Multiservice Function Chain Embedding With Delay Guarantee: A Game-Theoretical Approach

Junling Li, *Graduate Student Member, IEEE*, Weisen Shi, *Graduate Student Member, IEEE*, Qiang Ye, *Member, IEEE*, Ning Zhang, *Senior Member, IEEE*, Weihua Zhuang, *Fellow, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Through network function virtualization (NFV), virtual network functions (VNFs) can be mapped onto substrate networks as service function chains (SFCs) to provide customized services with guaranteed Quality of Service (QoS). In this article, we solve a multi-SFC embedding problem by a game-theoretical approach considering the heterogeneity of NFV nodes, the effect of processing-resource sharing among various VNFs, and the capacity constraints of NFV nodes. Specifically, each SFC is treated as a player whose objective is to minimize the overall latency experienced by the supported service flow, while satisfying the capacity constraints of all NFV nodes. Due to processing-resource sharing, additional delay is incurred and incorporated into the overall latency for each SFC. The capacity constraints of NFV nodes are considered by adding a penalty term into the cost function of each player, and are guaranteed by a prioritized admission control mechanism. We prove that the formulated resource-constrained multi-SFC embedding game (RC-MSEG) is an exact potential game admitting at least one pure Nash equilibrium (NE) and has the finite improvement property (FIP). Two iterative algorithms are developed, namely, the best response (BR) algorithm with fast convergence and the spatial adaptive play (SAP) algorithm with great potential to obtain the best NE. Simulations are conducted to demonstrate the effectiveness of the proposed game-theoretical approach.**

***Index Terms*—End-to-end (E2E) delay, Nash equilibrium (NE), potential game, service function chain (SFC) embedding.**

I. Introduction

With the evolution of Internet of Things (IoT), the next-generation communication networks are foreseen to demonstrate new and unique features [1]. Billions of smart sensors will be deployed on heterogeneous types of end terminals, such as self-driving vehicles, home appliances, public facilities, and wearable devices, to realize smart data collection and ubiquitous information sharing [2]. Ultrareliable and low-latency communication (URLLC) IoT services, such as remote control, industrial automation, and e-health care, are expected to be supported, while enhanced mobile broadband (eMBB) data communication services (e.g., augmented reality, virtual reality, and high-definition video conferencing) are foreseen to prevail. Supporting an integration of IoT and mobile broadband data services requires heterogeneous types of resources (e.g., communication, computing, and storage resources) and various Quality-of-Service (QoS) guarantee in terms of reliability, latency, and data rate [3]. Some computing-intensive IoT services, e.g., environment sensing for autonomous driving, not only demands reliable data transmission but also requires fast response for task processing/computation. Therefore, the new network and service features bring about new technical challenges for the network design and resource allocation. First, to support massive and differentiated services, network deployment should be highly cost effective; second, to accommodate high traffic volume for service delivery, the resource utilization should be significantly enhanced; and third, to support IoT applications with heterogeneous resource demands and multidimensional QoS requirements, new technologies are needed for fine-grained resource orchestration and customized QoS provisioning.

Network slicing is one of the enablers to achieve this goal [4], [5]. Through network slicing, the heterogeneous network resources are virtualized and partitioned to create multiple virtual networks, and each customized service with specific QoS requirement is supported by one virtual network. Multiple virtual networks are conceptually isolated for differentiated QoS satisfaction, but can coexist over the same virtualized resource pool to improve the overall utilization. Network function virtualization (NFV) is recognized as an indispensable component of network slicing for supporting customized IoT applications with heterogeneous types of resources and QoS requirements [6]. Through NFV, network functions can be virtualized, embedded, and chained among high capacity commodity servers, rather than dedicated devices in current networks. As a result, one virtual network is regarded as a service function chain (SFC) composed by virtual nodes interconnected by virtual links in between. Driven by the NFV platform, the future network architecture is foreseen to feature reduced capital and operational cost, flexible network function placement, and enhanced service quality [7].

Manuscript received May 22, 2020; revised August 5, 2020; accepted January 7, 2021. Date of publication January 18, 2021; date of current version July 7, 2021. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, and in part by the Faculty Research Grant (FRG) from Minnesota State University, Mankato. *(Corresponding author: Weisen Shi.)*

Junling Li, Weisen Shi, Weihua Zhuang, and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: j742li@uwaterloo.ca; w46shi@uwaterloo.ca; wzhuang@uwaterloo.ca; sshen@uwaterloo.ca).

Qiang Ye is with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, MN 56001 USA (e-mail: qiang.ye@mnsu.edu).

Ning Zhang is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada (e-mail: ning.zhang@uwindsor.ca).

Digital Object Identifier 10.1109/JIOT.2021.3051905

Efficient management of physical resources is one of the most crucial issues for NFV-enabled networks. Specifically, given the QoS requirements, traffic statistics of heterogeneous services, and resource constraints of the substrate network, SFCs are embedded onto the substrate network, referred to as *SFC embedding*. During this procedure, the physical resources need to be allocated efficiently to meet the resource and QoS requirements of each SFC [8]. The SFC embedding determines the routing path from the source node to the destination node and finds the locations of NFV nodes to host the virtual network functions (VNFs) along the path, which increases the resource utilization with guaranteed QoS [9].

The existing research devoted to the SFC embedding problem can be categorized into centralized and distributed approaches. For the centralized approaches, a central entity holds a global view over the whole substrate network and orchestrates the VNFs to support all the SFCs [10]–[15]. These works either seek for optimal solutions through optimization or attempt to obtain near-optimal solutions by proposing heuristic algorithms. Centralized approaches are typically developed from a service provider's point of view to achieve the network-wide optimization on resource utilization with high computational complexity, which may not be scalable for large-scale networks. Distributed approaches, however, do not require a centralized controller, and allow each SFC to find its own embedding strategy independently [16]–[24]. Among these approaches, game theory has been adopted as a powerful mathematical tool to address the SFC embedding problem for rational entities with conflicting objectives [25]. Once the system reaches the Nash equilibrium (NE), no player has the intention to change the strategy unilaterally. In that sense, the SFC embedding problem is usually formulated from the user's perspective in a distributed manner with the objective of minimizing/maximizing their individual costs/payoffs. Compared with centralized approaches, performing SFC embedding in a distributed manner requires less information exchange between the centralized controller and the users, thereby greatly reducing the signaling overhead. However, the global optimality of the solution is not easily guaranteed. For achieving customized service provisioning in complex and large-scale networks, distributed approaches can be effective in balancing the tradeoff between computational complexity and solution optimality.

In this article, we propose a game-theoretical approach for multi-SFC embedding, considering the effect of processing-resource sharing among different VNFs and the capacity constraints of different NFV nodes. Each SFC is treated as a player aiming to minimize the overall latency experienced by the traffic flow traversing the SFC. The formulated game is a resource-specific congestion game in that the NFV nodes are considered to be heterogeneous in terms of the ability of hosting VNFs, the time for processing data traffic of each VNF, and the latency parameter used to evaluate the effect of processing-resource sharing. Our main contributions are fourfold.

1) We formulate the multi-SFC embedding problem as a resource-specific congestion game, which is proved to be an exact potential game. In the multi-SFC embedding game (MSEG), the effect of processing-resource sharing among VNFs belonging to different SFCs is included. Considering the additional delay caused by processing-resource sharing, we establish an accurate end-to-end (E2E) delay model for each SFC with delay satisfaction.
2) To guarantee the capacity constraints on different NFV nodes, we formulate a new resource constrained MSEG (RC-MSEG) where a penalty term is added into the original cost function of each player. Both MSEG and 0 are proved to be exact potential games admitting at least one pure NE.
3) To obtain the NE of RC-MSEG, we design two iterative algorithms, namely, the best response (BR) algorithm with fast convergence and the spatial adaptive play (SAP) algorithm with great potential to obtain the best NE. The two algorithms obtain local and global optimal solutions to the original problem, respectively.
4) To further guarantee the capacity constraints of NFV nodes as well as to achieve better load balancing, we propose a prioritized admission control (AC) mechanism where the maximum number of embedded SFCs is predicted. Then, the size of the original SFC set is adjusted to accommodate the maximum number of SFCs with high bit rates, while all capacity constraints are satisfied.

The remainder of this article is organized as follows. In Section II, an overview of the related works is provided. In Section III, system model and the formulation of the potential game are presented in detail. In Section IV, we propose two iterative algorithms to obtain the NE of the resource-constrained MSEG. A prioritized AC mechanism is designed to deal with over-loaded service requests. In Section V, simulation results are presented to validate the effectiveness of the proposed game-theoretical approach, followed by conclusions of this study in Section VI.

II. Related Work

A. Centralized Approaches for SFC Embedding

A centralized approach usually addresses SFC embedding by formulating the embedding process as an optimization problem. The objectives of the problem can be: 1) maximizing the number of accommodated SFCs; 2) minimizing the average long-run embedding cost; and 3) minimizing the E2E latency of the flow passing each SFC. For example, in [10], the SFC embedding problem has been formulated as an integer linear program (ILP) whose objective is to minimize the number of VNF instances deployed onto the substrate network. To enhance scalability, a binary search-based heuristic algorithm has been presented to quickly obtain feasible and high-quality solutions to ILP. In [11], a coordinated approach has been studied to jointly optimize SFC composition, embedding, and scheduling. The former two phases are dealt with by a mixed integer linear programming (MILP) formulation, where the objective is to minimize the total embedding cost, while the last phase is addressed by a one-hop scheduling algorithm. In [12], SFC embedding is modeled as a combination of two separate problems, i.e., the facility location problem and the generalized assignment problem. The overall objective is to minimize the total system cost that includes the cost for network function initiation and the cost for traffic delivery. The linear relaxation technique is used to address the NP-hardness of the established ILP. In [13], the SFC embedding problem has been formulated as an integer program, where the objective is to minimize the latency. While these studies focus on SFC embedding, some other works jointly address SFC embedding with other issues of NFV resource allocation. In [14], SFC topology design and SFC embedding have been jointly studied to minimize the total link bandwidth consumption. A heuristic algorithm is proposed to coordinate the two processes using the feedback from embedding the key submodules of an SFC.

The existing centralized approaches face the challenges of how to deal with the computational complexity that increases exponentially as the network size expands. Moreover, the availability of a centralized entity collecting the information of all SFCs existing in the entire network needs to be stated. In reality, different SFCs could be controlled by operators from different network domains, which motivates researchers to rethink the SFC embedding problem and investigate the possibility of conducting SFC embedding in a distributed manner.

B. Distributed Approaches for SFC Embedding

Among all the distributed approaches for SFC embedding, game theoretical approach is the most widely adopted one due to its great potential to model the strategic interaction among a number of decision makers. Recently, some research activities have been carried out to develop game-theoretical approaches for SFC embedding [16]–[22].

In [16], the process of NFV servers providing network functions to users has been modeled as a two-stage Stackelberg game, where users and servers are regarded as followers and leaders of the game, respectively. Similar to [16], Chen *et al.* [17] have leveraged a mixed-strategy gaming approach to facilitate SFC provisioning over interdatacenter elastic optical networks, where resource brokers and users play the leader game and the follower game, respectively. This work has been extended in [18] from a single-broker scenario to a multibroker scenario. In [19], the SFC embedding problem has been envisioned as a cooperative graph partitioning game and a heuristic algorithm has been designed to achieve the NE corresponding to the optimal solution.

On the other hand, the SFC embedding problem has been formulated as a congestion game in [20]–[22]. D'Oro *et al.* first modeled the SFC embedding problem as a weighted congestion game in [20]. The game is proved to be a weighted potential game, and therefore admits at least one pure strategy NE. Following the study in [20], Bian *et al.* have proved in [21] that with the consideration of user and resource failures in the NFV system, the weighted congestion game is still a weighted potential game, in which the NE can be obtained with their proposed distributed algorithm. Le *et al.* [22] have developed a specific type of congestion game, called congestion game with player-specific (CGPS) utility functions, which has been proved to be a weighted potential game. Then, they apply CGPS to formulate the SFC embedding problem where different traffic flows have different priorities.

