inserted mdules of *DT-Slice-Soft-6G* are introduced sequentially.

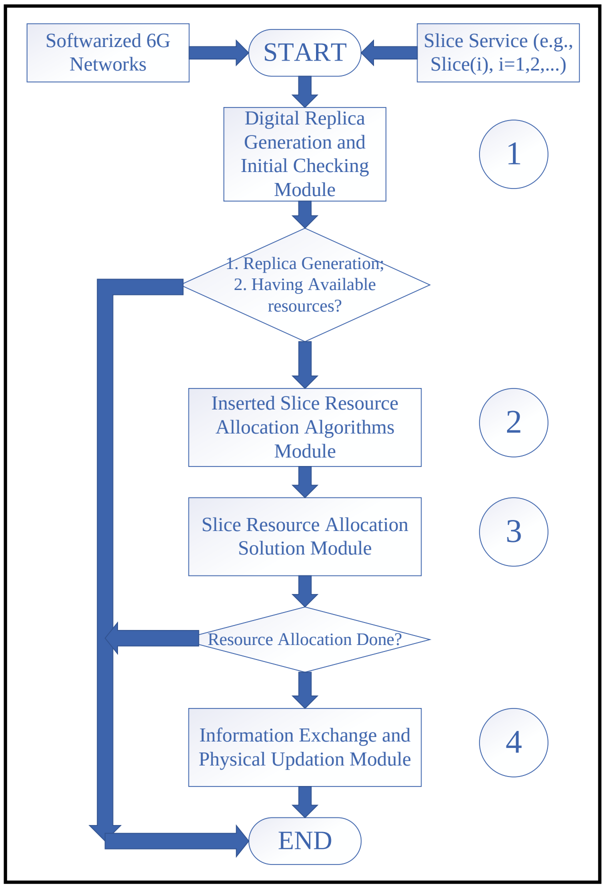

Fig. 3: Flow Diagram of *DT-Slice-Soft-6G* Framework Design

A. Digital Replica Generation and Initial Checking Module

The first module of *DT-Slice-Soft-6G* consists of two major procedures. Take note that two procedures in this module are two unique features of our *DT-Slice-Soft-6G*, comparing with existing softwarized resoruce allocation frameworks. The first procedure is to generate the digital replica of softwarized 6G networks. The second procedure is to do the initial resource checking of requested slice service. We abbreviate the slice example as $Slice(i)$ for easy description.

With respect to the first procedure, digital replica of softwarized 6G networks will be generated. We label the generated digital replica as $6GNetworks$. Within the digital replica, all attributes and state information of softwarized 6G networks will be replicated and stored, especially to resource and function attributes.

With respect to the second procedure, we focus on doing the initial resource checking of $Slice(i)$ and determine whether to continue to do softwarized resource allocation of $Slice(i)$. We firstly define four temporary variables and one temporary function set. Four temporary variables have the function of store the maximum resource values of $6GNetworks$: maximum computing resource, maximum storage resource, maximum capacity resource, and maximum bandwidth resource. The temporary function set stores the types of network functions that $6GNetworks$ have. In addition, another four temporary variables and one temporary function set will be defined for $Slice(i)$. These temporary variables and function set are adopted to store maximum required resources and required network function types of $Slice(i)$. Thereafter, we compare four variables and function set of $6GNetworks$ with four variables and function set of $Slice(i)$, respectively. Only if all resource maximum values and number of network function types of $6GNetworks$ are larger than that of $Slice(i)$, next module of *DT-Slice-Soft-6G* will continue to work. If certain one variable comparison is violated, $Slice(i)$ will be rejected.

B. Inserted Slice Resource Allocation Algorithms Module

Within the second module of *DT-Slice-Soft-6G*, a variety of slice resource allocation algorithms are inserted. Inserted algorithms can be classified into two major types [21][23]: the exact type and the heuristic type. Take note that the known meta-heuristic type [23] (e.g., reinforcement learning, transfer learning) strictly belongs to the heuristic type. The function of this module is selecting the suitable resource allocation algorithm from inserted algorithms to calculate the resource allocation solution of $Slice(i)$. The selecting criterion is the concrete goal of the requested tailored $Slice(i)$.

If the primary goal of $Slice(i)$ is to find the optimal resource allocation solution without worrying about exponential-level algorithm running time, the exact algorithms, such as integer linear programming-based approach, will be selected to serve $Slice(i)$. If $Slice(i)$ requires to get the resource allocation solution as soon as possible, it indicates that one feasible slice resource allocation solution is okay. Hence, the heuristic algorithms will be considered to select. If the balance between solution optimality and algorithm running algorithm is required, the meta-heuristic algorithms, such as the generic method, will be considered. After selecting the suitable slice resource allocation algorithm, $Slice(i)$ will skip to the third module of *DT-Slice-Soft-6G*.

C. Slice Resource Allocation Solution Module

The third module of *DT-Slice-Soft-6G* is responsible for running the selected algorithm so as to calculate the resource allocation solution of $Slice(i)$ from the exposed resource and function attributes of digital replica $6GNetworks$.

Details and procedures of selected slice resource allocation algorithms are not detailed in this subsection. While running the resource allocation algorithm, resource and function demands of $Slice(i)$ must be executed and fulfilled. If certain one demand is not fulfilled, $Slice(i)$ will be rejected. In addition, time attributes of $Slice(i)$ must be carefully considered. That is to say, the algorithm execution time of the selected algorithm must not violate the maximum allowed waiting time of $Slice(i)$. If violated, $Slice(i)$ will be rejected.

D. Information Exchange and Physical Updation Module

When the third module of *DT-Slice-Soft-6G* runs successfully, the resource allocation solution of $Slice(i)$ will be achieved. Consequently, the last module of *DT-Slice-Soft-6G* will start to work.

