

Joint Route Selection and Split Level Management for 5G C-RAN

Cristian C. Erazo-Agreto*, Mario Garza-Fabre*, Ramón Agüero†, Luis Diez†, Joan Serrat§, Javier Rubio-Loyola*

* Center for Research and Advanced Studies

Cinvestav Campus Tamaulipas
87130, Cd. Victoria, Tamaulipas, Mexico

{cristian.erazo, mario.garza, javier.rubio}
@cinvestav.mx

† Dept. of Communications Engineering

University of Cantabria, Santander 39005, Spain

{ramon, ldiez}@tlmat.unican.es

§ Dept. of Network Engineering

Universitat Politècnica de Catalunya, Barcelona, Spain

juan.serrat@upc.edu

Abstract—This work tackles the problem faced by network/infrastructure providers of jointly selecting routing and functional split level to satisfy requests from virtual mobile network operators (vMNOs). We build a novel system model that brings together all the involved elements and features, embracing split levels defined by the 3GPP and packet switch fronthaul network. To our best knowledge, this is the first work that provides a solution for multiple vMNO requests considering the two aforementioned sub-problems (i.e. split selection and routing). We use the model defined to formulate an optimization problem, which is characterized by the exponential size of its search space. We propose two heuristic approaches to address this problem: (1) a greedy scheme, and (2) an evolutionary algorithm, which is also improved with a specialized initialization. We conduct extensive experiments to assess the performance and behavior of the proposed methods, over varying network instances. When possible, we also perform comparisons with respect to the optimal solution and a well-known commercial solver. Our results indicate that the proposed techniques represent appropriate trade-offs between solution quality and execution time, and can serve complementary goals: the quality of the results yielded by our evolutionary method are better, but at the cost of longer execution times; in contrast, our greedy algorithm offers a reasonably appropriate performance, with an execution time that is notably lower. Our experiments show that it is possible to produce near-optimal results to the above complex problem through computationally efficient algorithmic solutions.

Index Terms—Route selection, functional split, 5G, C-RAN.

I. INTRODUCTION

The stringent requirements posed by 5G services lead to profound transformations in all network segments. On top of it, 5G networks aim to provision heterogeneous services with varying features and requirements. Along with the increasing capacity demanded by enhanced Mobile Broadband services, massive Machine Type Communications (mMTC) and Ultra Reliable Low Latency Communications (URLLC) services will require a tailored network configuration. Moreover, future 5G networks and beyond will need to dynamically adapt their capabilities to efficiently offer the aforementioned services.

One of the main advancements, from the architectural perspective, comes from the capacity to virtualize and re-allocate

network functions, leveraging software defined network (SDN) and network function virtualization (NFV) techniques. Under this new paradigm, the functionalities of conventional base stations (BSs) are split, so that part of the functions traditionally performed by Baseband Units (BBUs) are virtualized.

Initially, fully centralized solutions were proposed, leading to the so-called cloud-RAN (C-RAN) [1], [2]. In these architectures, the whole BBU is virtualized while a remote radio head (RRH) performs basic RF functions. Although this approach would allow a tight coordination of the access points, it also poses highly demanding requirements on the fronthaul links connecting the RRH and BBU [3]. As a result, C-RAN networks require a vast deployment of fiber links, leading to high deployment costs, which might not be affordable in some scenarios. In order to overcome these limitations, the use of different functional splits was proposed [4], [5]. The BBU is divided in a distributed unit (DU) placed close to the RRH, now renamed as radio unit (RU), and a central unit (CU) that is located close to the edge of the aggregation network. In recent years, flexible functional split solutions have been proposed [6], so that functions in the DU and CU can be dynamically shifted [7], [8] to convey the service requirements.

Fig. 1 depicts the splits defined by the 3GPP [9] over the protocol stack, so that each split defines the functionalities that are moved to the CUs (on the left) and those that remain at the DUs (on the right). As can be observed, most of the splits are defined in the boundaries of each protocol or layer, while three of them are specified within the protocol itself. In addition, for some of the splits depicted in Fig. 1, different alternatives are defined. For instance, the Intra-MAC split is defined within the MAC layer, dividing it in low and high sub-layers [9]. The high-MAC sub-layer, placed in the CU, would be responsible for managing multiple low-MAC sub-layers, leveraging the use of coordination techniques, such as centralized schedulers or Coordinated Multipoint (CoMP) transmission and reception solutions. On the other hand, the low-MAC sub-layer would implement stringent time-constrained functionalities, such as hybrid automatic response request (HARQ). The reader may refer to [9] for a detailed discussion of the functionalities that

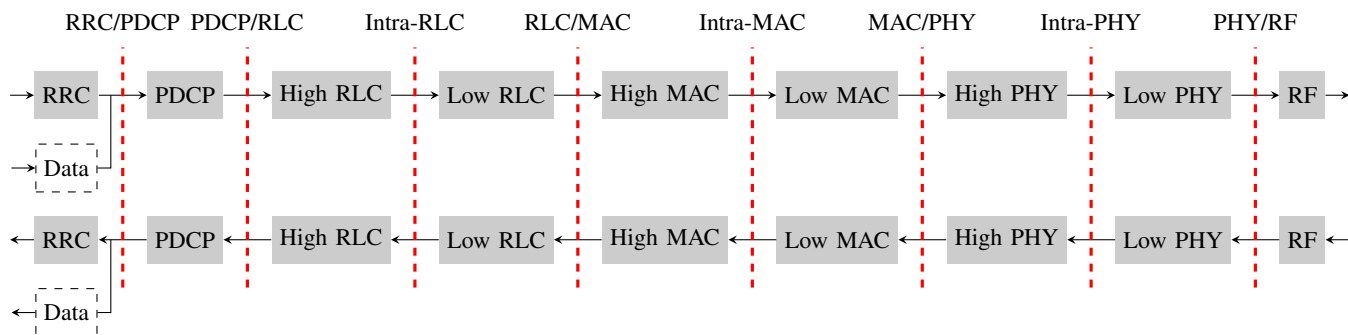


Fig. 1: Possible functional splits between the CU and DU according to 3GPP [9].

are considered by the various splits. It is worth noting that, at the time of writing, the applicability of the different splits is being analyzed by standardization and industrial initiatives like 3GPP or O-RAN Alliance. In particular, special attention is being paid to lower layer split possibilities [10], and open implementation have started by assuming PDCP/RLC split between DU and RU¹.

The selection of the different splits will be conditioned to both, the required quality of service (QoS) and the communication capabilities of the underlying fronthaul network that communicates DUs and CUs. Table I summarizes the main requirements of the splits depicted in Fig. 1 (i.e. bandwidth and latency), as well as some benefits and cons. As can be seen, the communication requirements show a large variation, and the split selection process thus needs to be tightly coordinated with other management techniques of the fronthaul network. As we discuss later, the proposed solution is based on the split requirements shown in Table I. In this sense, we will analyze in our future work studying how some particular splits might be preferred, based on the functionalities they execute at the CU, and so the cooperation techniques they enable.

This architectural evolution has increased the complexity of the fronthaul, that has shifted from a set of dedicated links to a packet-based network that needs to be appropriately managed. As a consequence, academia, industry and standardization bodies are working together on the definition of the so-called next generation fronthaul interface (NGFI) [11], [9]. According to [11], the conventional fronthaul segment in NGFI is further divided in the NGFI-I, which connects DUs and RUs, and the NGFI-II, also known as midhaul, which interconnects the DU and CU. As an example, Fig. 2 depicts an illustrative network, embracing three virtualized base stations, whose components are attached through an NGFI network. In the figure, the colored rectangles at the DUs and CUs represent, respectively, the functions to the right and to the left of the split selection according to Fig. 1. In addition, NGFI-I is depicted with dotted lines, which connect DUs and RUs, while NGFI-II is shown with colored dashed lines, communicating DUs and CUs through a network having various intermediate nodes.

As can be seen, NGFI-I connects RUs and DUs, which host the corresponding base station functions. Then, a NGFI-

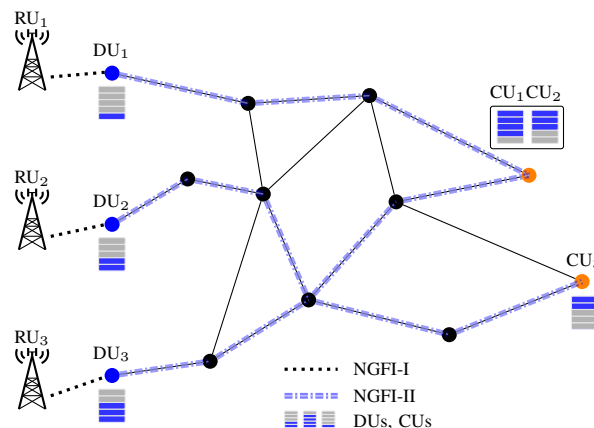


Fig. 2: Example of virtualized RAN with NGFI.

II packet based network connects DUs and CUs, which implement the remaining functions, according to the selected functional split. Considering the non-deterministic nature of a packet-based network, the intermediate nodes in the NGFI may need to employ buffering techniques. This might increase the delay, thus hindering the QoS, which in turn depends on the particular split. In addition, the different links of the NGFI-II may use different technologies, with varying capabilities.

In order to ensure that the requirements shown in Table I are fulfilled, it becomes necessary to develop techniques that can appropriately manage the traffic flow, jointly combining routing and split selection. Furthermore, the resulting optimization problems usually have a combinatorial nature [12], while the NGFI imposes stringent latency requirements [11]. Altogether, the proposed solutions need to find a trade-off between complexity and performance. It is worth noting that the concept of RAN functional split is not exclusive of 5G networks (in Fig. 1 the LTE protocol stack is actually used), but it is becoming more relevant (in 5G) along with the required evolution of the fronthaul and midhaul segments.