Some machine learning-based approaches that adapt to dynamic environments and uncertain network conditions have been proposed [23], [24]. For example, in [24], a decentralized optimization framework that addresses multidomain SFC embedding for wireless networks has been presented. The framework exploits a alternating directions dual decomposition (AD^3) algorithm to achieve consensus among different operators with guaranteed convergence.

The latency requirement is one of the most important metrics for NFV-enabled service provisioning. Due to processing-resource sharing, additional latency is introduced during the execution of network services, which involve multiple VNFs running on the same NFV node [26]. Thus, the effect of processing-resource sharing needs to be incorporated when dealing with the multi-SFC embedding problem. Moreover, to achieve better load-balancing among NFV nodes, the capacity constraints of NFV nodes need to be considered in the problem formulation.

In this article, we aim to address a multiservice function chain (SFC) embedding problem by using a game-theoretical approach, which is executed in a distributed manner. In comparison with existing centralized approaches (such as [16]–[24]), the proposed approach does not require a centralized controller to perform central optimization, and thus has low computational complexity and better scalability. Compared with existing game-theoretical approaches (such as [16]–[22]), the proposed approach has a different focus to minimize the E2E delay of each SFC. Moreover, we incorporate the effect of processing-resource sharing into the E2E delay modeling and consider the capacity constraints of NFV nodes in the multi-SFC embedding problem formulation. A novel prioritized AC mechanism is further proposed to deal with overloaded service requests.

III. System Model and Game Formulation

In this section, we present the system model and the game formulation for the multi-SFC embedding problem. Table I lists the important notations used in the following sections.

A. Network Model

We consider an NFV system with a set of NFV nodes/servers denoted by $\mathcal{V}$, where $\mathcal{V} = \{1, 2, \ldots, V\}$, and a set of SFCs denoted by $\mathcal{N}$, where $\mathcal{N} = \{1, 2, \ldots, N\}$. V and N represent the number of NFV nodes and the number of SFCs in the system, respectively. Let $\mathcal{M}$ be the set of VNF types available in the whole system, where $\mathcal{M} = \{1, 2, \ldots, M\}$. Each SFC $i \in \mathcal{N}$ and is composed by a set of VNFs $F_i = \{f_{i1}, f_{i2}, \ldots, f_{i|F_i|}\}$ which has a one-to-one mapping on a set of VNF types $\mathcal{M}_i = \{m_{i1}, m_{i2}, \ldots, m_{i|F_i|}\}$, where $m_{ij} \in \mathcal{M}$ ($j = 1, \ldots, |F_i|$). Each SFC is supporting one service associated with a bit rate λ_i. Each NFV node is capable of hosting multiple VNFs of different types. Let $\mathcal{M}(v) \subset \mathcal{M}$ denote the set of VNF types that can be supported by NFV node $v \in \mathcal{V}$. Each VNF type $m \in \mathcal{M}$ can be hosted by a set of NFV nodes, denoted by $\mathcal{V}(m) \subset \mathcal{V}$. Let $\rho_{v,m}$ be the processing time for one information bit for a certain VNF type $m \in \mathcal{M}$ on NFV node $v \in \mathcal{V}(m)$. Accordingly, the processing time of VNF f_{ij} on

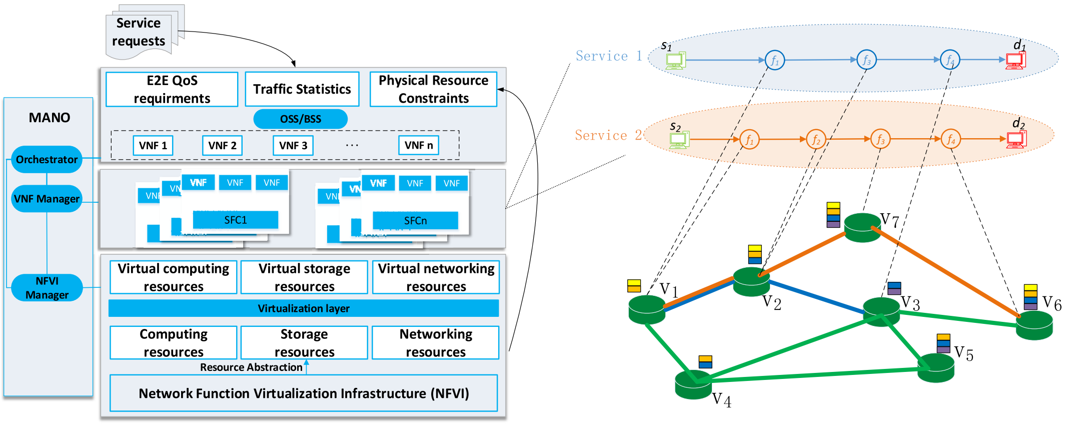

Fig. 1. Illustration of multi-SFC embedding where VNFs from different SFCs can share the physical resources of common NFV node.

TABLE I
SUMMARY OF IMPORTANT NOTATIONS

Notation	Description
$\mathcal{N}$	set of players (SFCs)
N	number of players (SFCs) in the system
F_i	set of VNFs in SFC i
f_{ij}	jth VNF in SFC i
$\mathcal{M}$	set of VNF types in the whole system
$\mathcal{M}_i$	set of VNF types in SFC i
$\mathcal{M}(v)$	set of VNF types that can be supported by v
m_{ij}	type of jth VNF in SFC i
$\mathcal{V}$	set of NFV nodes in the system
V	number of NFV nodes in the system
$\mathcal{V}(m)$	set of NFV nodes that can support VNF type m
λ_i	bit rate of player $i \in N$
$\delta(v_1, v_2)$	link propagation delay over the link between $v_1 \in \mathcal{V}$ and $v_2 \in \mathcal{V}$
μ_v	maximum service rate of NFV node $v \in \mathcal{V}$
$\rho_{v,m}$	processing time for one information bit for VNF type $m \in \mathcal{M}$ on NFV node $v \in \mathcal{V}$
$\rho(v, f_{ij})$	processing time for VNF f_{ij} on NFV node $v \in \mathcal{V}$
K_v	context-switching latency parameter for $v \in \mathcal{V}$
$\lvert\Upsilon_v\rvert$	number of SFCs that chooses server v to support a certain VNF
$d_i^{(prop)}(\bar{s}_i)$	total link propagation delay experienced by traffic flow i of strategy $\bar{s}_i$
$d_i^{(proc)}(\bar{s}_i)$	total VNF processing delay experienced by traffic flow i of strategy $\bar{s}_i$
$d_i^{(cs)}(\bar{s}_i, \boldsymbol{s}_{-i})$	additional context switching delay experienced by traffic flow i of strategy profile $(\bar{s}_i, \boldsymbol{s}_{-i})$

node $v \in \mathcal{V}(m_{ij})$ is given by

$$\rho\left(v, f_{ij}\right) = \rho_{v,m_{ij}}\lambda_i. \tag{1}$$

Note that different NFV nodes have different abilities of processing VNFs of the same type. Denote the maximum service rate that NFV node v can offer by μ_v. Let $\delta(v_1, v_2)$ represent the link propagation delay over the link between two adjacent NFV nodes v_1 and v_2, where $v_1 \in \mathcal{V}$ and $v_2 \in \mathcal{V}$.

B. Multi-SFC Processing-Resource Sharing Model

To reduce the VNF provisioning cost, we consider that multiple VNFs belonging to different SFCs can be embedded onto a common NFV server. However, embedding multiple VNFs on the same NFV node may result in performance degradation due to context switching among different VNFs [26]. The performance degradation is reflected in an increased VNF processing latency, referred to as the *context switching* delay. Therefore, to manage the physical resources efficiently and to achieve a better latency control over the traffic flows, the effect of processing-resource sharing should be taken into account when dealing with the multi-SFC embedding problem.

Let $d^{(cs)}(v)$ denote the additional context-switching delay imposed on the processing of all the VNFs embedded onto v. According to [26], this additional delay can be modeled as

$$d^{(\mathrm{cs})}(v) = |\Upsilon_v| K_v \tag{2}$$

where $|\Upsilon_v|$ represents the number of SFCs that chooses server v to support a certain VNF; and K_v is the context-switching latency parameter of NFV node $v \in \mathcal{V}$.

Fig. 1 gives an example of embedding multiple SFCs onto the same substrate network, where the processing resources of a common NFV node can be shared by multiple VNFs from different SFCs. In the figure, we assume that all the substrate nodes are NFV nodes equipped with multiple CPU cores. Suppose that we have two SFCs to be embedded onto the substrate network. For a given service, the source and the destination nodes in the substrate network are known. Let $S_1 = \{s_1, f_1, f_3, f_4, d_1\}$ and $S_2 = \{s_2, f_1, f_2, f_3, f_4, d_2\}$ denote the first and the second SFCs, respectively (other available information associated with the SFC are omitted for brevity). f_1 in both SFC 1 and SFC 2 is embedded on the NFV node v_1. They will share the processing and other physical (such as storage and disk) resources of v_1. The impact of this resource sharing among different VNFs is an increase on the E2E latency of both services. Meanwhile, f_3 in SFC 1 and f_2 in SFC 2 are both embedded onto NFV node v_2, which also leads to latency increase for the two services.

C. Delay Modeling and Capacity Constraints

We refer to the ith SFC as player i with its embedding strategy being denoted by $\bar{s}_i$. Specifically, $\bar{s}_i = (s_{i1}, s_{i2}, \ldots, s_{i|F_i|})$, where $s_{ij} \in \mathcal{V}$ represents the NFV node chosen by player i

to process f_{ij}, $j = 1, \ldots, |F_i|$. Let S_i represent the strategy set of player i, i.e., $\bar{s}_i \in S_i$. We assume that different VNFs from the same SFC cannot be embedded onto the same NFV node. Therefore, the total number of elements in S_i is given by $|S_i| = |\mathcal{V}(m_{i1})| \times |\mathcal{V}(m_{i2})| \times \cdots \times |\mathcal{V}(m_{i|F_i|})|$.

For any player $i \in \mathcal{N}$ and any strategy $\bar{s}_i \in S_i$, the total link propagation delay experienced by the traffic flow for player i is given by

$$d_i^{(\text{prop})}(\bar{s}_i) = \sum_{j=1}^{|F_i|-1} \delta\big(s_{ij}, s_{i,j+1}\big) \tag{3}$$

where $\delta(s_{ij}, s_{i,j+1})$ represents the link propagation delay between the two NFV nodes s_{ij} and $s_{i,j+1}$.

Based on (1), the overall delay for processing all the VNFs in SFC i under strategy $\bar{s}_i$ is given by

$$d_i^{(\text{proc})}(\bar{s}_i) = \sum_{j=1}^{|F_i|} \rho_{s_{ij}, m_{ij}} \lambda_i. \tag{4}$$

Let $\boldsymbol{s}_{-i}$ be the set containing the strategies chosen by all the players expect for player i, i.e., $\boldsymbol{s}_{-i} = \{\bar{s}_1, \bar{s}_2, \ldots, \bar{s}_{i-1}, \bar{s}_{i+1}, \ldots, \bar{s}_N\}$. Considering that multiple VNFs managed by different players can be embedded onto the same NFV node, we denote $\Upsilon_v(\bar{s}_i, \boldsymbol{s}_{-i})$ as the set containing the players that choose server v to execute a certain VNF. That is

$$\begin{aligned}\Upsilon_v(\bar{s}_i, \boldsymbol{s}_{-i}) = \big\{i' \in \mathcal{N} : \exists j \in \{1, \ldots, |F_{i'}|\}, s_{i'j} = v \\ s_{i'j} \in \bar{s}_{i'}, \bar{s}_{i'} \in (\bar{s}_i, \boldsymbol{s}_{-i})\big\}.\end{aligned} \tag{5}$$

According to (2), the additional delay experienced by the traffic flow of player i due to context switching is given by

$$d_i^{(\text{cs})}(\bar{s}_i, \boldsymbol{s}_{-i}) = \sum_{v \in \bar{s}_i} |\Upsilon_v(\bar{s}_i, \boldsymbol{s}_{-i})| K_v \tag{6}$$

where $|\cdot|$ represents the cardinality of a set.