In this module, (resource and function) attributes, information and element states of digital replica of 6G networks $6GNetworks$ will be updated in the first place. In addition, the digital replica will send back current attributes, information and element states to the original softwarized 6G networks through the bidirectional communication links. Slice resource allocation solution of $Slice(i)$ will be sent to the original softwarized 6G networks, too. All temporary variables and network function sets for $6GNetworks$ and $Slice(i)$ will be erased and waiting for serving the next slice. TSP will update softwarized 6G networks and deploy the $Slice(i)$, according to the calculated solution. After that, all information and states of softwarized 6G networks will be sent to the digital replica so as to further update and correct the differences.

When $Slice(i)$ is deployed successfully in softwarized 6G networks, next slice, such as $Slice(i+1)$ will be served by *DT-Slice-Soft-6G*. New circle of softwarized resource management and allocation starts again.

IV. Technical Details of *Heu-DT-Slice-6G* in *DT-Slice-Soft-6G* Design

In this subsection, we present the general formulation of exact type algorithm. Then, we detail one designed efficient and fine-grained softwarized resource allocation algorithm (*Heu-DT-Slice-6G*). This *Heu-DT-Slice-6G* algorithm is inserted in the Slice Resource Allocation Algorithms Module of *DT-Slice-6G*. Strictly speaking, *Heu-DT-Slice-6G* belongs to the heuristic type. We aim at getting efficient softwarized resoruce allocation solution in polynomial time. Algorithm complexity discussion of *Heu-DT-Slice-6G* is also presented.

A. Details of General Exact Type Algorithm

Details of general exact type algorithm are presented in this subsection. The general exact type algorithm is proposed based on the integer linear programming approach [25-27]. At first, we define two kinds of integer variables for the association of slice: X and Y. $X = (x_M^m, m\epsilon Slice(i), M\epsilon 6GNodes,)$. If $x_M^m = 1$, it indicates that slice node m is allocated to digital node M. Otherwise versa. $Y = (y_{MN}^{mn}, mn\epsilon Slice(i), MN\epsilon 6GLinks,)$. If $y_{MN}^{mn} = 1$, it indicates that slice link mn is allocated to digital path MN. Otherwise versa.

Then, we continue to construct 'Candidate Digital Nodes'(CDN) set for each slice node. With respect to Slice(i), the number of its CDN sets is equal to the number of nodes in $Nodes(i)$, labeled as $|Nodes(i)|$. With respect to certain one slice node m, if its required function demand is $Func1(m)$, its criterion of selecting candidate node is whether digital node has the network function $Func1(\ )$. Digital nodes having $Func1(\ )$ will be selected and constitute the CDN for slice node m, labeled as $CDN(m)$.

Next, we formulate the Slice(i) resource allocation in digital replica of softwarized 6G networks. Concrete mathematical expressions of general exact type algorithm are presented below:

$$maximize\ \alpha * \sum_{M\epsilon CDN(m)} \sum_{m\epsilon Nodes(i)} x_M^m * (Comp(m) + Stor(m) + Capa(m)) + \beta * \sum_{MN\epsilon 6GPaths} \sum_{mn\epsilon Links(i)} y_{MN}^{mn} * Band(mn) \tag{6}$$

$$subject\ to$$

$$\forall m\epsilon Nodes(i), \forall M\epsilon CDN(m),\ x_M^m\ \epsilon \{0,\ 1\} \tag{7}$$

$$\forall mn\epsilon Links(i), \forall MN\epsilon 6GPaths,\ y_{MN}^{mn}\ \epsilon \{0,\ 1\} \tag{8}$$

$$\forall m\epsilon Nodes(m), \sum_{M\epsilon CDN(m)} x_M^m = 1 \tag{9}$$

$$\forall M\epsilon CDN(m), \sum_{m\epsilon Nodes(m)} x_M^m \leq 1 \tag{10}$$

$$\forall mn\epsilon Links(i), \sum_{MN\epsilon 6GPaths} (y_{MN}^{mn} - y_{MN}^{nm}) = x_M^m - x_N^m \tag{11}$$

$$\forall mn\epsilon Links(i), \sum_{MN\epsilon 6GPaths} (y_{MN}^{mn} - y_{MN}^{nm}) = x_N^n - x_M^n \tag{12}$$

$$\forall m\epsilon Nodes(i), \sum_{M\epsilon CDN(m)} x_M^m * Comp(m) \leq Comp(M) \tag{13}$$

$$\forall m\epsilon Nodes(i), \sum_{M\epsilon CDN(m)} x_M^m * Stor(m) \leq Stor(M) \tag{14}$$

$$\forall m\epsilon Nodes(i), \sum_{M\epsilon CDN(m)} x_M^m * Capa(m) \leq Capa(M) \tag{15}$$

$$\forall m\epsilon Nodes(i), \sum_{M\epsilon CDN(m)} x_M^m * Delay(M) \leq Delay(m) \tag{16}$$

$$\forall mn\epsilon Links(i), \sum_{MN\epsilon 6GPaths} (y_{MN}^{mn} + y_{MN}^{nm}) * Band(mn) \leq Band(MN) \tag{17}$$

where the objective function (Expression (6)) indicates maximizing the number of successfully allocated slices. This function aims at maximizing the benefits. Since owner (e.g., telecommunication providers, service provider) constructs the underlying 6G networks and generates the digital replica, owner can serve signed end users directly. Thus, the owner needs to pay for renting fee. Cost part for allocating slices does not need to be included in the Expression (6). α and β are unit prices of node type resource and link type resource. Take note that $6GPaths$ is the physical set of $6GNetworks$, storing loop-free paths. Number of elements in $6GPaths$ set is larger than that in $6GLinks$ set. It is owing to the fact that certain one path may have intermediate nodes. A link can be regarded as a path while a path cannot be regarded as a link.