In this paper, we consider the critical issue of providing a solution to the problem of jointly determining appropriate functional splits and the routes of the midhaul network to support the quality of service requirements of virtual Mobile Network Operators (vMNOs) optimizing the physical infrastructure. This problem encompasses the following two sub-problems: (1) the problem of determining the location of each

¹Architecture of the O-RAN alliance software components: <https://docs.o-ran-sc.org/en/latest/architecture/architecture.html>

TABLE I: Split requirements, benefits and cons [9]. The particular requirements are defined with 100 MHz channel bandwidth, 256 QAM modulation and 8 MIMO layers for both downlink and uplink [9, Tables A-1, A-2], and benefits and cons are summarized from the document body. Information from different split alternatives is merged.

| Split | One-way latency | DL/UL bandwidth | Benefits | Cons |
|---------------------|-----------------|------------------------------|--|---|
| RRC/PDCP (f_1) | 10 ms | $\sim 4/3$ Gbps | U-plane separation. Potential benefits for edge computing. | Unclear performance with security. Potential issues with aggregation. |
| PDCP/RLC (f_2) | 1.5 – 10 ms | 4/3 Gbps | U-plane separation and traffic aggregation. | Security configuration issues. |
| Intra-RLC (f_3) | 1.5 – 10 ms | lower than PDCP/RLC | Traffic aggregation and better flow control. Potential handling of more connected mode UEs | Latency requirements and duplication of buffers. |
| RLC/MAC (f_4) | ~ 100 us | $\sim 4/3$ Gbps | No benefit foreseen with LTE protocol stack. | – |
| Intra-MAC (f_5) | hundreds of us | $\sim 4/3$ Gbps | Traffic aggregation and better interference management. | Additional scheduling complexity. |
| MAC/PHY (f_6) | 250 us | $\sim 4/5$ Gbps | Traffic aggregation, COMP joint transmission, centralized scheduling. | Stringent timing between CU and DU. |
| Intra-PHY (f_7) | 250 us | 10.1 – 22.2/16.6 – 86.1 Gbps | Allows implementation of advanced receivers | – |
| PHY/RF (f_8) | 250 us | 157.3/157.3 Gbps | More efficient resource management. Improvement of RF/PHY scalability. | High requirement in fronthaul for latency and bandwidth |

DU and CU composing the BS, as well as the optimal split distribution between them, according to such location; and (2) the problem of selecting the best routes for the traffic coming from each of the DUs. The first sub-problem consists of assigning, to each DU, the functional split that optimizes both, the level of centralization and the use of network resources, and associating each DU with an appropriate CU that will implement the remaining virtualized functions. The second sub-problem aims at establishing the best route to connect each DU with its respective CU, which will process and forward the traffic to the core network. The routes should meet the delay and bandwidth requirements of the functional splits, and the network connections of the DU-to-CU assignments. It is worth mentioning that the solution proposed in this paper intends to address both sub-problems jointly.

Even when considered individually, each of the above sub-problems poses a significant computational challenge, since both have been proved to be NP-hard [12]. On the one hand, the optimal allocation of a functional split depends on the capacity and latency of the paths connecting the DU to the CU. These values depend on the routes of the whole network. On the other hand, identifying the best routes for the traffic for each DU requires information about the functional splits, because the latter determine the bandwidth and latency requirements. Addressing these two problems hierarchically, that is, solving one of the problems first and using the solution obtained as a starting point to solve the second problem, might yield sub-optimal or even unfeasible solutions [12].

This paper addresses the problem described above, introducing a solution for joint route and functional split selection for 5G C-RAN architectures. It is worth noting that the solution proposed in this work is not intended to be applied in real-time. Contrarily, it would be part of the mechanisms for network configuration and orchestration that might be executed upon requests from vMNOs. We can summarize the main contributions of this paper as follows:

- A novel model that brings together all the elements and features to address the whole problem and to handle its complexity. The problem is modeled as a virtual network embedding problem, supporting multiple requests from vMNOs, heterogeneity in the use of resources, diversity of DU-to-CU assignments, constraint-aware routes selection, and functional split level management.
- An optimization problem formulation that is meant to evaluate the quality of potential solutions based on the level of centralization given by the functional splits, as well as the number and types of network links.
- Two algorithmic approaches to address the aforementioned problem. Due to the complexity of the problem, exact algorithms can find optimal solutions, but their execution time drastically increases with practical instances, i.e., instances consisting of large substrate and virtual networks. Therefore, exact algorithms can be efficiently used only for small problem instances. We introduce two heuristic algorithms: a greedy approach tailored to optimize the execution time, and an evolutionary algorithm customized to provide more promising solutions.
- With the aim of facilitating future comparative analyses, this paper also contributes with a full collection of test problem instances, as well as the source codes of our proposed methods, which are available to the research community through our GitHub repository².

The remainder of this paper is organized as follows. Section II discusses the related work in the area, identifying the contributions of the research presented herewith. The system model, problem formulation and search space analysis are described in Section III, while algorithmic solutions are introduced in Section IV. Section V describes the evaluation setup, which is afterwards used in Section VI, where we analyze the performance of the algorithms proposed by means of an

²https://github.com/cristian-erazo/Joint_Route_Selection_and_Split_Level_Management_for_5G_C-RAN

extensive experiment campaign. Finally, Section VII highlights the most relevant outcomes of our work and provides some possible directions for future research.

II. RELATED WORK

Recently, several studies have focused on optimizing the fronthaul performance assuming functional splits. As a general approach, the authors of [13] analyze the convergence time of split decision algorithms according to data rate performance.

As for particular solutions, there exist a variety of application scenarios and use-cases. Energy-aware solutions are proposed in [14], [15], and an energy-constrained functional split solution for scenarios with unmanned autonomous vehicles is proposed in [16]. Content-aware solutions have been also studied. In particular, content placement along with split selection and energy minimization, are considered in [17], jointly analyzing flexible functional split and mobile edge computing (MEC). Other works focus on the interplay of functional split selection and fronthaul communication technology. In this regard, functional split management over optical networks is analyzed in [18], [19], [20]. On the other hand, joint optimization of split selection and network slicing is addressed in [21], while machine learning is used in [22] to develop a delay-constrained split selection solution. Also, some recent works have focused on the implementation aspects and implications of dynamic functional split. Worthy to mention is the work presented in [23], where baseband signal precoding is analyzed to enable functional split. Similarly, a flexible functional split implementation is proposed and evaluated in [24].

Although the aforementioned works share aspects with our proposal, none of them jointly addresses split selection and routing over the fronthaul network. However, some works have partially analyzed routing and path selection in fronthaul networks. The authors of [25] study the impact of the optical routing on fully centralized C-RAN architectures, paying special attention to load balancing. Other works consider the impact of functional split in the routing algorithms, so that delay requirements are satisfied for pre-established splits. In [26], for instance, the authors consider the minimization of the worst case, while a machine learning solution is proposed in [27] to minimize the delay.

Taking into account the increasing convergence of fronthaul and backhaul networks, it is worth mentioning recent backhaul routing solutions for 5G scenarios. Works [28] and [29] apply backpressure routing to communicate cellular base stations through a wireless backhaul network. Also, routing over millimeter wave (mmWave) links has been analyzed, jointly with link scheduling [30] and with beam alignment [31]. However, these proposals do not consider functional split.

Only very few works have jointly analyzed the problem of functional split selection and routing. For instance, the authors of [32] present a framework that integrates heuristic solutions for energy-efficient routing in converged fronthaul/backhaul networks. The framework also allows the re-allocation of virtual network functions. As an example, a simple application use-case of the framework is described in [33]. Similarly, the authors of [34] propose an architecture for joint resource management and energy efficiency in backhaul/fronthaul networks.

In particular, the resource management approach considers path computation and placement of virtual networks functions, including functional split. The general optimization problem is simplified, since the aim of this work is on the architecture, rather than in an algorithmic solution itself.

The authors of [35] propose a method to find the optimal location of CUs, jointly addressing computation resource management and routing. It is worth noting that a chain of CUs is considered, rather than a central one. In this sense, the split configuration is not constrained by traffic requirements (e.g. delay), but it is a consequence of route selection and network costs. Work by Koutsopoulos [36] proposes a set of solutions for joint split selection and traffic scheduling, considering different network characteristics. However, the envisioned model assumes that CUs and DUs are connected by a single link, where in more realistic scenarios these units are connected by networks.

Much closer to our proposal are the approaches described in [12] and [37]. The authors of [12] tackle the problem of jointly deciding the split shift and route selection to maximize centralization. To solve such problem the authors propose two algorithms: one based on backtracking branch and bound, and a second one with a greedy behavior. One step beyond, [37] proposes a mathematical framework to jointly select functional split, fronthaul routing and placement of MEC functions. The authors thoroughly describe the resulting optimization problem and propose a solution based on Benders decomposition. Unlike our proposal, these works assume that the placement of CUs and DUs is known beforehand, notably simplifying the underlying problem, since the potential number of routes is much lower. On the other hand, different to our work, it may not find the overall optimal fronthaul behavior, and it does not minimize the number of active processing entities where virtualized components (DUs and CUs) are allocated.

Finally, some works have considered the functional split selection as part of some type of network embedding. In particular, in [38] the authors propose slice embedding strategies for 5G considering functional split. Similarly, in [7] virtual network embedding was already addressed along with split selection. Nevertheless, none of these works consider the routing through the packet-based fronthaul.

III. SYSTEM MODEL AND PROBLEM FORMULATION

This section presents the model proposed to address the problem of jointly selecting the routes of the midhaul network to connect DUs and CUs of the NGFI-II, and determining the appropriate functional splits to support the quality of service requirements of virtual Mobile Network Operators (vMNOs). It is assumed that the vMNOs do not own physical resources, and their requests thus need to be embedded in a common substrate network. The problem is modeled as a virtual network embedding problem (VNE) for which we first model the physical network, the virtual network requests, the mapping of virtual and physical resources, and the candidate solutions. Further, the optimization problem is formulated based on our VNE model. Finally, the size of the search space for this problem is analyzed at the end of this section. It

TABLE II: Main components of the proposed system model.

| Component | Description |
|----------------------------------|--|
| G | Physical (substrate) network. |
| V | Set of physical network nodes. |
| E | Set of connections between physical network nodes. |
| C | Set of possible physical CUs, $ C = n$. |
| R | Set of possible physical DUs, $ R = m$. |
| T | Set of intermediate (transmission) physical nodes, $ T = l$. |
| $P_{a,b}^j$ | The j -th physical path available between nodes a and b . |
| $P_{a,b}^*$ | Set of all physical paths available between nodes a and b . |
| F | Set of all functional splits. |
| Q | Set of all virtual resource requests. |
| G_i^v | The i -th virtual resource request. |
| V_i^v | Set of requested virtual nodes for the i -th request. |
| C_i^v | Set of virtual CUs requested for the i -th request, $ C_i^v = n'_i$. |
| R_i^v | Set of virtual DUs requested for the i -th request, $ R_i^v = m'_i$. |
| E_i^v | Set of requested virtual links for the i -th request. |
| $\Omega_{r_j^v}^i$ | Set of suitable candidate physical DUs for virtual DU r_j^v . |
| $\Delta_{c_j^v}^i$ | Set of suitable candidate physical CUs for virtual CU c_j^v . |
| $\Lambda_{e_{r_j^v, c_k^v}^v}^i$ | Set of all candidate physical paths for the virtual link $e_{r_j^v, c_k^v}^v$. |
| $s_{j,k}^i$ | Assignment of physical nodes, physical path, and functional split for the i -th request, j -th virtual DU, and k -th virtual CU. |
| X | Candidate solution (assignment configuration for all the requests). |
| R_{L1}^v | Set of assignments of physical layer functional split. |
| R_{L2}^v | Set of assignments of data link layer functional split. |
| R_{L3}^v | Set of assignments of network layer functional split. |
| $BBU_{pool_\rho}^v$ | Set of CUs that represent the ρ -th pool of resources. |

is worth mentioning that the fronthaul elements (e.g. model, physical and electrical features of the NGFI-I network) are out of the scope of this paper and they are not considered in the problem formulation.

A. System Model

The main components of our proposed system model are summarized in Table II. A comprehensive description of this model is presented below, which is organized based on the correspondence of model components to the physical network (Section III-A1), to the virtual resources requested (Section III-A2), to the mapping between physical and virtual resources (Section III-A3), and to the definition of a full candidate solution to the problem (Section III-A4).