Finally, to ensure that the capacity constraint on each NFV node can be satisfied, the total bit rates from different SFCs embedded onto the same server should not exceed the maximum service rate that can be provided by that server, i.e.,

$$\sum_{i' \in \Upsilon_v(\bar{s}_i, \boldsymbol{s}_{-i})} \lambda_{i'} \leq \mu_v \quad \forall v \in \mathcal{V}. \tag{7}$$

D. Game Formulation

The objective of player i is to minimize the overall latency experienced by its supported traffic flow, while satisfying the capacity constraints of all the NFV nodes, i.e.,

$$\begin{aligned} &\min_{\bar{s}_i \in S_i} D_i(\bar{s}_i, \boldsymbol{s}_{-i}) \\ \text{s.t.} &\sum_{j \in \Upsilon_v(\bar{s}_i, \boldsymbol{s}_{-i})} \lambda_j \leq \mu_v \quad \forall v \in V \end{aligned} \tag{8}$$

where $D_i(\bar{s}_i, \boldsymbol{s}_{-i})$ represents the overall latency experienced by the traffic flow managed by player i with the strategy profile being $(\bar{s}_i, \boldsymbol{s}_{-i})$. The overall latency for an embedded SFC is the sum of the total link propagation delay, the total processing delay, and the total additional context-switching delay imposed on all the VNFs in the SFC, i.e.,

$$D_i(\bar{s}_i, \boldsymbol{s}_{-i}) = d_i^{(\text{prop})}(\bar{s}_i) + d_i^{(\text{proc})}(\bar{s}_i) + d_i^{(\text{cs})}(\bar{s}_i, \boldsymbol{s}_{-i}). \tag{9}$$

Accordingly, the multi-SFC embedding problem without considering the capacity constraints can be formulated as a congestion game: $\mathcal{G} = \{\mathcal{N}, \mathcal{V}, \mathcal{S}, (D_i(\bar{s}_i, \boldsymbol{s}_{-i}))_{i \in \mathcal{N}}\}$, where $\mathcal{N}$ represents the set of players, $\mathcal{V}$ the set of NFV servers, and $D_i(\bar{s}_i, \boldsymbol{s}_{-i})$ the cost function of player i as given in (9). $\mathcal{S}$ denotes the set of all the possible strategy profiles, i.e., $\mathcal{S} = S_1 \bigotimes S_2 \bigotimes \cdots \bigotimes S_N$. We refer to $\mathcal{G}$ as MSEG in this article. Notably, MSEG is a resource-specific congestion game in that different NFV nodes have different ability of hosting VNFs ($\mathcal{M}(v)$), different processing times for the same VNF type ($\rho_{v,m}$), different maximum service rates (μ_v), and different context-switching latency parameters (K_v).

To relax the capacity constraints of NFV nodes, we modify the original cost function of each player by adding a penalty term as [27]–[29]

$$\bar{D}_i(\bar{s}_i, \boldsymbol{s}_{-i}) = D_i(\bar{s}_i, \boldsymbol{s}_{-i}) + wP(\bar{s}_i, \boldsymbol{s}_{-i}) \tag{10}$$

where w is the weighting parameter for the penalty term $P(\bar{s}_i, \boldsymbol{s}_{-i})$. The penalty term reflects the violation degree of the capacity constraints for all the NFV nodes, given by

$$P(\bar{s}_i, \boldsymbol{s}_{-i}) = \sum_{v \in V} p_v(\bar{s}_i \boldsymbol{s}_{-i}) \tag{11}$$

where $p_v(\bar{s}_i, \boldsymbol{s}_{-i})$ is defined as

$$p_v(\bar{s}_i, \boldsymbol{s}_{-i}) = \begin{cases} M, & \text{if } \sum_{i' \in \Upsilon_v(\bar{s}_i, \boldsymbol{s}_{-i})} \lambda_{i'} > \mu_v \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Here, M is a positive number used to penalize the strategy $\bar{s}_i$ if the capacity constraint on server v is violated. Based on (10)–(12), we reformulate the multi-SFC embedding problem subject to the capacity constraints as a new congestion game: $\bar{\mathcal{G}} = \{\mathcal{N}, \mathcal{V}, \mathcal{S}, (\bar{D}_i(\bar{s}_i, \boldsymbol{s}_{-i}))_{i \in \mathcal{N}}\}$, where $\bar{D}_i(\bar{s}_i, \boldsymbol{s}_{-i}))$ is the modified cost function given by (10). We refer to $\bar{\mathcal{G}}$ as RC-MSEG in this article.

Definition 1: A strategy profile $(\bar{s}_1^*, \bar{s}_2^*, \ldots, \bar{s}_N^*) \in \mathcal{S}$ is an NE if and only if

$$D_i\big(\bar{s}_i^*, \boldsymbol{s}_{-i}^*\big) \leq D_i\big(\bar{s}_i, \boldsymbol{s}_{-i}^*\big) \quad \forall i \in \mathcal{N} \quad \forall \bar{s}_i \in S_i \tag{13}$$

which means that $(\bar{s}_1^*, \bar{s}_2^*, \ldots, \bar{s}_N^*)$ is a strategy profile where no player has incentive to change its strategy since its overall latency cannot be decreased unilaterally.

Definition 2: A game is an exact potential game if there is a function, Φ, such that $\forall \boldsymbol{s}_{-i} \in S_{-i}$ $\forall \bar{a}, \bar{b} \in S_i$, the following equation holds:

$$D_i(\bar{a}, \boldsymbol{s}_{-i}) - D_i\big(\bar{b}, \boldsymbol{s}_{-i}\big) = \Phi(\bar{a}, \boldsymbol{s}_{-i}) - \Phi\big(\bar{b}, \boldsymbol{s}_{-i}\big) \tag{14}$$

where D_i is the cost function of player i; $\bar{a}$ and $\bar{b}$ are two strategies chosen by player i. The function Φ is called the exact potential function of the game.

Theorem 1: The formulated game MSEG ($\mathcal{G}$) is an exact potential game admitting at least one pure strategy NE.

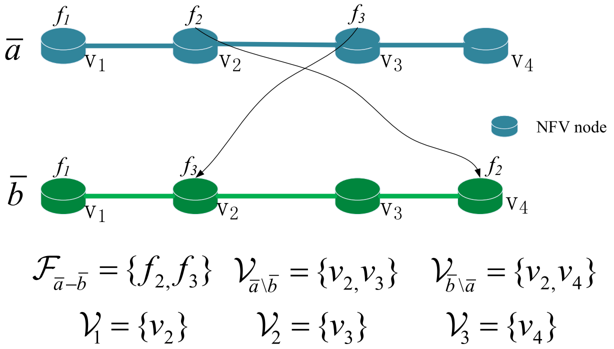

Fig. 2. Illustration of the definitions of $\mathcal{F}_{\bar{a}-\bar{b}}$, $\mathcal{V}_{\bar{a}\backslash\bar{b}}$, and $\mathcal{V}_{\bar{b}\backslash\bar{a}}$.

Proof: Define the potential function Φ for game $\mathcal{G}$ as follows:

$$\Phi(\bar{s}_i,\boldsymbol{s}_{-i})=\sum_{i\in\mathcal{N}}\tilde{d}_i(\bar{s}_i)+\sum_{v\in\mathcal{V}}\sum_{n=1}^{|\Upsilon_v(\bar{s}_i,\boldsymbol{s}_{-i})|}nK_v \tag{15}$$

where $|\Upsilon_v(\bar{s}_i,\boldsymbol{s}_{-i})|$ represents the number of players that choose v to process a certain VNF, and $\tilde{d}_i(\bar{s}_i)$ is defined as

$$\tilde{d}_i(\bar{s}_i)=d_i^{(\text{prop})}(\bar{s}_i)+d_i^{(\text{proc})}(\bar{s}_i). \tag{16}$$

Suppose that player i changes its strategy from $\bar{a}$ to $\bar{b}$ and that other players keep their strategies unchanged. From (9), we obtain the change of the cost function for player i as follows:

$$\begin{aligned}&D_i(\bar{a},\boldsymbol{s}_{-i})-D_i\big(\bar{b},\boldsymbol{s}_{-i}\big)\\&\quad=\tilde{d}_i(\bar{a})-\tilde{d}_i\big(\bar{b}\big)+d_i^{(\text{cs})}(\bar{a},\boldsymbol{s}_{-i})-d_i^{(\text{cs})}\big(\bar{b},\boldsymbol{s}_{-i}\big)\\&\quad=\tilde{d}_i(\bar{a})-\tilde{d}_i\big(\bar{b}\big)\\&\qquad+\sum_{v\in\bar{a}}|\Upsilon_v(\bar{a},\boldsymbol{s}_{-i})|K_v-\sum_{v\in\bar{b}}\big|\Upsilon_v\big(\bar{b},\boldsymbol{s}_{-i}\big)\big|K_v.\end{aligned} \tag{17}$$

From (15), we obtain the change of the potential function Φ between two strategy profiles $(\bar{a},\boldsymbol{s}_{-i})$ and $(\bar{b},\boldsymbol{s}_{-i})$ as

$$\begin{aligned}&\Phi(\bar{a},\boldsymbol{s}_{-i})-\Phi\big(\bar{b},\boldsymbol{s}_{-i}\big)\\&\quad=\sum_{i\in\mathcal{N}}\tilde{d}_i(\bar{a})-\sum_{i\in\mathcal{N}}\tilde{d}_i\big(\bar{b}\big)\\&\qquad+\sum_{v\in\mathcal{V}}\left(\sum_{n=1}^{|\Upsilon_v(\bar{a},\boldsymbol{s}_{-i})|}nK_v-\sum_{n=1}^{|\Upsilon_v(\bar{b},\boldsymbol{s}_{-i})|}nK_v\right).\end{aligned} \tag{18}$$

Denote $\mathcal{F}_{\bar{a}-\bar{b}}$ as the set containing the VNFs of player i that are processed by different NFV nodes between $\bar{a}$ and $\bar{b}$, i.e., $\mathcal{F}_{\bar{a}-\bar{b}}=\{f_{ij}\in\mathcal{F}_i:\bar{a}(f_{ij})\neq\bar{b}(f_{ij})\}$, where $\bar{a}(f_{ij})$ and $\bar{b}(f_{ij})$ are the NFV nodes chosen to support f_{ij} of strategy $\bar{a}$ and strategy $\bar{b}$, respectively. Denote $\mathcal{V}_{\bar{a}\backslash\bar{b}}$ as the set of NFV nodes that are chosen to process the VNFs in $\mathcal{F}_{\bar{a}-\bar{b}}$ of strategy $\bar{a}$, i.e., $\mathcal{V}_{\bar{a}\backslash\bar{b}}=\{\bar{a}(f_{ij})\in\mathcal{V}:f_{ij}\in\mathcal{F}_{\bar{a}-\bar{b}}\}$. Similarly, denote $\mathcal{V}_{\bar{b}\backslash\bar{a}}$ as the set of NFV nodes that are chosen to process the VNFs in $\mathcal{F}_{\bar{a}-\bar{b}}$ of strategy $\bar{b}$, i.e., $\mathcal{V}_{\bar{b}\backslash\bar{a}}=\{\bar{b}(f_{ij})\in\mathcal{V}:f_{ij}\in\mathcal{F}_{\bar{a}-\bar{b}}\}$. Fig. 2 illustrates $\mathcal{F}_{\bar{a}-\bar{b}}$, $\mathcal{V}_{\bar{a}\backslash\bar{b}}$, and $\mathcal{V}_{\bar{b}\backslash\bar{a}}$ using a simple example with $|F_i|=3$ and $V=4$.

