Network Function Virtualization in Dynamic Networks: A Stochastic Perspective

Xiangle Cheng, Yulei Wu, *Member, IEEE,* Geyong Min, *Member, IEEE,* and Albert Y. Zomaya, *Fellow, IEEE*

***Abstract*—As a key enabling technology for 5G network softwarization, Network Function Virtualization (NFV) provides an efficient paradigm to optimize network resource utility for the benefits of both network providers and users. However, the inherent network dynamics and uncertainties from 5G infrastructure, resources and applications are slowing down the further adoption of NFV in many emerging networking applications. Motivated by this, in this paper, we investigate the issues of network utility degradation when implementing NFV in dynamic networks, and design a proactive NFV solution from a fully stochastic perspective. Unlike existing deterministic NFV solutions, which assume given network capacities and/or static service quality demands, this paper explicitly integrates the knowledge of influential network variations into a two-stage stochastic resource utilization model. By exploiting the hierarchical decision structures in this problem, a distributed computing framework with two-level decomposition is designed to facilitate a distributed implementation of the proposed model in large-scale networks. The experimental results demonstrate that the proposed solution not only improves 3∼5 folds of network performance, but also effectively reduces the risk of service quality violation.**

***Index Terms*—Network function virtualization, 5G, decomposition method, stochastic network optimization.**

I. Introduction

The increasing mobility of humans and connected devices are actuating the explosive growth of mobile Internet traffic. According to [1], by 2021, global mobile data traffic will grow 7-fold, and the number of mobile users will be up to 5.5 Billion. To meet the extreme traffic demands, the next-generation networks (5G) are expected to be equipped with 5x as many as base stations and utilize 200x more spectrum than 4G [2]. This makes the orchestration of so many 5G elements to achieve the desired objectives get even more challenging than before.

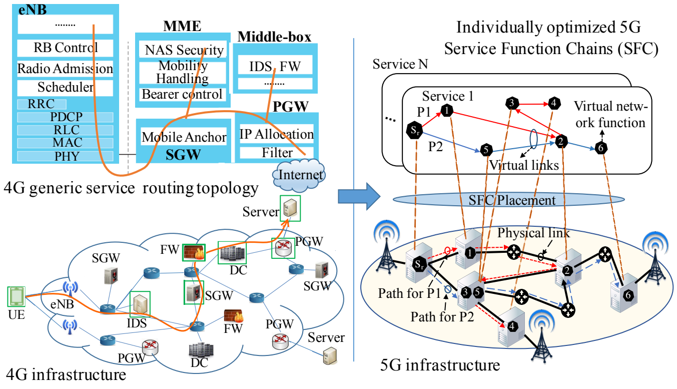


Fig. 1. Network softwarization for next-generation network evolution.

To improve the situation, many organizations are resorting to network softwarization [3], [4] for the next-wave network evolution through the technologies of Network Function Virtualization (NFV) and Software Defined Networking (SDN). As illustrated in Fig. 1, in such a transition, more commodity servers and shared hardware devices will be introduced to replace these special-purpose devices in 4G. As a result, services will be constructed as individually optimized Service Function Chains (SFCs) [5]. These SFCs are then implemented with the isolated network resources sliced from the underlying network infrastructure. This enables prompt delivery of new services with better flexibility, agility and lower capital and operating expenditures [2].

As shown in Fig. 1, in an NFV based network service, Virtualized Network Functions (VNFs) will be dynamically chained as a specific SFC topology according to its service offerings. In order to implement such a service, one critical task is to perform the SFC placement in the underlying physical network bound to diverse resource and service requirements. This is achieved by placing VNFs in hosting servers and then connecting the placed VNFs with physical links through the proper allocation of infrastructure resources (*e.g.*, CPUs, cable bandwidth, radio spectrum).

The SFC placement problem (or service chain composition problem as termed in related work [6], [7]) is similar to the Virtual Network Embedding (VNE) problem [8] for network virtualization. Essentially, both problems aim to make efficient implementations of virtual requests in a physical network infrastructure. However, as discussed in [6], existing solutions to VNE problems are not sufficient for the SFC placement problems due to the specific features and applications of NFV.

Inheriting the methodologies built upon Integer Linear Programming (ILP) or Mixed Integer Linear Programming (MILP) for VNE, many preliminary studies have attempted to model and solve this new problem under a deterministic resource or traffic condition. However, with the penetration of NFV into more emerging networking applications [2], more network dynamics and uncertainties are expected than the current networks, which will make many existing deterministic NFV solutions not directly applicable. These network dynamics can be summarized as the following three classes:

a) ***Architecture level:*** In order to achieve an efficient utilization of higher and wider frequency spectrum beyond 6GHz, 5G networks will be heterogeneously constructed with more Radio Access Technologies (RAT), such as GSM, W-CDMA, LTE, W-LAN and new 5G RAT(s) [2]. The coexistence and cooperation of ever-increasing radio systems will highly oscillate the network topology and bring more dynamics and uncertainties to the network operation and management.

b) ***Traffic level:*** In a virtualized operation environment in 5G, the traffic of each request may be collected from an individual mobile user, multi-tenant users, or a group of sensors. In many emerging 5G scenarios, the traffic profiles and demanded Service Level Agreements (SLAs) are to be highly varied and unpredictable. For example, in Internet-of-Things (IoT) applications, with the asynchronous activation and silence, node failures and mobility of large dynamic numbers of interconnected sensors and actuators, the collective traffic load injected into the traffic aggregation point (*e.g.*, gateway) of a network slice is always time-varying.

c) ***Resource level:*** With the increasing network cloudification in NFV, more globally controlled resources will be pooled together for more efficient utilization. The frequent resource scaling by the expected network self-management and -optimization procedures [9] for 5G will result in high instability and uncertainty on the resource distributions in each substrate node and link. With the expected coexistence of legacy and millimeter Wave (mmWave) spectrum bands [10], the 5G radio environment is also becoming increasingly unpredictable because of fast-fading, shadowing effects and interference.

When these dynamics are presented, the resource and/or traffic conditions are always time-varying. In this case, the deterministic SFC placement decisions can only customize the networking performance over the instantly observed information. However, this would leave the system vulnerable to potential network changes after decisions are made.

A straightforward solution to handle this situation is to migrate and re-route the VNF instances reactively by re-invoking a deterministic model (*e.g.*, MILP) to compute a new or recourse solution against each network change. However, no matter whether they are executed dynamically or online, such a solution can lead to frequent network reconfiguration and instability. In addition, it is also unaffordable in terms of the additional service latency incurred by the expensive re-computation of these usually NP-hard models. For example, in many delay-sensitive 5G applications [2], a millisecond-order system response is required, which is even beyond the time required for most existing model solvers to find a converged solution. To maintain a seamless service provisioning in dynamic networks, it is desirable to have a service deployment strategy that can handle the network changes proactively.

Therefore, in this paper, we highlight the network utility degradation problem for the implementation of NFV in dynamic networks and aim to design a proactive NFV deployment solution in both centralized and distributed fashions. Different from the posterior scaling or migration based dynamic algorithms in the literature, this paper alternatively resorts to reinforce the temporal robustness of the obtained SFC placement decisions from every expensive attempt of model solving by integrating future stochastic information at the initial placement decision phase. The contributions of this paper can be summarized as follows:

- We implement the SFC placement in dynamic networks with a carefully designed Stochastic Resource Allocator (SRA) that: 1) jointly exploits the already-observed and future stochastic information to infer the placement decisions, and 2) balances of the immediate reward with the impact of each decision on future rewards.
- We provide a centralized optimal solution by solving the SRA model with a two-stage stochastic program and identify the hardness involved in solving SRA in a large instance, including the need of enumerating an exponentially expanding constraint set, and computing the expected random functions.
- A distributed computing framework with two-level decomposition is developed to facilitate a distributed implementation of the SRA in large-scale networks. Supported by the classic decomposition theory, the complicated combinatorial program only needs to be solved during service initialization, while the subsequent service running only involves the solving of a simple linear program. This significantly reduces the computation complexity involved in the whole duration of service running controls.
- Extensive simulation experiments are conducted with the settings in accordance with 5G expectations. Through the comparisons with the incumbent placement solutions, the results confirm that the proposed solution not only achieves significant performance improvement, but also effectively reduces risks of service quality violation in dynamic networks.

The rest of this paper is organized as follows. Section II summarizes the related work. In Section III, we formally derive the resource utility model for SFC placement in dynamic networks and analyze its optimal implementation. The distributed implementation based on two-level decomposition is presented in Section IV. Section V illustrates the performance evaluation results. Finally, in Section VI, conclusions are made.

II. Related Work

As a key enabling 5G technology, NFV has been gaining momentum among an ever-growing community of researchers from both academia and industry. It has been also the focus of different standardization bodies (*e.g.*, 3GPP, ETSI and 5GMF) [3]. A global architecture can be found in [11], which defines the modules and interfaces that ensure the life-cycle management of NFV services. In the envisioned architecture, the technical implementation of NFV plays a critical role on the provisioning efficiency and service performance. Next, we provide a concrete review to highlight the differences between our work and the existing NFV solutions.

A. Deterministic NFV Modelling and Solutions

The preliminary efforts for the SFC placement problem mainly focus on optimizing the SFC placement with acceptable computing complexity in static settings. In this era, many centralized solutions based on deterministic optimization methods were proposed. For example, based on the multiple-objective Mixed Integer Quadratic Programming (MIQP), an initial study on placing VNFs was provided in [12]. However, its solution is developed through a Pareto set analysis while the solution scalability issue is left unattended. In [13], the authours formulated the VNF orchestration problem as an ILP, and a dynamic programming-based heuristic was provided to solve the problem in large instances. Based on the tools of MILP and game theory, extensive studies on deterministic VNF placement algorithms can also be found in [14]– [19]. Nevertheless, as aforementioned, these centralized solutions usually have scalability and/or accuracy issues and are often insufficient for large-scale dynamic networks.

