Near-optimal Cloud-Network Integrated Resource Allocation for Latency-Sensitive B5G

Masoud Shokrnezhad and Tarik Taleb
Centre for Wireless Communications (CWC)
Oulu University, Oulu, Finland
{masoud.shokrnezhad; tarik.taleb}@oulu.fi

Abstract—Nowadays, while the demand for capacity continues to expand, the blossoming of Internet of Everything is bringing in a paradigm shift to new perceptions of communication networks, ushering in a plethora of totally unique services. To provide these services, Virtual Network Functions (VNFs) must be established and reachable by end-users, which will generate and consume massive volumes of data that must be processed locally for service responsiveness and scalability. For this to be realized, a solid cloud-network Integrated infrastructure is a necessity, and since cloud and network domains would be diverse in terms of characteristics but limited in terms of capability, communication and computing resources should be jointly controlled to unleash its full potential. Although several innovative methods have been proposed to allocate the resources, most of them either ignored network resources or relaxed the network as a simple graph, which are not applicable to Beyond 5G because of its dynamism and stringent QoS requirements. This paper fills in the gap by studying the joint problem of communication and computing resource allocation, dubbed CCRA, including VNF placement and assignment, traffic prioritization, and path selection considering capacity constraints as well as link and queuing delays, with the goal of minimizing overall cost. We formulate the problem as a non-linear programming model, and propose two approaches, dubbed B&B-CCRA and WF-CCRA respectively, based on the Branch & Bound and Water-Filling algorithms. Numerical simulations show that B&B-CCRA can solve the problem optimally, whereas WF-CCRA can provide near-optimal solutions in significantly less time.

Index Terms—Beyond 5G, 6G, Computing First Networking, Cloud-Network Integration, Resource Allocation, Path Selection, Traffic Prioritization, VNF Placement, and Optimization Theory.

I. INTRODUCTION

As of today, the major reason for the evolution of networks has been a surge in data flow, which has resulted in a continuous 1000x gain in capacity. While this demand for capacity will continue to expand, the blossoming of Internet of Everything is forging a paradigm shift to new-born perceptions bringing a range of entirely novel services with rigorous deterministic criteria, such as connected robotics, smart healthcare, autonomous transportation, and extended reality [1]. The provision of these services will be accomplished by establishing several functional components, namely Virtual Network Functions (VNFs), which will generate and consume vast amounts of data that must be processed locally for service responsiveness and scalability.

A distributed cloud architecture is critical in these situations [2], which could be realized through a solid cloud-network

integrated infrastructure built of distinct domains in Beyond 5G (B5G) [3]. These domains can be distinguished by the technology employed, including radio access, transport, and core networks, as well as edge, access, aggregation, regional, and central clouds. Additionally, these resources can be virtualized through the use of technologies such as Network Function Virtualization (NFV), which allows for the creation of isolated virtual entities atop this physical infrastructure. Since distributed cloud and network domains would be diverse in terms of characteristics but limited in terms of capability, communication and computing resources should be jointly allocated, prioritized, and scheduled to ensure maximum Quality of Service (QoS) satisfaction while maximizing resource sharing and maintaining the system in a deterministic state, resulting in energy savings as one of the most significant examples of cost minimization objectives [4].

The joint problem of resource allocation in cloud-network integrated infrastructures has been extensively studied in the literature. In [5], the authors examined the VNF placement problem as an Integer Linear Programming (ILP) model that assures the minimal End-to-End (E2E) latency while maintaining QoS requirements by not exceeding an acceptable latency violation limit. They suggested an approach based on neural networks and demonstrated that it can produce near-optimal solutions in a timely manner. The authors in [6] investigated the same problem and proposed a hierarchical reinforcement learning method that includes local level prediction modules as well as a global learning component. They demonstrated that their method significantly outperforms conventional approaches. The similar topic was investigated in [7], with the goal of maximizing the number of accepted requests, and a Markov decision process design was presented. They asserted that the proposed method provides efficient placements.

Although innovative approaches are presented in [5]–[7] to address computing resource constraints, the network is solely viewed as a pipeline in these papers with no cognitive ability to the cloud domains. However, there are also some studies in the literature that have been concentrating on communication and computing resources jointly. In [8], the joint problem of VNF placement and path selection was investigated to better utilize the network resources, and a heuristic approach was proposed to tackle it. The authors of [9] and [10] addressed the problem of VNF placement with the goal of maximizing the sum rate of accepted requests. In [9], an optimization solver is used to

find the optimal solution, while the solution approach offered in [10] is a heuristic strategy. The authors of [11] formulated the latency-optimal placement of functions as an ILP problem and proposed a genetic meta-heuristic algorithm to solve it. In [12], to reduce the cost of computing resources, the problem of VNF placement and scheduling was addressed, and a latency-aware heuristic algorithm was devised.

The methods proposed in the cited studies are clearly effective in addressing the resource allocation problem. They cannot, however, be used in B5G systems. Because of the stringent QoS requirements in the delay-reliability-rate space [13], the large number of concurrent services and requests, and the ever-changing dynamics of both infrastructure and end-user service consumption behavior across time and space, every detail of communication and computing resources should be decided and controlled towards achieving a deterministic B5G system [3]. In [8], latency-related constraints and requirements are simply disregarded. Despite the fact that delay is addressed in the rest of these studies, they simplified it to be a link feature, and queuing delay is completely eliminated. Furthermore, path selection is ignored in [9]–[11], and cost optimization is overlooked in [10], [11].