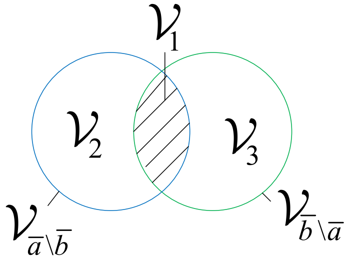
$\mathcal{V}_1$
$\mathcal{V}_2$
$\mathcal{V}_3$
$\mathcal{V}_{\bar{a}\backslash\bar{b}}$
$\mathcal{V}_{\bar{b}\backslash\bar{a}}$

Fig. 3. Illustration of the relationships between the sets $\mathcal{V}_{\bar{a}\backslash\bar{b}}$, $\mathcal{V}_{\bar{b}\backslash\bar{a}}$, $\mathcal{V}_1$, $\mathcal{V}_2$, and $\mathcal{V}_3$.

We can rewrite the change of the cost function of player i in (17) into

$$\begin{aligned}D_i(\bar{a},\boldsymbol{s}_{-i})-D_i\big(\bar{b},\boldsymbol{s}_{-i}\big)&=\tilde{d}_i(\bar{a})-\tilde{d}_i\big(\bar{b}\big)\\&\quad+\sum_{v\in\mathcal{V}_{\bar{a}\backslash\bar{b}}}|\Upsilon_v(\bar{a},\boldsymbol{s}_{-i})|K_v\\&\quad-\sum_{v\in\mathcal{V}_{\bar{b}\backslash\bar{a}}}\big|\Upsilon_v\big(\bar{b},\boldsymbol{s}_{-i}\big)\big|K_v.\end{aligned} \tag{19}$$

The change of the potential function Φ given by (18) can be now expressed as

$$\begin{aligned}&\Phi(\bar{a},\boldsymbol{s}_{-i})-\Phi\big(\bar{b},\boldsymbol{s}_{-i}\big)\\&\quad=\tilde{d}_i(\bar{a})-\tilde{d}_i\big(\bar{b}\big)\\&\qquad+\sum_{v\in\mathcal{V}_{\bar{a}\backslash\bar{b}}\cup\mathcal{V}_{\bar{b}\backslash\bar{a}}}\left(\sum_{n=1}^{|\Upsilon_v(\bar{a},\boldsymbol{s}_{-i})|}nK_v-\sum_{n=1}^{|\Upsilon_v(\bar{b},\boldsymbol{s}_{-i})|}nK_v\right).\end{aligned} \tag{20}$$

Now, let $\mathcal{V}_0=\mathcal{V}_{\bar{a}\backslash\bar{b}}\cup\mathcal{V}_{\bar{b}\backslash\bar{a}}$. Then, $\mathcal{V}_0$ can be considered as the union of three disjoint sets, i.e., $\mathcal{V}_0=\mathcal{V}_1\cup\mathcal{V}_2\cup\mathcal{V}_3$, where $\mathcal{V}_1=\mathcal{V}_{\bar{a}\backslash\bar{b}}\cap\mathcal{V}_{\bar{b}\backslash\bar{a}}$, $\mathcal{V}_2=\mathcal{V}_{\bar{a}\backslash\bar{b}}\setminus\mathcal{V}_1$, and $\mathcal{V}_3=\mathcal{V}_{\bar{b}\backslash\bar{a}}\setminus\mathcal{V}_1$. Fig. 3 illustrates the relationships among the sets $\mathcal{V}_{\bar{a}\backslash\bar{b}}$, $\mathcal{V}_{\bar{b}\backslash\bar{a}}$, $\mathcal{V}_1$, $\mathcal{V}_2$, and $\mathcal{V}_3$. Given an NFV node $v\in\mathcal{V}_0$, depending on which subset (among $\mathcal{V}_1$, $\mathcal{V}_2$, and $\mathcal{V}_3$) v belongs to, we have the following three cases to describe the relationship between $|\Upsilon_v(\bar{a},\boldsymbol{s}_{-i})|$ and $|\Upsilon_v(\bar{b},\boldsymbol{s}_{-i})|$.

Case 1: $v\in\mathcal{V}_1$, i.e., the NFV server v is chosen by player i of both $\bar{a}$ and $\bar{b}$, but to process two different VNFs. For the example shown in Fig. 2, $\mathcal{V}_1=\{v_2\}$ and this case refers to the NFV node v_2. In such a case, the number of VNFs received by v in the old strategy $\bar{a}$ and that in the new strategy $\bar{b}$ are the same. Accordingly, we have $|\Upsilon_v(\bar{a},\boldsymbol{s}_{-i})|=|\Upsilon_v(\bar{b},\boldsymbol{s}_{-i})|$.

Case 2: $v\in\mathcal{V}_2$, i.e., the NFV server v is chosen by player i to process a certain VNF in the old strategy $\bar{a}$, but that VNF is moved to a different server in the new strategy $\bar{b}$. As for the example shown in Fig. 2, this case refers to the NFV node v_3. In such a case, the number of VNFs received by v of strategy $\bar{a}$ is one more than that of strategy $\bar{b}$. Therefore, we have $|\Upsilon_v(\bar{a},\boldsymbol{s}_{-i})|=|\Upsilon_v(\bar{b},\boldsymbol{s}_{-i})|+1$.

Case 3: $v\in\mathcal{V}_3$, i.e., in the new strategy $\bar{b}$, player i places a certain VNF on server v, which was not chosen by the old strategy $\bar{a}$. This case refers to the NFV node v_4 for the example shown in Fig. 2. In this case, the number of VNFs received by v in $\bar{a}$ is one less than that in $\bar{b}$, i.e., $|\Upsilon_v(\bar{a},\boldsymbol{s}_{-i})|=|\Upsilon_v(\bar{b},\boldsymbol{s}_{-i})|-1$.

With the above three cases, we obtain the change of the potential function Φ in (20) as

$$\begin{aligned}
&\Phi(\bar{a}, \boldsymbol{s}_{-i}) - \Phi(\bar{b}, \boldsymbol{s}_{-i}) \\
&= \tilde{d}_i(\bar{a}) - \tilde{d}_i(\bar{b}) \\
&\quad + \sum_{v \in \mathcal{V}_2} |\Upsilon_v(\bar{a}, \boldsymbol{s}_{-i})| K_v - \sum_{v \in \mathcal{V}_3} \left|\Upsilon_v(\bar{b}, \boldsymbol{s}_{-i})\right| K_v \\
&= \tilde{d}_i(\bar{a}) - \tilde{d}_i(\bar{b}) \\
&\quad + \left(\sum_{v \in \mathcal{V}_2} |\Upsilon_v(\bar{a}, \boldsymbol{s}_{-i})| K_v + \sum_{v \in \mathcal{V}_1} |\Upsilon_v(\bar{a}, \boldsymbol{s}_{-i})| K_v \right) \\
&\quad - \left(\sum_{v \in \mathcal{V}_1} \left|\Upsilon_v(\bar{b}, \boldsymbol{s}_{-i})\right| K_v + \sum_{v \in \mathcal{V}_3} \left|\Upsilon_v(\bar{b}, \boldsymbol{s}_{-i})\right| K_v \right) \\
&= \tilde{d}_i(\bar{a}) - \tilde{d}_i(\bar{b}) \\
&\quad + \sum_{v \in \mathcal{V}_{\bar{a} \setminus \bar{b}}} |\Upsilon_v(\bar{a}, \boldsymbol{s}_{-i})| K_v - \sum_{v \in \mathcal{V}_{\bar{b} \setminus \bar{a}}} \left|\Upsilon_v(\bar{b}, \boldsymbol{s}_{-i})\right| K_v \\
&= D_i(\bar{a}, \boldsymbol{s}_{-i}) - D_i(\bar{b}, \boldsymbol{s}_{-i}). \qquad (21)
\end{aligned}$$

Therefore, the potential function Φ is an exact potential function for the original game $\mathcal{G}$. ■

Theorem 2: The new game RC-MSEG ($\bar{\mathcal{G}}$) is also an exact potential game admitting at least one pure strategy NE.

Proof: See Appendix A for the proof. ■

Lemma 1: Every finite ordinal potential game has a pure strategy equilibrium.

Lemma 2: Every finite ordinal potential game has the finite improvement property (FIP).

Lemmas 1 and 2 provide two important properties for potential games. Lemma 1 shows the existence of pure-strategy NE of any potential game, while Lemma 2 provides an efficient way to find an NE. The FIP is a special property of potential games in which players can choose a better strategy to unilaterally update their strategies and *improvement path* is defined as a sequence of unilateral strategy updates of players. Based on FIP, NE can be found by simple BR dynamics [30].

Since the game $\bar{\mathcal{G}}$ is proved to be an exact potential game, according to Lemmas 1 and 2, $\bar{\mathcal{G}}$ has at least one pure NE and also has the FIP property [30]–[32]. Therefore, it is guaranteed to find an NE in a finite number of iterations with unilateral improvement dynamics.

IV. Algorithm Design

In this section, we first employ the BR algorithm to seek for an NE of the game RC-MSEG. The BR algorithm, in general, achieves an NE fast, but may result in a local optimal solution at which the potential function is not maximized. Then, to improve the optimality, we further design a learning algorithm, namely, the SAP algorithm, to find the best NE of RC-MSEG.

A. Best Response Iterative Algorithm

For every finite ordinal potential game, an NE point exists and every maximal improvement path will terminate at a pure NE. With the FIP property, the basic BR algorithm can be employed to find the NE of the potential game $\bar{\mathcal{G}}$. The BR algorithm is executed in a round-robin manner. In each round, one player is chosen to update its strategy to the best strategy such that the cost function is minimized, while other players keep their strategies unchanged. The algorithm is terminated once the stopping criterion is met (e.g., the value of potential function does not change for a number of successive rounds, or the maximum number of rounds is reached). The details of BR algorithm to find an NE of RC-MSEG are given by Algorithm 1.

```
Algorithm 1: BR Algorithm to Find the NE of RC-MSEG
Initialize the round counter $r \leftarrow 0$;
for $i$ in $\mathcal{N}$ do
    Randomly select a strategy from the whole strategy set to initialize $\bar{s}_i(0)$;
    Execute the initial strategy;
Set $r \leftarrow r + 1$;
while stopConditionNotMet() do
    for $i$ in $\mathcal{N}$ do
        Keep $\boldsymbol{s}_{-i}$ unchanged, i.e., $\boldsymbol{s}_{-i}(r) = \boldsymbol{s}_{-i}(r-1)$
        Loop all the possible strategies of player $i$ to find the best response strategy $\bar{s}_i^*$, i.e.
        $\bar{s}_i^* = \arg\min_{\bar{s}_i} \bar{D}_i(\bar{s}_i, \boldsymbol{s}_{-i}(r)), \forall \bar{s}_i \in S_i$;
        Set $\bar{s}_i(r) = \bar{s}_i^*$;
        Execute the best response strategy $\bar{s}_i^*$;
        Update the round counter $r \leftarrow r + 1$;
```

Due to the FIP feature of the ordinal potential game, after a finite number of rounds, the BR algorithm can converge to a stable solution that corresponds to an NE. The solution obtained from BR is at least local optimal.