In addition to these centralized solutions, the authors in [7] investigated the drawbacks of centralized placement solutions and proposed a distributed NFV solution by exploiting congestion games. A similar attempt is the work in [20] which provides a Markov approximated algorithm to solve the centralized placement model in a distributed way. In these existing studies, their solutions are solved only over the already observed network and service conditions. Consequently, these deterministic placement solutions, no matter in a centralized or distributed fashion, are not directly applicable to dynamic networks. In contrast, the SFC placement problem in dynamic networks is more complex. Beyond the considerations for a deterministic problem, more network dynamics are needed to handle in order to guarantee the effectiveness of the obtained solutions.

B. *Dynamic Resource Utility for NFV Networks*

There also exist a few studies in the literature striving to address similar resource utility problems for dynamic NFV networks.

Among the very few studies, Jia *et al.* in [21] proposed an online NFV scaling solution to handle the time-varying traffic volumes in geo-distributed Datacenters. However, the cost and drawbacks of dynamic scaling of VNF instances are not addressed in their solution. With a similar motivation, the resource provisioning solution proposed by Li *et al.* [22] is also proactive, although its objective is to assign requests with bounded response time. This is achieved by using SFC consolidation with timing abstraction, but the placement of SFCs is still based on deterministic models and VNF instance migration. Ghaznavi *et al.* in [23] optimized the VNF placement under changing workload. This is achieved by dynamically migrating the VNF instances on the basis of the migration costs in the current instant. This work was then extended in [24] by taking into account the benefits and penalties of these migrations in successive instants. However, different from the work in these existing studies, this paper highlights the challenges of network dynamics that potentially limit the application of reactive scaling or migration strategies. Alternatively, we focus on generating SFC placement policies that can work robustly even when network state changes.

C. *Network Applications of Decomposition Methods*

As a complement, the applications of decomposition theories are also surveyed to show both the wide theoretical effectiveness and the differences between our application and other related networking problems. Decomposition methods are widely used in large-scale networks where a centralized solution is infeasible, non-scalable or too costly. Deniel *et al.* in [25] developed a generic application framework of decomposition methods for network utility maximization problem. A related survey of decomposition methods on many practical network applications can be found in [26]. Among these applications, the work in [8] comes closest to ours, in which they applied the Column Generation technique to decompose the deterministic centralized VNE model into two smaller subproblems. As a similar resource utility problem in supply chain network design, the authors in [27] proposed an accelerated benders' decomposition approach to expedite the solution time of the centralized MILP model.

Decomposition approaches require a good decomposable model structure. By exploiting the problem-specific structures in the proposed SRA model, we build a two-level decomposition framework to facilitate the distributed implementation of the proposed SRA solution.

III. A Proactive Placement Model

As implied in Fig. 1, the SFC placement is essentially a graph-embedding problem. That is, mapping VNF nodes into substrate nodes and connecting VNFs with substrate links to implement the SFC graphs in the physical network topology. As discussed in Section I, this is non-trivial as many problem-specific features are presented in the target problem. In what follows, we will design a stochastic resource utility model to implement the SFC placement with fully respect to the features of NFV paradigms and network dynamics. In this paper, we treat the networking system as a discrete time stochastic system in which the network dynamics are assumed to follow stationary random processes. Moreover, operational scaling or migration of VNF instances are not considered due to the aforementioned challenges. The symbol notations used in this paper are listed in Table I.

A. *Model Formulation*

In this paper, we consider a resource limited network system, in which partial admission control is applied. As a result, service requests are able to be accepted by allocating compromised service rates rather than directly rejected when the available physical resources are not enough to fully meet the required demands. In addition, once accepted, each service will occupy isolated resources to instantiate its sliced network until new scheduling decisions are made or service is terminated. Such a pay-as-you-go admission policy is more practical in dynamic networks or when the resource demands of a service are aggregated from a group of users (*e.g.* IoT or multi-tenant applications).

To present the network dynamics, we consider a discrete time stochastic networking system, in which the service rate demands $\beta^s(t)$ and the available amounts of wireless resources at access nodes (*e.g.*, wireless transmission capacity), $c_v(t)$, are subject to random variations. The Probability Distribution Functions (PDF) of random variables are assumed known as a priori (via *e.g.*, estimations from historical statistics). At the

TABLE I. Notations

System parameters			
$(V,\mathcal{L})$	Directed graph for physical network topology with node $v\in V$ and link $l_{uv}\in\mathcal{L}$ connecting u to v	c_{vr}	Residual resource capacity on physical node $v\in V$ for resource $r\in R$
c_l	Residual bandwidth capacity on physical link l	c_v	Residual wireless capacity on access node v
k_l, k_r	Usage price for per-unit link & node resources	T	Scheduling interval
Request parameters			
S	Received service requests	$f\in\mathcal{F}^s$	VNFs in service s, and let $\mathcal{F}=\cup_{s\in S}\mathcal{F}^s$
d_{fr}	Resource demand of $r\in R$ required to instantiate an instance of VNF f on a hosting node	$e_{ij}\in E^s$	Virtual link connecting VNF i to j for service s, and let $E=\cup_{s\in S}E^s$
b^s	Service price or benefit when unit rate of s is routed	$\beta^s\le\beta^s_0$	Rate demand requested by s, which is assumed to be up bounded by β^s_0
Decision variables			
π_s	Binary, 1 *iff* service request $s\in S$ is accepted	$\pi_{e\to l}$	Binary, 1 *iff* the routing of virtual link $e\in E^s$ uses physical link $l\in\mathcal{L}$
$\pi_{f\to v}$	Binary, 1 *iff* $f\in\mathcal{F}$ is placed on node $v\in V$	γ^s	Allocated rate for request s
Key auxiliary mathematical operators and symbols			
$(\cdot)^T$	Vector transpose	$\|\cdot\|$	Return the cardinality of a vector
$\mathbf{0},\mathbf{1}$	All-one and all-zero vectors, respectively	$\mathbb{E}_\gamma[\cdot]$	Take expectation over γ
$u_s(\cdot)$	Revenue function for $s\in S$	$U_s(\cdot)$	Weighted revenue function for $s\in S$
$Q_s(\cdot)$	Utility function of s for policy approximation	w	Weight for future revenue
C^s_{nd}	Node resource cost for $s\in S$	$\Phi_{e\to l}$	Auxiliary variable for linearization
$\tilde{\pi}$	Fractional placement solution	$\bar{\pi}$	Approximate binary placement solution

beginning of every time slot $t\in\{0,1,2,\cdots\}$, the network controller observes a state update $\omega(t)=(\beta^s(t)_{s\in S}, c_v(t)_{v\in V})$, which specifies the current realizations of rate demands and resource state. Depending on $\omega(t_0)$ and the statistics about network dynamics, the controller, at the beginning of every scheduling interval T, decides a proactive placement policy $\pi=(\pi_s,\pi_{f\to v},\pi_{e\to l})^T_{s\in S}$ for the running duration $[t_0, t_0+T]$, and then adapts users' service rates to real-time observations at each time t.

From an algorithmic point of view, the design of placement policy in such a system requires to decide policies for admission control, VNF placement, and VNF chaining in a sequential order. This is fundamentally a combinatorial optimization process, which decides a long-term optimal placement policy π under a stochastic environment. Following the pay-as-you-go billing model for network services [28], this paper defines the following revenue oriented utility function for each $s\in S$:

$$u_s(\gamma^s,\pi^s)=\gamma^s\Big(b^s-\sum_{\substack{e\in E^s\\ l\in\mathcal{L}}}k_l\pi_{e\to l}\Big)-\sum_{\substack{f\in\mathcal{F}^s\\ v\in V,r\in R}}\pi_{f\to v}d_{fr}k_r \quad (1)$$

where $\pi^s=(\pi_s,\pi_{f\to v},\pi_{e\to l})^T$ is the placement policy for s, $C^s_{lk}=\gamma^s\sum_{\substack{e\in E^s\\ l\in\mathcal{L}}}k_l\pi_{e\to l}$ and $C^s_{nd}=\sum_{\substack{f\in\mathcal{F}^s\\ v\in V,r\in R}}\pi_{f\to v}d_{fr}k_r$ are the cost for using the link and node resources, respectively.

Once a service is instantiated, a fixed node installation cost C^s_{nd} is charged, but the practical link cost $C^s_{lk}(t)$ and the benefit from this service are dynamically decided by the allocated service rate $\gamma^s(t)$ at runtime. Clearly, only services with benefit larger than all costs will be accepted by providers. Based on this insight, the concept of beneficial placement is defined as follows:

Definition 1 ***(Beneficial placement)***. *Given a placement policy π^s, the placement of service request s is beneficial if the placement action for this request incurs a positive collective revenue (i.e., $\sum_{t\in[t_0,t_0+T]}u_s(\gamma^s(t)|\pi^s)>0$).*

Based on the observed information $\omega(t_0)$, an *immediate revenue* can be counted as follows:

$$U(\gamma(t_0),\pi)=\sum\nolimits_{s\in S}u_s(\gamma^s(t_0),\pi^s) \quad (2)$$

For any future time $t_f>t_0$, the realizations of $\omega(t_f)$ are to be observed after the placement decisions. Consider the stochastic nature of the network, an *expected future revenue* under a given placement policy π made at t_0 can be calculated as follows:

$$\bar{U}(\gamma(t_f)|\pi)=\mathbb{E}_\gamma\Big[\sum_{s\in S}\gamma^s(t_f)\big(b^s-\sum_{\substack{e\in E^s\\ l\in\mathcal{L}}}k_l\pi_{e\to l}\big)-C^s_{nd}\Big] \quad (3)$$

where $\gamma=\gamma^s(t_f)_{s\in S}$ is a random vector dependent on the random outcome of $\omega(t_f)$.