This paper fills in the gap in the existing works by studying the joint problem of allocating communication and computing resources, including VNF placement and assignment, traffic prioritization, and path selection while taking into account capacity constraints as well as link and queuing delays, with the goal of minimizing overall cost. Our main contributions in this paper are as follows

- Formulating the joint resource allocation problem of the cloud-network integrated infrastructure as a Mixed Integer Non-Linear Programming (MINLP) problem.
- Proposing a method based on Branch & Bound (B&B) algorithm to find the optimal solution of the problem.
- Devising a heuristic approach based on the Water-Filling (WF) algorithm in order to identify near-optimal solutions to the problem.

The reminder of this paper is organized as follows. Section II introduces the system model. Formulating the resource allocation problem is provided in Section III. Next, the B&B and heuristic approaches are provided in Sections IV and V, respectively. Numerical results are illustrated and analyzed in Section VI, followed by concluding remarks in Section VII.

II. SYSTEM MODEL

In the following, we describe the main components of the system studied in this paper: infrastructure, services, and requests. The system model is also depicted in Fig 1.

A. Infrastructure Model

The considered infrastructure is composed of the access and core network domains (non-radio domains) consisting of $\mathcal{V}$ nodes, $\mathcal{L}$ links, and $\mathcal{P}$ paths denoted by $\mathcal{G} = \langle \mathcal{V}, \mathcal{L}, \mathcal{P} \rangle$. $\mathcal{V} = \{v | v \in \{1, 2, ..., \mathcal{V}\}\}$ is the set of nodes. $\mathcal{L} \subset \{l : (v, v') | v, v' \in \mathcal{V}\}$ indicates the set of links, and for each l, its bandwidth capacity is constrained by $\widehat{B_l}$, and it costs Ξ_l per
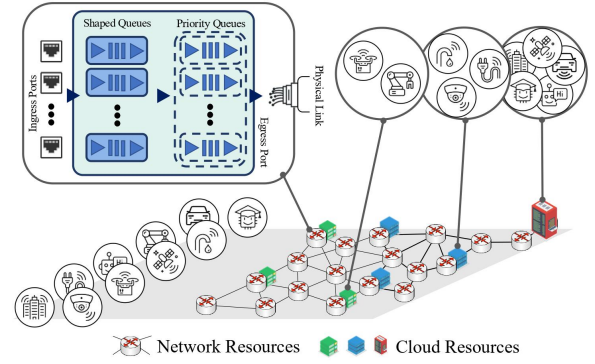

Fig. 1. System model

capacity unit. $\mathcal{P} = \{p : (\vdash_p, \dashv_p) | p \subset \mathcal{L}\}$ denotes the set of all paths in the network, where $\vdash_p$ and $\dashv_p$ are the head and tail nodes of path p, and $l'_{l,p}$ is a binary constant equal to 1 if path p contains link l.

Each node in the network is an IEEE 802.1 Time-Sensitive Networking (TSN) device comprising an IEEE 802.1 Qcr Asynchronous Traffic Shaper (ATS) at each of their egress ports. An ATS consists of two hierarchical queuing steps [14]: interleaved shaping, and scheduling through a set of prioritized queues. We consider $\mathcal{K} = \{k | k \in \{1, 2, ..., \mathcal{K}\}\}$ as the set of priority levels and assume that k_r is the assigned priority of the traffic associated with request r, and the size of the shaping queues for priority level k is the same and equal to $\widehat{\mathcal{T}_k}$. Note that lower levels have higher priorities. Moreover, each node v is equipped with computing resources as one of the prospective hosts to deploy service VNFs and limited to a predefined capacity threshold $\widehat{\zeta_v}$ which costs Ψ_v per capacity unit.

It is worth mentioning that the network is divided into a number of tiers, with nodes distributed across them so that the entry nodes of requests are located in tier 0. The higher the tier index, the greater the capacity of the associated nodes, and the lower their cost. In other words, the nodes closest to end-users (or to the nodes that serve as entry points) are provisioned with high-cost, limited-capacity computing facilities, while low-cost, high-capacity depots are deployed in the core.

B. Service Model

The set of services obtainable to order is dubbed by $\mathcal{S} = \{s | s \in \{1, 2, ..., \mathcal{S}\}\}$, where $\mathcal{S}$ indicates the number of services. If an end-user requests a service, its VNF has to be replicated in the network-embedded computing resources. Each VNF is empowered to serve more than one request, and $\widehat{\mathcal{C}_s}$ indicates the maximum capacity of each VNF of service s.