Complexity Analysis: In each round of the BR algorithm, player i is chosen to calculate the potential cost over the whole strategy space S_i. The computational complexity is $O(|S_i|) = O(|\mathcal{V}(m_{i1})| \times |\mathcal{V}(m_{i2})| \times \cdots \times |\mathcal{V}(m_{i|F_i|})|)$, which is upper bounded by $O(V^{|F_i|})$, where $|F_i|$ is the number of VNFs managed by player i. Let $r_{\max}$ be the maximum number of rounds. Then, the total complexity of BR is given by $O(r_{\max} \cdot V^{|F_i|})$.

B. Spatial Adaptive Play Algorithm

We now present a learning algorithm, i.e., the SAP algorithm [34]–[36], to find the best NE of our game $\bar{\mathcal{G}}$. In the SAP algorithm, an exploration parameter is used to determine the probability of escaping from a local minimum of the potential function.

The strategies of the players are updated in a round-robin manner. Let $(\bar{s}_i(r), \boldsymbol{s}_{-i}(r))$ be the strategy profile at round r. Initially ($r = 0$), each player selects a strategy from S_i following a uniform distribution $p_{i,j}(r = 0) = 1/|S_i|$, where $j = 1, \ldots, |S_i|$ and $p_{i,j}$ is the probability of player i selecting the jth strategy from S_i. At round $r+1$, one player (say player i) is to update its strategy from $\bar{s}_i(r)$ to $\bar{s}_i(r+1)$. The new strategy is selected from the whole strategy set S_i according to the following probability distribution $p_{i,j}(r+1)$:

$$p_{i,j}(r+1) = \frac{\exp\left(-\beta \bar{D}_i\left(\bar{s}_{i,j}, \boldsymbol{s}_{-i}(r)\right)\right)}{\sum_{\bar{s}_{i,j} \in S_i} \exp\left(-\beta \bar{D}_i\left(\bar{s}_{i,j}, \boldsymbol{s}_{-i}(r)\right)\right)} \qquad (22)$$

Algorithm 2: SAP Algorithm to Find the Best NE of RC-MSEG

```
Initialize the round counter $r \leftarrow 0$;
Initialize the exploration parameter $\beta$;
Initialize the probability vector $p_{i,j}(r=0) = 1/|S_i|$,
    $\forall i \in \mathcal{N}, \forall j = 1, \ldots, |S_i|$;
for $i$ in $\mathcal{N}$ do
    Select an initial strategy from the whole strategy set
        according to $p_{i,j}(r=0)$;
    Execute the initial strategy;
Set $r \leftarrow r+1$;
while stopConditionNotMet() do
    for $i$ in $\mathcal{N}$ do
        Keep $\boldsymbol{s}_{-i}$ unchanged, i.e., $\boldsymbol{s}_{-i}(r) = \boldsymbol{s}_{-i}(r-1)$;
        Loop all the possible strategies of player $i$ to calculate
            $\bar{D}_i(\bar{s}_{i,j}, \boldsymbol{s}_{-i}(r))$, $\forall j = 1, \ldots, |S_i|$;
        Update the probability vector:
        $p_{i,j}(r) = \frac{\exp(-\beta \bar{D}_i(\bar{s}_{i,j}, \boldsymbol{s}_{-i}(r)))}{\sum_{\bar{s}_{i,j} \in S_i} \exp(-\beta \bar{D}_i(\bar{s}_{i,j}, \boldsymbol{s}_{-i}(r)))}$;
        Select the new strategy $\bar{s}_i(r)$ according to the updated
            probability vector $p_{i,j}(r)$;
        Execute the updated strategy $\bar{s}_i(r)$;
        Update the round counter $r \leftarrow r+1$;
```

where $\bar{s}_{i,j}$ represents the jth strategy in S_i; $\bar{D}_i(\bar{s}_{i,j}, \boldsymbol{s}_{-i}(r))$ is the cost when the strategy profile is $(\bar{s}_{i,j}, \boldsymbol{s}_{-i}(r))$; and $\beta \geq 0$ is the exploration parameter. A large β will force the players to select the BR strategy with high probability, while a small β in general leads to slower convergence. The SAP algorithm terminates if the stopping criterion is satisfied (e.g., the value of potential function does not change for a number of successive rounds, or the maximum number of rounds is reached). The details of the SAP algorithm to find the best NE of RC-MSEG are shown in Algorithm 2.

Theorem 3: The proposed SAP algorithm can converge to the following stationary distribution $\pi(\boldsymbol{s})$:

$$\pi(\boldsymbol{s}) = \frac{\exp\left(-\beta(\boldsymbol{s})\bar{\Phi}(\boldsymbol{s})\right)}{\sum_{\boldsymbol{s}' \in \mathcal{S}} \exp\left(-\beta \bar{\Phi}(\boldsymbol{s}')\right)}. \tag{23}$$

Proof: See Appendix B for the proof. ■

Complexity Analysis: In each round of the SAP algorithm, one player i is chosen to calculate the potential cost over the whole strategy space S_i. The computational complexity is $O(|S_i|) = O(|\mathcal{V}(m_{i1})| \times |\mathcal{V}(m_{i2})| \times \cdots \times |\mathcal{V}(m_{i|F_i|})|)$, which is upper bounded by $O(V^{|F_i|})$, where $|F_i|$ is the number of VNFs managed by player i. Then, player i updates the probability vector $p_{i,j}(r)$, which has a complexity of $O(|S_i|)$. Therefore the complexity in each round of the SAP algorithm is upper bounded by $O(V^{|F_i|})$.

Optimality Analysis: As for optimality, given sufficiently large β, the global optimal solution can be achieved with an arbitrarily large probability using the SAP algorithm [34]–[36].

C. Proposed Prioritized Admission Control Mechanism

When the physical resources are scarce or the traffic load is high, simply adding a penalty term to the cost function may not be able to ensure the capacity constraints of NFV nodes. Therefore, we further design a prioritized AC mechanism to address overloaded scenarios where the capacity constraints of NFV nodes are difficult to be guaranteed. The details of the proposed AC mechanism are shown in Algorithm 3.

Algorithm 3: Prioritized AC Mechanism

```
1  Calculate $N_{\max}$ using (24);
2  if $N > N_{\max}$ then
3      Sort $\mathcal{N}$ in descending order according to $\lambda_i$;
4      Choose the first $N_{\max}$ players in $\mathcal{N}$ to form $\tilde{\mathcal{N}}$;
5  else
6      Set $\tilde{\mathcal{N}} \leftarrow \mathcal{N}$;
7  All players in $\tilde{\mathcal{N}}$ play game $\bar{\mathcal{G}}$ to obtain an NE $\boldsymbol{s}^*$;
8  if $P(\boldsymbol{s}^*) = 0$ is true then
9      if $N > N_{\max}$ then
10         while $P(\boldsymbol{s}^*) = 0$ is true do
11             Set $\boldsymbol{s}^{\mathrm{AC}} \leftarrow \boldsymbol{s}^*$;
12             Add one more player to $\tilde{\mathcal{N}}$, i.e.,
13             $i^* = \arg\min_{i \in \mathcal{N} \setminus \tilde{\mathcal{N}}} \lambda_i$; $\tilde{\mathcal{N}} = \tilde{\mathcal{N}} \cup \{i^*\}$;
14             All players in $\tilde{\mathcal{N}}$ re-play the game $\bar{\mathcal{G}}$ to obtain a
                   new NE $\boldsymbol{s}^*$;
15         Remove the lastly added player from $\tilde{\mathcal{N}}$;
16         return $\boldsymbol{s}^{\mathrm{AC}}$, $\tilde{\mathcal{N}}$;
17     else
18         Set $\boldsymbol{s}^{\mathrm{AC}} \leftarrow \boldsymbol{s}^*$;
19         return $\boldsymbol{s}^{\mathrm{AC}}$, $\tilde{\mathcal{N}}$;
20 else
21     while $P(\boldsymbol{s}^*) = 0$ is false do
22         Remove one player from $\tilde{\mathcal{N}}$, i.e.,
23         $i^* = \arg\min_{i \in \tilde{\mathcal{N}}} \lambda_i$; $\tilde{\mathcal{N}} = \tilde{\mathcal{N}} \setminus \{i^*\}$;
24         All players in $\tilde{\mathcal{N}}$ re-play the game $\bar{\mathcal{G}}$ to obtain a new
               NE $\boldsymbol{s}^*$;
25     Set $\boldsymbol{s}^{\mathrm{AC}} \leftarrow \boldsymbol{s}^*$;
26     return $\boldsymbol{s}^{\mathrm{AC}}$, $\tilde{\mathcal{N}}$;
```

First, given the statistical characteristics of the NFV system and the SFCs, the maximum allowed number of SFCs that can be embedded, $N_{\max}$, is predicted as

$$N_{\max} \approx \frac{E(\mu_v) \cdot V}{E(\lambda_i) \cdot E(|F_i|)} \tag{24}$$

where $E(\mu_v)$ denotes the expectation of the service rate of an NFV node, $E(\lambda_i)$ denotes the expectation of the bit rate of an SFC, and $E(|F_i|)$ is the average number of VNFs in an SFC. In (24), the numerator approximates the total amount of physical resources available in the whole system, while the denominator approximates the amount of physical resources demanded by an SFC. Next, the mechanism checks if the actual number of SFCs to embedded (N) is greater than $N_{\max}$. If we have $N \leq N_{\max}$, all the players in $\mathcal{N}$ are sorted in a descending order according to their bit rates (λ_i). Then, the first $N_{\max}$ players form the set of SFCs (denoted by $\tilde{\mathcal{N}}$) that are chosen to be embedded (lines 2–4). Otherwise, we keep the original SFC set $\mathcal{N}$ unchanged (lines 5 and 6). After that, all the players in $\tilde{\mathcal{N}}$ play the game $\bar{\mathcal{G}}$ (with either BR or SAP algorithm) to obtain an NE $\boldsymbol{s}^*$ (line 7). If all the capacity constraints are satisfied and we have $N \leq N_{\max}$, the AC mechanism terminates and outputs the final accommodated SFC set $\tilde{\mathcal{N}}$ and

TABLE II
PARAMETERS USED IN SIMULATIONS

Parameter	Minimum	Maximum
Total number of VNF types $\lvert\mathcal{M}\rvert$	10	10
Number of VNF types in each SFC $\lvert\mathcal{M}_i\rvert$	3	5
Service rate μ_v	100	200
Latency parameter K_v	0.1	0.5
VNF processing time $\rho(v, m_{ij})$	0.5	1.0
Link propagation delay $\delta(v_1, v_2)$	10.0	20.0
Bit rate λ_i	10.0	20.0

their corresponding embedding strategies $\boldsymbol{s}^{\text{AC}}$ (lines 17–19). Otherwise, the mechanism attempts to find the maximum number of SFCs that can be accommodated. Each time the player with the smallest bit rate is chosen and added into $\tilde{\mathcal{N}}$, and the game is replayed to find a new NE (lines 9–16). Similarly, if at least one capacity constraint is violated, one player is removed from $\tilde{\mathcal{N}}$ each time and the game is replayed until all the capacity constraints are satisfied (lines 20–26). Notably, by executing the proposed prioritized AC mechanism, a maximum set of SFCs with relatively high bit rates is accommodated by the system where all the capacity constraints of NFV nodes are satisfied.

Note that different from centralized approaches that require a controller to gather the global information and perform centralized optimization to determine the embedding strategies, we propose distributed solutions where the SFCs in the system locally make decisions based on strategic interactions. The proposed two iterative algorithms (i.e., the BR algorithm and the SAP algorithm) are both distributed algorithms in which the players/SFCs make the embedding decisions locally and independently. To make the two algorithms work properly, some necessary information should be broadcast and exchanged among the players. In particular, each player has to collect the chosen embedding strategies from other players at the beginning of its round. Then, it calculates the BR, updates its embedding strategy, and broadcasts the new strategy to other players at the end of its round if his strategy has been changed [33]. The prioritized AC mechanism can be executed in a network orchestrator before performing multi-SFC embedding. The network orchestrator broadcasts a message to each player if it is allowed to play the MSEG $\bar{\mathcal{G}}$.