The current placement decision has an impact not only on the immediate revenue, but also on the future revenues. From the network provider's point of view, the objective is always to maximize the long-term revenue under as minimum resource cost as possible. Consequently, efficient policies have to balance the benefits of an immediate reward with the expected impact of each decision on future rewards. This leads to the following global objective function designed to achieve the long-term revenue maximization:

$$\begin{aligned}U(\gamma,\pi)&=\overbrace{U(\gamma(t_0),\pi)}^{\text{immediate exploitation}}+\text{w}\,\overbrace{\bar{U}(\gamma(t_f)|\pi)}^{\text{future exploration}}\\&=\sum_{s\in S}\underbrace{\Big\{\big(b^s-\sum_{e\in E^s,l\in\mathcal{L}}k_l\pi_{e\to l}\big)\big(\gamma^s(t_0)+\text{w}\mathbb{E}_\gamma[\gamma^s(t_f)]\big)-(1+\text{w})C^s_{nd}\Big\}}_{U_s(\gamma^s,\pi^s)}\end{aligned} \quad (4)$$

where $U_s(\cdot)$ is the weighted utility function for an individual service, and $\text{w}\ge 0$ is a weighting factor to control the decisions' balance between exploiting immediate revenue and exploring the potentially better future revenue after network state changes.

Then, the intended SFC placement process can be readily formulated as the Stochastic Resource Allocation (SRA) program in Algorithm 1.

Algorithm 1 The stochastic resource allocation for SFC placement in dynamic networks

Input: resource and traffic states at t_0, PDFs of $\omega(t_f)$, network and SFC topologies.

Output: placement policy π^* and running rate $\gamma^*(t_0)$.

$$U(\gamma^*, \pi^*) = \max_{\substack{\pi \in \{\mathbf{0},\mathbf{1}\} \\ \gamma \geq \mathbf{0}}} \sum_{s \in S} U_s(\gamma^s, \pi^s) \tag{5a}$$

$$s.t. \qquad \sum_{e \in E^s, s \in S} \pi_{e \to l} \gamma^s(t) \leq c_l, \forall l \in \mathcal{L}, t \in \{t_0, t_f\} \tag{5b}$$

$$\sum_{s \in S_v} \gamma^s(t) \leq c_v(t), \forall v \in V, t \in \{t_0, t_f\} \tag{5c}$$

$$\gamma^s(t) \leq \pi_s \beta^s(t), \forall s \in S, t \in \{t_0, t_f\} \tag{5d}$$

$$\sum_{f \in \mathcal{F}} \pi_{f \to v} d_{fr} \leq c_{vr}, \forall v \in V, r \in R \tag{5e}$$

$$\sum_{v \in V} \pi_{f \to v} = \pi_s, \forall f \in \mathcal{F}^s, s \in S \tag{5f}$$

$$\sum_{l_{uv} \in O(u)} \pi_{e_{ij} \to l_{uv}} - \sum_{l_{vu} \in I(u)} \pi_{e_{ij} \in l_{vu}} = \pi_{i \to u} - \pi_{j \to u}, \quad \forall e_{ij} \in E^s, s \in S, u \in V \tag{5g}$$

In Algorithm 1, (5b) and (5e) are the capacity upper bounds for link and node resources, respectively. (5c) guarantees that the total allocated rates for the set of services S_v attached to access node v will not overload its real-time wireless resource capacity. (5d) sets the rate upper bound that should be allocated for each service. (5f) imposes the variable dependencies and guarantees that each VNF will be placed at most once. In this paper, unsplittable flow [29] is considered for constructing each virtual link. Let $O(u)$ and $I(u)$ denote the outgoing and incidental edges of node u, respectively. Then, the correlated connection between VNF placement decisions and VNF chaining decisions is finally expressed as (5g). Dependent on the practical applications, this model is versatile enough to integrate more problem-specific constraints.

For any deterministic realization (*i.e.*, a problem instance with all parameters determined), the model in Algorithm 1 corresponds to an MIQP. However, by exploiting the binary structure, this model can be readily linearized to a pure MILP. Let us define auxiliary variable $\Phi_{e \to l} = \pi_{e \to l} \gamma^s$ to substitute the quadratic expressions in (5a) and (5b) with the following two extra constraints:

$$\Phi_{e \to l} \leq \gamma^s, \forall e \in E^s, s \in S, l \in \mathcal{L} \tag{6}$$

$$\frac{\gamma^s}{\beta_0^s} - 1 + \pi_{e \to l} \leq \frac{\Phi_{e \to l}}{\beta_0^s} \leq \frac{\gamma^s}{\beta_0^s} + 1 - \pi_{e \to l} \tag{7}$$

Note that (6) is redundant when the link cost term is counted in the objective function.

In contrast to the dominant deterministic NFV resource utilization models in the literature, the proposed SRA jointly refers to both currently observed network information and future variation information at the placement decision phase. The added extra information can help exclude non-beneficial service placement more accurately, but also drive the model to the following more challenging dilemma when solving it.

Exploitation-exploration dilemma: *One needs to balance the exploitation of the placement action currently optimal with the exploration of other actions that currently appear suboptimal but may turn out to be superior in the long run.*

Algorithm 1 can be directly solved with all possible realizations of $\omega(t_f)$. However, this may require solving the resultant MILP model under an unmanageably large set of realizations of these random parameters, which is usually intractable. Next, we attempt to address this problem through a two-stage equivalent process.

B. The Global Optimality Solved through A Two-Stage Equivalence

Recall the structure of the model in Algorithm 1, the whole program can be re-arranged to a hierarchical two-stage process by separating the binary variables from continuous variables. Let us treat the case at t_0 as a special realization of future randomness. Then, the maximization problem in SRA can be reformulated as the following equivalent two-stage minimization problem (in linearized format):

$$U(\gamma^*, \pi^*) = \min_{\pi \in \{\mathbf{0},\mathbf{1}\}} \{(1 + \mathrm{w}) \sum_{s \in S} C_{nd}^s + \min_{\substack{\gamma \geq \mathbf{0} \\ \Phi \geq \mathbf{0}}} \mathbb{E}_\gamma [\sum_{\substack{s \in S \\ t \in \{0,1\}}} \mathrm{w}_t (\sum_{\substack{e \in E^s \\ l \in \mathcal{L}}} k_l \Phi_{e \to l}(t) - b^s \gamma^s(t))]\} \tag{8}$$

where w_t is the weighting factor for current and future time. Thus, we have $\mathrm{w}_0 = 1, \mathrm{w}_1 = \mathrm{w}$.

At the first stage, the program in (8) manages to decide a placement policy π with the constraints solely related to binary variables. Under the given policy π, a policy evaluation program with only continuous variables (*i.e.,*γ) is then applied at the second stage to evaluate the achievable average revenue.

For any determined placement policy $\bar{\pi}$ at each t, the inner minimization problem in (8) can be reduced to the following linear subproblem (policy evaluation program):

$$\text{(SP)} \quad \min_{\gamma \geq \mathbf{0}, \Phi \geq \mathbf{0}} \mathbb{E}_\gamma \Big[\sum_{s \in S} \mathrm{w}_t \Big(\sum_{e \in E^s, l \in \mathcal{L}} k_l \Phi_{e \to l}(t) - b^s \gamma^s(t) \Big) \Big] \tag{9a}$$

$$s.t. \qquad \sum_{e \in E^s, s \in S} \Phi_{e \to l}(t) \leq c_l, \forall l \in \mathcal{L} \tag{9b}$$

$$\sum_{s \in S_v} \gamma^s(t) \leq c_v(t), \forall v \in V \tag{9c}$$

$$\gamma^s(t) \leq \bar{\pi}_s \beta^s(t), \forall s \in S \tag{9d}$$

$$\Phi_{e \to l}(t) - \gamma^s(t) \leq \beta_0^s(1 - \bar{\pi}_{e \to l}), \forall e \in E^s, l \in \mathcal{L}, s \in S \tag{9e}$$

$$\gamma^s(t) - \Phi_{e \to l}(t) \leq \beta_0^s(1 - \bar{\pi}_{e \to l}), \forall e \in E^s, l \in \mathcal{L}, s \in S \tag{9f}$$

Define column vector $\mu := (\mu_l^0, \mu_v^1, \mu_s^2, \mu_{sel}^3, \mu_{sel}^4)^T$ as dual variables associated with each constraint in SP. Then, the dual of SP can be formulated as:

$$\text{(DSP)} \qquad \max_{\mu \leq \mathbf{0}} \ \mathbb{E}_{c_v, \beta^s}\left[- D(\mu, \bar{\pi}, t)\right] \tag{10a}$$

$$s.t. \quad \mu_{sel}^4 - \mu_{sel}^3 - \mu_l^0 \leq \mathrm{w}_t k_l, \forall e \in E^s, l \in \mathcal{L}, s \in S \tag{10b}$$

$$\sum_{e \in E^s, l \in \mathcal{L}} (\mu_{sel}^s - \mu_{sel}^4) - \mu_s^2 - \mu_{v_s}^1 \leq -\mathrm{w}_t b^s, \forall s \in S \tag{10c}$$

where v_s is the attached access node[1] for s, and $D(\mu, \bar{\pi}, t)$ is defined as follows:

$$\begin{aligned} D(\mu, \bar{\pi}, t) = & \sum_{l \in \mathcal{L}} c_l \mu_l^0 + \sum_{v \in V} c_v(t) \mu_v^1 + \sum_{s \in S} \bar{\pi}_s \beta^s(t) \mu_s^2 \\ & + \sum_{\substack{l \in \mathcal{L} \\ s \in S, e \in E^s}} (\mu_{sel}^3 + \mu_{sel}^4) \beta_0^s (1 - \bar{\pi}_{e \to l}) \end{aligned} \tag{11}$$

The SP in (9) is a parametric linear program and always has a feasible solution under any given policy $\bar{\pi}$. In this case, according to duality theory [30], the DSP in (10) has always a bounded optimal solution corresponding to an extreme point of the polyhedron in dual space. After the transition from the primal SP to its dual, we can see that the uncertain parameters only exist in the objective function of dual problem, but the constraints of dual problem constitute a fixed polyhedron whose space is independent of the network variations and the chosen placement policy. Therefore, through the complete enumeration of extreme points, the original problem in (8) can be equivalently solved by the following master problem:

$$\text{(MP)} \qquad \min_{\substack{U, \pi \in P_\pi \\ \pi \in \{\mathbf{0}, \mathbf{1}\}}} \ U \tag{12a}$$

$$\begin{aligned} s.t. \quad U \geq - \sum_{t \in \{0,1\}} \mathbb{E}_{c_v, \beta^s}[D(\bar{\mu}_i, \pi, t)] + (1 + \mathrm{w}) \sum_{s \in S} C_{nd}^s, \\ \forall \bar{\mu}_i \in P_\Delta \end{aligned} \tag{12b}$$

where P_π is the policy space defined by (5e)-(5g), and P_Δ is the set of extreme points in the DSPs polyhedron.