C. Request Model

The set of requests asking for services is represented by $\mathcal{R} = \{r | r \in \{1, 2, ..., \mathcal{R}\}\}$, where $\mathcal{R}$ is the number of requests. Each request r arrives in the network through node v_r, one of the nodes through which the infrastructure connects to the radio access network, and intends a service s_r specifying

its minimum necessitated service capacity, network bandwidth, and maximum tolerable delay, indicated by $\widetilde{\mathcal{C}}_r$, $\widetilde{\mathcal{B}}_r$, and $\widetilde{\mathcal{D}}_r$, respectively. In addition, $\widetilde{\mathcal{T}}_r$ and $\widehat{\mathcal{H}}_r$, denoting the burstiness of traffic and the largest packet size for request r, are also assumed to be known a priori. Utilizing historical data along with predictive data analytics methods is one of the viable options for obtaining such accurate and realistic statistical estimates of traffic.

III. PROBLEM DEFINITION

In this section, the joint problem of VNF placement and assignment, traffic prioritization, and path selection is described. The constraints and objective function are formulated as a MINLP problem in what follows, and the problem is stated at the end of the section.

A. VNF Placement and Assignment Constraints

To begin, each request must be assigned a single node as its service location (C1). This assignment is acceptable if the assigned node hosts a VNF for the requested service (C2). Following that, it must be ensured that the assigned requests do not violate the capacity constraints of VNFs and nodes (C3 and C4). Note that the capacity constraints are intrinsically linked to avoiding congestion and ensuring the system's reliability. The formulation is as follows:

$$\sum_{\mathcal{V}} g_{r,v} = 1, \forall r \in \mathcal{R}, \tag{C1}$$

$$g_{r,v} \le z_{s_r,v}, \forall r \in \mathcal{R}, \forall v \in \mathcal{V}, \tag{C2}$$

$$\widetilde{\mathcal{C}}_r \sum_{\{r|r\in\mathcal{R}\wedge s_r=s\}} g_{r,v} \le \widehat{\mathcal{C}}_s, \forall v \in \mathcal{V}, \forall s \in \mathcal{S}, \tag{C3}$$

$$\widehat{\mathcal{C}}_s \sum_{\mathcal{S}} z_{s,v} \le \widehat{\zeta}_v, \forall v \in \mathcal{V}, \tag{C4}$$

where $g_{r,v}$ and $z_{s,v}$ are binary variables. $g_{r,v}$ is 1 if node v is selected as the service node of request r, and $z_{s,v}$ is 1 if service s is replicated on node v.

B. Traffic Prioritization and Path Selection Constraints

First, we must ensure that each request is assigned to exactly one priority level (C5). Then, the request and reply paths of each request are determined (C6 and C7). For each request, a single request path is chosen that starts at the request's entry node and ends at the request's VNF node. The reply path follows the same logic but in reverse order. The following two constraints guarantee that the two paths are chosen on the priority level assigned to each request (C8 and C9). Finally, the constraints maintaining the maximum capacity of links and shaping queues are enforced (C10 and C11). With C10, the sum of the required bandwidth for all requests whose request or reply path, or both, contains link l is guaranteed to be less than or equal to the link's capacity, and in C11, the capacity

of shaping queues is guaranteed in the same way for each link and each priority level. The set includes:

$$\sum_{\mathcal{K}} \varrho_{r,k} = 1, \forall r \in \mathcal{R}, \tag{C5}$$

$$\sum_{\{p|p\in\mathcal{P}\wedge\vdash_p=v_r\wedge\dashv_p=v\},\mathcal{K}} \overrightarrow{f_{r,p,k}} = g_{r,v}, \forall r \in \mathcal{R}, \forall v \in \mathcal{V}, \tag{C6}$$

$$\sum_{\{p|p\in\mathcal{P}\wedge\vdash_p=v\wedge\dashv_p=v_r\},\mathcal{K}} \overleftarrow{f_{r,p,k}} = g_{r,v}, \forall r \in \mathcal{R}, \forall v \in \mathcal{V}, \tag{C7}$$

$$\sum_{\mathcal{P}} \overrightarrow{f_{r,p,k}} = \varrho_{r,k}, \forall r \in \mathcal{R}, \forall k \in \mathcal{K}, \tag{C8}$$

$$\sum_{\mathcal{P}} \overleftarrow{f_{r,p,k}} = \varrho_{r,k}, \forall r \in \mathcal{R}, \forall k \in \mathcal{K}, \tag{C9}$$

$$\sum_{\mathcal{R}} \widetilde{\mathcal{B}}_r \sum_{\mathcal{P},\mathcal{K}} l'_{l,p}(\overrightarrow{f_{r,p,k}} + \overleftarrow{f_{r,p,k}}) \le \widehat{B}_l, \forall l \in \mathcal{L}, \tag{C10}$$

$$\sum_{\mathcal{R}} \widetilde{\mathcal{T}}_r \sum_{\mathcal{P}} l'_{l,p}(\overrightarrow{f_{r,p,k}} + \overleftarrow{f_{r,p,k}}) \le \widehat{\mathcal{T}}_k, \forall k \in \mathcal{K}, \forall l \in \mathcal{L}, \tag{C11}$$

where $\varrho_{r,k}$ is a binary variable representing the assigned priority level of request r, and $\overrightarrow{f_{r,p,k}}$ and $\overleftarrow{f_{r,p,k}}$ are binary variables that reflect the request and reply paths for request r on priority level k, respectively.