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the two proposed algorithms. We develop a platform based on C++ to carry out all the simulations. Detailed simulation parameters are listed in Table II.

A. Optimality at Convergence of the Proposed Algorithms

We first validate the optimality at convergence of the proposed algorithms. The exhaustive search (ES) algorithm is used as the benchmark to obtain the global optimal solution. To make the comparison feasible, we consider a small network instance with five NFV nodes and four SFCs. Each SFC is associated with three VNFs. We consider that each NFV node can support all the VNF types in the system, i.e.,

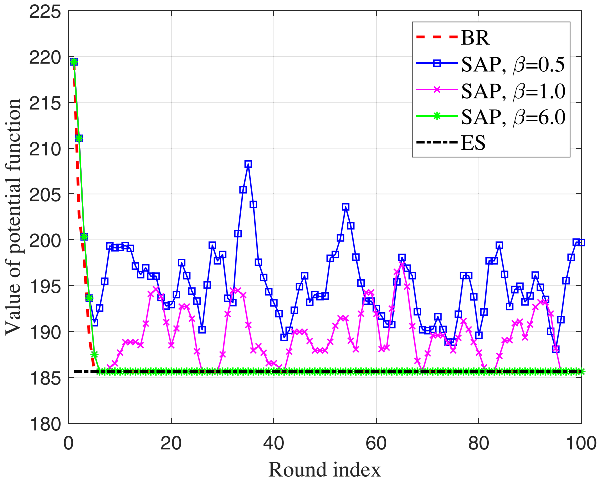

Fig. 4. Performance of the BR and SAP algorithms at convergence, where $N = 4$, $V = 5$, and $|F_i| = 3\ \forall i \in \mathcal{N}$.

$\mathcal{M}(v) = \mathcal{M}\ \forall v \in \mathcal{V}$. Then, the size of search space for the ES algorithms is $|\mathcal{S}| = (5 \times 4 \times 3)^4 = 12,960,000$. For the SAP algorithm, the exploration parameter β is selected from [0.5, 1.0, 6.0]. The maximum number of rounds is set as 100. The parameters of the penalty term in the cost function are set as $w = 1.0$ and $M = 10.0$. The convergence behavior of the proposed BR and SAP algorithms is shown in Fig. 4. As can be seen from the figure, after a small number of rounds, both the BR and SAP algorithms (with $\beta = 6.0$) converge to the global optimal solution found by the ES algorithm. Notice that for this small network scenario, the SAP algorithm converges to the global optimal solution as quickly as the BR algorithm does. On the other hand, after 100 rounds, the SAP algorithm with $\beta = 0.5$ or $\beta = 1.0$ has not converged. To achieve convergence, a larger number of rounds are needed. Notably, a larger value of exploration parameter (β) leads to a faster convergence of the SAP algorithm.

B. Performance Comparison Between Proposed Algorithms

Next, the performance of the two proposed algorithms are compared in terms of the ability of finding the best NE. Consider an NFV system with 8 nodes and 20 SFCs (i.e., $V = 8$ and $N = 20$). The number of VNFs in each SFC is a random number chosen from [3, 4, 5]. The parameters in the penalty term are set as $w = 1.0$ and $M = 10.0$. For both algorithms, the total number of rounds is set to be 250 to ensure convergence. For the SAP algorithm, the initial value of the exploration parameter is chosen from [0.5, 1.0, 2.0, 4.0]. Figs. 5 and 6 compare the performance of minimizing the potential function and the sum of SFCs' latency between the two algorithms, respectively. To achieve an NE of the RC-MSEG, the BR algorithm takes about 50 rounds, while the SAP algorithm (with $\beta = 2.0$ or $\beta = 4.0$) requires about 110 rounds. However, the values of the potential function and the sum of overall latency achieved by SAP are both smaller than that achieved by BR, which demonstrates that the SAP algorithm has greater potential to achieve the best NE of the formulated potential game RC-MSEG. Moreover, it is seen

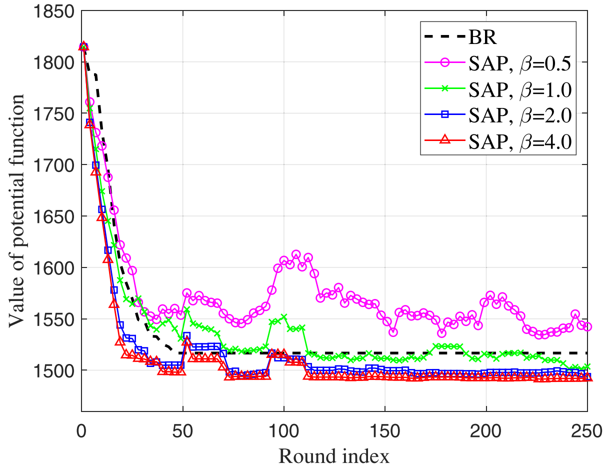

Fig. 5. Comparison between BR and SAP algorithms in terms of the value of potential function.

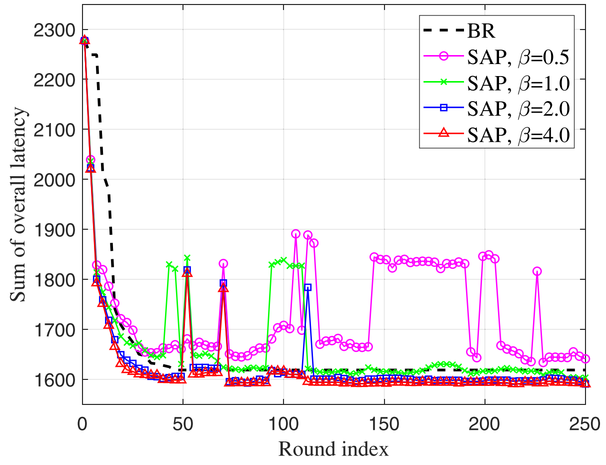

Fig. 6. Comparison between BR and SAP algorithms in terms of the sum of overall latency for all the SFCs.

that a larger value of β leads to a faster convergence of the SAP algorithm.

Figs. 7 and 8 show the dynamic evolution of the costs of five chosen players (i.e., Player 1, 3, 5, 7, and 9) for BR and SAP (with $\beta = 4.0$) algorithms, respectively. From Fig. 7, we can see that with the BR algorithm, the costs of the five players keep decreasing as the round index increases, reaching to an equilibrium after about 50 rounds. The final cost of each player, which corresponds to the total delay experienced by each SFC, at the NE depends on the final embedding strategy chosen by SFC. A similar tendency can be observed from Fig. 8, where the cost of each player oscillates at the initial phase and converges to a stable value after 100 rounds.

Figs. 9 and 10 illustrate the evolution of the VNF embedding strategies of two selected players (i.e., player 1 and player 6) using BR and SAP algorithms, respectively. Both players are associated with three VNFs. With the BR algorithm, it can be seen that the VNF embedding strategies of both players keep unchanged after 40 rounds. With the SAP algorithm, it is shown in Fig. 10 that the strategies of both players remain unchanged after about 150 rounds. These results further demonstrate that both algorithms are able to obtain an NE of the MSEG.

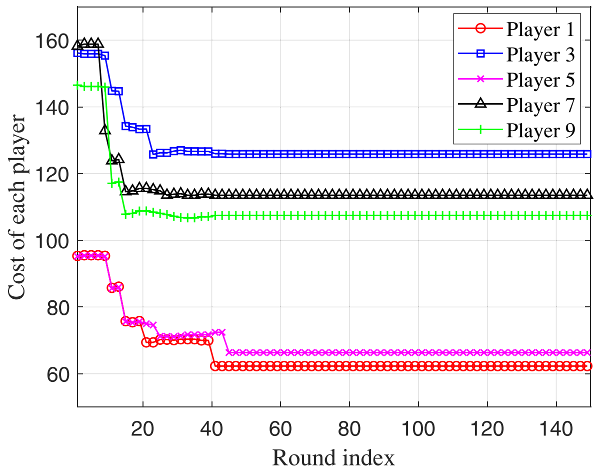

Fig. 7. Dynamic evolution of the costs of the five chosen players when using the BR algorithm for the scenario $N = 20$, $V = 8$, and $|F_i| \in \{3, 4, 5\}$.

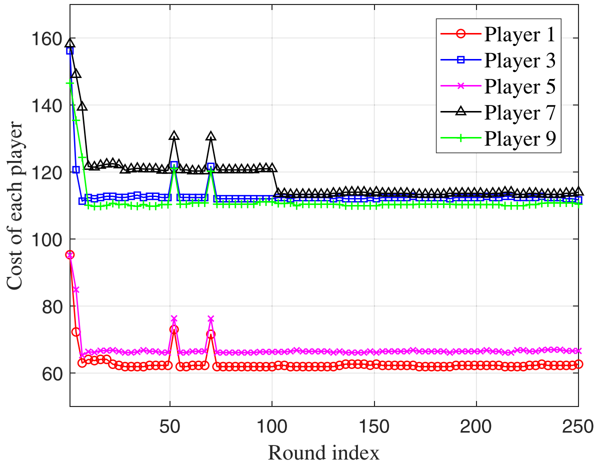

Fig. 8. Dynamic evolution of the costs of the five chosen players when using the SAP algorithm with $\beta = 4.0$ for the scenario $N = 20$, $V = 8$, and $|F_i| \in \{3, 4, 5\}$.

To show that the formulated potential game can effectively handle capacity constraints, we present the dynamic evolution of the penalty term (i.e., $P(\bar{s}_i, \boldsymbol{s}_{-i})$) for BR and SAP algorithms in Figs. 11 and 12, respectively. A larger value of penalty term indicates a higher violation degree of the capacity constraints of the NFV nodes. It can be observed from Fig. 11 that the capacity constraints of two NFV nodes (we set $w = 1.0$ and $M = 10.0$) are violated with the initial strategies of the players. After 15 rounds, the BR algorithm finds an embedding strategy where the capacity constraints of all the NFV nodes can be satisfied (i.e., $P(\bar{s}_i, \boldsymbol{s}_{-i}) = 0$). For the SAP algorithm, we can see from Fig. 12 that the number of NFV nodes with violated capacity constraints fluctuates between 0 and 2, and stabilizes at 0 after about 120 rounds for all the three exploration parameter settings. These results demonstrate that our proposed game-theoretical approach is able to ensure the capacity constraints of NFV nodes effectively.

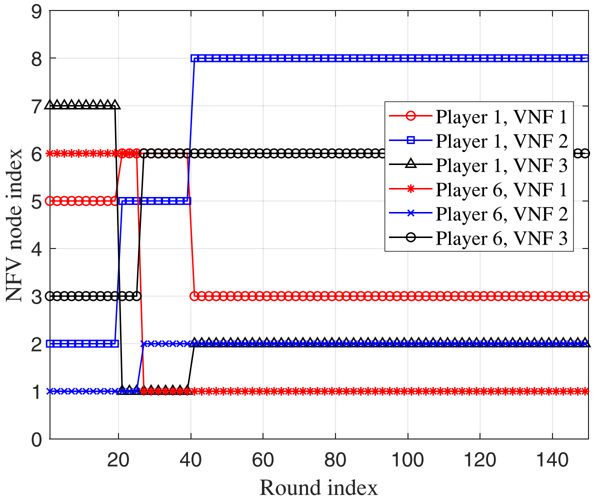

Fig. 9. VNF embedding strategy evolution using the BR algorithm.