Recall the structure of the function $D(\cdot)$ in (11), we can see that random variable c_v is independent of π and other random variables. Therefore, we have

$$\mathbb{E}_{c_v, \beta^s}[D(\bar{\mu}_i, \pi, t)] = \mathbb{E}_{\beta^s}[D(\bar{\mu}_i, \pi, t) | c_v = \bar{c}_v] \tag{13}$$

where $\bar{c}_v$ is the mean value of c_v.

As a consequence, under the above separated two-stage structure, we only need to know the mean values of resource variations, although the detailed realization distributions of rate demands are still required. This property significantly reduces the number of random samples that are required to calculate the expectation of future average revenue.

When the problem is presented in a small instance, the MP can be solved efficiently with global optimality by enumerating all extreme points and possible realizations of β^s in (12b). Compared with the state-of-the-art methods, *e.g.,* directly adapting Column Generation [8] or Sample Average Approximation [27] to solve the SRA model, the above two-stage strategy requires less samples and thus results in a smaller problem to solve.

However, it would get very hard to do this for large-scale networks. Tri-fold challenges can be identified when solving the SRA model in a large instance with global optimality.

First, by removing the constraints (12b), the original problem is relaxed to the combination of classic facility location and multi-commodity flow problems [31], which are both NP-hard. Consequently, any single attempt of solving the problem with global optimality in a large instance is time consuming if not impossible. Second, to solve the problem in a large instance, there is typically an exponentially increased number of extreme points in the dual polyhedron. However, a more challenging problem is the computation of the expected value for the random function $D(\bar{\mu}_i, \pi, t)$. When the number of service requests gets large, this may involve an unmanageably large set of realization combinations for $(\beta^s)_{s \in S}$. All of these factors make it essentially impractical to completely enumerate all the constraints in (12b).

Consequently, in the following sections, we consider to design distributed models and approximate algorithms to alleviate the computational challenges of implementing this model in large-scale networks.

IV. A Distributed Implementation Based on Two-Level Decomposition

It is well known that the computation load and the required memory for solving an optimization program increase exponentially with the number of variables and constraints. Therefore, by harvesting the above separated two-stage structure of the SRA model, we first design the following higher-level decomposition to reduce the large scale of the SRA model brought by the enumeration of extreme points and possible realizations of $(\beta^s)_{s \in S}$ in (12b). This is achieved based on the theory of stochastic decomposition [32].

A. Higher-Level Decomposition

By exploiting the method of stochastic decomposition, it is possible to decompose the complicated monolithic model in a large instance into a series of solvable submodules in a distributed way. The solution of original model can then be reached by solving these submodules in an iterative manner. This is implemented through the concepts of variable partition and constraint delay.

Following the two-stage structure of the SRA model in Section III-B, we first construct the Higher-level Sub-Problem (HSP) exactly same as the SP model in (9). The HSPs are a series of linear programs with only continuous variables. Then, a dimension-reduced Higher-level Master Problem (HMP) can be initially constructed as a relaxed version of the MP model in (12) without the constraint (12b).

Instead of a complete enumeration of constraints in (12b), the method of stochastic decomposition alternatively solves the HMP model to generate a trial decision for the placement policy. The trial placement policy is then fed into the HSPs

[1] For simplifying exposition, single access node is considered for each s.

with randomly sampled parameters. Accordingly, the associated DSPs are solved to obtain the resultant extreme point and an approximation towards the original objective function under current samples. Next, a constraint of (12b) related to this extreme point will be inserted into the HMP, which is then solved again until a predefined termination criterion achieved. The overall progress is outlined in Algorithm 2.

Note that the extra terms in (14) are introduced to exclude unnecessary policy trials. Normally, we have $\sum_{s\in S}\{\pi_s - \sum_{e\in E^s, l\in\mathcal{L}} \frac{k_l \pi_{e\to l}}{\beta^s}\} \ll U$, thus the impact of introduced terms on the optimality of U is negligible.

Algorithm 2 Approximate stochastic decomposition algorithm for SRA

```
    Input: resource and traffic states at $t_0$, $(\bar{c}_v)_{v\in V}$, PDFs of $(\beta^s)_{s\in S}$, network and SFC topologies.
    Output: placement policy $\bar{\pi}$ and running rate $\gamma^*(t_0)$.
 1: Initialization: set $m = 0$, collect current observation $\omega(t_0)$.
 2: do $m = m + 1$ and solve the following HMP to obtain trail policy $\bar{\pi}_m$:
      $$U_m^l(\bar{\pi}_m) = \min_{\substack{U,\pi\in P_\pi\\ \pi\in\{\mathbf{0},\mathbf{1}\}}} U - \sum_{s\in S}\{\pi_s - \sum_{e\in E^s, l\in\mathcal{L}} \frac{k_l \pi_{e\to l}}{b^s}\} \quad (14)$$
 3: draw random samples for the future realization of $(\beta^s)_{s\in S}$ according to their PDFs.
 4: solve the DSPs in (10) for each $t$ with the generated samples and policy $\bar{\pi}_m$ to obtain the extreme point $\bar{\mu}_m$ and an empirical estimation to the original expected objective value:
      $$U_m^u = -\sum_{t\in\{0,1\}} D(\bar{\mu}_m, \bar{\pi}_m, t) + (1 + \mathrm{w}) \sum_{s\in S} C_{nd}^s \quad (15)$$
 5: if termination criteria meet then
 6:     solve HSPs with $\bar{\pi}_m$ for $t_0$ to get the allocated rate $\gamma^*$ for current time.
 7: else
 8:     add the following optimality constraint to the program in (14):
          $$U \geq -\frac{1}{m} \sum_{t\in\{0,1\}} D(\bar{\mu}_m, \pi, t) + (1 + \mathrm{w}) \sum_{s\in S} C_{nd}^s \quad (16)$$
 9:     update the coefficients in previous optimality cuts as follows and go to step 2:
          $$D(\bar{\mu}_i, \pi, t) = \frac{m-1}{m} D(\bar{\mu}_i, \pi, t), \forall t \in \{0,1\}, i = 1, \cdots, m-1 \quad (17)$$
10: end if
```

Termination criteria: For discrete distributions of random parameters, the sample space is deterministic. Therefore, by using the whole sample space at each iteration, the lower and upper bounds of original objective function can be derived precisely from the results of U_m^l and U_m^u, respectively. In this case, a deterministic criterion can be used by monitoring the optimality gap between upper and lower bounds. However, for continuous distributions, the bound gap is subject to statistical variation due to the random sampling outcomes. Then, an alternative criterion is to monitor the progress of incumbent solutions. For example, the iteration stops when the incumbent solution has remained unchanged for a certain number of iterations within a tolerant variation on the expected objective value. Such criteria can provide better safeguards to prevent the sensitivity of the selected solution to additional sampling.

The proof of the asymptotic optimality of Algorithm 2 is similar to many existing stochastic approximation applications [34], thus is not explicitly presented here.

For each iteration in Algorithm 2, all sampled subproblems are linear programs, which can be solved easily via many standard LP solvers [30]. However, the HMP corresponds to an MILP problem, which is still NP-hard, although the dimension has already been reduced compared with the original problem. Therefore, the handicap regarding solving the NP-hard HMP still persists for large-scale networks. In the following, we will build a lower-level decomposition to turn the HMP into a decoupled resource utility problem for each service request, which can then be solved with better scalability.

B. Lower-Level Decomposition

Recall the structure of HMP model, we can see that the constraints in (12b) and (5e) are coupled among all service requests. This results in an exponentially increased computation complexity as more service requests are required to schedule. However, if these coupling constraints are relaxed, the original HMP model naturally turns into an individual resource utility problem for each service request. Each of these problems can then be, independently and parallelly, solved. This is implemented as follows through linear relaxation and dual decomposition.

Since the HMP is only introduced to generate the bound and trial placement policy for HSPs, the accurate but expensive solving for the optimal solution at every iteration is not essential. Alternatively, simple and faster approximation algorithms can be adopted to generate a new trial policy. The trial policy can then be gradually improved through fast iterations. Therefore, instead of directly getting an optimal solution for the HMP in (14) at each iteration, a linear relaxed version of HMP can be first solved as follows:

$$\text{(RHMP)} \quad \tilde{\pi} = \operatorname*{argmin}_{\substack{U,\pi\in P_\pi\\ \pi\in[\mathbf{0},\mathbf{1}]}} U - \sum_{s\in S}\{\pi_s - \sum_{e\in E^s, l\in\mathcal{L}} \frac{k_l \pi_{e\to l}}{b^s}\} \quad (18)$$

The RHMP relaxes the binary constraint in HMP to the continuous value ranging from $[0, 1]$. The fractional placement solution $\tilde{\pi}$ obtained from the RHMP conveys the globally coordinated resource allocation information when all requests compete for the shared resources. We can anticipate that larger fractional values of decision variables would suggest a better revenue if the corresponding service is accepted and placed accordingly. Therefore, by proportionally weighting the placement selections with the corresponding fractional solutions, an approximate solution to the HMP can be solved from a weighted HMP program as follows:

$$\text{(WHMP)} \qquad \bar{\pi} = \operatorname*{argmax}_{\pi\in P_\pi, \pi\in\{\mathbf{0},\mathbf{1}\}} \sum_{s\in S} Q_s(\pi^s) \quad (19)$$

where $Q_s(\pi^s)$ is the utility function for individual service, which is defined as

$$Q_s(\pi^s) = \alpha_1 \tilde{\pi}_s \pi_s - \sum_{\substack{f\in\mathcal{F}^s\\ v\in V}} \frac{\alpha_2 \pi_{f\to v}}{\tilde{\pi}_{f\to v}+1} - \sum_{\substack{e\in E^s\\ l\in\mathcal{L}}} \frac{\alpha_3 \pi_{e\to l}}{\tilde{\pi}_{e\to l}+1} \tag{20}$$

where $\alpha_1 \gg \alpha_2 \gg \alpha_3$ are weights to preserve the hierarchical decision order along $\pi_s \to \pi_{f\to v} \to \pi_{e\to l}$.