C. Delay Constraints

The final set guarantees the minimum delay requirement of requests as follows:

$$D_{r,s_r} = \frac{\widetilde{\mathcal{H}}_r}{\widetilde{\mathcal{C}}_r}, \forall r \in \mathcal{R}, \tag{C12}$$

$$D_{r,k,l} = \frac{\sum_{\mathcal{R}_1} \widetilde{\mathcal{T}}_{r'} + \bigwedge_{\mathcal{R}_2} \widetilde{\mathcal{H}}_{r'}}{\widehat{B}_l - \sum_{\mathcal{R}_3} \widetilde{\mathcal{B}}_{r'}} + \frac{\widetilde{\mathcal{H}}_r}{\widehat{B}_l}, \forall k \in \mathcal{K}, \forall l \in \mathcal{L}, \tag{C13}$$

$$D_r = \sum_{\mathcal{P},\mathcal{L},\mathcal{K}} D_{r,k,l} l'_{l,p}(\overrightarrow{f_{r,p,k}} + \overleftarrow{f_{r,p,k}}) + D_{r,s_r}, \forall r \in \mathcal{R} \tag{C14}$$

$$D_r \le \widetilde{\mathcal{D}}_r, \forall r \in \mathcal{R}, \tag{C15}$$

where $D_{r,k,l}$, D_{r,s_r} and D_r are continues variables denoting the delay experienced by a given flow of request r associated to priority level k passing through ATS-based link l [14], its computing delay, and the corresponding E2E delay calculated as the sum of the delays on the links that comprise both paths of the request and its computing delay. Besides, $\bigwedge$ is a function which returns the max value over the given set, $\mathcal{R}_1$ represents $\{r'|r' \in \mathcal{R} \wedge k_{r'} \le k \wedge l'_{l,p}(\overrightarrow{f_{r',p,k_{r'}}} + \overleftarrow{f_{r',p,k_{r'}}}) > 0\}$, $\mathcal{R}_2$ is $\{r'|r' \in \mathcal{R} \wedge k_{r'} > k \wedge l'_{l,p}(\overrightarrow{f_{r',p,k_{r'}}} + \overleftarrow{f_{r',p,k_{r'}}}) > 0\}$, and $\mathcal{R}_3$ denotes $\{r'|r' \in \mathcal{R} \wedge k_{r'} < k \wedge l'_{l,p}(\overrightarrow{f_{r',p,k_{r'}}} + \overleftarrow{f_{r',p,k_{r'}}}) > 0\}$. In other words, these sets represent requests that share the same link as r, whereas $\mathcal{R}1$ includes requests with a higher or equal priority, $\mathcal{R}_2$ contains requests with a lower priority, and $\mathcal{R}_3$ shares requests with a higher priority.

D. Objective Function

The objective function is to minimize the total cost of allocated resources, that is:

$$\sum_{\mathcal{R},\mathcal{V}} \Psi_v g_{r,v} + \sum_{\mathcal{R},\mathcal{L}} \Xi_l \sum_{\mathcal{P},\mathcal{K}} l'_{l,p}(\overrightarrow{f_{r,p,k}} + \overleftarrow{f_{r,p,k}}). \tag{OF}$$

Given that the cost of links and computing resources would be easily linked to their energy consumption, the objective function could be interpreted as minimizing energy consumption as a critical goal of developing B5G sustainable communication and computing systems [1].

E. Problem

Considering the constraints and objective function, the problem of Communication and Computing Resource Allocation (CCRA) is:

$$\text{CCRA: } min \text{ OF } s.t. \text{ C1 - C15.} \tag{1}$$

IV. B&B-CCRA

The problem specified in (1) is NP-hard (the multidimensional knapsack problem [15] can be reduced to it, as detailed in [16]), and finding its optimal solution in polynomial time is mathematically intractable. One potential strategy for addressing such a problem is to restrict its solution space using the B&B algorithm, which relaxes and solves the problem to obtain lower bounds, and then improves the bounds using mathematical techniques to reach acceptable solutions. The method is described in Algorithm 1. In this algorithm, the solution space is discovered by maintaining an unexplored candidate list $\mathcal{N} = \{N_t | t \geq 1\}$, where each node N_t contains a problem, denoted by Φ_t, and t is the iteration number. This list only contains the root candidate N_1 at the beginning with the primary problem to be solved. To reduce its enormous computational complexity, instead of directly applying the B&B algorithm to CCRA, we consider its integer linear transformation as the problem of N_1.

CCRA comprises non-linear constraints C13 and C14. To linearize C13, the summations and max function with variable boundaries should be converted to a linear form. A simple, effective technique is to replace each term with an approximated upper bound. Since the aggregated traffic burstiness is bounded by $\widehat{\mathcal{T}_k}$ for each priority level k in C11, $\sum_{\mathcal{R}_1} \widetilde{\mathcal{T}_{r'}}$ can be replaced by the sum of this bound for all priority levels greater than or equal to k, that is $\sum_{\{k' | k' \leq k\}} \widehat{\mathcal{T}_{k'}}$. In a similar way, we define a new constraint (C13′) for the aggregated bandwidth allowed on priority level k over link l, dubbed $\widehat{f_{l,k}}$, and replace the sum of allocated bandwidths with $\sum_{\{k' | k' < k\}} \widehat{f_{l,k'}}$. Besides, the maximum packet size for a particular subset of requests can be replaced by the maximum permitted packet size in the network, denoted by $\widehat{\mathcal{H}}$. Therefore, the followings define the linear transformation of C13:

$$\sum_{\mathcal{R}} \widetilde{\mathcal{B}_r} \sum_{\mathcal{P}} l'_{l,p}(\overrightarrow{f_{r,p,k}} + \overleftarrow{f_{r,p,k}}) \leq \widehat{f_{l,k}}, \forall k \in \mathcal{K}, \forall l \in \mathcal{L}, \tag{C13′}$$

$$\widehat{D_{r,k,l}} = \frac{\sum_{\{k'|k' \leq k\}} \widehat{\mathcal{T}_{k'}} + \widehat{\mathcal{H}}}{\widetilde{B_l} - \sum_{\{k'|k' < k\}} \widehat{f_{l,k'}}} + \frac{\widetilde{\mathcal{H}_r}}{\widetilde{B_l}}, \forall r \in \mathcal{R}, \forall k \in \mathcal{K}, \tag{C13″}$$
$$\forall l \in \mathcal{L},$$

where $\widehat{D_{r,k,l}}$ is the delay upper bound for request r on link l with priority level k. Since D_{r,s_r} is linear, C14 can be linearized by substituting the actual delay for the upper bound derived in C13″, and the new constraint for E2E delay is:

$$D_r = \sum_{\mathcal{P},\mathcal{L},\mathcal{K}} \widehat{D_{r,k,l}} l'_{l,p}(\overrightarrow{f_{r,p,k}} + \overleftarrow{f_{r,p,k}}) + D_{r,s_r}, \forall r \in \mathcal{R}. \tag{C14′}$$

Given this, the linear transformation of CCRA, dubbed LiC-CRA, is as follows:

$$\text{LiCCRA: } min \text{ OF } s.t. \text{ C1 - C12, C13′, C13″, C14′, C15.} \tag{2}$$

Algorithm 1 B&B-CCRA
___
1: $\mathcal{N} \leftarrow \{N_1\}$, $\eta^\star \leftarrow +\infty$, $t \leftarrow 0$
2: **while** $\mathcal{N}$ is not empty **do**
3: | $t \leftarrow t + 1$
4: | $N_t \leftarrow$ selects a node form $\mathcal{N}$
5: | $\mathcal{N} \leftarrow \mathcal{N} \backslash \{N_t\}$
6: | $(\boldsymbol{\mu}_t^\star, \boldsymbol{\lambda}_t^\star), \phi_t^\star \leftarrow$ solve the relaxed problem of Φ_t
7: | **if** $\boldsymbol{\mu}_t^\star$ is integer for all elements **then**
8: | | $\eta^\star \leftarrow min(\eta^\star, \phi_t^\star)$
9: | **else if** $\phi_t^\star < \eta^\star$ is preserved **then**
10: | | $N_t^1, N_t^2 \leftarrow$ two children of N_t
11: | | $\mathcal{N} \leftarrow \mathcal{N} \cup \{N_t^1, N_t^2\}$
___

Now, with LiCCRA as Φ_1, each iteration of the B&B algorithm begins with the selection and removal of a candidate from the unexplored list. Then, the problem of this candidate is naturally relaxed and solved, i.e., all the integer variables ($\in \{0, 1\}$) are replaced with their continues equivalents restricted by the box constraint ($\in [0, 1]$), and the relaxed problem is solved using a Linear Programming (LP) solver to obtain the solution of the relaxed problem $(\boldsymbol{\mu}_t^\star, \boldsymbol{\lambda}_t^\star)$ and the optimal objective value $\phi_t^\star$, where $\boldsymbol{\mu}$ is the relaxed integer variables set, and $\boldsymbol{\lambda}$ is the set of continuous variables. Next, if all relaxed variables have integer values, the obtained objective in this iteration is considered to update the best explored integer solution. Otherwise, a variable index j is selected such that $\boldsymbol{\mu}_t^\star[j]$ is fractional, and the feasible constraints set π_t is divided into two parts as $\pi_t^1 = \pi_t \cap \{\boldsymbol{\mu}_t[j] \leq \lfloor \boldsymbol{\mu}_t^\star[j] \rfloor\}$ and $\pi_t^2 = \pi_t \cap \{\boldsymbol{\mu}_t[j] \geq \lceil \boldsymbol{\mu}_t^\star[j] \rceil\}$. Then, two problems are formed as $\Phi_t^1 = min \text{ OF } s.t. \pi_t^1$ and $\Phi_t^2 = min \text{ OF } s.t. \pi_t^2$. Now, two child nodes N_t^1 and N_t^2, whose problems are Φ_t^1 and Φ_t^2 respectively, are put into the unexplored list. The B&B algorithm is iterated until $\mathcal{N}$ is empty.

Alternatively, we can run this algorithm until a desired solving time is reached or an acceptable objective value is acquired. The prime advantage of this algorithm is that it produces at least a lower bound even when the solving time is limited. As a result, it may be used to establish baselines allowing for the evaluation of alternative approaches.