In both Expression (7) and Expression (8), they are definitions of two binary variables. Two state values (0 and 1) are stored in each variable state set.

In both Expression (9) and Expression (10), they are relationship constraints, aiming at expression the relationship between digital node and slice node. Certain one slice node must be allocated to one digital node while certain one digital node may not be allocated to one slice node.

In both Expression (11) and Expression (12), they indicate that each slice link must be allocated to one single physical path. Meanwhile, two end nodes of this slice link must be allocated to two end nodes of the selected path. In both expressions, X type variables and Y type variables are adopted to guarantee their matching relationships.

From Expression (13) to Expression (16), they are constraints of node resources (computing, storage and capacity) and processing data delay. The selected digital node must reserve abundant resources to accommodate the allocated slice node. In addition, the maximum required node processing delay of slice node cannot be violated by the selected digital node. With respect to the last constraint (Expression (17)), it indicates that the selected digital path must reserve available bandwidth resource to fulfill the required bandwidth of the allocated slice link.

Finally, these formulations can be addressed by using open-source (e.g., CPLEX [28]) or non-open-source (e.g., GLPK [29], Matlab CVX) mathematical tools. Since these formulations (objective function and constraints) are linear, this optimization problem can be directly addressed without doing transformation and integer variable relaxation [26]. In addition, a variety of optimization algorithms, such as the branch and bound [26], can be adopted to solve this optimization problem. In the third subsection, we will further discuss the complexity related contents of this optimization problem.

As presented above, they are technical details of general exact type algorithm for calculating slice (e.g., $Slice(i)$) resource allocation solution in the digital replica of softwarized 6G networks. When allocated positions and results in digital $6GNetworks$ are output successfully, we can get the optimal slice resource allocation solution ($Slice(i)$).

B. Details of Heu-DT-Slice-6G Algorithm

The *Heu-DT-Slice-6G* is detailed in this subsection. The *Heu-DT-Slice-6G* belongs to the heuristic type and consists of three parts: Node-Value Vector Calculation Part, Slice Node Resource Allocation Part, and Slice Link Resource Allocation Part. Resource allocation solutions of some application scenarios, such as the online shopping and meeting, can be adopted by our *Heu-DT-Slice-6G*. These scenarios allowed medium latency. The scenarios, such as the remote surgery, cannot be adopted as they have rigid time limitation of 1 $ms1$[1]. With respect to the first part, it is designed based on the Markov random model and extracted crucial topology and resource attributes. Based on gained knowledge [30][31] and previous attempts [32][33], we select crucial topology attributes.

For easy understanding, node M and link MN are chosen. Two crucial attributes (node degree $Degree(M)$, node farness $Farness(M)$) are chosen. In ref. [33], definitions and formulation methods of chosen attributes are presented. We also choose another five resoruces: $Comp(M)$, $Stor(M)$, $Capa(M)$, $Delay(M)$, $Band(MN)$. By using these attributes and resources, we can derive another metric, marked as $NodeValue$, for quantifying node value (e.g., M):

$$\begin{aligned}NodeValue(M) = (Comp(M) + Stor(M) + Capa(M))*\\ Degree(M) * Farness(M) * Delay(M)\\ * \sum_{MN \epsilon MXLinks} Band(MN)\end{aligned} \tag{18}$$

where $MXLinks$ is the adjacent link set of M. Available resource sum (Section II) is the hyperedge attribute of node M. We select it to help calculating value.

As formulated in Expression (18), $NodeValue$ can be adopted straightly. In the next section, the *Direct-DT-Slice-6G* adopts this kind metric. However, the straight value metric could not reveal accurately resource allocation ability. Therefore, we need to find another alternative method. Learning from Markov random model method, accurate node value can be achieved by extensive calculation. We present details of the Markov method below:

In the first stage, the initialization procedure will be executed for all nodes in the digital replica. Concrete expression is presented below:

$$\begin{aligned}NodeValue(M)\% = NodeValue(M)^0\\ = \frac{NodeValue(M)}{\sum_{M\epsilon 6GNodes} NodeValue(M)}\end{aligned} \tag{19}$$

In the second statge, we will use all initialized values to produce another vector. The vector is in Expression (20).

$$\begin{aligned}\overrightarrow{NodeVector^0} =\\ (NodeValue(A)\%...NodeValue(M)\%, ...)^{num}\end{aligned} \tag{20}$$

where the dimension num indicates the number of nodes in $6GNetworks$. Its value is the $\overrightarrow{NodeVector^0}$'s dimension.

In the third stage, it is necessary to quantify two kinds of relationships that can be adopted to express node M and its neighboring nodes. We formulate Expression (21) having the gooal of quantifying the probability of jumping from node M to certain one node (except M). This is the first kind of relationship.

$$JProb(MN)\% = \frac{NodeValue(N)}{\sum_{M\epsilon 6GNodes} NodeValue(M)} \tag{21}$$

We further formulate Expression (22) having the goal of indicating probability of forwarding M to its nearest node (e.g., node P).

$$FProb(MP)\% = \frac{NodeValue(P)}{\sum_{P\epsilon NeiNodes(M)} NodeValue(P)} \tag{22}$$

where $NeiNodes(M)$ is neighboring set of M. Learning from noth expressionns, we can construct two matrices. Two are labeled as $JumpMatrix$ and $ForMatrix$, respectively.