In WHMP, the objective function has a separable structure, but the resource constraint (5e) in P_π is coupled across all $s \in S$. Since strong duality holds when solving the WHMP through its Lagrange dual problem, this constraint can be decoupled for each $s \in S$ by means of dual decomposition [25].

We define the Lagrangian of the WHMP by relaxing the coupling constraint (5e) in P_π as

$$\begin{aligned} L(\pi,\lambda) &= \sum_{s\in S} Q_s(\pi^s) + \sum_{v\in V, r\in R} \lambda_{vr}\big(c_{vr} - \sum_{s\in S}\sum_{f\in\mathcal{F}^s} \pi_{f\to v} d_{fr}\big) \\ &= \sum_{s\in S}\big\{Q_s(\pi^s) - \sum_{\substack{f\in\mathcal{F}^s\\ v\in V, r\in R}} \lambda_{vr}\pi_{f\to v} d_{fr}\big\} + \sum_{\substack{v\in V\\ r\in R}} \lambda_{vr} c_{vr} \end{aligned} \tag{21}$$

where λ is the non-negative Lagrange multiplier associated with the constraint (5e).

Clearly, for a given λ, the resulted Lagrangian dual problem can be decomposed into solving, independently for each $s \in S$, the following Lower-level Sub-Problem (LSP):

$$\text{(LSP)} \quad L_s(\bar{\pi}^s,\lambda) = \max_{\pi^s\in\{\mathbf{0},\mathbf{1}\}} Q_s(\pi^s) - \sum_{\substack{f\in\mathcal{F}^s\\ v\in V, r\in R}} \lambda_{vr}\pi_{f\to v} d_{fr} \tag{22a}$$

$$s.t. \quad \sum_{v\in V} \pi_{f\to v} = \pi_s, \forall f \in \mathcal{F}^s \tag{22b}$$

$$\sum_{l_{uv}\in O(u)} \pi_{e_{ij}\to l_{uv}} - \sum_{l_{vu}\in I(u)} \pi_{e_{ij}\to l_{vu}} = \pi_{i\to u} - \pi_{j\to u}, \quad \forall e_{ij} \in E^s, u \in V \tag{22c}$$

LSPs can be directly solved through existing integer program solvers since only several VNFs and virtual links are involved in the placement process. Additionally, linear relaxation and rounding based approximate solvers (*e.g.,* [31]) are also applicable in order to further reduce the computation complexity.

Physically, the Lagrange multiplier λ corresponds to the resource congestion price, on which each service has to depend to decide the amount of resources to be used at their own benefits. Then, in order to achieve the original global optimality, the minimum congestion price can be solved by using the following Lower-level Master Problem (LMP) to coordinate all LSPs:

$$\text{(LMP)} \quad \hat{\lambda} = \operatorname*{argmin}_{\lambda\geq\mathbf{0}} \sum_{s\in S} L_s(\bar{\pi}^s,\lambda) + \sum_{v\in V, r\in R} \lambda_{vr} c_{vr} \tag{23}$$

When the analytical expression of L_s is absent, the (LMP) can be recursively solved through the following gradient method:

$$\lambda_{vr}(n+1) = \lceil \lambda_{vr}(n) - \delta(c_{vr} - \sum_{f\in\mathcal{F}} \bar{\pi}_{f\to v} d_{fr}) \rceil^+, \forall v \in V, r \in R \tag{24}$$

where n is the iteration index, $\delta > 0$ is a positive step size, and $\lceil\cdot\rceil^+$ denotes the projection onto the non-negative orthant.

In Algorithm 3, we provide the detailed implementation of the approximate dual decomposition for solving the HMP.

Algorithm 3 Approximate dual decomposition algorithm for solving the HMP

```
Input: coefficient matrix for HMP, mean values of network
states ($(\mathbb{E}_{\beta^s}[\beta^s]_{s\in S}, (c_v)_{v\in V})$).
Output: trial placement policy $\bar{\pi}$.
1: set $n = 0$ and $\lambda_{vr}(0)$ equal to some non-negative value
   for all $(v, r)$.
2: solve RHMP to obtain the fractional solution $\tilde{\pi}$.
      ▷ Solving WHMP under $\tilde{\pi}$
3: do $n = n+1$, and solve all LSPs to obtain a local placement
   policy $\bar{\pi} = \{\bar{\pi}^s\}_{s\in S}$.
4: update congestion prices according to (24) and broadcast
   the new prices to all LSPs.
5: go to step 3 until maximum iterates or tolerant variation
   on the collective utility $L(\pi, \lambda)$ reached.
      ▷ Revenue evaluation
6: solve HSP with $\bar{\pi}$ and statistical mean values, i.e., $\omega(t_f) =$
   $(\mathbb{E}_{\beta^s}[\beta^s]_{s\in S}, (c_v)_{v\in V})$.
7: calculate the individual revenue as follows based on the
   solution obtained in step 6:
      $U_s = u_s(\gamma^s(t_0), \bar{\pi}^s) + T u_s(\gamma^s(t_f), \bar{\pi}^s)$     (25)
8: reject requests with non-beneficial placement (i.e., $U_s \leq 0$)
   and return all accepted $\bar{\pi}^s$.
```

Note that in Algorithm 3, the WHMP will only generate a placement solution that fully respects the fractional placement results obtained in RHMP. But this does not guarantee that every admitted request has a beneficial placement for the duration $[t_0, t_0 + T]$ since the optimality constraints are not evaluated in WHMP. Therefore, under the placement solution from the WHMP, a final revenue evaluation process is invoked to exclude the requests with non-beneficial placement when the network is in the mean state.

Finally, by putting all together, the overall distributed computing framework for the SRA model is illustrated in Fig. 2 and summarized as follows:

Step 0: Preprocessing the request information and collect the coefficient matrix of SRA program.

Step 1: Solve HMP with lower-level decomposition algorithm:

1.1: Solve RHMP to obtain fractional solution;

1.2: For each $s \in S$, solve LSPs with given congestion price λ;

1.3: Update congestion price and return to **1.2** until termination;

1.4: Individual revenue evaluation and return beneficial placement policy.

Step 2: Solve all DSPs to evaluate the optimality gap of current trial policy.

Step 3: Add new optimality constraint to HMP and return to **Step 1** until termination.

Step 4: Network slice running management until next-round scheduling.

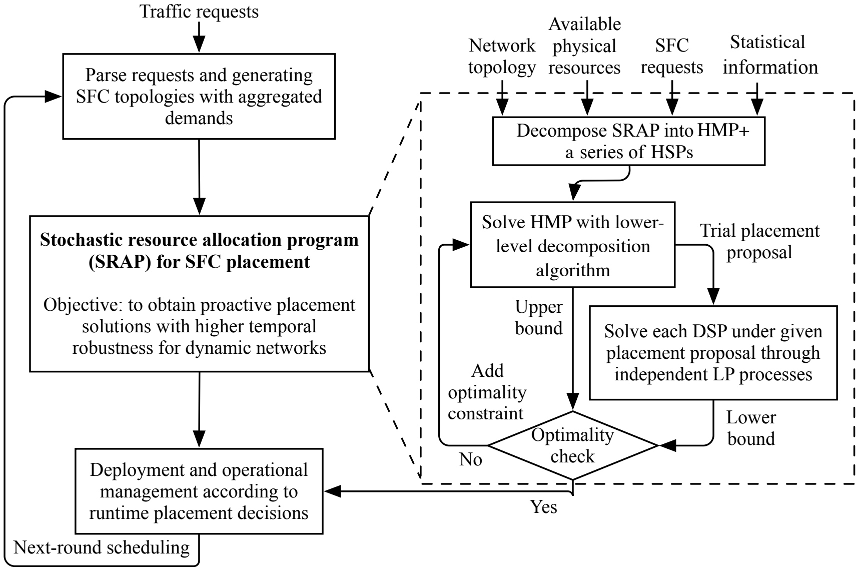

Traffic requests
Parse requests and generating SFC topologies with aggregated demands
Stochastic resource allocation program (SRAP) for SFC placement
Objective: to obtain proactive placement solutions with higher temporal robustness for dynamic networks
Deployment and operational management according to runtime placement decisions
Next-round scheduling
Network topology
Available physical resources
SFC requests
Statistical information
Decompose SRAP into HMP+ a series of HSPs
Solve HMP with lower-level decomposition algorithm
Trial placement proposal
Solve each DSP under given placement proposal through independent LP processes
Upper bound
Lower bound
Add optimality constraint
Optimality check
No
Yes

Fig. 2. Distributed SFC placement diagram based on two-level decomposition.

V. Simulation Results

In this section, we conduct extensive simulation experiments with the settings in accordance with 5G expectations to evaluate the proposed solution.

A. Simulation Setup

Following the similar setups used in the existing NFV/VNE experimental studies, *e.g.* [31], [35], we generate synthetic network slices and random resource demands to support the following simulation experiments. Current BT's IP network topology within Europe[2] is considered as the physical network, which includes 21 nodes and 34 bidirectional links. 5 nodes from these 21 nodes are randomly selected to act as the wireless access nodes. For each node in the network, a fixed amount of computing resources is configured. For each fibre link, the transmission capacity is set proportionally scaled from the practical BT core network bandwidth[3].

In the following simulations, we emulate the envisioned 5G small cells with mmWave spectrum to model the capacities of wireless access links connected to each access node. Considering the fast transitions among Line-of-Sight (LOS), non-LOS (NLOS) and outage network stages in mmWave channels [10], we use the Rician fading for LOS stage and Rayleigh fading [36] for NLOS and outage stages. The transition probabilities between any two channel states are set as equal. The channel parameters are configured so that the resulting wireless capacity of each access node is on average within the envisioned capacity range for a 5G cell [2]. Table II lists the main configurations for the simulation experiments below.