V. WF-CCRA

Since the B&B method searches the problem's solution space for the optimal solution, its complexity can grow up to the size of the solution space in the worst case [17]. Given that the size of the solution space in CCRA (or LiCCRA) for each request is $\mathcal{V}^2|\mathcal{P}|^2\mathcal{K}^3$ considering its integer variables, the problem's overall size is $\mathcal{R}^{\mathcal{V}^2|\mathcal{P}|^2\mathcal{K}^3}$. Therefore, finding its optimal solution for large-scale instances using B&B is impractical in a timely manner, and the goal of this section is to devise an efficient approach based on the WF concept in order to identify near-optimal solutions for this problem.

The WF-CCRA method is elaborated in Algorithm 2. The first step is to initialize the vectors of parameters and variables used in (1) (or (2)). Following that, two empty sets, $\mathcal{R}'$ and Ω, are established. The former maintains the set of

Algorithm 2 WF-CCRA

1: initialize variable and parameter vectors
2: $\mathcal{R'} \leftarrow \{\}$, $\mathbf{\Omega} \leftarrow \{\}$
3: sort $\mathcal{R}$ in ascending order according to $\widetilde{\mathcal{D}_r}$
4: **while** $\mathcal{R}$ is not empty **do**
5: | **for** $v \in \mathcal{V}$ **do**
6: | | **if** $z_{s_r,v} == 1$ **and** $\widetilde{\mathcal{C}_r} \leq \widehat{\mathcal{C}_{s_r}}$ on v **then**
7: | | | $g_{r,v} = 1$
8: | | **if** $z_{s_r,v} \neq 1$ **and** $\widehat{\mathcal{C}_{s_r}} \leq \widehat{\zeta_v}$ **then**
9: | | | $z_{s_r,v} = 1$, $g_{r,v} = 1$
10: | | **else go to the next iteration**
11: | | **for** $k \in \mathcal{K}$ **do**
12: | | | $\varrho_{r,k} = 1$
13: | | | **for** $p \in \mathcal{P} \wedge \vdash_p = v_r \wedge \dashv_p = v$ **do**
14: | | | | **if** $\widetilde{\mathcal{B}_r} \leq \widehat{\mathcal{B}_l}$ & $\widetilde{\mathcal{T}_r} \leq \widetilde{\mathcal{T}_k}$ on l $\forall l \in \mathcal{L} \wedge l'_{l,p} = 1$ **then**
15: | | | | | $\overrightarrow{f_{r,p,k}} = 1$
16: | | | | | **for** $p' \in \mathcal{P} \wedge \vdash_{p'} = v \wedge \dashv_{p'} = v_r$ **do**
17: | | | | | | **if** $\widetilde{\mathcal{B}_r} \leq \widehat{\mathcal{B}_l}$ & $\widetilde{\mathcal{T}_r} \leq \widetilde{\mathcal{T}_k}$ on l $\forall l \in \mathcal{L} \wedge l'_{l,p'} = 1$ **then**
18: | | | | | | | $\overleftarrow{f_{r,p',k}} = 1$
19: | | | | | | | calculate D_r based on (C14) (or C14′)
20: | | | | | | | **if** $D_r \leq \widetilde{\mathcal{D}_r}$ **then**
21: | | | | | | | | $\mathbf{\Omega} \leftarrow \mathbf{\Omega} \cup \{(z_{s_r,v}, g_{r,v}, \varrho_{r,k}, \overrightarrow{f_{r,p,k}}, \overleftarrow{f_{r,p',k}})\}$
22: | fix assignments of $argmin_{\mathbf{\Omega}}$ (OF) for r
23: | update capacities, $\mathbf{\Omega} \leftarrow \{\}$, $\mathcal{R} \leftarrow \mathcal{R}/\{r\}$, $\mathcal{R'} \leftarrow \mathcal{R'} \cup \{r\}$

Parameter	Value
number of links ($\mathcal{L}$)	$\sim \mathcal{U}\{3\mathcal{V}, 5\mathcal{V}\}$, where the resulted graph is connected.
Number of priority levels ($\mathcal{K}$)	4
Number of services ($\mathcal{S}$)	3
Number of tiers	3
Capacity of each link ($\widehat{B_l}$)	$\sim \mathcal{U}\{100, 150\}$ Gbps
Cost of each link (Ξ_l)	$\sim \mathcal{U}\{10, 20\}$
Capacity of each node ($\widehat{\zeta_v}$)	$\sim 50\,\mathcal{U}(\alpha, \alpha+1)$ Gbps
Cost of each node (Ψ_v)	$\sim 50^{\mathcal{U}(\alpha,\alpha+1)}$
Priority bandwidth upper bound ($\widehat{f_{l,k}}$)	$\widehat{B_l}/\mathcal{K}$

α is the number of tiers minus the tier number of the node.

accepted requests, and the latter stores the feasible resource combinations for each request during its iteration. Now, the algorithm iterates through each request in $\mathcal{R}$, starting with the one with the most stringent delay requirement, and keeps track of the feasible allocations of VNF, priority, as well as request and reply paths based on the constraints of (1) (or (2)). The final steps of each iteration are to choose the allocation with the lowest cost and fix it for the request, as well as to update remaining resources and the set of pending and accepted requests. When there is no pending request, the algorithm terminates.