In the fourth stage, we select the Markov random model to calculate the all node values in vector form. To help understanding, node M is selected. $NodeValue(N)^1$ can be achieved by following expression.

$$NodeValue(N)^1 = \lambda * \sum_{P \epsilon NeiNodes(M)} FProb(MP)\% * NodeValue(N)^0 + (1-\lambda) * \sum_{M \epsilon 6GNodes} JProb(MN)\% * NodeValue(N)^0 \quad (23)$$

In the above expression, λ is called as balancing factor and can be adjusted. Through t rounds of calculation, we can calculate the $NodeValue(N)^{t+1}$.

$$NodeValue(N)^{t+1} = \lambda * \sum_{P \epsilon NeiNodes(M)} FProb(MP)\% * NodeValue(N)^t + (1-\lambda) * \sum_{M \epsilon 6GNodes} JProb(MN)\% * NodeValue(N)^t \quad (24)$$

With respect to the vector form, it is presented in Expression (25).

$$\overrightarrow{NodeVector^{t+1}} = \lambda * JumpMatrix * \overrightarrow{NodeVector^{t}} + (1-\lambda) * ForMatrix * \overrightarrow{NodeVector^{t}} \quad (25)$$

$$\begin{aligned}\overrightarrow{NodeVector^{t+1}} &= [\lambda * JumpMatrix + (1-\lambda) * ForMatrix] * \overrightarrow{NodeVector^{t}} \\ &= M * \overrightarrow{NodeVector^{t}} \end{aligned} \quad (26)$$

where $M = [\lambda * JumpMatrix + (1-\lambda) * ForMatrix]$.

Apparently, the Expression (26) [25][26] is converged after multiple rounds of calculation. Concrete proof is availble in existing publications [25][26][31]. In addition, uniqueness of Expression (26) is apparent due to the eigenvalue value of M. Consequently, accurate node-value vector of $6GNetworks$ can be achieved eventually. In this paper, we adopt the convenient iterative method [31].

Finally, node values of all digital nodes can be achieved by using above approach. Furthermore, digital nodes are sorted in descending sequence and prepared for softwarized resource allocation.

In order to determine the allocating resource order to nodes in $Slice(i)$, we formulate Expression (27) (hyperedge attribute). Slice node m is chosen in Expression (27). According to calculated values, we arrange all slice node. For example, m has higher value and is allocated in the first place.

$$NodeValue(m) = Comp(m) + Stor(m) + Capa(m) \quad (27)$$

With respect to the second part, it indicates the slice node resource allocation of $Slice(i)$. Slice node, has the highest value (Expression (27)), has the highest priority to be processed. We set another five variables for storing this slice node's demands: network function, computing, storage, capacity, data processing delay. We select slice node m as the example. The digital nodes are selected one by one to attempt to process the slice node m, following the descending order of node values. We assume that digital node M has the highest value. Thus, M is selected to try to accommodate m. We define another four variables, having the function of storing its available resources, and one set, having the function of its equipped network functions. Then, we compare five variables of m with corresponding variables and set of M. If all demands of m are fulfilled, we can determine to allocated M to m. If certain one resource or function demand cannot be fulfilled, we will erase all variables and set for M and store the next digital node's resources and functions. Thereafter, we will do the comparison procedure again. Repeat this until m is allocated successfully. If m cannot be allocated eventually, $slice(i)$ will be rejected to be served. That is to say, our ***Heu-DT-Slice-6G*** cannot serve $Slice(i)$.

Algorithm 1 Node Resource Allocation of $Slice(i)$

Input: Node-Value Vector of $6GNetworks$, $Slice(i)$
Output: Node Resource Allocation Results of $Slice(i)$

```
Define one node set Set1 storing all slice nodes in descending order, according to Expression (27).
Define one node set Set2 storing all digital nodes in descending order, according to Expression (26).
Define five variables for Slice(i) and define four variables and one set for 6GNetworks
Define two integer variables, NUM1 and NUM2, with initial values 0.
Define one integer slicenum for storing the number of nodes in Slice(i).
while NUM1 ≤ slicenum do
    Select the (NUM2 + 1)th digital node from Set2;
    Store of (computing, storage, capacity, node processing delay) resource values of (NUM2 + 1)th digital node into four defined variables and equipped network functions into the defined set;
    Select the (NUM1 + 1)th slice node from Set1;
    Store of (computing, storage, capacity, node processing delay, function) demand values of (NUM1 + 1)th digital node into five defined variables;
    Do the comparison work between (NUM2 + 1)th digital node and (NUM1 + 1)th slice node;
    if certain one demand of (NUM1 + 1)th slice node cannot be fulfilled then
        Reject the Slice(i);
        if all five kinds of demands can be fulfilled then
        Allocate (NUM1+1)th slice node to (NUM2+1)th digital node and remove (NUM2+1)th digital node from the Set2;
    end if
    Set value of NUM2 to be 0;
    NUM1 + +;
end while
Output node resource allocation results of Slice(i) and update 6GNetworks.
Erase all values in defined variables and set and wait for serving next slice's node resource allocation.
```

Above all are softwarized resource allocation of one slice node m. With respect to the remaining slice nodes in $Slice(i)$, the above procedure will repeat until remaining nodes are processed successfully. Take note that if certain one digital node is allocated to one slice node, it will not to be considered to allocate another slice node. We must guarantee that one digital node is allowed to serve at most one slice node in the same slice service. We present ***Algorithm 1*** to assist understanding. When all slice nodes of $Slice(i)$ are done, our *Heu-DT-Slice-6G* algorithm will run its third and last part.