TABLE II. Simulation Setup

Parameters	Value
Node resource capacity c_{vr}	5–10, uniformly distributed
Fixed link capacity c_l	$10Gbps$
Resource prices $[k_l, k_r]$	$[20/Gbps, 20]$
# node resource type $\lvert R\rvert$	1
# VNFs $\lvert\mathcal{F}^s\rvert$	2
Node resource demand d_{fr}	1–3, uniformly distributed
Aggregated rate demand β^s	1–$3Gbps$, uniformly distributed
Service price b^s	100–300, uniformly distributed
Rician factors K	$1dB$
Radio bandwidth B	$1GHz$
Normalized power allocation ρ	LOS: $31.3dB$; NLOS: $9.3dB$; Outage: -$4.3dB$

B. The Compared Algorithms and Performance Metrics

Two reference algorithms, CG_SP [37] and CMG_SP are compared. In CG_SP, the placement decisions are made only to optimize the immediate revenue at t_0 based on already observed network information. However, in CMG_SP, mean state information, $\mathbb{E}_{\beta^s}[\beta^s]_{s\in S}$ and $(\bar{c}_v)v \in V$ are used to represent their future states.

The decisions of CMG_SP are then made to optimize the same objective as the proposed SRA under the current observations and the mean states of futures. CMG_SP is a widely used policy to handle with system dynamics [38]. This comparison can provide an insight to the difference between the exploration of complete PDF knowledge and simple statistic knowledge. For both reference algorithms, the corresponding placement models are solved through the greedy node mapping with shortest path based link mapping [37].

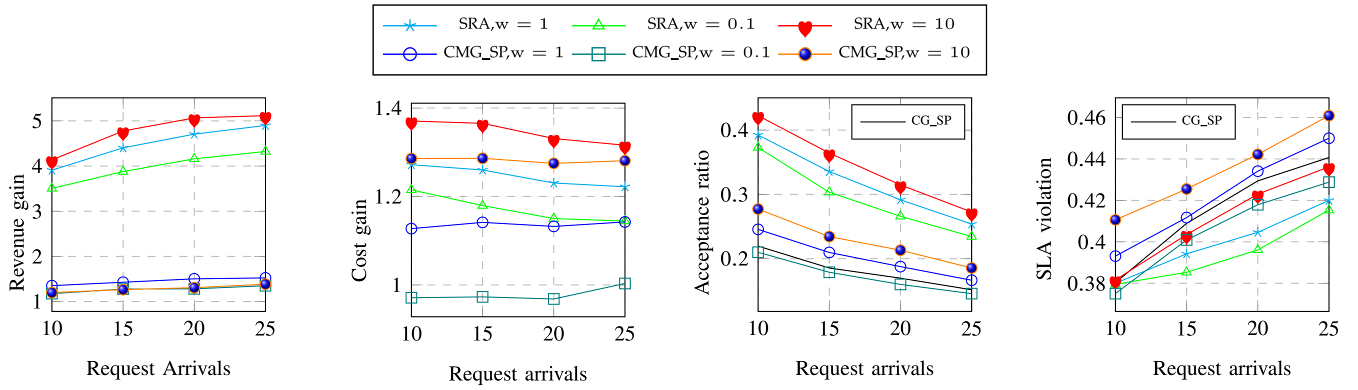
In SRA, the iterative progress is set to stop when the incumbent solution has remained unchanged for 5 iterates. In the lower-level decomposition, the tolerant variation on the monitored utility is 10%. For both levels, the maximum number of iterations is limited to 50.

The following four metrics are used to evaluate the performance of our algorithms against the compared ones.

1) ***Average revenue gain:*** This is defined as the ratio of average revenue achieved by SRA (or CMG_SP) and that by CG_SP within the running period $[0, T]$.

[2] http://www.topology-zoo.org/maps/BtEurope.jpg

[3] https://www.globalservices.bt.com/static/assets/pdf/products/optical_connect/BT_Optical_Connect_datasheet.pdf



(a)

(b)

(c)

(d)

Fig. 3. Performance comparisons with $T = 10$: a) Average revenue gain, b) Provisioning cost gain, c) Acceptance ratio, and d) SLA violation.

2) ***Provisioning cost gain:*** Provisioning cost defines the average cost for occupying physical node and link resources under each placement policy. Accordingly, the provisioning cost gain is the ratio between the provisioning cost in SRA (or CMG_SP) and that in CG_SP.
3) ***Acceptance ratio:*** The acceptance ratio of an algorithm measures the percentage of total requests accepted by different algorithms. Combined with the revenue metric, the acceptance ratio gives a sense of how well an algorithm performs on excluding non-beneficial placements.
4) ***SLA violation:*** This is calculated as $\sum_{s\in S}\sum_{t\in[0,T]}(\pi_s - \gamma^s(t)/\beta^s(t))/\sum_{s\in S}(T+1)\pi_s$. SLA violation measures the average offset degree of the allocated running service rates within $[0,T]$ from the requested rate demands over all **accepted** requests, which is an important metric reflecting users' quality of experience towards the provisioned services.

C. Performance Analysis

Fig. 3 depicts the compared performance under different settings. All performance metrics are calculated by generating 1000 random samples to evaluate each placement policy. Based on the simulation results, our key observations are summarized in the following.

1) From long-term consideration, extra future statistic information can enhance the placement policy with $3 \sim 5$x *better revenue.* Fig. 3a shows the average revenues collected from different algorithms. Under the given settings, the simulation results confirm that the significant revenue improvement of the proposed SRA approach over the referenced two algorithms. Compared with the only 1.5x revenue gain made by the deterministic algorithm in [31] over the same CG_SP benchmark, the proposed SRA presents a more positive results with up to $3 \sim 5$x revenue gain when network dynamics are considered.

Specifically, when the available physical resources are abundant, the possible network variations have little impact on the placement decisions. In this case, the performance gain in SRA are mainly contributed by the more efficient policy computing than the greedy policies. With the increased requests, however, the resource competition among requests gets intensified. As a result, any over-optimistic or -pessimistic placement decisions in CG_SP would be detrimental to the long-term revenue performance. This is avoided in SRA with the joint reference of future statistic information, thus creating a higher gain as resources become scarce.

Benefiting from the calibration to the decisions by the statistical mean values, CMG_SP, on the other hand, also achieves around 1.5x revenue gain over CG_SP. However, due to the non-convexity of the achievable revenue under each combinatorial policy option, the accuracy from statistical mean values is highly compressed than that when complete PDFs are used to capture the future network variations.

2) The nearly 5x *better revenue of SRA only raises about 30% more resource cost.* Fig. 3b shows the compared results on the provisioning cost gain over CG_SP. Combined with the revenue results in Fig. 3a, we can see that SRA achieves up to 5x better revenue but using only 30% more resources than CG_SP. This indicates that the available physical resources are coordinated more efficiently to serve more requests when physical resources become more scarce. However, with the greedy placement policies, about 20% more resource investment only contributes 1.5x revenue gain for CMG_SP.

3) SRA makes more good-quality acceptances. As depicted in Fig. 3c, SRA accepts 2x more services than CM_SP, and the ratio for CMG_SP is 1.5x. Then, we can get that the revenue gains contributed by unit acceptance are 5/2 and 1.5/1.5 for SRA and CMG_SP, respectively. This shows that the acceptances made by SRA are more beneficial, which collectively contribute the higher revenue gain. The degradation of the compared algorithms results from both the acceptance to the requests that are currently beneficial but long-term non-beneficial and the exclusion of requests that are temporarily non-beneficial but long-term beneficial.

4) Services deployed according to SRA policy present lower SLA violation risk. Fig. 3d shows that benefiting from the accurate capture of future network variations, the SLA violation is significantly lower in SRA than the compared ones. This reflects a better long-term robustness and users' quality of experience towards the provisioned services in SRA when network dynamics are presented. In contrast, the statistical mean values only decrease a little the SLA violation risk in CMG_SP.

TABLE III. Performance of SRA under Different w, $T = 10$.

Weight w	Average revenue gain	Provisioning cost gain	Acceptance ratio	SLA violation
0.1	4.32	1.14	23.4%	41.5%
1.0	4.90	1.22	25.4%	42.0%
10	5.12	1.32	27.3%	43.6%
15	4.23	1.10	23.7%	38.3%
20	3.90	1.08	23.3%	38.1%

D. Effect of Different Weighting Balance

In SRA, the weight settings of parameter w show different emphasis on the future expected revenue, which results in a performance tradeoff. The optimal value of w is subject to parameter tuning, depending on the runtime estimation towards the quality of current and future network states. For example, when the observed current network state is believed overwhelmingly better than the average cases, setting a small value of w is more reasonable so that the current good state can be fully exploited. Conversely, if the network state is currently observed to be very bad, a large value of w should be set to leave more spaces to explore potentially better performance in the future. However, estimating the quality of an observed network state is non-trivial when the explicit expression of the system performance over observations is absent.

In this section, we evaluate the effect of different weight settings on the performance of SRA. Table III and Fig. 4 enumerate the compared performance when the system is loaded with request arrivals = 25. Based on the simulation results, the following two behaviors of the proposed SRA solution are observed.

1) Selecting an appropriate weighting balance for each decision is a tradeoff. We observe from Table III that the considered performance metrics exhibit different changes over the setting of weight w. Under the given settings, the difference gap among these average revenue gains is up to 1. Moreover, the average revenue gain and provisioning cost gain show more sensitivity towards the setting of w. In contrast, the variation of w only makes little changes on the SLA violation. This stands to the reason that the SLA violation is averaged over all the accepted requests, thus is normally less sensible to the changes of w than the other metrics.

2) When the network variations follow stationary processes, best weight is not around T, but a value approximately ranging between $[1, 10]$. Stationarity is the property of a stochastic process whose probability distribution is the same at all times [39]. In this case, the averaged long-term observations will finally converge to the mean values of network variations. As a consequence, if weighting according to the average revenue contribution of the immediate and future ones in the objective function (5a), $w = T$ should be a reasonable weight option to balance the immediate and future revenues in the objective function (5a). However, according to the simulation results in Fig. 4, $w = T$ is not always better when T takes different values. This shows the non-convexity property for the average revenue of SRA in terms of w. The calculation of the optimal weight requires the accurate modelling of the system performance over observations, which is complicated in the considered combinatorial optimization scenario. Another alternative option is to set a dynamic weight through some heuristic rules according to every observation. For the practical industrial application of the proposed SRA solution, taking a value between $[1, 10]$ could be a mild weighting option since this setting retains nearly 90% revenue gain in Fig. 4.