1) *Physical Network*: Our model considers a scenario in which a physical (substrate) network is used to support a set of requests for resources from virtual Mobile Network Operators (vMNOs). An example is graphically illustrated in Fig. 3. The substrate network is represented as a non-directed graph $G = (V, E)$, where the network nodes correspond to the graph vertices, V , and the connections between the network nodes correspond to the graph edges, E . The set of physical nodes is defined as $V = C \cup R \cup T$, where C corresponds to the set of n possible CUs, R corresponds to the set of m DUs, and T corresponds to the set of l intermediate (transmission) nodes. The CUs are nodes with processing capabilities, in which the functionalities of the different sub-layers of the radio protocol stack can be virtualized. Each CU c_i is characterized by its coordinates (x_i, y_i) , and its maximum amount of computational resources w_{prc_i} . Similarly, DUs are nodes where the radio frequency functions can be virtualized. Each DU r_i is characterized by its location (x_i, y_i) , its coverage radius θ_i , its maximum capacity of computational

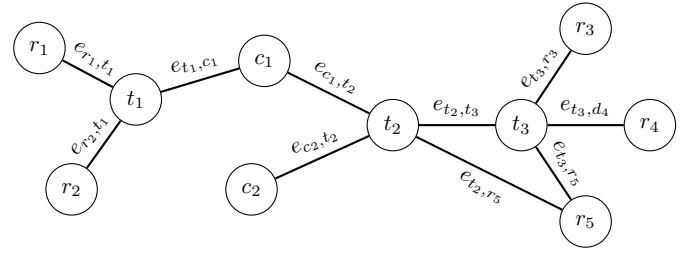


Fig. 3: Example of a physical (substrate) network. The example comprises possible CUs (c), DUs (r) and intermediate nodes (t), all connected by heterogeneous links (e).

resources w_{prc_i} , its number of antennas w_{ant_i} , and its number of physical resource blocks w_{prb_i} . We assume that each DU r_i is associated with one or multiple RUs³. Each intermediate network node t_i is modeled by its position (x_i, y_i) .

A link $e_{v_i, v_j} \in E$ between two physical nodes v_i and v_j is modeled by its maximum capacity $b_{i,j}$, its induced delay $d_{i,j}$, and its cost $\gamma_{i,j}$. Parameter $\gamma_{i,j}$ is modeled as a weight which captures the relative cost of using fiber optic, copper, or wireless links. It is worth mentioning that the concept of cost is generic (i.e. not necessarily monetary cost) and can be used to model any preference or policy to be implemented by the physical substrate owner. A path between two nodes a and b is defined as a set of k links connecting a succession of $k+1$ nodes, such that the first node corresponds to a and the last one corresponds to b ; formally, $P_{a,b}^j = \{e_{v_i, v_{i+1}} \mid 1 \leq i \leq k, v_1 = a, v_{k+1} = b\}$. The set of all possible paths between nodes a and b is given by $P_{a,b}^* = \{P_{a,b}^1, \dots, P_{a,b}^p\}$.

2) *Virtual Resource Requests*: The set of all virtual resource requests⁴ is denoted by a set of q non-directed graphs, $Q = \{G_1^v, \dots, G_q^v\}$, where $G_i^v = (V_i^v, E_i^v)$ represents the i -th request. Virtual nodes V_i^v and virtual links E_i^v are modeled to support the QoS requirements of the vMNO. Here, $V_i^v = C_i^v \cup R_i^v$, where C_i^v and R_i^v denote the sets of n'_i virtual CUs and m'_i virtual DUs requested, respectively. Each virtual CU c_j^v specifies the required processing capacity $w_{prc_j}^v$ to be set at some physical CU. Analogously, each virtual DU r_j^v defines a desired location (x_j, y_j) , the required processing capacity $w_{prc_j}^v$, the number of antennas $w_{ant_j}^v$ (each virtual DU r_j^v might be associated with one or multiple RUs), and the number of required physical resource blocks $w_{prb_j}^v$, to be set at some physical DU.

A virtual link $e_{r_j^v, c_k^v} \in E_i^v$, connecting virtual DU $r_j^v \in R_i^v$ and virtual CU $c_k^v \in C_i^v$ of the i -th request, is defined by the tuple $e_{r_j^v, c_k^v} = (r_j^v, c_k^v, b_{j,k}^v, d_{j,k}^v)$, where terms $b_{j,k}^v$ and $d_{j,k}^v$ correspond to the minimum bandwidth required and the maximum acceptable delay, respectively, $b_{j,k}^v, d_{j,k}^v \in \mathbb{N}$.

Finally, it is assumed that vMNOs cannot exploit other resources but those from the common substrate network, so that solutions like Licensed-Assisted-Access (LAA) or unlicensed

³It is important to note, however, that in practice the same physical node may assume either the role of a CU, that of a DU, or the two roles simultaneously. Such a scenario is supported by our system model, as the same node can be specified to be a member of either C , R , or both sets.

⁴In this paper, the term virtual resource request is used to refer to the requirements of resources (namely, DUs, CUs, and links) from a vMNO.

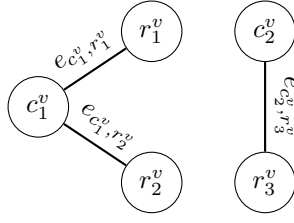


Fig. 4: Example of a resource request from a vMNO. r_j^v are virtual DUs, c_j^v are virtual CUs, and $e_{r_j^v, c_j^v}$ are virtual links.

spectrum (i.e. LTE-U) are out of the scope of this work. An example of a virtual resource request is shown in Fig. 4.

3) *Mapping of Virtual and Physical Resources:* For any given virtual request G_i^v , we define the set of all candidate physical DUs that can support r_j^v , its j -th virtual DU, as $\Omega_{r_j^v}^i = \{r_k \mid w_{prc_j}^v \leq w_{prc_k}, w_{ant_j}^v \leq w_{ant_k}, w_{prb_j}^v \leq w_{prb_k}, distance(r_j^v, r_k) \leq \theta_k\}$; that is, suitable physical DUs are those nodes r_k whose coverage area θ_k and resources are compatible with the desired location and with the resources required by the virtual DU r_j^v . Similarly, the set of candidate physical CUs for the virtual CU c_k^v consists of all nodes c_l with sufficient computational resources to serve c_k^v ; formally, $\Delta_{c_k^v}^i = \{c_l \mid w_{prc_k}^v \leq w_{prc_l}\}$. Regarding suitable assignments for a virtual link $e_{r_j^v, c_k^v}$, candidate physical paths comprise all possible paths that connect suitable physical nodes for r_j^v and c_k^v , as given by $\Lambda_{e_{r_j^v, c_k^v}}^i = \{P_{r_x, c_y}^l \in P_{r_x, c_y}^* \mid r_x \in \Omega_{r_j^v}^i, c_y \in \Delta_{c_k^v}^i\}$. To illustrate the above, Fig. 5 shows an illustrative example of the mapping of a virtual resource request to suitable physical resources of the substrate network.

A central topic of this work is the appropriate selection of functional splits (also referred to as split levels), for which we can define the set of possible alternatives as $F = \{f_i \mid f_i = (\beta_i, \delta_i), \beta_i, \delta_i \in \mathbb{N}, 1 \leq i \leq 8\}$. In this definition, β_i and δ_i denote the bandwidth and delay requirements of functional split f_i , respectively.

Having defined all the above concepts, we can now formalize the assignment of the physical nodes, physical path, and split level for the i -th virtual request G_i^v , j -th virtual DU r_j^v , and k -th virtual CU c_k^v , as the tuple $s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l)$, where $r_x \in \Omega_{r_j^v}^i$, $c_y \in \Delta_{c_k^v}^i$, $f_z \in F$, and $P_{r_x, c_y}^l \in \Lambda_{e_{r_j^v, c_k^v}}^i$. It is worth noting the correspondence between the previous definition of an assignment and that of a virtual link; for each virtual link $e_{r_j^v, c_k^v} \in E_i^v$, the goal is to find an assignment $s_{j,k}^i$ so that suitable physical nodes, physical path, and functional split can be configured, according to the resource requirements. Therefore, the set containing all of the assignments for the i -th request can be defined as $S_i = \{s_{j,k}^i \mid e_{r_j^v, c_k^v} \in E_i^v\}$.

4) *Candidate Solution:* A full candidate solution, denoted by X , is given by a particular set of assignment configurations for all the q virtual requests; formally, $X = \{S_i \mid 1 \leq i \leq q\}$. Once a candidate solution has been formally defined, we need to introduce additional elements that account for the usage of network resources.

Recalling our definition of the assignment of resources for a given virtual DU-CU pair of the i -th request as a tuple

$s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l)$, we can now divide and classify the set of all assignments in a solution X based on the specific functional splits f_z configured. Three disjoint subsets are defined: (i) assignments of physical layer split level, $R_{L_1}^v = \{s_{j,k}^i \in S_i \mid S_i \in X, f_z \in \{f_6, \dots, f_8\}\}$; (ii) assignments of data link layer split level, $R_{L_2}^v = \{s_{j,k}^i \in S_i \mid S_i \in X, f_z \in \{f_2, \dots, f_5\}\}$; and (iii) assignments of network layer split level, $R_{L_3}^v = \{s_{j,k}^i \in S_i \mid S_i \in X, f_z \in \{f_1\}\}$.

For convenience, we can also define subsets of assignments depending on whether they use a particular physical resource. The set of assignments that use a specific physical link e_{v_a, v_b} is defined as $\mathcal{E}_{v_a, v_b} = \{s_{j,k}^i \in S_i \mid S_i \in X, e_{v_a, v_b} \in P_{r_x, c_y}^l\}$. Also, the sets of assignments that involve physical CU c_y and physical DU r_x are respectively defined as $\mathcal{C}_{c_y} = \{s_{j,k}^i \in S_i \mid S_i \in X, s_{j,k}^i = (\dots, c_y, \dots)\}$ and $\mathcal{R}_{r_x} = \{s_{j,k}^i \in S_i \mid S_i \in X, s_{j,k}^i = (r_x, \dots)\}$. Moreover, we can define several sets representing all the assigned physical resources as follows. The set of all physical links that have been assigned is given by $E^A = \bigcup_{i=1}^q E_i^A$, where E_i^A denotes the set of links that are in use as part of the physical paths assigned to the i -th request. Similarly, the set that contains all the physical DUs that have been assigned is defined as $R^A = \bigcup_{i=1}^q R_i^A$, where R_i^A is the subset of physical DUs assigned to the i -th request. Likewise, the set that contains all the physical CUs that have been assigned is defined as $C^A = \bigcup_{i=1}^q C_i^A$, where C_i^A is the subset of physical CUs assigned to the i -th request.

Finally, a pool of resources, denoted by $BBU_{pool, \rho}$, represents a group of assigned CUs, $c_j \in C^A$, such that for every pair of nodes in this group there exists a path connecting them, either directly or through other intermediate assigned CUs. An example illustrating two pools of resources is shown in Fig. 6. The collection of all the resource pools defined by a given candidate solution is referred to as $BBU_{pools} = \{BBU_{pool_1}, \dots, BBU_{pool_\lambda}\}$.

B. Optimization Problem Formulation

With the elements considered and described above, this section formalizes the definition of the problem studied in this paper as a *constrained discrete optimization problem*.