With respect to the last part, it indicates the slice link resource allocation of $Slice(i)$. In this part, we will set another set, abbreviated as $LinkSet$, for storing all links of $Slice(i)$. Within the link set, all links are sorted, according to their required bandwidth. Slice link, such as mn, has the highest bandwidth request, will be placed in the first place of $LinkSet$. Thus, mn is selected to allocate in the first place. In the second part of *Heu-DT-Slice-6G*, two end nodes of mn are successfully allocated. In the third part, we simply need to find the most suitable from two fixed digital nodes. The selecting criterion in this part are: having minimum number of intermediate nodes and fulfilling this slice link's bandwidth demand. Two potential and dominant benefits hide behind this selecting criterion. The first benefit is saving total consumed bandwidth resource. The other benefit is decreasing the possibility of data packet loss and transmission delay in prototype implementation.

Following the above strategy to complete all slice links' allocation of $Slice(i)$. If certain one slice link is not fulfilled successfully, our *Heu-DT-Slice-6G* algorithm will reject to serve $Slice(i)$. Only when this part works successfully, our *Heu-DT-Slice-6G* algorithm successfully allocate slice links successfully. According to allocated positions and results in digital $6GNetworks$, we can get the slice resource allocation solution of *Heu-DT-Slice-6G*.

C. Algorithm Complexity Discussion

In this subsection, algorithm complexity work is conducted and discussed. At first, we discuss the algorithm complexity of general exact algorithm. Since the exact algorithm is proposed based on the integer linear programming method [27], the nature of integer programming method will incur exponential-level complexity [25][26]. Considering the dynamic and time-variant attributes of slice services, it is not suitable to be evaluated in CTS. Thus, exact type algorithm will be evaluated in DTS. In addition, network scale is the dominant factor of binary variable number that will be defined in exact algorithm. Hence, the network scale of general exact type algorithm must be limited.

Secondly, we discuss the algorithm complexity of *Heu-DT-Slice-6G*. As *Heu-DT-Slice-6G* is made up of three parts, we focus on discussing complexity of these three parts. With respect to the first part, its complexity lies in the node-value vector calculation. According to published references, the complexity of Markov model, addressed by iterative method, is not more than $O(num * log(1/\delta))$ [30], where num is the number of nodes in $6GNetworks$. Thus, this part can be completed in polynomial time. With respect to the complexity of second part, it is not more than $O(num*|Node(i)|)$, where $|Node(i)|$ is the number of nodes in $Slice(i)$. This part is a polynomial-time level procedure. With respect to the last part, it is similar to the procedure of finding shortest path [31] between two fixed locations. This shortest path problem can be addressed within polynomial time. In general, the time complexity of *Heu-DT-Slice-6G* belongs to the polynomial-time level. Hence, it can be applied in CTS and DTS.

V. Comprehensive Evaluation Work

A. Evaluation Settings of Continuous Time Scheme

With respect to the softwarized 6G networks and their digital replica, the network scale is set to have 100 nodes and 0.5 connectivity possibility per pair of nodes. This network scale is large in slicing resource allocation research. With respect to each node, its available resource (computing, storage, capacity) is an integer, following the uniform distribution between 80 and 100. With respect to equipped network functions of each node, four types of functions are included in each node. With respect to the data processing delay of each node, it is an integer and uniformly distributed between 1 time unit and 4 time units. With respect to each link, its available bandwidth is an integer, following the uniform distribution of [70, 100].

With respect to each requested slice, its network scale is an arbitrary integer and follows the uniform distribution [2, 6]. Each pair of nodes have a connecting possibility of 0.5. With respect to each node, its resource (computing, storage, capacity) demand is an integer, following the uniform distribution between 2 and 10. Each node has one network function demand. The maximum allowed data processing delay is an integer and uniformly distributed between 2 time units and 5 time units. The bandwidth demand of each link is an integer uniformly distributed between 10 and 15.

With respect to selected algorithms in CTS, typical slice algorithms are selected from existing publications [4][6]. Selected algorithms are abbreviated as *Greedy-DT-Slice-6G*, *Direct-DT-Slice-6G*, *RelaxedILP-DT-Slice-6G*, *DirMar-DT-Slice-6G*. The *Greedy-DT-Slice-6G* directly adopts maximum node resource value (direct sum of computing, storage and capacity resource value) as the allocating basis. Following node resource allocation strategy and link resource allocation strategy are same to that of our *Heu-DT-Slice-6G*. The *Direct-DT-Slice-6G* adopts Expression (18) (direct product of resource sum, crucial attributes and bandwidth sum) to be the allocating basis. Following node resource allocation strategy and link resource allocation strategy are same to that of our *Heu-DT-Slice-6G*. *DirMar-DT-Slice-6G* adopts maximum node resource sum value as the initial node value. Other algorithm details are similar to our *Heu-DT-Slice-6G*. In both our *Heu-DT-Slice-6G* and *DirMar-DT-Slice-6G*, the δ is set to be 0.001. *RelaxedILP-DT-Slice-6G* relaxes the integer variables and gets the linear programming solution as the feasible resource allocation solution.

In CTS, the simulation time will last 1000 time units. Take note that 1 time unit is equal to 1 minute in this evaluation work. Within 100 time units, 5 slices will arrive and require to be processed. Maximum allowed waiting time of each slice is set to be 5 time unit. The simulation work is run on the Desktop that is equipped with Intel Core CPU i7-4790 3.6GHz Processor, 16.00G RAM and Window 8 operation system. GLPK [29] tool is adopted to solve linear programming model of *RelaxedILP-DT-Slice-6G*. The evaluation is conducted in our self-developed simulator [34].

B. Evaluation Results and Discussion in Continuous Time Scheme

We aim at validating the softwarized resource allocation ability of our *DT-Slice-Soft-6G* framework and designed *Heu-DT-Slice-6G* algorithm in the long run. Achieved results are plotted, ranging from Fig. 4 to Fig. 7.