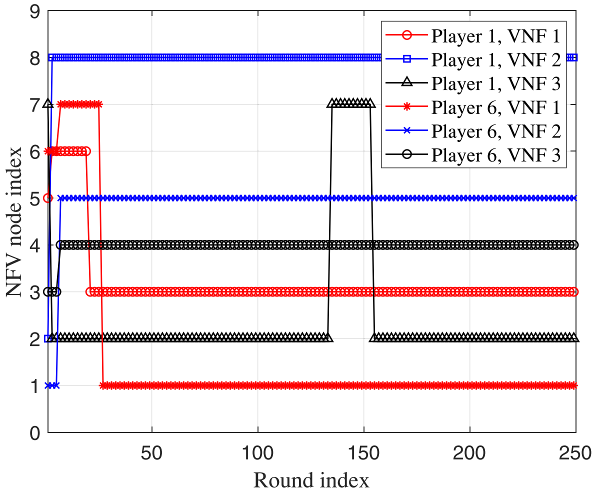

Fig. 10. VNF embedding strategy evolution using SAP algorithm ($\beta = 4.0$).

C. Performance Verification of the Proposed Prioritized Admission Control Mechanism

To validate the effectiveness of the proposed prioritized AC mechanism, we consider another network scenario where the number of SFCs to be embedded is 22. We adopt the resource utilization ratio at each NFV node (defined as the ratio of total traffic load on the NFV node over the service rate of that node) as the performance metric. Fig. 13 compares the performance between the BR algorithm with the proposed AC mechanism and that without the AC mechanism. By using the AC mechanism, 20 SFCs are embedded at the NE point. It can be seen from the figure that without the AC mechanism, there are three NFV nodes whose the capacity constraints are violated at the NE. With the AC mechanism applied, all the NFV nodes have their capacity constraints satisfied at NE, demonstrating the effectiveness of our proposed AC mechanism. Moreover, with the proposed AC mechanism, the resource utilization ratios of all NFV nodes are quite similar to each other and close to 100%, which demonstrate better traffic load balancing among NFV nodes for the proposed approach.

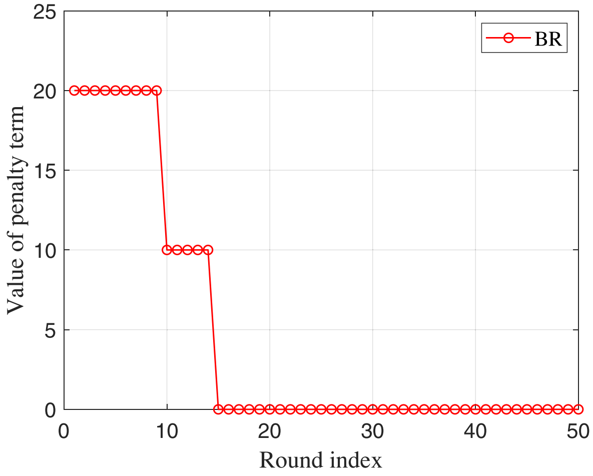

Fig. 11. Dynamic evolution of the penalty term of the BR algorithm.

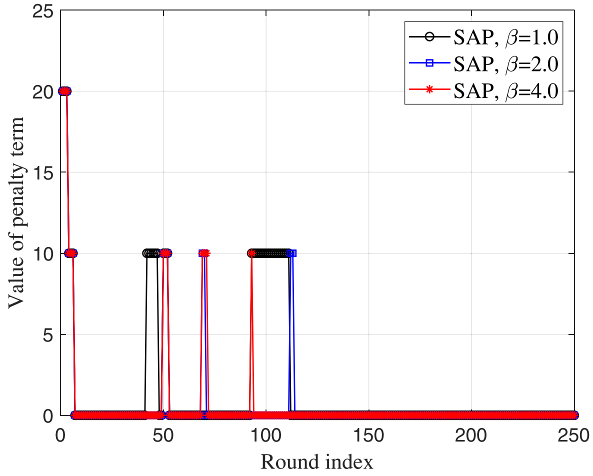

Fig. 12. Dynamic evolution of the penalty term of the SAP algorithm.

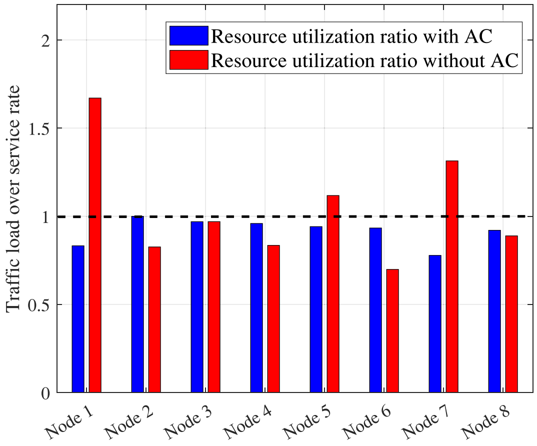

Fig. 13. Comparison of resource utilization ratio between the BR algorithm with AC and that without AC.

VI. Conclusion

In this article, we have investigated a multi-SFC embedding problem to efficiently map SFCs onto a substrate network to provide customized service provisioning with guaranteed QoS. Capturing the effect of processing-resource sharing among various VNFs and the capacity constraints of NFV nodes, a game-theoretical approach has been proposed to solve the multi-SFC embedding problem with reduced amount of information exchange and computational complexity. A novel prioritized AC mechanism has been designed to deal with overloaded service requests. Simulation results have been provided to demonstrate the effectiveness of the proposed game approach in achieving QoS-guaranteed multiservice provisioning. For future work, we will study multi-SFC embedding in a dynamic network environment.

Appendix A

Let us construct a potential function as follows:

$$\bar{\Phi}(\bar{s}_i, \boldsymbol{s}_{-i}) = \Phi(\bar{s}_i, \boldsymbol{s}_{-i}) + wP(\bar{s}_i, \boldsymbol{s}_{-i}) \tag{25}$$

where the definitions of $P(\bar{s}_i, \boldsymbol{s}_{-i})$ and $p_v(\bar{s}_i, \boldsymbol{s}_{-i})$ are given by (11) and (12), respectively. Suppose that the strategy of player i is changed from $\bar{a} \in S_i$ to $\bar{b} \in S_i$, then, the change of the modified cost function $\bar{D}_i(\bar{s}_i, \boldsymbol{s}_{-i})$ is given by

$$\begin{aligned}\bar{D}_i(\bar{a}, \boldsymbol{s}_{-i}) - \bar{D}_i(\bar{b}, \boldsymbol{s}_{-i}) &= D_i(\bar{a}, \boldsymbol{s}_{-i}) - D_i(\bar{b}, \boldsymbol{s}_{-i}) \\ &\quad + w\big(P(\bar{a}, \boldsymbol{s}_{-i}) - P(\bar{b}, \boldsymbol{s}_{-i})\big).\end{aligned} \tag{26}$$

Based on (21) and (25), we obtain the change of the potential function $\bar{\Phi}$ as follows:

$$\begin{aligned}\bar{\Phi}(\bar{a}, \boldsymbol{s}_{-i}) - \bar{\Phi}(\bar{b}, \boldsymbol{s}_{-i}) &= \Phi(\bar{a}, \boldsymbol{s}_{-i}) - \Phi(\bar{b}, \boldsymbol{s}_{-i}) \\ &\quad + w\big(P(\bar{a}, \boldsymbol{s}_{-i}) - P(\bar{b}, \boldsymbol{s}_{-i})\big) \\ &= \bar{D}_i(\bar{a}, \boldsymbol{s}_{-i}) - \bar{D}_i(\bar{b}, \boldsymbol{s}_{-i}).\end{aligned} \tag{27}$$

That is, the potential function $\bar{\Phi}$ is an exact potential function for RC-MSEG.

Appendix B

Following similar proofs in [34]–[36], let us denote the network state at the rth round as $\boldsymbol{s}(r) = (\bar{s}_1(r), \bar{s}_2(r), \ldots, \bar{s}_N(r))$, where $\bar{s}_i(r)$ represents the strategy of player i. Obviously, $\boldsymbol{s}(r)$ is a discrete time, irreducible, and aperiodic Markov process, which has a unique stationary distribution. Let $\boldsymbol{s}_1 \in \mathcal{S}$ and $\boldsymbol{s}_2 \in \mathcal{S}$ represent any two arbitrary network states. Specifically, let $\boldsymbol{s}_1 = (\bar{s}_1, \ldots, \bar{s}_{m-1}, \bar{s}_m, \bar{s}_{m+1}, \ldots, \bar{s}_N)$ and $\boldsymbol{s}_2 = (\bar{s}_1, \ldots, \bar{s}_{m-1}, \bar{s}'_m, \bar{s}_{m+1}, \ldots, \bar{s}_N)$, where $m \in \mathcal{N}$ is the player chosen to change her strategy between states $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$. Denote the transition probability from $\boldsymbol{s}_1$ to $\boldsymbol{s}_2$ by $P(\boldsymbol{s}_2|\boldsymbol{s}_1)$. To prove that the unique distribution of $\boldsymbol{s}(r)$ must be (23), we only need to prove that the following balanced equation holds:

$$\pi(\boldsymbol{s}_1)P(\boldsymbol{s}_2|\boldsymbol{s}_1) = \pi(\boldsymbol{s}_2)P(\boldsymbol{s}_1|\boldsymbol{s}_2). \tag{28}$$

Given $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ defined earlier, the L.H.S. of the above equation is given by

$$\begin{aligned}&\pi(\boldsymbol{s}_1)P(\boldsymbol{s}_2|\boldsymbol{s}_1) \\ &= \frac{\exp\big(-\beta\bar{\Phi}(\boldsymbol{s}_1)\big)}{\sum_{\boldsymbol{s}_1\in\mathcal{S}}\exp\big(-\beta\bar{\Phi}(\boldsymbol{s}_1)\big)} \cdot \frac{\exp\big(-\beta\bar{D}_m(\bar{s}'_m, \boldsymbol{s}_{-m})\big)}{\sum_{\bar{s}'_m\in S_m}\exp\big(-\beta\bar{D}_m(\bar{s}'_m, \boldsymbol{s}_{-m})\big)} \\ &= \eta\exp\big\{-\beta\big(\bar{\Phi}(\boldsymbol{s}_1) + \bar{D}_m(\bar{s}'_m, \boldsymbol{s}_{-m})\big)\big\}\end{aligned} \tag{29}$$

where the parameter η is defined as

$$\eta = 1\bigg/\left\{\sum_{\boldsymbol{s}_1\in\mathcal{S}}\exp\big(-\beta\bar{\Phi}(\boldsymbol{s}_1)\big) \cdot \sum_{\bar{s}'_m\in S_m}\exp\big(-\beta\bar{D}_m(\bar{s}'_m, \boldsymbol{s}_{-m})\big)\right\}.$$

In a similar manner, we can also obtain the R.H.S. of (28) as follows:

$$\pi(\boldsymbol{s}_2)P(\boldsymbol{s}_1|\boldsymbol{s}_2) = \eta\exp\big\{-\beta\big(\bar{\Phi}(\boldsymbol{s}_2) + \bar{D}_m(\bar{s}_m, \boldsymbol{s}_{-m})\big)\big\}. \tag{30}$$

Considering that from $\boldsymbol{s}_1$ to $\boldsymbol{s}_2$, there is only one element (i.e., player m's strategy) changed, and that $\bar{\mathcal{G}}$ is an exact potential game, we have

$$\bar{\Phi}(\boldsymbol{s}_1) - \bar{\Phi}(\boldsymbol{s}_2) = \bar{D}_m(\bar{s}_m, \boldsymbol{s}_{-m}) - \bar{D}_m(\bar{s}'_m, \boldsymbol{s}_{-m}). \tag{31}$$

According to (29)–(31), we can conclude that the stationary distribution shown in (23) satisfies the balanced (28) and therefore is the unique stationary distribution of the proposed SAP algorithm.

References

[1] W. Zhuang, Q. Ye, F. Lyu, N. Cheng, and J. Ren, "SDN/NFV-empowered future IoV with enhanced communication, computing, and caching," *Proc. IEEE*, vol. 108, no. 2, pp. 274–291, Feb. 2020.