The complexity of the WF-CCRA algorithm is $O(\mathcal{RVK}|\mathcal{P}|^2)$. Although this approach is significantly more efficient than the B&B algorithm in terms of complexity (it can be executed within milliseconds), its complexity can be further reduced by restricting the number of valid paths between each pair of nodes to the $\mathcal{P}$ paths with the lowest costs or smallest number of links.

VI. SIMULATION RESULTS

In this section, the accuracy of the B&B-CCRA and WF-CCRA methods is numerically investigated. The simulation parameters are listed in Table I. As long as the problem remains feasible, the values for the remaining parameters can be chosen arbitrarily. Note that the results were obtained on a computer with 8 processing cores, 16 GB of memory, and a 64-bit operating system.

The results are illustrated in Fig 2. The proposed methods are evaluated based on the accuracy of the solutions they provide. Note that the accuracy of a solution for a scenario (η) is defined as $1 - ((\eta - \eta^\star)/\eta^\star)$, where $\eta^\star$ is the scenario's optimal solution, which is obtained by solving it with CPLEX 12.10. In Fig 2-A, the accuracy of B&B-CCRA is plotted vs. the solving time for five scenarios with different network sizes. In this simulation, the number of requests is set to 200. As illustrated, the accuracy of B&B-CCRA starts at 80% after the first iteration, which is obtained by solving the LP transformation of LiCCRA with CPLEX 12.10 in just a few milliseconds, and increases as the solving time passes, reaching 92% for all samples after 100 seconds. It proves that this method can be easily applied to provide baseline solutions for small and medium size use cases. However, the accuracy growth is slowed by increasing the network size, which is expected given the problem's NP-hardness and complexity.

In the two remaining sub-figures, the accuracy of WF-CCRA is depicted against the number of requests and network size. In addition, these sub-figures illustrate the outcomes of two more approaches, called DlyMin and Rnd. In the DlyMin method, allocations are performed to minimize delay regardless of other constraints, while Rnd is used to allocate resources randomly to requests. Note that the number of requests in Fig 2-B is 200, and the number of network nodes in Fig 2-C is 20. For each number of nodes or requests, 50 random systems are formed, and the problem is solved for them using the aforementioned techniques. It is evident that regardless of network size, WF-CCRA has an average accuracy of greater than 99%, implying that it can be used to allocate resources in a near-optimal manner even for large networks. For different numbers of requests, the average accuracy remains significantly high and greater than 96%. It does, however, slightly decrease as the number of requests increases, which is the cost of decomplexifying the problem by allocating the resources through separating requests. For the Rnd method, because it consumes the resources of all tiers uniformly, its accuracy is slightly above 50%. DlyMin is the least efficient method according to the results. The reason is that this method always utilizes the costly tier one nodes to minimize E2E delay. In conclusion, it is shown that the WF-CCRA algorithm is capable of efficiently allocating resources for large numbers of requests compared to other approaches.
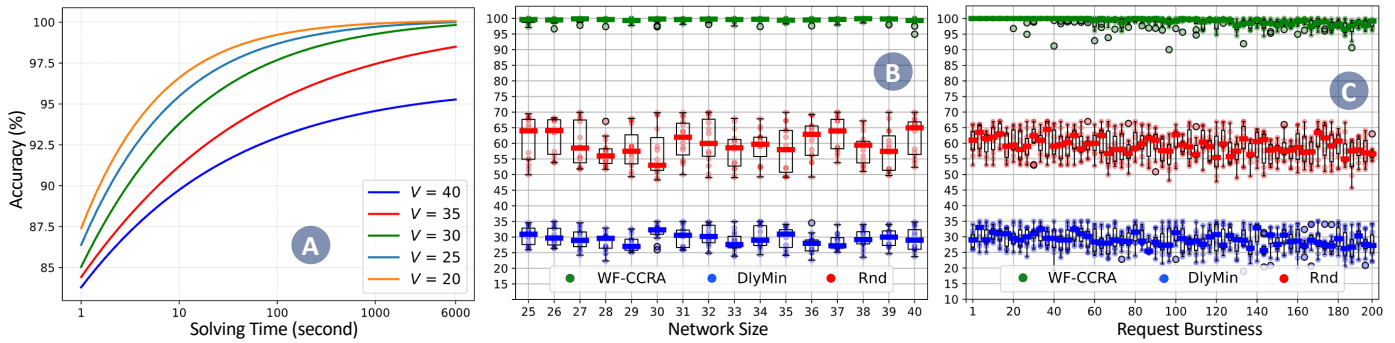
Fig. 2. B&B-CCRA accuracy vs. solving time (A), accuracy of WF-CCRA, DlyMin, and Rnd vs. network size (B) and request burstiness (C)

VII. CONCLUSION

In this paper, the joint problem of communication and computing resource allocation including VNF placement and assignment, traffic prioritization, and path selection considering capacity and delay constraints was studied. The primary goal was to minimize the overall cost of allocations. We first formulated the problem as a MINLP model, and used a method, named B&B-CCRA, to solve it optimally. Then, due to the complexity of B&B-CCRA, a WF-based approach was developed to find near-optimal solutions in a timely manner. Numerical results demonstrated that the proposed methods can efficiently offer accurate baselines and near-optimal solutions for large numbers of requests and nodes.