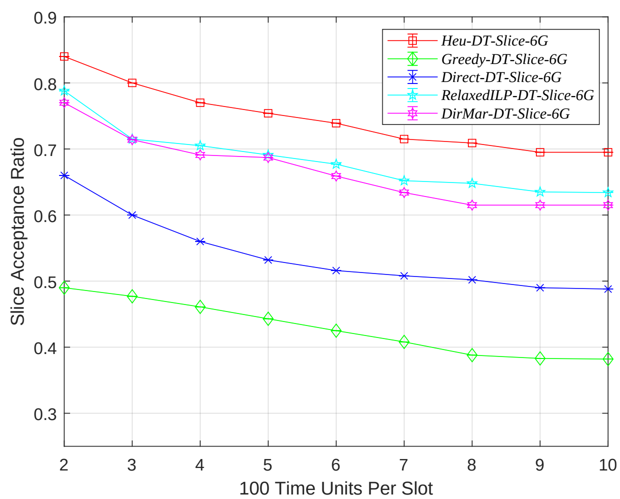

Fig. 4: Slice Acceptance Results in Continuous Time Event

Slice acceptance results of all selected algorithms are plotted in Fig. 4. By scanning the Fig. 4, our *Heu-DT-Slice-6G* achieves the highest slice acceptance results throughout the whole simulation event. We mainly discuss the advantages of our *Heu-DT-Slice-6G* from three groups of performance comparison. With respect to the first group of comparison, it consists of *Heu-DT-Slice-6G*, *Greedy-DT-Slice-6G* and *Direct-DT-Slice-6G*. As shown in Fig. 4, *Greedy-DT-Slice-6G* achieves the lowest slice acceptance among all selected algorithms. It is owing to the fact that its allocating criterion is the resource sum of node. This criterion is too simple to reveal the node importance in the while digital replica and the resource allocation. In addition, our *Heu-DT-Slice-6G* achieves an advantage of nearly 20%, comparing with *Direct-DT-Slice-6G*. Though the resource allocating criterion of *Direct-DT-Slice-6G* is the product of node resource sum and node's adjacent bandwidth sum, node importance and crucial role in the network are not well quantified. Quantifying method of our *Heu-DT-Slice-6G* is based on nodes' resource and links' bandwidth and crucial topology attributes. In addition, the iterative method further converges the quantified value after limited calculation. Hence, the calculated values can reveal node importance in the whole network [31]. Consequently, our *Heu-DT-Slice-6G* can efficiently do slice resource allocation. With simulation time extending, node values can be updated in time. This further guarantees the performance advantage.

With respect to the second group of comparison, it consists of *Heu-DT-Slice-6G* and *DirMar-DT-Slice-6G*. As shown in Fig. 4, *Heu-DT-Slice-6G* achieves higher slice acceptance than *DirMar-DT-Slice-6G*. Since around 600 time units, the performance gap is stable and approaches 8%. The reason for performance advantage is the selected crucial topology attributes of *Heu-DT-Slice-6G*. Our *Heu-DT-Slice-6G* incorporates crucial topology attributes so as to quantify node values accurately. Node values directly indicate the node importance in the whole network. Consequently, our *Heu-DT-Slice-6G* can guarantee its slice acceptance ratio high, comparing with the *DirMar-DT-Slice-6G*.

With respect to the third group of comparison, it consists of *Heu-DT-Slice-6G* and *RelaxedILP-DT-Slice-6G*. As shown in Fig. 4, *RelaxedILP-DT-Slice-6G* cannot perform as well as our *Heu-DT-Slice-6G* though it is based on the integer linear programming method. It is owing to that fact that relaxation procedure is included in *RelaxedILP-DT-Slice-6G*. Though algorithm execution time of *RelaxedILP-DT-Slice-6G* can be greatly decreased, the optimality of slice solution cannot be guaranteed. Throughout the whole simulation event, its slice acceptance is greatly weakened. The performance gap is more than 5% in the long run. In general, our *Heu-DT-Slice-6G* is proved to be an efficient slice resource allocation in the continuous time event and can do near-optimal slice allocation in DT-empowered softwarized 6G networks.

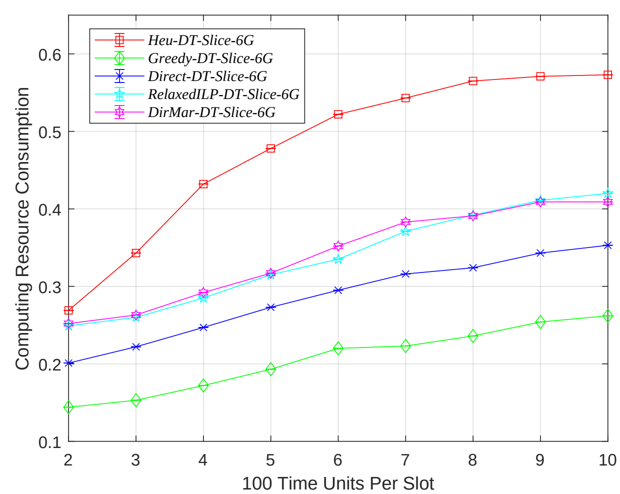

Fig. 5: Computing Resource Consumption Results in Continuous Time Event

With respect to remaining figures (Fig. 5, Fig. 6, Fig. 7), they are resource-consumption results. It can be easily found that our *Heu-DT-Slice-6G* achieves the highest (node and link) resource consumption results. These behaviors further prove the efficiency of our *Heu-DT-Slice-6G*. Our *Heu-DT-Slice-6G* allocates and deploys more slices successfully. Consequently, more resources are consumed. These results run as expected. With respect to other results illustration, such as the average