Our goal is to obtain a resource assignment, X , for all the virtual requests from the vMNOs (as defined in Section III-A4), which maximizes the degree of centralization and at the same time optimizes the usage of the network resources. We therefore define this optimization problem formally as the task of determining the best possible assignment X^* such that:

$$X^* = \arg \max_{X \in \mathcal{F}} f(X), \quad (1)$$

where $\mathcal{F} \subset \mathcal{X}$ denotes the set of all possible feasible assignments in the search space \mathcal{X} , as defined by a set of constraints (see below), and $f: \mathcal{F} \rightarrow \mathbb{R}$ is the following objective function, to be maximized:

$$f(X) = [DoC(X) \cdot (1 - UoL(X))], \quad (2)$$

where

$$DoC(X) = \frac{\omega_1 \cdot |R_{L_1}^v| + \omega_2 \cdot |R_{L_2}^v| + \omega_3 \cdot |R_{L_3}^v|}{|BBU_{pools}| \cdot \sum_{i=1}^q m_i'}, \quad (3)$$

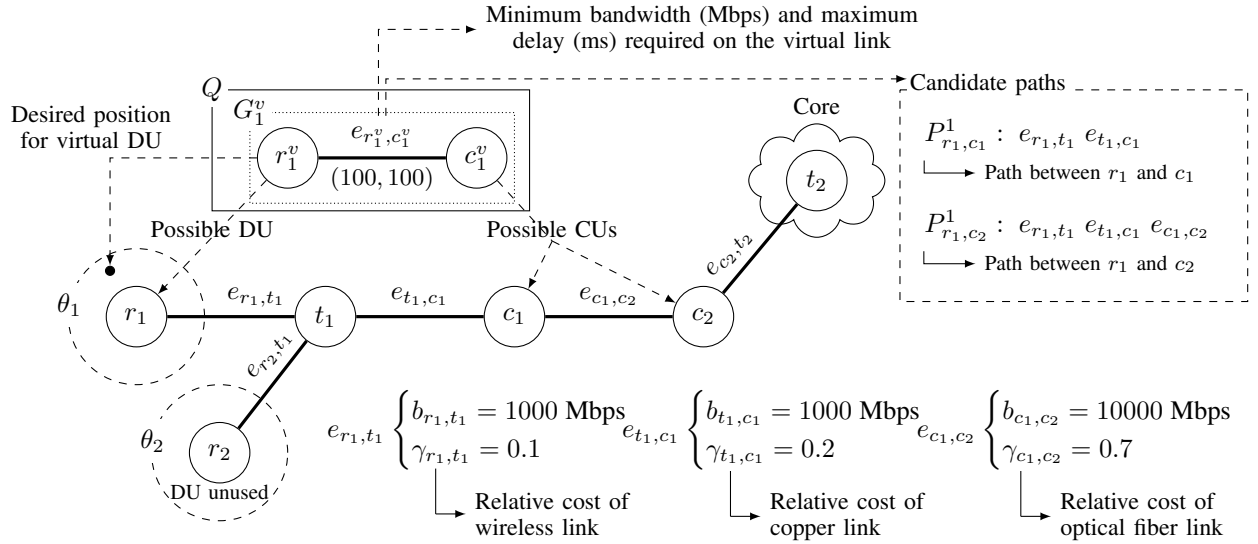


Fig. 5: Example of a potential resource assignment for a virtual request G_1^v , which involves a single CU, c_1^v , and a single DU, r_1^v . Virtual DU r_1^v can be assigned to physical node r_1 , as the desired position of r_1^v is within the coverage area of r_1 . Virtual CU c_1^v can be assigned either to c_1 or to c_2 . Thus, the virtual link $e_{r_1^v, c_1^v}^v$ can be assigned either to path P_{r_1, c_1}^1 or to path P_{r_1, c_2}^1 .

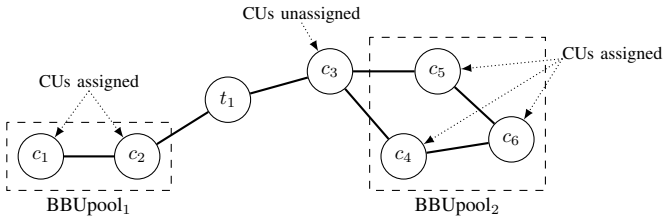


Fig. 6: The pools of resources BBU_{pool_1} and BBU_{pool_2} represent groups of assigned physical CUs connected to each other either directly or through other assigned physical CUs.

$$UoL(X) = \sum_{e_{v_a, v_b} \in E^A} \frac{\gamma_{a,b} \cdot \beta(v_a, v_b)}{|E| \cdot b_{a,b}}, \quad (4)$$

and

$$\beta(v_a, v_b) = \sum_{s_{j,k}^i \in \mathcal{E}_{v_a, v_b}} \beta_z. \quad (5)$$

The objective function in (2) involves two factors. The first one, given in (3), evaluates the degree of centralization (DoC) of the network, which depends on the different functional splits of the assignment solution X . The DoC is defined as a weighted sum of the number of virtual DUs assigned to each layer of the radio protocol stack (as given by the cardinality of sets $R_{L_1}^v$, $R_{L_2}^v$, and $R_{L_3}^v$). The weights ω_1 , ω_2 , and ω_3 are used to represent the degree with which each split selection contributes to the centrality level of the RAN network. Since the goal is to optimize the level of centralization, the following relationship is defined to correlate these three weights: $\omega_1 > \omega_2 > \omega_3$. In this study, the specific values adopted for these weighting parameters are as follows: $\omega_1 = 0.7$, $\omega_2 = 0.2$, and $\omega_3 = 0.1$. It is worth noting that

this setup may yield an unbalanced centralization among the different vMNOs requests. The analysis of other definitions for the DoC , such as those fostering proportional fairness, is left for future work. By dividing this weighted sum in (2) by the total number of resource pools and the total number of virtual DUs assigned in solution X , we can keep the values of DoC within the range $[0, 1]$.

The second factor in our objective function, $(1 - UoL(X))$, is used to account for network links usage (UoL) in the assignment solution X . The goal is to decrease link usage, by minimizing the number of links and the bandwidth required to satisfy the requests of the vMNOs. The UoL in (4) evaluates the total bandwidth used for each physical link assigned (i.e., link in set E^A). As defined in (5), and once again leveraging our previous definition of an assignment as a tuple $s_{j,k}^i = (r_x, c_y, f_z, P_{r_x, c_y}^l)$, the total capacity used depends on the specific bandwidth requirement β_z of the functional split f_z , which is accumulated for all the assignments $s_{j,k}^i \in \mathcal{E}_{v_a, v_b}$ using a particular link. Due to the diversity of links in the midhaul network, a hierarchical approach is taken to assess the cost of the links. Given the relative cost of each type of physical connection ($\gamma_{a,b}$), the cost of optical fiber links is assumed to be higher than that of wired links, which in turn is assumed to be higher than that of wireless links. Similar to DoC in (3), link usages in (4) are divided by the total number of physical links and the maximum capacity of the link in order to define the values of UoL in the range $[0, 1]$. In (2), we subtract $UoL(X)$ from 1 so that the factor can be maximized, as required by our objective function.

The objective function defined in (2) is subject to the following constraints:

$$\forall e_{v_a, v_b} \in E^A \left(\sum_{s_{j,k}^i \in \mathcal{E}_{v_a, v_b}} \beta_z \leq b_{a,b} \right), \quad (6)$$

$$\forall s_{j,k}^i \in S_i \left(\sum_{e_{v_a, v_b} \in P_{r_x, c_y}^l} d_{a,b} \leq \delta_z \right), \quad 1 \leq i \leq q, \quad (7)$$

$$\forall c_y \in C^A \left(\sum_{s_{j,k}^i \in C_{c_y}} w_{prc_k}^v \leq w_{prc_y} \right), \quad (8)$$

$$\forall r_x \in R^A \left(\sum_{s_{j,k}^i \in \mathcal{R}_{r_x}} w_{prc_j}^v \leq w_{prc_x} \right), \quad (9)$$

$$\forall r_x \in R^A \left(\sum_{s_{j,k}^i \in \mathcal{R}_{r_x}} w_{ant_j}^v \leq w_{ant_x} \right), \quad (10)$$

$$\forall r_x \in R^A \left(\sum_{s_{j,k}^i \in \mathcal{R}_{r_x}} w_{prb_j}^v \leq w_{prb_x} \right), \quad (11)$$

$$\forall s_{j,k}^i \in S_i \left(b_{j,k}^v \leq \beta_z \right), \quad 1 \leq i \leq q, \quad (12)$$

$$\forall s_{j,k}^i \in S_i \left(\sum_{e_{v_a, v_b} \in P_{r_x, c_y}^l} d_{a,b} \leq d_{j,k}^v \right), \quad 1 \leq i \leq q, \quad (13)$$

$$|C_i^A| = |C_i^v|, \quad 1 \leq i \leq q, \quad (14)$$

$$|R_i^A| = |R_i^v|, \quad 1 \leq i \leq q, \quad (15)$$

$$|BBU_{pools}| > 0. \quad (16)$$

The expression in (6) models the bandwidth constraint of physical connections. For each physical link e_{v_a, v_b} in use by solution X , the link capacity $b_{a,b}$ must be greater than or equal to the sum of the bandwidths β_z required by the functional splits f_z of all assignments $s_{j,k}^i \in \mathcal{E}_{v_a, v_b}$ using this link. Functional split f_z also involves parameter δ_z , which imposes a constraint on the maximum acceptable delay. The sum of the delays produced by all the links in the path P_{r_x, c_y}^l of an assignment $s_{j,k}^i$ must not exceed δ_z , which must hold for all the assignments of the q vMNO requests, as stated in (7).

The constraint in (8) refers to the computational resource capacity of CUs. For any physical CU used $c_y \in C^A$, this constraint implies that c_y must have a capacity w_{prc_y} at least as high as the sum of the capacities $w_{prc_k}^v$ required by all the virtual CUs c_k^v assigned to this physical node. Similarly, the computational resource capacity constraint for all the DUs assigned $r_x \in R^A$ is modeled in (9). This constraint implies that the physical node r_x must have a capacity w_{prc_x} greater than or equal to the sum of the requirements $w_{prc_j}^v$ of all the virtual DUs r_j^v assigned to r_x . Additional constraints for assigned DUs $r_x \in R^A$ refer to the number of antennas and physical resource blocks, as modeled respectively by (10) and (11). The total number of requested antennas $w_{ant_j}^v$ and resource blocks $w_{prb_j}^v$ for all the assignments $s_{j,k}^i$ involving physical node r_x must not exceed its number of available antennas w_{ant_x} and available resource blocks w_{prb_x} , respectively.

Recall that, as stated at the end of Section III-A3, $s_{j,k}^i$ represents the assignment of physical resources to the virtual link $e_{r_j^v, c_k^v}$. Thus, as captured by (12), the minimum bandwidth

β_z required by the functional split f_z specified in $s_{j,k}^i$ must be at least as high as the bandwidth requirement of the virtual link, namely $b_{j,k}^v$. Similarly, the total delay of the path P_{r_x, c_y}^l specified in $s_{j,k}^i$ must be at least as low as the maximum delay $d_{j,k}^v$ defined by the virtual link $e_{r_j^v, c_k^v}$, as expressed in (13).