[2] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of Things applications," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, Feb. 2018.

[3] "5G: Study on scenario and requirements for next generation access technologies, V14.2.0," 3GPP, Sophia Antipolis, France, Rep. TR 38.913, 2018.

[4] Q. Ye, J. Li, K. Qu, W. Zhuang, X. Shen, and X. Li, "End-to-end quality of service in 5G networks: Examining the effectiveness of a network slicing framework," *IEEE Veh. Technol. Mag.*, vol. 13, no. 2, pp. 65–74, Jun. 2018.

[5] A. J. Gonzalez, G. Nencioni, A. Kamisinski, B. E. Helvik, and P. E. Heegaard, "Dependability of the NFV orchestrator: State of the art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3307–3329, 4th Quart., 2018.

[6] X. Tian, W. Huang, Z. Yu, and X. Wang, "Data driven resource allocation for NFV-based Internet of Things," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8310–8322, Oct. 2019.

[7] J. Li, W. Shi, N. Zhang, and X. Shen, "Delay-aware VNF scheduling: A reinforcement learning approach with variable action set," *IEEE Trans. Cogn. Commun. Netw.*, early access, Apr. 13, 2020, doi: 10.1109/TCCN.2020.2988908. .

[8] J. Wang, H. Qi, K. Li, and X. Zhou, "PRSFC-IoT: A performance and resource aware orchestration system of service function chaining for Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1400–1410, Jun. 2018.

[9] J. Li, W. Shi, Q. Ye, W. Zhuang, X. Shen, and X. Li, "Online joint VNF chain composition and embedding for 5G networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2018, pp. 1–6.

[10] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, "Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manag. (IM)*, May 2015, pp. 98–106.

[11] L. Wang, Z. Lu, X. Wen, R. Knopp, and R. Gupta, "Joint optimization of service function chaining and resource allocation in network function virtualization," *IEEE Access*, vol. 4, pp. 8084–8094, Nov. 2016.

[12] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *Proc. IEEE Conf. Comput. Commun.*, May 2015, pp. 1346–1354.

[13] B. Martini, F. Paganelli, P. Cappanera, S. Turchi, and P. Castoldi, "Latency-aware composition of virtual functions in 5G," in *Proc. 1st IEEE Conf. Netw. Softwarization (NetSoft)*, Apr. 2015, pp. 1–6.

[14] Z. Ye, X. Cao, J. Wang, H. Yu, and C. Qiao, "Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization," *IEEE Netw.*, vol. 30, no. 3, pp. 81–87, May/Jun. 2016.

[15] Q. Ye, W. Zhuang, X. Li, and J. Rao, "End-to-end delay modeling for embedded VNF chains in 5G core networks," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 692–704, Feb. 2019.

[16] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "A game theoretic approach for distributed resource allocation and orchestration of softwarized networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 721–735, Mar. 2017.

[17] X. Chen *et al.*, "Leveraging mixed-strategy gaming to realize incentive-driven VNF service chain provisioning in broker-based elastic optical inter-datacenter networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 2, pp. A232–A240, Feb. 2018.

[18] X. Chen, Z. Zhu, R. Proietti, and S. J. B. Yoo, "On incentive-driven VNF service chaining in inter-datacenter elastic optical networks: A hierarchical game-theoretic mechanism," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 1–12, Mar. 2019.

[19] A. Leivadeas, G. Kesidis, M. Falkner, and I. Lambadaris, "A graph partitioning game theoretical approach for the VNF service chaining problem," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 890–903, Dec. 2017.

[20] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, "Exploiting congestion games to achieve distributed service chaining in NFV networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 2, pp. 407–420, Feb. 2017.

[21] S. Bian, X. Huang, Z. Shao, X. Gao, and Y. Yang, "Service chain composition with failures in NFV systems: A game-theoretic perspective," in *Proc. IEEE Int. Conf. Commun.*, May 2019, pp. 1–6.

[22] S. Le, Y. Wu, and X. Sun, "Congestion games with player-specific utility functions and its application to NFV networks," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1870–1881, Oct. 2019.

[23] X. Fu, F. R. Yu, J. Wang, Q. Qi, and J. Liao, "Dynamic service function chain embedding for NFV-enabled IoT: A deep reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 507–519, Jan. 2020.

[24] P. T. A. Quang, A. Bradai, K. D. Singh, G. Picard, and R. Riggio, "Single and multi-domain adaptive allocation algorithms for VNF forwarding graph embedding," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 1, pp. 98–112, Mar. 2019.

[25] Y. Gu, Q. Cui, Q. Ye, and W. Zhuang, "Game-theoretic optimization for machine-type communications under QoS guarantee," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 790–800, Feb. 2019.

[26] M. Savi, M. Tornatore, and G. Verticale, "Impact of processing-resource sharing on the placement of chained virtual network functions," *IEEE Trans. Cloud Comput.*, early access, May 2, 2020, doi: 10.1109/TCC.2019.2914387.

[27] Y. Pan and L. Pavel, "Global convergence of an iterative gradient algorithm for the nash equilibrium in an extended OSNR game," in *Proc. 26th IEEE Int. Conf. Comput. Commun.*, May 2007, pp. 206–212.

[28] G. Arslan, M. F. Demirkol, and S. Yuksel, "On games with coupled constraints," *IEEE Trans. Autom. Control*, vol. 60, no. 2, pp. 358–372, Feb. 2015.

[29] W. Yuan, P. Wang, W. Liu, and W. Cheng, "Variable-width channel allocation for access points: A game-theoretic perspective," *IEEE Trans. Mobile Comput.*, vol. 12, no. 7, pp. 1428–1442, Jul. 2013.

[30] N. Zhang, S. Zhang, J. Zhang, X. Fang, J. W. Mark, and X. Shen, "QoE driven decentralized spectrum sharing in 5G networks: Potential game approach," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 7797–7808, Sep. 2017.

[31] D. Monderer and L. S. Shapley, "Potential games," *Games Econ. Behav.*, vol. 14, no. 1, pp. 124–143, 1996.

[32] J. Zheng, Y. Cai, Y. Liu, Y. Xu, B. Duan, and X. Shen, "Optimal power allocation and user scheduling in multicell networks: Base station cooperation using a game-theoretic approach," *IEEE Trans. Wireless Commun.*, vol. 13, no. 12, pp. 6928–6942, Dec. 2014.

[33] R. Deng, Z. Yang, J. Chen, N. R. Asr, and M.-Y. Chow, "Residential energy consumption scheduling: A coupled-constraint game approach," *IEEE Trans. Smart Grid*, vol. 5, no. 3, pp. 1340–1350, May 2014.

[34] H. P. Young, *Individual Strategy and Social Structure: An Evolutionary theory of Institutions*. Princeton, NJ, USA: Princeton Univ. Press, 1998.

[35] Y. Xu, J. Wang, Q. Wu, A. Anpalagan, and Y. Yao, "Opportunistic spectrum access in cognitive radio networks: Global optimization using local interaction games," *IEEE J. Sel. Topics Signal Process.*, vol. 6, no. 2, pp. 180–194, Apr. 2012.

[36] J. Marden, G. Arslan, and J. Shamma, "Cooperative control and potential games," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1393–1407, Dec. 2009.

Junling Li (Graduate Student Member, IEEE) received the B.S. degree from Tianjin University, Tianjin, China, in 2013, the M.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2016, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2020.

She is currently a Joint Postdoctoral Research Fellow with Shenzhen Institute of Artificial Intelligence and Robotics for Society, Chinese University of Hong Kong, Shenzhen, China, and the University of Waterloo. Her interests include game theory, machine learning, software-defined networking, network function virtualization, and vehicular networks.

Dr. Li received the Best Paper Award at the IEEE/CIC International Conference on Communications in China in 2019.

Weisen Shi (Graduate Student Member, IEEE) received the B.S. degree from Tianjin University, Tianjin, China, in 2013, the M.S. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2016, and the Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, in 2020.

His interests include space-air-ground integrated networks, UAV communication and networking, and RAN slicing.

Qiang Ye (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2016.

He had been with the Department of Electrical and Computer Engineering, University of Waterloo, as a Postdoctoral Fellow and then a Research Associate from December 2016 to September 2019. He has been an Assistant Professor with the Department of Electrical and Computer Engineering and Technology, Minnesota State University, Mankato, MN, USA, since September 2019. His current research interests include 5G networks, software-defined networking and network function virtualization, network slicing, artificial intelligence and machine learning for future networking, protocol design, and end-to-end performance analysis for the Internet of Things.

Ning Zhang (Senior Member, IEEE) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2015.

He is an Associate Professor with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON. After that, he was a Postdoctoral Research Fellow with the University of Waterloo and the University of Toronto, Toronto, ON. He is a Highly Cited Researcher (Web of Science). His research interests include connected vehicles, mobile-edge computing, wireless networking, and machine learning.

Dr. Zhang received an NSERC PDF Award in 2015 and the Six Best Paper Awards from IEEE Globecom in 2014, IEEE WCSP in 2015, IEEE ICC in 2019, IEEE ICCC in 2019, IEEE Technical Committee on Transmission Access and Optical Systems in 2019, and the *Journal of Communications and Information Networks* in 2018. He also serves/served as a TPC Chair for IEEE SAGC 2020, a Track Chair for several international conferences, including IEEE VTC 2020, IEEE ICC 2022, AICON 2020, and CollaborateCom 2020, and a co-chair for numerous international workshops. He serves as an Associate Editor of IEEE INTERNET OF THINGS JOURNAL, IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING, and IEEE SYSTEMS JOURNAL; and a Guest Editor of several international journals, such as IEEE WIRELESS COMMUNICATIONS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, and IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING.

Weihua Zhuang (Fellow, IEEE) received the Ph.D. degree from the University of New Brunswick, Fredericton, NB, Canada.

She has been with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada, since 1993, where she is currently a Professor and a Tier I Canada Research Chair of wireless communication networks.

Dr. Zhuang was a recipient of the 2017 Technical Recognition Award from the IEEE Communications Society Ad Hoc and Sensor Networks Technical Committee, and several best paper awards from IEEE conferences. She was the Technical Program Chair/Co-Chair of the IEEE VTC Fall 2016 and Fall 2017, the Editor-in-Chief of IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2007 to 2013, and an IEEE Communications Society Distinguished Lecturer from 2008 to 2011. She is an Elected Member of the Board of Governors and VP Publications of the IEEE Vehicular Technology Society. She is a fellow of the Royal Society of Canada, the Canadian Academy of Engineering, and the Engineering Institute of Canada.

Xuemin (Sherman) Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990.

He is currently a University Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular *ad hoc* and sensor networks.

Dr. Shen received the R. A. Fessenden Award in 2019 from IEEE, Canada, the Award of Merit from the Federation of Chinese Canadian Professionals (Ontario) in 2019, the James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, the Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society, and the Technical Recognition Award from Wireless Communications Technical Committee in 2019, and AHSN Technical Committee in 2013. He has also received the Excellent Graduate Supervision Award in 2006 from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He served as the Technical Program Committee Chair/Co-Chair for the IEEE Globecom 2016, the IEEE Infocom 2014, the IEEE VTC 2010 Fall, the IEEE Globecom 2007, the Symposia Chair for the IEEE ICC 2010, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He is the elected IEEE Communications Society Vice President for Technical and Educational Activities, the Vice President for Publications, a Member-at-Large on the Board of Governors, the Chair of the Distinguished Lecturer Selection Committee, and a Member of IEEE Fellow Selection Committee. He was/is the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL, IEEE NETWORK, *IET Communications*, and *Peer-to-Peer Networking and Applications*. He is a registered Professional Engineer of Ontario, Canada, an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society.