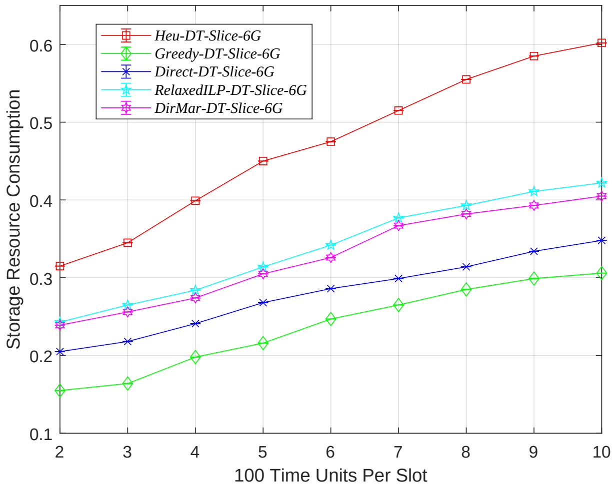

Fig. 6: Storage Resource Consumption Results in Continuous Time Event

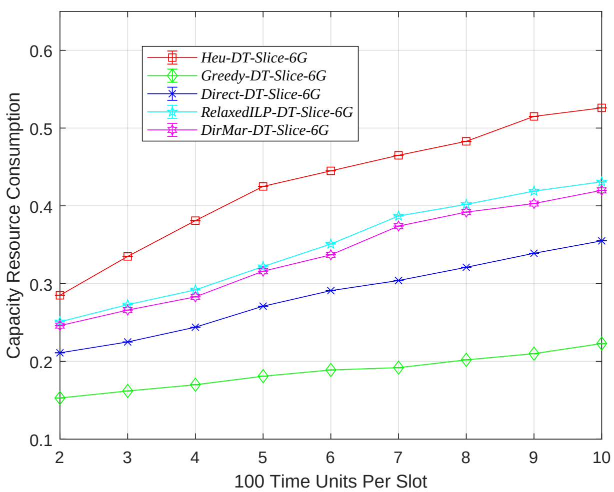

Fig. 7: Capacity Resource Consumption Results in Continuous Time Event

node processing delay, we do not plot them in this paper. We focus on validating the strong slice resource allocation ability of our *Heu-DT-Slice-6G*.

VI. Conclusion Work

Digital twins (DT), originated from the manufacturing, has been applied to a variety of industrial domains and emerging areas. With DT empowering, the digital replica of original physical object can be generated and adopted to simulate behaviors, exchange information and assist decision making. In addition, DT has the characteristic of scalability. Hence, the performance of softwarized 6G networks' digital replicas can be researched in more economical and efficient manners. Based on these backgrounds, we research the resource allocation of requested slices in DT-empowered softwarized 6G networks. We firstly model the DT-empowered softwarized 6G networks and tailored slices by using graph and hypergraph theory. Then, we design one softwarized resource management framework, labeled as *DT-Slice-Soft-6G*. Within the *DT-Slice-Soft-6G* framework, one efficient and fine-grained slice resource allocation algorithm (*Heu-DT-Slice-6G*) is developed and inserted. Next, we present technical details of this developed algorithm, such as formulation, pseudo codes and time complexity discussion. In order to validate the feasibility and merits of our *DT-Slice-Soft-6G* framework and the proposed algorithm *Heu-DT-Slice-6G*, we conduct the comprehensive evaluation work. Typical and classic slice resource allocation algorithms are selected to constitute the evaluation work. Evaluation results reveal that our *Heu-DT-Slice-6G* achieves the best long-term performance among all evaluated algorithms.

For the future work, we will investigate it from three aspects. First aspect is that we will design more algorithms and approaches to do softwarized resource allocation in digital replica. Second aspect is that we will further refine the resoruce threads of requested services. Based on the refined resource threads, more efficient algorithm can be designed. Third aspect is that we will explore the scalability of simulation work. We will explore the performance of proposed algorithm in large-scale network settings. Since the integration of DT and softwarization for future emerging applications is in infancy, the scalability of proposed algorithm is not investigated in this paper.

References

[1] X. You, C. Wang, et al., "Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts," Sci. China Inf. Sci., vol. 64, no. 1, 110301, Jan. 2021.

[2] C. Wu, C. Cheng, Y. Uwate, G. Chen, S. Mumtaz, and F. Tsang, "State-of-the-art and research opportunities for next-generation consumer electronics," IEEE Trans. Cons. Elec., vol. 69, no. 4, pp. 937-948, Nov. 2023.

[3] A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," Computer Networks, vol. 167, pp. 1-40, 2020.

[4] S. Yang, F. Li, S. Trajanovski, R. Yahyapour, and X. Fu, "Recent advances of resource allocation in network function virtualization," IEEE Trans. Para. Distr. Sys., vol. 32, no. 2, pp. 295-314, Feb. 2021.

[5] T. Li, X. Zhu, and X. Liu, "An end-to-end network slicing algorithm based on deep Q-learning for 5G network," IEEE Access, vol. 8, pp. 122229-122240, 2020.

[6] H. Cao et al., "Resource-ability assisted service function chain embedding and scheduling for 6G networks with virtualization," IEEE Trans. Veh. Tech., vol. 70, no. 4, pp. 3846-3859, Apr. 2021.

[7] Y. Chen, Y. Wang, M. Liu, J. Zhang, and L. Jiao, "Network slicing enabled resource management for service-oriented ultra-reliable and low-latency vehicular networks," IEEE Trans. Veh. Tech., vol. 69, no. 7, pp. 7847-7862, Jul. 2020.

[8] H. Cao et al., "Toward tailored resource allocation of slices in 6G Networks with softwarization and virtualization," IEEE Internet of Things Journal, vol. 9, no. 9, pp. 6623-6637, May, 2022.