In (14), we model the constraint on the number of CUs assigned, which implies that for each of the q vMNO requests, the number of physical CUs assigned $|C_i^A|$ must equal the number of virtual CUs requested $|C_i^v|$. Analogously, the number of physical DUs assigned $|R_i^A|$ must match the number of virtual DUs requested $|R_i^v|$, as indicated in (15). Finally, (16) models the resource pool constraint, which states that there must be at least one resource pool in the assignment.

C. Search Space Analysis

The search space of the above optimization problem, denoted by \mathcal{X} , represents the set of all possible solutions. The size of \mathcal{X} can be computed as:

$$|\mathcal{X}| = \prod_{i=1}^q \left[\left(|\Delta_{c_k^v}^i| \right)^{n_i'} \cdot \left(|\Omega_{r_j^v}^i| \cdot |F| \cdot |P_{r_x, c_y}^*| \right)^{m_i'} \right]. \quad (17)$$

As can be seen in (17), the size of \mathcal{X} is determined by the number of possible assignment configurations that can (independently) take each of the virtual requests. That is, each possible solution $X \in \mathcal{X}$ represents a potential assignment configuration for all the q requests. As defined in Section III-A, the i -th request involves a total of n_i' virtual CUs, and each of these virtual CUs is assigned to a physical node for which there exists a total of $|\Delta_{c_k^v}^i|$ available choices. Similarly, the request involves m_i' virtual DUs, for which the assignment consists of a selection of one of the $|\Omega_{r_j^v}^i|$ available physical DUs, one of the $|F|$ possible functional splits, and one of the $|P_{r_x, c_y}^*|$ existing physical paths.

The above analysis reveals that the optimization problem studied in this paper defines a very large search space, whose size grows exponentially with the number of vMNO requests, as well as with the amount of virtual resources involved in these requests (namely, virtual CUs and DUs). The consideration of large physical networks also poses a challenge, as it is the physical network that defines the sets of choices available to satisfy the requests (namely, physical CUs, DUs, and paths). It is worth noting that our definition of the search space in terms of sets $\Delta_{c_k^v}^i$ and $\Omega_{r_j^v}^i$ entails a meaningful reduction; that is, rather than considering the full sets C and R of physical nodes, we narrow the search in order to focus only on these subsets of nodes that are deemed appropriate for each specific requested virtual node. Furthermore, in order to reduce the cardinality of set P_{r_x, c_y}^* as well, we have employed the generalized version of Dijkstra's algorithm to compute the subset of k shortest paths between the physical nodes, as detailed in Section V-A. Shorter paths are considered more promising choices, as they favor the optimization of link usage, which is the goal pursued by the second factor of our objective function, UoL , as described above. A similar strategy to reduce the number of possible paths has been adopted in other studies (refer, for example, to [12]). These considerations

are intended to make the problem more manageable, although it keeps the exponential nature of its search space.

The above analysis fuels the need for solution approaches that can efficiently explore the enormous search space of this problem in order to identify promising resource assignments within reasonable times. In the next section, we introduce two such approaches to cope with this challenge.

IV. SOLUTION TECHNIQUES

The problem addressed in this paper has a very large search space, as has been analyzed in Section III-C. The use of exact optimization techniques thus becomes prohibitive, even for relatively small problem instances, namely, physical networks with very few nodes and links, and few requests of vMNOs. Such a complexity demands the development of efficient methods, capable of producing high-quality results within reasonable execution times. This section introduces the two heuristic-based approaches that we propose to tackle our problem. First, in Subsection IV-A, we present a greedy algorithm that iteratively constructs a (suboptimal) solution in a rapid manner. Then, Subsection IV-B describes an evolutionary algorithm that implements a specialized initialization routine, based on the greedy strategy introduced in Section IV-A, and uses genetic operators that facilitate a more effective exploration of the search space. The proposed methods thus represent two distinct trade-offs between computational efficiency and the ability to yield higher-quality solutions.

A. Greedy Algorithm

Greedy algorithms construct solutions iteratively, by making the decisions that seem to be the best at each step of the process. Although such locally optimal choices do not usually lead to the global optimum, these heuristics tend to produce satisfactory results in a reasonable amount of time and have been successfully applied in a range of practical scenarios.

Our greedy algorithm is driven by a resource-demanding priority policy. That is to say, the algorithm starts by sorting the requests in descending order based on the amount of resources they involve. Then, it processes all the requests one by one in this order, thus prioritizing those requiring the most resources. Such resource-demanding requests might otherwise be difficult to satisfy if they were considered after some network resources have been already allocated to other requests. Our greedy algorithm is illustrated in Fig. 7.

The processing of a request starts by assigning, one by one, each of the virtual CUs to a physical CU. In particular, virtual CUs that require the most resources are considered first. When two or more suitable physical CUs are available for a given virtual CU, the algorithm always favors the physical CU that would be left with the minimum amount of (unallocated) resources after such an assignment. In other words, the algorithm would choose a physical CU that meets the requirements, but that in turn exceeds them as little as possible. This conservative strategy is intended to reserve resource-rich physical CUs for other more demanding virtual CUs that may arise in future requests.

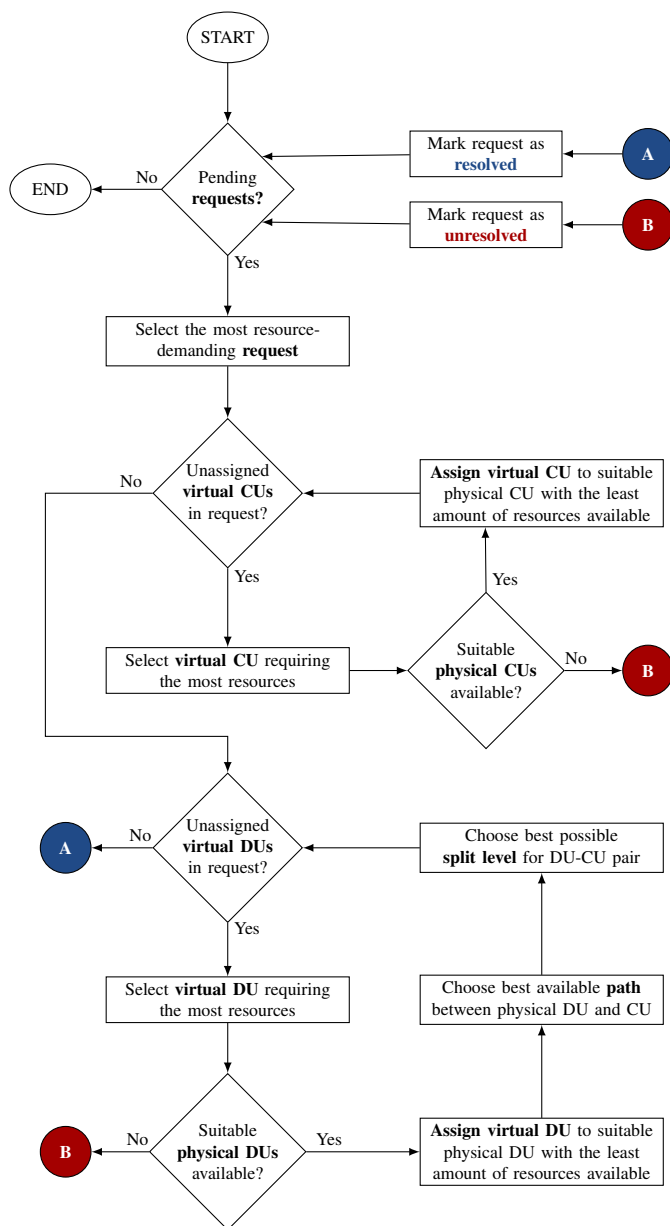


Fig. 7: Flowchart of the proposed greedy algorithm.

Once all the virtual CUs in the request have been assigned to suitable physical nodes, the algorithm attempts the assignment of virtual DUs. Just as for virtual CUs, the most resource-demanding virtual DUs are handled first, and the assignment process targets suitable physical nodes minimizing the amount of available resources. Here, a physical DU is considered suitable if and only if: (i) it meets the resource requirements of the virtual DU; and (ii) there exists at least one path from the physical DU to the corresponding physical CU that satisfies the bandwidth and latency constraints of the request. Once a suitable physical DU is identified, the algorithm completes the assignment by choosing the best available path (minimizing the delay as well as the excess of bandwidth) and the best possible split level (achieving the maximum centralization).

When there does not exist a physical node that can meet the requirements of at least one of the virtual nodes of the request

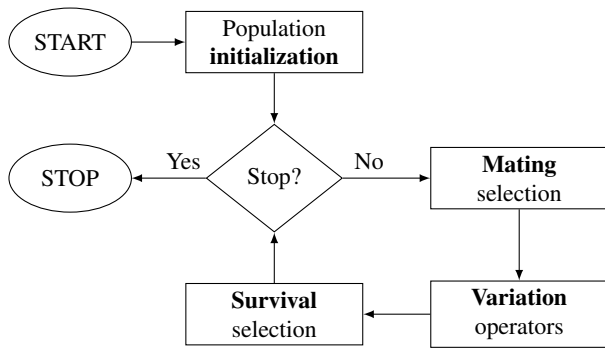


Fig. 8: Flowchart of the proposed evolutionary algorithm.

(this applies both to CUs and to DUs), the whole request is regarded as *unsatisfiable*, and so is marked as unresolved. The algorithm stops once all the requests have been processed and marked either as resolved or unresolved.

B. Evolutionary Approach

Evolutionary algorithms (EAs) are metaheuristic techniques inspired by the principles and mechanisms of natural evolution [39], [40]. Our proposed method follows the generalized scheme of EAs outlined in Fig. 8 [41]. The EA starts by initializing a population of individuals (candidate solutions), and then enters an iterative process that evolves these individuals until a stopping criterion is met (in this case, until a maximum number of generations G_{max} is reached). The evolutionary process consists of three key components: (i) the *mating selection* mechanism, which identifies suitable parent individuals for reproductive purposes; (ii) the *variation operators*, that create new offspring individuals from the selected parents; and (iii) the *survival selection* strategy, which determines the parent and offspring individuals that survive from one generation to the next. All the elements that allow us to instantiate such a generalized scheme in order to address our problem are separately described below.

1) *Solution Representation*: The genetic representation (also referred to as encoding) is a key component of EAs, as it has to be devised so as to suit the problem and facilitate the design of operators that can effectively sample the solution space [42], [43]. An integer representation is adopted in this study, where the genotype of a candidate individual encodes a potential physical resource assignment for each of the requests. An encoding example is depicted in Fig. 9, where the physical network and the virtual request are those illustrated earlier in Figs. 3 and 4, respectively. As can be seen, the assignment configuration for any given request is encoded as a string of integers. Each encoding position (gene) corresponds to a particular decision and it can take any value from a finite set of possible choices (alleles). These decisions include the physical nodes each of the virtual nodes will be assigned to (CUs and DUs), as well as the choice of a physical path and the functional split level to use for each DU-CU pair selected.