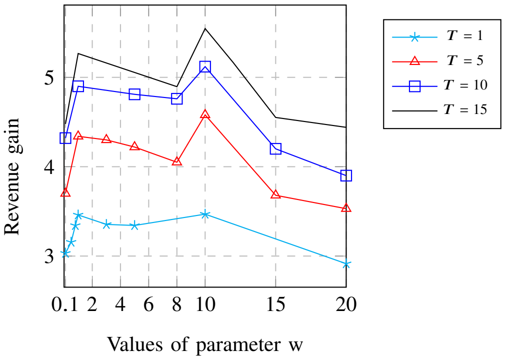


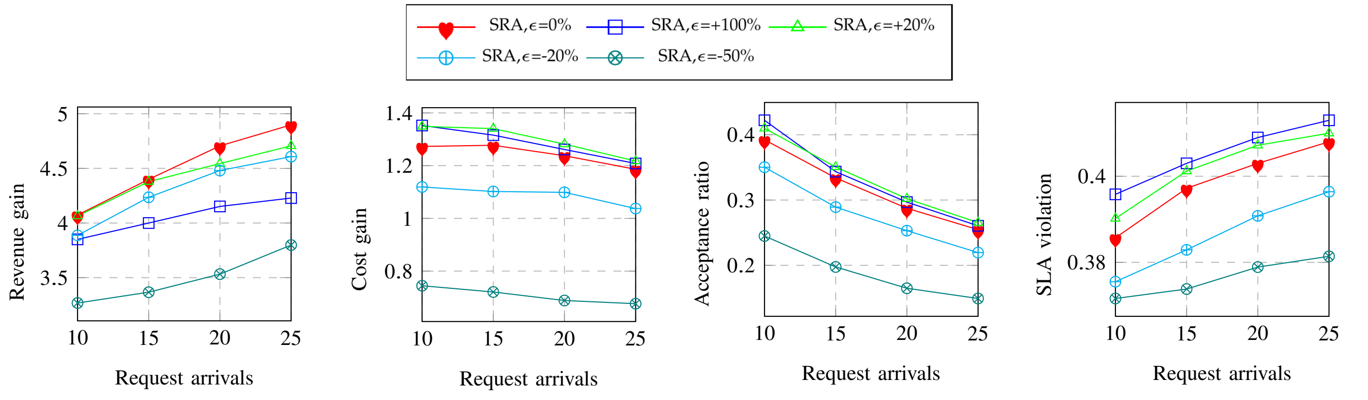
Fig. 4. Weighting effects under different scheduling intervals.

E. Effect of Statistical Error for Future

Stationary random processes are assumed for the network variations in the SRA model. Therefore, a proactively improved decision can be made to create a better long-term revenue based on the given PDFs of network variations. In this section, we release the stationarity assumption and evaluate the performance of SRA when the estimated PDFs are subject to statistical errors or temporal evolution. The error is presented by setting an offset between the mean values of the practical and estimated PDFs. We summarize the observed behaviors of the proposed SRA as follows.

1) The superiority of the proposed SRA is preserved even when the estimated network variations are subject to 50% statistical errors. The results for the considered performance metrics with $w = 1, T = 10$ are depicted in Fig. 5. We use a positive ϵ to denote the optimistic case when the statistical mean of future network variation is estimated 20% more than its practical value. Likewise, pessimistic estimations are evaluated with a negative ϵ to show the effect when the statistical mean of future network variation is estimated 20% less than its practical value. In Fig. 5, multi-fold performance gains are still presented for the SRA under statistical errors. However, it also causes a degradation up to 1 in terms of the revenue gain for the case of $\epsilon = -50\%$ when compared with the results under accurate PDF information (*i.e.*,$\epsilon = 0$).

2) Pessimistic estimation decreases the overall service capacity, while optimistic estimation leads to more bad-quality acceptances. Also depicted in Fig. 5 are the performances of SRA when ϵ takes different offset values. In the pessimistic estimation case, we can observe that with the increased estimation errors, the accepted maximum loads are gradually plummeted from the amount that the system can really serve. The decreased acceptance directly results in the under utilization of network resources and considerable revenue loss. On the other hand, SRA can make more acceptances in the optimistic case than the amount that the system can really serve. However, the increased acceptances only take negative effects, resulting in more revenue loss, provisioning cost and also SLA violation. This turns out that the extra acceptances are actually non-beneficial that should not have been accepted.



(a)

(b)

(c)

(d)

Fig. 5. Performance comparisons under statistical error with w = 1, T = 10: a) Average revenue gain. b) Provisioning cost gain. c) Acceptance ratio. d) SLA violation.

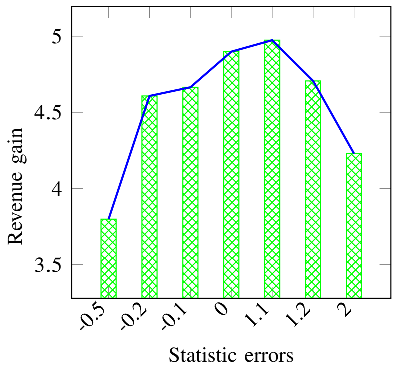


Fig. 6. Revenue performance under different statistic errors with w = 1, T = 10, request arrivals = 25.

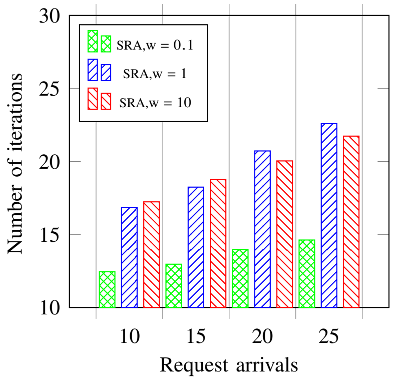


Fig. 7. Number of iterations under different weights.

3) The revenue loss due to the sub-optimality of SRA can be compensated by pre-setting $\epsilon = +10\%$ statistical error. In the case of $\epsilon = +10\%$, we can observe from Fig. 6 that the SRA solver can load more requests than the case with error-free resource estimation. The slightly increased acceptances turn out to be beneficial and finally contribute nearly 10% revenue improvement. This confirms that due to the sub-optimality of the solution in SRA, there are nearly 10% potential revenue loss. However, such loss can be compensated by pre-setting the resource estimation with $\epsilon = +10\%$ statistical error.

F. Convergence Analysis

The main computation cost of the proposed SRA solution comes from recursively calculating the approximate HMP and sample averaged approximation in SPs. In the designed termination criteria, the maximum number of iterations is limited to 50 so that the solution can be generated with a controlled time budget. Fig. 7 shows the average number of iterations under different SRA weighting balance. The following two behaviors can be observed.

1) More iterations are required to make the solution converge when the future revenue is considered with a higher weight than the immediate revenue. The future revenue in the proposed solution is evaluated through random samples at each iteration. When the future revenue takes a higher weight than the immediate revenue, the achieved objective value will get more sensitive to the outcomes of random sampling at each iteration. Consequently, more samplings are required to converge the objective of the model to the tolerant value variation.

2) The required iterations to converge increase when more requests are presented. As shown in Fig. 7, with the increase of request arrivals, more iterations are required to find the accepted placement policies. This is reasonable, since more arrivals lead to more similar policy options to compare with. However, with the fractional placement information used in solving HMP, many unnecessary placement policy trials can be avoided. We can see from Fig. 7 that all experiments finish the computation within an average number of 20 iterations. Combined with the results provided in Fig. 5, the performance achieved under such iteration criteria retains nearly 90% optimality.

Although multiple iterations are required, the complicated combinatorial program in the proposed solution only needs to be solved during service initialization. The proactive deployment decisions, once solved, can be used with robustness across the whole scheduling interval. However, the subsequent service running controls only need to solve a simple linear program. This significantly reduces the computation complexity involved in the course of service running controls.

Moreover, this proposed solution can be further accelerated by harvesting distributed and parallel computing technologies in the proposed computing framework.

Similar to the analysis to the VNE problem in [31], for the general version of the considered problem, theoretical bounds do not exist. It is quite challenging to model the analytic optimality bounds and convergence rate, due to possibly random termination of the iteration progress. A reasonable direction is to explore stochastic and approximate ratio analysis [40] in future work.

VI. Conclusions

In this paper, we have highlighted the network utility degradation problem for NFV in dynamic networks, and accordingly proposed a proactive NFV deployment solution SRA that is robust against network state changes within a certain running period. By exploiting the problem-specific structures, a distributed computing framework with two-level decomposition has been designed to facilitate a distributed implementation of the proposed SRA model in large-scale networks. The simulation experiments have confirmed the performance degradation of existing NFV solutions in dynamic networks, and demonstrated that the proposed SRA solution can achieve up to 5x performance improvement against the compared algorithms. The obtained solution has presented low sensitivity towards parameter errors and even worked robustly with up to 50% statistical errors.

For the future work, more considerations are to be explored in terms of the more general implementation of the SRA algorithm and the challenges in theoretical analysis.

First, in this paper, a fixed pricing strategy is used to control the whole admission and placement decisions. Considering the dynamics in networking market and traffic patterns, more dynamic pricing model is expected. For example, if a request has a long lease time and the market price of resources allocated to that request keeps fluctuating over the lease period, a full-fledged economic model will be required in order to model the revenue function.

Additionally, the model or statistical information of networking environment, such as traffic patterns and the PDFs of network variations, are required in this paper. The accurate and timely acquisition of these knowledge in a dynamic networking environment is non-trivial. Therefore, a knowledge-free model extension is expected for future work to release the assumption of complete and stationary PDF information with technologies *e.g.*, multi-armed bandit learning theory [38], reinforcement learning [41], *etc*.

Acknowledgment

The work of Xiangle Cheng is partially supported by the China Scholarship Council for the study at the University of Exeter. This work is also partially supported by the UK EPSRC project (Grant No.: EP/R030863/1).