[9] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey", IEEE Trans. Netw. Service Manag., vol. 13, no. 3, pp. 518-532, Sep. 2016.

[10] P. Zhang, C. Wang, C. Jiang, and A. Benslimane, "Security-aware virtual network embedding algorithm based on reinforcement learning," IEEE Trans. Net. Sci. Engi., vol. 8, no. 2, pp. 1095-1105, Apr. 2021.

[11] A. Rasheed, O. San, and T. Kvamsdal, "Digital twin: Values, challenges and enablers from a modeling perspective," IEEE Access, vol. 8, pp. 21980-22012, 2020.

[12] M. Shafto et al., Modeling, Simulation, Information Technology and Processing Roadmap, National Aeronautics and Space Administration, Washington, DC, USA, 2012.

[13] E. J. Tuegel, A. R. Ingraffea, T. G. Eason, and S. M. Spottswood, "Reengineering aircraft structural life prediction using a digital twin," Int. J. Aerosp. Eng., vol. 2011, Oct. 2011, Art. no. 154798.

[14] "Global horizons final report," United States Air Force, Glob. Sci. Technol. Vis., Washington, DC, USA, Tech. Rep. AF/ST TR 13-01, 2013.
[15] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable," in Undesirable Emergent Behavior in Complex Systems, Cham, Switzerland: Springer, 2017, pp. 85–113.
[16] Y. Wu, K. Zhang, Y. Zhang, "Digital twin networks: A survey," IEEE Internet of Thnigs Journal, vol. 8, no. 18, pp. 13789-13804, Sept. 2021.
[17] H. Cao, Z. Lin, L. Yang, J. Wang, and M. Guizani, "DT-SFC-6G: Digital twins assisted service function chains in softwarized 6G networks for emerging V2X," IEEE Network, vol. 37, no. 4, pp. 289-296, July/August 2023.
[18] X. Xu, Z. Liu, M. Bilal, S. Vimal, and H. Song, "Computation offloading and service caching for intelligent transportation systems With digital twin," IEEE Trans. Intell. Trans. Sys., vol. pp. no. 99, pp. 1-16, Jul. 2022.
[19] H. Wang, Y. Wu, et al., "A graph neural network-based digital twin for network slicing management," IEEE Trans. Indus. Inform., vol. 18, no. 2, pp. 1367-1376, Feb. 2021.
[20] F. Naeem, G. Kaddoum, and M. Tariq, "Digital twin-empowered network slicing in B5G networks: Experience-driven approach," in Proc. of IEEE GC Wkshps, pp. 1-5, Dec. 2021.
[21] G. Mirjalily, and Z. Luo, "Optimal network function virtualization and service function chaining: A survey," Chinese Jour. Elec., vol. 27, no. 4, pp. 704-717, 2018.
[22] Q. Li, C. Li, J. Zhang, H. Chen, and S. Wang, "A review of deep neural network compression," Comput. Sci., vol. 46, no. 9, pp. 1-14, 2019.
[23] H. Cao, S. Wu, Y. Hu, Y. Liu, and L. Yang, "A survey of embedding algorithm for virtual network embedding," China Commun., vol. 16, no. 12, pp. 1-33, Nov. 2019.
[24] M. Newman, Networks: An Introduction. Oxford University Press, Oxford, UK, 2010.
[25] H. Tuy, Convex Analysis and Global Optimization, Kluwer Academic, 1998.
[26] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[27] A. Schrijiver, Theory of Linear and Integer Programming. New York, NY, USA: Wiley, 1998.
[28] (Jul. 2022). IBM ILOG Optimization Products. [Online]. Available: www=01.ibm.com/software/websphere/products/optimization
[29] (Jul. 2022). GLPK(GNU Linear Programming Kit). [Online]. Available: http://www.gnu.org/software/glpk/
[30] T. H. Cormen, C. Stein, R. Rivest, and C. Leiserson, Introduction to Algorithms, 2nd ed. McGraw-Hill Higher Education, 2001.
[31] M. Newman, Networks: An Introduction. Oxford University Press, Oxford, UK, 2010.
[32] H. Cao, S. Garg, G. Kaddoum, M. Alrashoud, and L. Yang, "Efficient resource allocation of slicing services in softwarized space-aerial-ground integrated networks for seamless and open access services," IEEE Trans. Veh. Tech., vol. pp, no. 99, no. 1-12, Oct. 2023.
[33] H. Cao, A. Xiao, Y. Hu, P. Zhang, S. Wu, and L. Yang, "On virtual resource allocation of heterogeneous networks in virtualization environment: A service oriented perspective," IEEE Trans. Netw. Sci. Eng., vol. 7, no. 4, pp. 4268-4280, Dec. 2020.
[34] H. Cao, S. Hu, and L. Yang, "New functions added to ALEVIN for evaluating virtual network embedding," in Proc. of IEEE ICCC, pp. 2411-2414, Oct. 2016.
[35] G. Sun, K. Xiong, G. Boateng, et al., "Resource slicing and customization in RAN with dueling deep Q-network," Journal of Network and Computer Applications, vol. 157, pp.102573, 2020.
[36] K. Singh, P. Wang, S. Biswas, S. Singh, S. Mumtaz, and C. Li, "Joint active and passive beamforming design for RIS-aided IBFD IoT communications: QoS and power efficiency considerations," IEEE Trans. Con. Elec., vol. 69, no. 2, pp. 170-182, May 2023.
[37] G. Sun, K. Xiong, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and W. Jiang, "Autonomous resource provisioning and resource customization for mixed traffics in virtualized radio access network," IEEE Systems Journal, vol. 13, no. 3, pp. 2454-2465, Sept. 2019.