In the example of Fig. 9, the specific integer string encodes that virtual nodes c_1^v , c_2^v , r_1^v , r_2^v , and r_3^v are assigned to physical nodes c_1 , c_2 , r_1 , r_2 , and r_5 , respectively. Configurations for

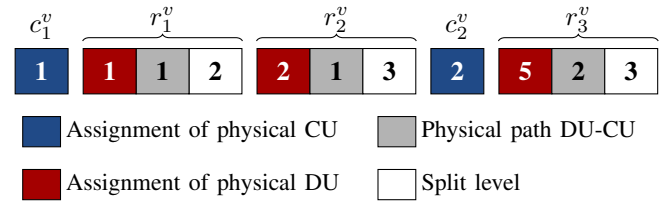


Fig. 9: Encoding of a possible resource assignment for the virtual request in Fig. 4 over the physical network in Fig. 3.

r_1^v and r_2^v indicate that the first (and only) available path between the assigned physical nodes will be used, and that the split levels will be f_2 and f_3 , respectively. Finally, the last three encoding positions indicate that r_3^v will use the second available path between the assigned physical nodes, and functional split f_3 . It is worth noting that multiple virtual DUs can be connected to the same virtual CU in the request, and so the encoding positions that refer to DU configurations are organized by CU in our genetic representation.

The set of all virtual resource requests is represented by a matrix, where the i -th row corresponds to the i -th request. The number of columns in this matrix corresponds to the maximum number of encoding positions required by any of the requests, which depends on the number of virtual CUs and DUs involved. More specifically, the number of columns in the matrix is given by $\max(n_i^v + 3 \cdot m_i^v)$, being n_i^v and m_i^v the number of virtual CUs and DUs of the i -th request, respectively (as defined in Section III-A). Given that some requests will only need a subset of the matrix columns, unnecessary columns are simply ignored in our method.

2) *Population Initialization*: The proposed EA employs a specialized initialization routine which aims to seed the population with a diverse set of N potentially good initial individuals. In order to do so, our initialization routine takes advantage of the greedy algorithm proposed in Section IV-A.

The first individual of the population is built by using the greedy algorithm, exactly as described in Section IV-A. The remaining $N - 1$ individuals, however, are generated using a modified version of this algorithm. Rather than processing, selecting, and assigning resources (namely, requests, CUs, DUs, paths, and split levels) in the order dictated by their priority, which as explained above is established by the total amount of demanded resources, the modified version of this algorithm makes all these decisions at random. Despite such random (non-greedy) decisions, the modified algorithm preserves the characteristic of only assigning virtual nodes whenever a suitable physical node can be found.⁵

The above modification thus converts the originally deterministic procedure into a non-deterministic one, which can generate a potentially different individual with each execution.

3) *Optimization Criterion*: The objective function defined in (2), Section III-B, which is to be maximized, is adopted as the fitness (evaluation) function to measure the quality of candidate solutions in our proposed method.

⁵Even a single virtual node for which no suitable physical node is available makes the whole request unsatisfiable, as indicated at the end of Section IV-A.

4) *Mating Selection*: We use a panmictic mating strategy [44], giving all individuals in the population the same probability of being selected as parents. The parents that will undergo recombination (see variation operators below) are randomly selected (without replacement). This is intended to reduce the selection pressure and increase the diversity.

5) *Variation Operators*: Variation operators produce new candidate solutions and are to a large extent responsible for the exploration and exploitation capabilities of every search algorithm [45]. Our method uses both recombination and mutation as variation operators. Recombination takes as input two parent individuals, and produces as output two offspring individuals by means of *uniform crossover* [46]. This operator works by creating exact copies of the input parents, and by either preserving or interchanging every piece of genetic information between these copies with equal probability. Recombination is applied with a given probability p_r .

All the offspring produced through recombination are subjected to *uniform mutation* [47]. This operator independently mutates each position in the encoding of the individual with a given probability p_m . When a position mutates, its original value is replaced by another one, uniformly chosen from the set of available options for that particular encoding position.

6) *Survival Selection and Constraint-Handling*: Our survival selection strategy relies on the use of stochastic ranking (SR), which is known to be an effective constraint-handling technique [48]. Once an offspring population has been created by using the variation operators, these new individuals compete against the parent individuals for a place in the population of the next generation. SR thus provides a suitable means to rank all the individuals (both parents and offspring), taking into account the constrained nature of our problem.

SR employs a bubble-sort-like procedure and a user-defined parameter, here denoted p_s , representing the probability of using either fitness or the degree of constraint violation as the basis for solution comparison. For a comprehensive description of SR, the reader is referred to [48]. In order to promote the diversity of the population, all duplicated individuals are filtered and discarded before applying the SR technique.

V. EVALUATION SETUP

This section provides details about the evaluation setup that was developed to validate the proposals of this paper. This includes the description of the test scenarios considered, the parameter settings adopted for the algorithms evaluated, and the performance assessment methodology employed.

A. Test Scenarios

We define a problem instance as a triplet that contains a substrate network, a set of resource requests from vMNOs, and a set of pre-calculated routes. For our experiments, we generated a total of 3,024 instances and classified them into three categories: small-size, medium-size, and large-size (there are 1,008 instances in each of these groups). For the network topologies, we considered Barabasi-Albert, Waxman, Hierarchical Top-Down, and Hierarchical Bottom-Up substrate topologies. The results reported in Section VI correspond to

TABLE III: Parameters used for generating physical networks.

| Group | # nodes min,max | # links min,max | # DUs min,max | # CUs min,max | # substrate networks |
|--------|--------------------|--------------------|------------------|------------------|-------------------------|
| small | 3, 10 | 3, 12 | 2, 8 | 1, 3 | 42 |
| medium | 100, 500 | 197, 1000 | 20, 50 | 14, 26 | 56 |
| large | 600, 1000 | 1197, 2000 | 40, 200 | 25, 50 | 56 |

TABLE IV: Parameters for the generation of virtual requests.

| Group | # request per set min, max | # virtual DUs min, max | # virtual CUs min, max | # sets of requests |
|--------|-------------------------------|---------------------------|---------------------------|-----------------------|
| small | 1, 5 | 2, 7 | 1, 3 | 8 |
| medium | 5, 50 | 2, 10 | 1, 5 | 6 |
| large | 50, 100 | 2, 10 | 1, 5 | 6 |

the average for all such topologies. For completeness, however, a detailed analysis of the performance of our proposals on a per-topology basis is presented in Appendix D.

For the small-size test instances, the network topology can be one out of 42 options. The substrate networks of all instances are generated with the BRITE toolkit⁶ with random values within the following ranges: the smallest network has 3 nodes with 3 links, and the largest network has 10 nodes and 12 links. The networks are generated with a random number of DUs between 2 and 8, and a random number of CUs between 1 and 3. The small-size instances include one of 8 sets of resource requests. Each of these sets is generated with a random number of resource requests in ranges from 1 to 5. Each resource request is generated with a random number of virtual DUs which ranges from 2 to 7, and with random numbers of virtual CUs in the range from 1 to 3.

For medium-size instances, the network topology can be one out of 56 options. Substrate networks are generated randomly considering the following ranges: the smallest network has 100 nodes with 197 links, and the largest network has 500 nodes and 1,000 links. The networks are generated with a random number of DUs between 20 and 50, and a random number of CUs between 14 and 26. These instances include one of 6 sets of resource requests, each of which is generated with a random number of resource requests ranging from 5 to 50. Each resource request is generated with a random number of virtual DUs with ranges from 2 to 10, and with a random number of virtual CUs in the range from 1 to 5.

For large-size instances, the network topology can be one out of 56 options. Substrate networks are generated at random using the following ranges: the smallest network has 600 nodes with 1,197 links, and the largest network has 1,000 nodes and 2,000 links. The networks are generated with a random number of DUs between 25 and 50, and a random number of CUs between 40 and 200. These instances include one of 6 sets of resource requests, each generated with a random number of requests in the range from 50 to 100. Each resource request has a random number of virtual DUs between 2 and 10, and a random number of virtual CUs between 1 and 5.

⁶BRITE - Boston University Representative Internet Topology Generator, software to generate network topologies. Source code: <https://github.com/unly/brite>, BRITE homepage: <https://www.cs.bu.edu/brite/index.html>

TABLE V: Parameter settings selected for EA^{gre} and EA^{rnd} . The table shows the specific values of the recombination (p_r), mutation (p_m), and stochastic ranking (p_s) probabilities that are used for small, medium, and large problem instances.

| Algorithm | Small-size instances | Medium-size instances | Large-size instances |
|------------|--|--|--|
| EA^{gre} | $p_r = 0.5$ $p_m = 3/\ell$ $p_s = 0.425$ | $p_r = 0.7$ $p_m = 1/\ell$ $p_s = 0.425$ | $p_r = 0.7$ $p_m = 1/\ell$ $p_s = 0.425$ |
| EA^{rnd} | $p_r = 0.5$ $p_m = 3/\ell$ $p_s = 0.475$ | $p_r = 0.5$ $p_m = 1/\ell$ $p_s = 0.475$ | $p_r = 0.7$ $p_m = 1/\ell$ $p_s = 0.475$ |

ℓ refers to the total number of encoding positions in the genotype.

For the three groups of test instances (small-size, medium-size and large-size), we generated three sets of pre-calculated routes. The k -shortest path routing version of Dijkstra’s algorithm was used, where each set corresponds to the k routes between each CU-DU pair of nodes, varying k from 1 to 3.

For all types of instances, the available resources and the requested resources were randomly generated using the following ranges of values: for each CU, w_{prc} takes a value from 10 to 20; for each physical link, $b_{i,j}$ takes a value from 1,000 and 10,000 and $\gamma_{i,j}$ has values among 0.1, 0.2 and 0.7; for each DU, w_{ant} takes a value from 10 to 50, w_{prc} takes a value from 10 to 20, w_{prb} takes a value from 50 to 100, and θ takes a value from 10 to 100; for each virtual link, $b_{i,j}^v$ takes a value from 100 to 1,000, and $d_{i,j}^v$ takes a value from 100 to 500; for each virtual CU, w_{prc}^v takes a value from 2 to 6; for each virtual DU, w_{ant}^v takes a value from 2 to 10, w_{prc}^v takes a value from 1 to 5, and w_{prb}^v takes a value from 6 to 15.

Table III summarizes the parameters used to generate the substrate networks, and Table IV summarizes the parameters used to generate the virtual resource requests.

B. Algorithms Evaluated and Parameter Settings

Our experiments seek to investigate and compare the performance of our two proposed methods: the greedy algorithm described in Section IV-A, and the evolutionary approach introduced in Section IV-B. In addition, we consider a modified version of the evolutionary approach, which is enhanced with a random initialization approach. This variant will allow us to evaluate to which extent the use of a specialized initialization routine contributes to the performance of our method. The three algorithms will be respectively referred to as GRE , EA^{gre} , and EA^{rnd} throughout our evaluation study.

A preliminary analysis was conducted to investigate appropriate parameter settings for EA^{gre} and EA^{rnd} (see Appendix B). Based on those preliminary experiments, the most promising settings for each of these algorithms were selected and used for the experiments whose results are presented in Section VI. Table V summarizes the settings for the probability of recombination, mutation, and stochastic ranking. In all cases, the population size and the number of generations were set to $N = 100$ and $G_{max} = 300$, respectively.