References

[1] Cisco White Paper, Cisco Visual Networking Index: Forecast and Methodology, 2016–2021, [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf.

[2] 5GMF White Paper, 5G Mobile Communications Systems for 2020 and beyond, 2017, [Online]. Available: https://5gmf.jp/en/whitepaper/5gmf-white-paper-1-1/.

[3] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing & softwarization: A survey on principles, enabling technologies & solutions,", *IEEE Commun. Surveys Tuts.*, DOI: 10.1109/COMST.2018.2815638. [Online]. Available: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8320765&isnumber=5451756.

[4] P. Rost *et al.*, "Network slicing to enable scalability and flexibility in 5G mobile networks," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 72-79, May 2017.

[5] R. Mijumbi *et al.*, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[6] Z. Liu, S. Wang, and Y. Wang, "Service function chaining resource allocation: A survey," *Cornell Univ. Library*, arXiv: 1608.00095, 2016. [Online]. Available: https://arxiv.org/pdf/1608.00095.pdf.

[7] 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.

[8] A. Jarray and A. Karmouch, "Decomposition approaches for virtual network embedding with one-shot node and link mapping," *IEEE/ACM Trans. Netw.*, vol. 23, no. 3, pp. 1012-1025, Jun. 2015.

[9] Z. Zhao *et al.*, "Autonomic communications in software-driven networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 11, pp. 2431–2445, Nov. 2017.

[10] M. R. Akdeniz *et al.*, "Millimeter wave channel modeling and cellular capacity evaluation," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1164–1179, Jun. 2014.

[11] ETSI NFV, Network Functions Virtualisation (NFV); Architectural framework , GS NFV 002, Oct. 2013.

[12] S.Mehraghdam, M.Keller, and H.Karl, "Specifying and placing chains of virtual network functions," in *Proc. 3rd IEEE Conf. Cloud Netw. (CloudNet)*, Oct. 2014, pp. 7–13.

[13] M. F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," in *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 725–739, Dec. 2016.

[14] M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini, and H. Flinck, "Coalitional game for efficient virtual evolved packet core in 5G networks," in *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 469–484, Mar. 2018.

[15] A. Laghrissi, T. Taleb, and M. Bagaa, "Conformal mapping for optimal network slice planning based on canonical domains," in *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 519–528, Mar. 2018.

[16] T. Taleb, M. Bagaa, and A. Ksentini, "User mobility-aware virtual network function placement for virtual 5G network infrastructure," in *Proc. IEEE ICC 2015*, Jun. 2015, pp. 3879–3884.

[17] M. Bagaa, T. Taleb, and A. Ksentini, "Service-aware network function placement for efficient traffic handling in carrier cloud," in *Proc. IEEE WCNC'14*, Apr. 2014, pp. 2402–2407.

[18] A. Laghrissi, T. Taleb, M. Bagaa, and H. Flinck, "Towards edge slicing: VNF placement algorithms for a dynamic & realistic edge cloud environment," in *Proc. IEEE Globecom 2017*, Dec. 2017, pp. 1–6.

[19] I. Benkacem, T. Taleb, M. Bagaa, and H. Flinck, "Optimal VNFs placement in CDN slicing over multi-cloud environment," in *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 616–627, Mar. 2018.

[20] P. Wang, J. Lan, X. Zhang, Y. Hu, and S. Chen, "Dynamic function composition for network service chain: Model and optimization," *Computer Networks*, vol. 92, Part 2, pp. 408–418, Dec. 2015.

[21] Y. Jia, C. Wu, Z. Li, F. Le, and A. Liu, "Online scaling of NFV service chains across geo-distributed datacenters," *IEEE/ACM Trans. Netw.*, vol. 26, no. 2, pp. 2008–2025, April 2018.

[22] Y. Li, L. T. X. Phan, and B. T. Loo, "Network functions virtualization with soft real-time guarantees," in *Proc. IEEE INFOCOM 2016*, Apr. 2016, pp. 1–9.

[23] M. Ghaznavi *et al.*, "Elastic virtual network function placement," in *Proc. 4th IEEE Conf. Cloud Netw. (CloudNet)*, Oct. 2015, pp. 1–6.

[24] V. Eramo, E. Miucci, M. Ammar, and F. Lavacca, "An approach for service function chain routing and virtual function network instance migration in network function virtualization architectures," *IEEE/ACM Trans. Netw.*, vol. 25, no. 4, pp. 2008–2025, Aug. 2017.

[25] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.

[26] Q.-V. Pham and W.-J Hwang, "Network utility maximization-based congestion control over wireless networks: A survey and potential directives," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1173–1200, 2nd Quart., 2017.

[27] H. M. Bidhandi, J. Patrick, “Accelerated sample average approximation method for two-stage stochastic programming with binary first-stage variables,” *Applied Mathematic Modelling,* vol. 41, pp. 582–595, 2017.

[28] S. Su *et al.*, “Energy-aware virtual network embedding,” in *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1607–1620, Oct. 2014.

[29] Y. Dinitz, N. Garg, M.X. Goemans, “On the single source unsplittable flow problem,” *Proceedings of the 39th Symposium on the Foundations of Computer Science*, Palo Alto, CA, 1998, pp. 290–299.

[30] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge Univ. Press, 2004.

[31] M. Chowdhury, M. R. Rahman, and R. Boutaba, “ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping,” in *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 206–219, Feb. 2012.

[32] A. J. Conejo, E. Castillo, R. Mnguez, and R. Garca-Bertrand, D*ecomposition Techniques in Mathematical Programming: Engineering and Science Applications*, Springer, 2006.

[33] V. Borkar, *Stochastic Approximation: A Dynamical Systems Viewpoint.* Cambridge, U.K.: Cambridge Univ. Press, 2008.

[34] J. Higle and S. Sen, “Stochastic decomposition: An algorithm for two stage linear programs with recourse,” *Math. Oper. Res.*, vol. 16, no. 3, pp. 650–669, 1991.

[35] T. W. Kuo, B. H. Liou, K. C. Lin, and M. J. Tsai, “Deploying Chains of Virtual Network Functions: On the Relation Between Link and Server Usage,” in *Proc. IEEE INFOCOM 2016*, Apr. 2016, pp.1–9.

[36] G. L. Stuber, *Principles of Mobile Communication*, Second Edition, Kluwer Academic Publishers, 2001.

[37] Y. Zhu and M. Ammar, “Algorithms for assigning substrate network resources to virtual network components,” in *Proc. IEEE INFOCOM 2006*, Apr. 2006, pp. 1–12.

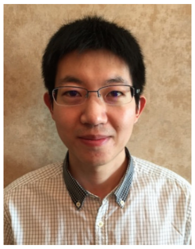
[38] Y. Gai, B. Krishnamachari, and R. Jain, “Combinatorial network optimization with unknown variables: multi-armed bandits with linear rewards and individual observations”, *IEEE/ACM TRANS. Netw.*, vol. 20, no. 5, pp. 1466–1478, Oct. 2012.

[39] P. Z. Peebles, *Probability, Random Variables and Random Signal Principles.* New York, NY, USA: McGraw-Hill, 1993.

[40] G. G. Yin and H. J. Kushner, *Stochastic Approximation and Recursive Algorithms and Applications*. New York, NY, USA: Springer-Verlag, 2003.

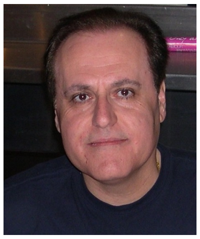
[41] X. Chen, J. Wu, Y. Cai, H. Zhang, and T. Chen, “Energy-efficiency oriented traffic offloading in wireless networks: A brief survey and a learning approach for heterogeneous cellular networks,” *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 627–640, April 2015.

Xiangle Cheng received the M.Sc. degree in communication and information system from Southwest Jiaotong University, Chengdu, China in 2015. He is currently a Ph.D. candidate in Computer Science at the University of Exeter, UK. His research interests include Future Internet Architecture and Technologies, Intelligent Wireless Networks and Mobile Computing, Network Virtualization, Stochastic Combinatorial Optimization, and Dynamic System Modelling and Performance Optimisation.



Yulei Wu is a Lecturer in Computer Science at the University of Exeter. He received his Ph.D. degree in Computing and Mathematics and B.Sc. (First Class Hons) degree in Computer Science from the University of Bradford, UK, in 2010 and 2006, respectively. His main research focuses on Future Internet Architecture and Technologies, Smart Network Management, Green Networking, Big Data for Networking, and Analytical Modelling and Performance Optimisation. His recent research has been supported by UK EPSRC, National Natural Science Foundation of China, University’s Innovation Platform and industries.

Geyong Min is a Professor of High Performance Computing and Networking in the Department of Computer Science within the College of Engineering, Mathematics and Physical Sciences at the University of Exeter, United Kingdom. He received the PhD degree in Computing Science from the University of Glasgow, United Kingdom, in 2003, and the B.Sc. degree in Computer Science from Huazhong University of Science and Technology, China, in 1995. His research interests include Future Internet, Computer Networks, Wireless Communications, Multimedia Systems, Information Security, High Performance Computing, Ubiquitous Computing, Modelling and Performance Engineering.



Albert Y. Zomaya is the *Chair Professor of High Performance Computing & Networking* in the School of Information Technologies, University of Sydney, and he also serves as the Director of the Centre for Distributed and High Performance Computing. Professor Zomaya published more than 550 scientific papers and articles and is author, co-author or editor of more than 20 books. He is the Founding Editor in Chief of the *IEEE Transactions on Sustainable Computing* and serves as an associate editor for more than 20 leading journals. Professor Zomaya served as an Editor in Chief for the *IEEE Transactions on Computers* (2011-2014).

Professor Zomaya is the recipient of the *IEEE Technical Committee on Parallel Processing Outstanding Service Award* (2011), the *IEEE Technical Committee on Scalable Computing Medal for Excellence in Scalable Computing* (2011), and the *IEEE Computer Society Technical Achievement Award* (2014), and the *ACM MSWIM Reginald A. Fessenden Award* (2017). He is a Chartered Engineer, a Fellow of AAAS, IEEE, and IET. Professor Zomayas research interests are in the areas of parallel and distributed computing and complex systems.