As a potential future work, we plan to solve the problem considering the ever-changing characteristics of end-users and infrastructure resources. We intend to devise an online machine learning approach for real-time adaptation of the allocation strategy in order to keep the cost of allocation minimized in such dynamic scenarios. Additionally, we are developing an access control strategy that will reduce overall cost over time by predicting future requests.

REFERENCES

[1] J. R. Bhat and S. A. Alqahtani, "6G Ecosystem: Current Status and Future Perspective," *IEEE Access*, vol. 9, pp. 43 134–43 167, 2021, conference Name: IEEE Access.

[2] L. Corneo, M. Eder, N. Mohan, A. Zavodovski, S. Bayhan, W. Wong, P. Gunningberg, J. Kangasharju, and J. Ott, "Surrounded by the Clouds: A Comprehensive Cloud Reachability Study," in *Proceedings of the Web Conference 2021*, ser. WWW '21. New York, NY, USA: Association for Computing Machinery, Apr. 2021, pp. 295–304.

[3] X. Yang, Z. Zho, and B. Huang, "URLLC Key Technologies and Standardization for 6G Power Internet of Things," *IEEE Communications Standards Magazine*, vol. 5, no. 2, pp. 52–59, Jun. 2021, conference Name: IEEE Communications Standards Magazine.

[4] Y. Li, J. Huang, Q. Sun, T. Sun, and S. Wang, "Cognitive Service Architecture for 6G Core Network," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 7193–7203, Oct. 2021, conference Name: IEEE Transactions on Industrial Informatics.

[5] M. Emu, P. Yan, and S. Choudhury, "Latency Aware VNF Deployment at Edge Devices for IoT Services: An Artificial Neural Network Based Approach," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6, iSSN: 2474-9133.

[6] X. Vasilakos, M. Bunyakitanon, R. Nejabati, and D. Simeonidou, "Towards Low-latent amp; Load-balanced VNF Placement with Hierarchical Reinforcement Learning," in *2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom)*, Sep. 2021, pp. 162–167.

[7] H. Sami, A. Mourad, H. Otrok, and J. Bentahar, "Demand-Driven Deep Reinforcement Learning for Scalable Fog and Service Placement," *IEEE Transactions on Services Computing*, pp. 1–1, 2021, conference Name: IEEE Transactions on Services Computing.

[8] T.-W. Kuo, B.-H. Liou, K. C.-J. Lin, and M.-J. Tsai, "Deploying Chains of Virtual Network Functions: On the Relation Between Link and Server Usage," *IEEE/ACM Transactions on Networking*, vol. 26, no. 4, pp. 1562–1576, Aug. 2018, conference Name: IEEE/ACM Transactions on Networking.

[9] B. E. Mada, M. Bagaa, T. Tale, and H. Flinck, "Latency-aware Service Placement and Live Migrations in 5G and Beyond Mobile Systems," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Jun. 2020, pp. 1–6, iSSN: 1938-1883.

[10] Q. Zhang, F. Liu, and C. Zeng, "Adaptive Interference-Aware VNF Placement for Service-Customized 5G Network Slices," in *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, Apr. 2019, pp. 2449–2457, iSSN: 2641-9874.

[11] Q. Yuan, X. Ji, H. Tang, and W. You, "Toward Latency-Optimal Placement and Autoscaling of Monitoring Functions in MEC," *IEEE Access*, vol. 8, pp. 41 649–41 658, 2020, conference Name: IEEE Access.

[12] T. Gao, X. Li, Y. Wu, W. Zou, S. Huang, M. Tornatore, and B. Mukherjee, "Cost-Efficient VNF Placement and Scheduling in Public Cloud Networks," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4946–4959, Aug. 2020, conference Name: IEEE Transactions on Communications.

[13] C. D. Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6G Frontiers: Trends, Applications, Requirements, Technologies and Future Research," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 836–886, 2021, conference Name: IEEE Open Journal of the Communications Society.

[14] J. Specht and S. Samii, "Urgency-Based Scheduler for Time-Sensitive Switched Ethernet Networks," in *2016 28th Euromicro Conference on Real-Time Systems (ECRTS)*, Jul. 2016, pp. 75–85, iSSN: 2159-3833.

[15] H. Kellerer, U. Pferschy, and D. Pisinger, "Multidimensional Knapsack Problems," in *Knapsack Problems*, H. Kellerer, U. Pferschy, and D. Pisinger, Eds. Berlin, Heidelberg: Springer, 2004, pp. 235–283.

[16] F. Faticanti, F. De Pellegrini, D. Siracusa, D. Santoro, and S. Cretti, "Cutting Throughput on the Edge:App-Aware Placement in Fog Computing," *arXiv:1810.04442 [cs]*, Oct. 2018, arXiv: 1810.04442.

[17] G. Pataki, M. Tural, and E. B. Wong, "Basis Reduction and the Complexity of Branch-and-Bound," in *Proceedings of the 2010 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, ser. Proceedings. Society for Industrial and Applied Mathematics, Jan. 2010, pp. 1254–1261.