C. Performance Assessment

Due to the stochastic nature of the approaches EA^{gre} and EA^{rnd} , a total of 31 independent executions of these algorithms was performed for every problem instance considered. Given that our proposed GRE method is deterministic, a single execution of this algorithm was considered in all cases. In our experiments, the performance of these algorithms is evaluated in terms of solution quality, request acceptance rate, execution time, and the statistical significance as further described below.

1) *Solution Quality*: In this study, solution quality refers to the degree of centralization of a given resource assignment. This is properly captured by the objective function defined in (2), Section III-B, which is to be maximized. In our experiments, thus, solution quality corresponds to the objective value of the (best) candidate solution reached by an algorithm. Even though objective values are defined in the range $[0, 1]$, the best attainable objective value can vary from one instance to another. Therefore, the objective values obtained for a particular problem instance have been normalized by dividing them between the best (highest) objective value observed for that instance (considering all the algorithms evaluated and executions performed). We report both, the average solution quality and the best solution quality achieved by an algorithm in the set of all independent executions performed.

2) *Acceptance Rate*: This metric is computed as the ratio of the number of virtual resource requests that were successfully assigned to physical resources, let this be denoted by q' , to the total number of requests, q . Formally, the acceptance rate is computed as q'/q , and it is defined in the range $[0, 1]$, and higher values of this metric indicate a better performance.

3) *Execution Time*: In order to conduct a fair comparison based on execution time, we ensured that all the algorithms were run under similar conditions. Experiments for measuring execution time were run in an exclusively dedicated server, equipped with 2 Intel Xeon E5-2630 (2.30 GHz) processors, 32 GB of RAM, and the Ubuntu 18.04.4 LTS operating system.

4) *Statistical Significance*: We investigate the statistical significance of the differences observed among the set of algorithms evaluated as follows. First, the (non-parametric) Kruskal-Wallis test is carried out in order to investigate the set of algorithms as a whole. If the test indicates that there exists a significant difference, a post hoc analysis based on the Mann-Whitney U test and Bonferroni correction is conducted to investigate specific pairs of algorithms. In all the cases, a significance level of $\alpha = 0.05$ is considered.

VI. RESULTS

This section presents the results of a series of experiments conducted in order to investigate the suitability of the two algorithms proposed in Section IV, namely, our greedy algorithm denoted by GRE , and our evolutionary method denoted by EA^{gre} . Also, as indicated in Section V-B, a modified version of our evolutionary method, here referred to as EA^{rnd} , is included in our comparative analysis as a baseline. The three algorithms are investigated in the context of a diverse

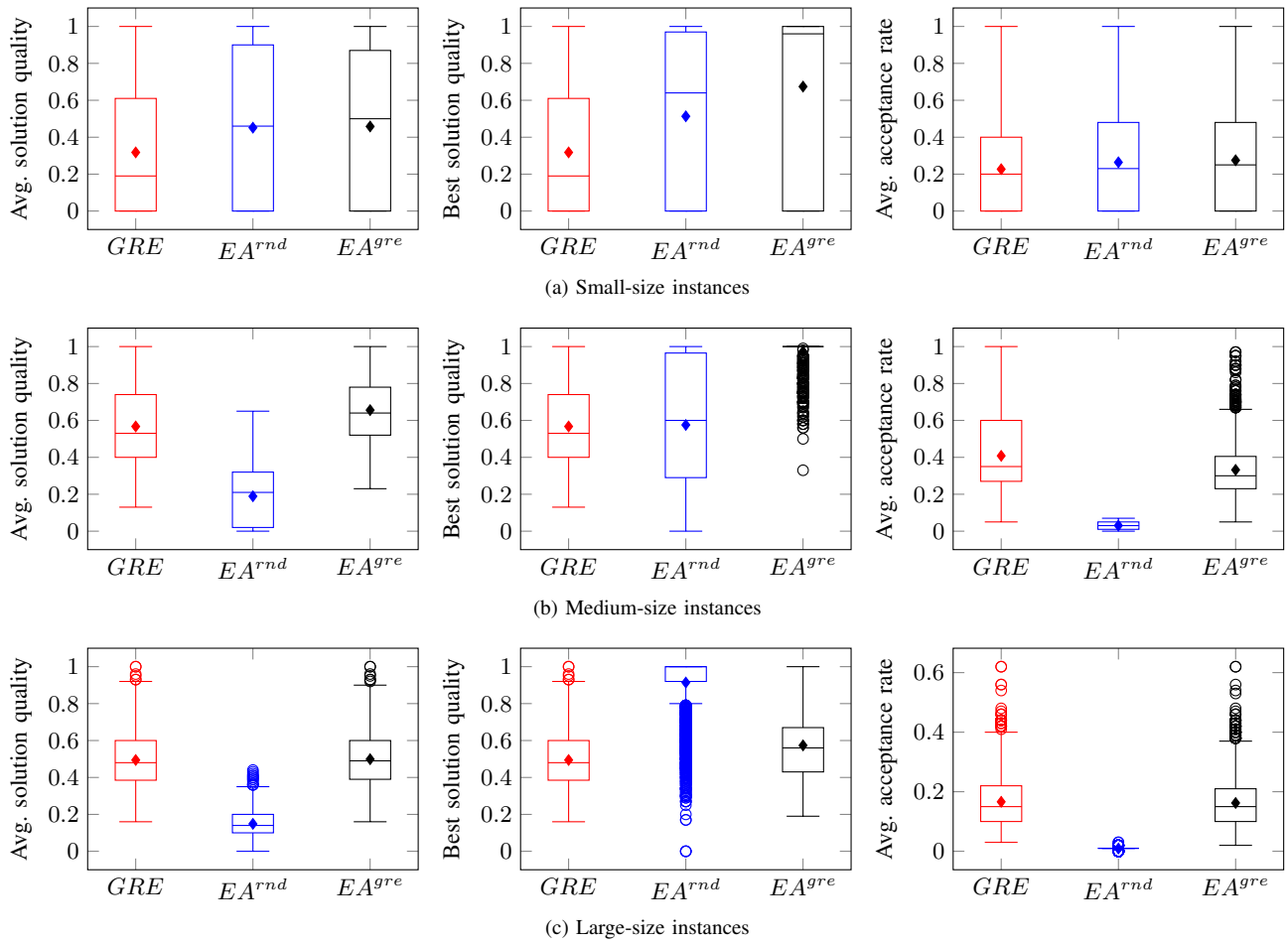


Fig. 10: Summary of the results obtained by approaches GRE , EA^{gre} , and EA^{rnd} . Results are shown separately for small-size instances (a), medium-size instances (b), and large-size instances (c). For each of the problem categories, results are evaluated in terms of the average solution quality (left), the best solution quality (center), and the average acceptance rate (right).

collection of test scenarios, which are organized into the small-size, medium-size, and large-size categories of problem instances, as described in Section V-A. The performance of these algorithms is evaluated on the basis of the average and best achieved solution quality, the request acceptance rate, and the execution time (refer to Section V-C for a description of these measures). The results obtained are analyzed and discussed from different perspectives in the following subsections.

A. Overall Performance

The results obtained during the testing of approaches GRE , EA^{gre} , and EA^{rnd} are shown in Fig. 10, with the findings of the statistical significance analysis summarized in Appendix A. As can be seen, the proposed evolutionary method, EA^{gre} , is found to be the best overall performer in our experiments, offering the best compromise for the set of evaluation measures employed. In all the three categories of test scenarios considered, EA^{gre} outperforms the other two algorithms, or at least achieves comparable results, in terms of average solution quality, best solution quality, and acceptance rate.

Our GRE method exhibits a significantly lower performance for the small problem instances in comparison to the

evolutionary approaches EA^{gre} and EA^{rnd} . Nevertheless, in the medium-size and large-size scenarios, GRE clearly surpasses EA^{rnd} in terms of average solution quality and acceptance rate (with statistically significant differences), while competing closely with EA^{gre} .⁷ As described in Section IV-B, GRE is the basis of the initialization routine of EA^{gre} ; the fact that GRE is able to produce, by itself, very satisfactory results, confirms that our specialized initialization provides EA^{gre} with a clear competitive advantage and explains, to a certain extent, the success of this method in our tests.

Despite the promising results obtained by EA^{rnd} for the small instances, this algorithm reports poor average solution qualities and acceptance rates for the medium-size and large-size scenarios. Such performance decreases are statistically significant when compared with respect to both GRE and EA^{gre} (see Appendix A). This suggests that the advantages of a specialized initialization routine become more evident as the size of the instances (and therefore the search space) increases. Given that EA^{rnd} starts with a population of random, most likely infeasible solutions, it can invest a significant amount

⁷Note that the proposed GRE method is deterministic. Hence, the average solution quality is the same as the best solution quality for this algorithm.

of effort just to reach the feasible region. This behavior is evidenced by the convergence curves of this algorithm, as analyzed in Section VI-B. However, in spite of such invested effort, EA^{rnd} usually fails at discovering even a single feasible solution (satisfying the requirements of at least one of the vMNO requests), as indicated by the very poor acceptance rates scored by this algorithm. Interestingly, though, the results for the large instances show that EA^{rnd} outperforms our two proposals when measuring the best solution quality. This suggests that the unbiased nature of a purely random initialization routine still has some merits, and allows EA^{rnd} to produce some successful runs for a number of test scenarios.

To further illustrate the performance of our EA^{gre} algorithm, Appendix C includes an example of a test instance as well as the highest-quality solution produced by this method.

B. Convergence Behavior

This section analyzes the convergence behavior of algorithms EA^{gre} and EA^{rnd} as a means of explaining their performance differences as well as highlighting the role and impact of initialization. Fig. 11 illustrates the evolution of the highest quality solution for both EA^{gre} and EA^{rnd} throughout the evolutionary process in a sample of test scenarios.

The examples in Fig. 11 confirm that the initial efforts of algorithm EA^{rnd} are mostly invested in locating the feasible region, as argued earlier in Section VI-A. This behavior can be observed as a flat region in the convergence curves of EA^{rnd} during the early stages of the search, which is particularly evident for the medium-size and large-size examples. As shown in the plots, after a number of generations have passed, EA^{rnd} manages to identify a feasible individual, and hence the algorithm can focus further on improving solution quality. The figure also confirms that this limitation of EA^{rnd} is accentuated as larger problem instances are considered.

As described in Section V-B, the initialization process is the only distinguishing factor between approaches EA^{gre} and EA^{rnd} . Therefore, our analysis confirms that this component plays a critical role and is a major contributor to our method's performance. From Fig. 11, it is possible to observe that our specialized initialization routine is effective at seeding the population of EA^{gre} with good-quality, feasible initial solutions. EA^{gre} is thus able to start the evolutionary process with a significant advantage over the reference method EA^{rnd} (see in Fig. 11 the differences between the performance of the initial populations of these methods, at generation 0).

Finally, it is worth mentioning that although initialization is a key component, it is not the only factor contributing to the performance of EA^{gre} . The convergence curves shown in Fig. 11 illustrate the ability of the method to gradually improve solutions over time. While initialization is responsible for setting up a good starting point for the search, the evolutionary process itself is the key to capitalize on such a favorable initial state in order to reach a high-quality final solution.

C. Ability to Produce High-Quality Solutions

The results presented in the previous subsections are suitable to compare the studied algorithms with respect to each other,

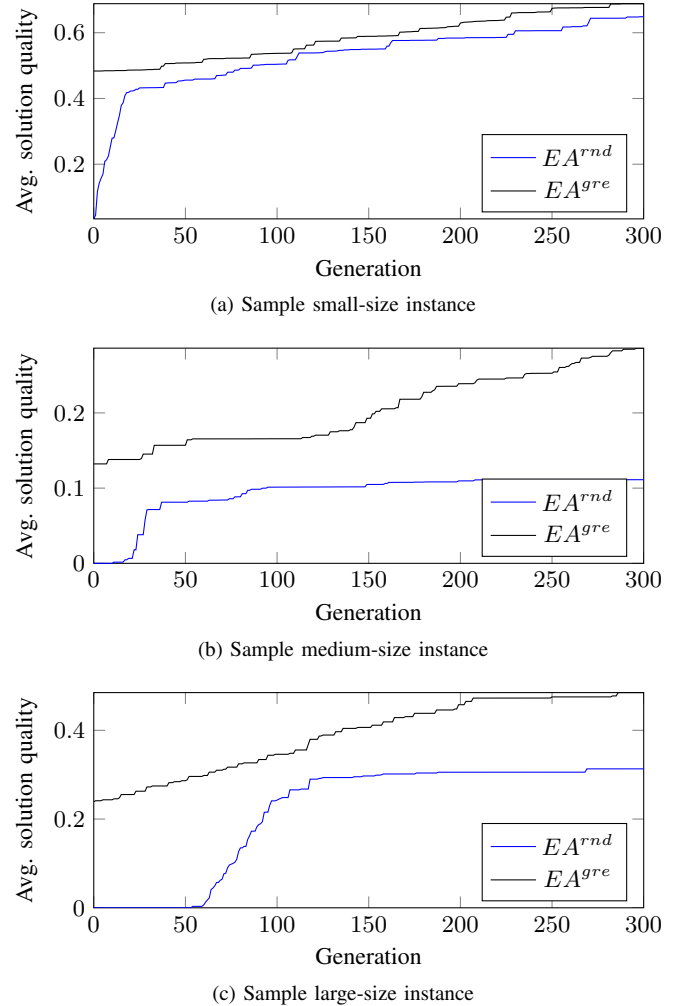


Fig. 11: Convergence behavior of approaches EA^{gre} and EA^{rnd} . The plots show the best solution quality observed at every generation, averaging the results of all the independent executions performed. Results are exemplified for a small-size (a), a medium-size (b), and a large-size (c) problem instance.

based on their relative performance. However, due to the heuristic nature of these approaches, there remains the question of how close the solutions produced by these algorithms are to the optimal solution of the problem instances considered.

The main challenge in addressing this question lies in the fact that the optimal solutions for our test instances are unknown and, more importantly, they are computationally expensive to calculate. Thus, our attempt to shed some light on this question necessarily focuses on a subset of our small-size instances, for which the optimum has been determined through the systematic, exhaustive enumeration of the solution space. Note that this analysis is limited to small instances, as it becomes prohibitive for larger instances due to the exponential growth of the solution space (see Section III-C). Accordingly, the analysis presented in this section is based on results computed for a selection of 129 of our small-size problem instances (originally described in detail in Section V-A). These selected instances involve a maximum of 4 virtual resource requests, where each request has at most 5 DUs and 3 CUs,

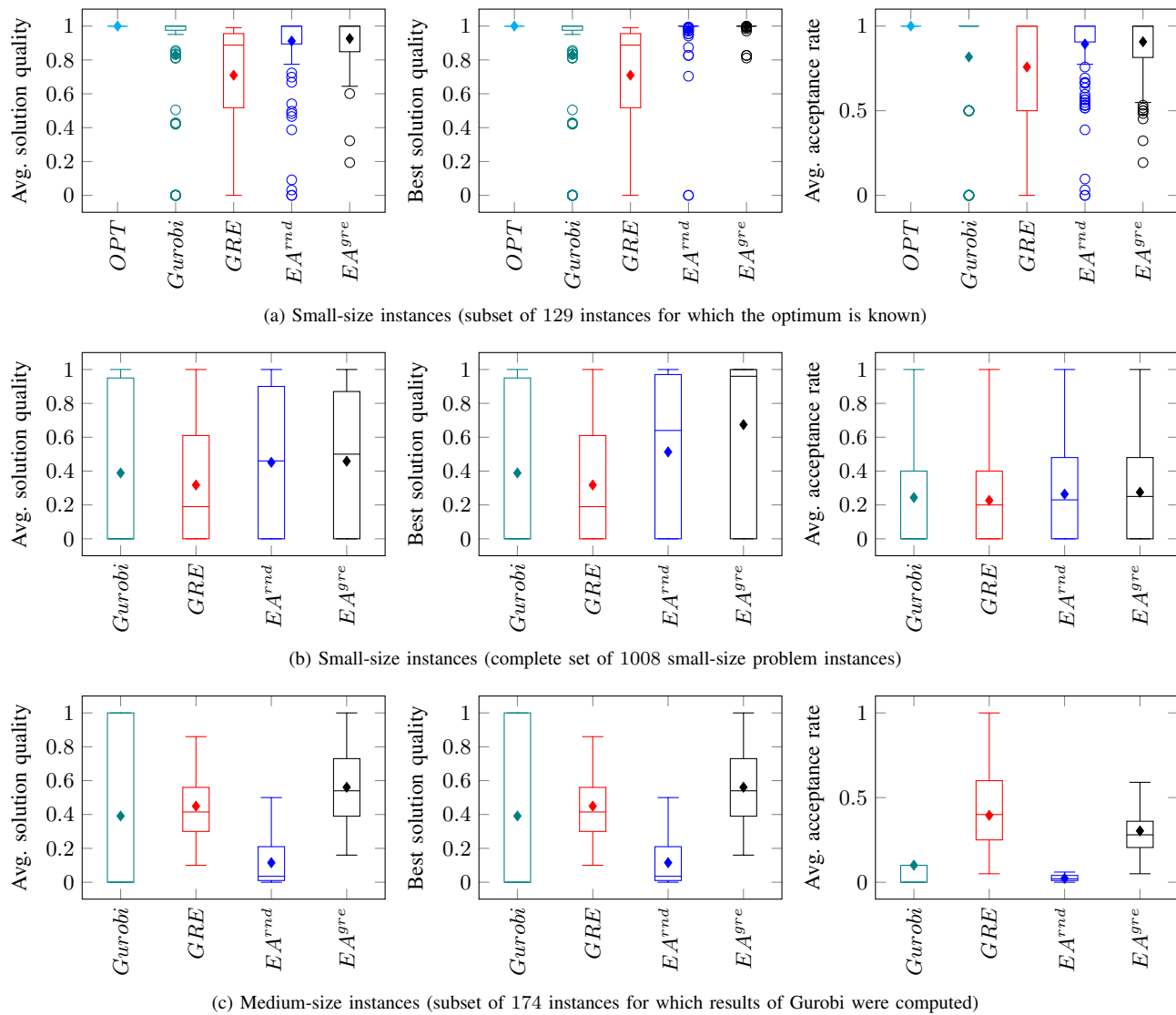


Fig. 12: Ability of approaches GRE , EA^{gre} , and EA^{rnd} to find good-quality solutions. In (a), the results of these approaches are evaluated with respect to the optimal solution (OPT) and the results of solver $Gurobi$ [49] in a subset of 129 small-size problem instances. In (b) and (c), comparisons with respect to solver $Gurobi$ are presented for the full set of 1008 small-size instances and for a subset of 174 medium-size instances, respectively. In all the cases, results are evaluated in terms of the average solution quality (left), the best solution quality (center), and the average acceptance rate (right). Results are normalized by dividing the obtained solution qualities and acceptance rates by those of the optimal (or best known) solution.

as well as substrate networks with at most 6 DUs, 2 CUs, and 2 available routes connecting each DU-CU pair.

In view of the above limitations, and with the aim of extending this analysis to a larger number of instances, results for solver $Gurobi$ [49] are included as an additional reference. $Gurobi$ is one of the most powerful, commercial mathematical programming solvers available, and one of the fastest. This allowed us to compute results for all the 1008 small-size instances, and for a subset of 174 instances of the medium-size category (instances with at most 20 virtual requests, each involving a maximum of 10 DUs and 5 CUs; and substrate networks with at most 500 nodes, a maximum of 30 DUs and 20 CUs, and a single available route for each DU-CU pair). Note that, as shown in Section VI-D, the execution times of $Gurobi$ increase significantly with the size of the problem, and

consequently the remaining medium-size instances as well as all large-size instances have been excluded from this comparison. It is also worth noting that $Gurobi$ does not always produce the optimum, but it often yields approximations that can serve the purposes of this analysis to evaluate the ability of our methods to find good-quality solutions.

The results of this analysis are summarized in Fig. 12. As can be seen from sub-figure (a), $Gurobi$ and evolutionary approaches EA^{gre} and EA^{rnd} report solution qualities that in most cases are comparable to that of the optimal solution. This suggests that the three methods are able to either locate the optimum or at least produce a close approximation for the particular subset of small-size instances considered. The results for the full set of small-size instances, shown in sub-figure (b), reveal that both EA^{gre} and EA^{rnd} perform

better in average than solver *Gurobi*, used as the reference. Similarly, sub-figure (c) indicates that in average EA^{gre} yields higher solution qualities and acceptance rates in comparison to *Gurobi* for the set of medium-size instances analyzed, highlighting the suitability of our proposed method.

On the other hand, our *GRE* algorithm is seen to produce good approximations for some of the small-size instances, but overall its solution qualities and acceptance rates are inferior when compared to those of the optimal solution and solver *Gurobi*. Note, however, that in average the solution qualities scored by *GRE* are comparable to those obtained by *Gurobi* on the medium-size scenarios, where *GRE* also reports the highest acceptance rates among all the approaches evaluated; this latter result suggests that the strategy of *GRE* of prioritizing resource-demanding requests may grant this method a better ability to satisfy a greater number of requests.

D. Computational Efficiency

A key aspect in determining the applicability of an algorithm in practice is certainly its computational efficiency. While the analyses presented earlier reveal that our methods are able to yield high-quality solutions, we now turn to investigate the efficiency with which such solutions are reached.

The execution times measured during our experiments are shown in Fig. 13. The results in sub-figure (a) make evident that the exponential size of the search space of this problem renders an exhaustive exploration impractical, even for very small instances. Although such an exhaustive approach was only applied to a subset of our smallest test cases (refer to Section VI-C for details), the times reported are relatively very high (the largest instances considered took more than 22 days to run, using the hardware described in Section V-C). Sub-figure (a) also shows that, in general for the small-size instances, evolutionary approaches EA^{gre} and EA^{rnd} are found to be more computationally expensive than solver *Gurobi* [49] (which is used as a reference, as detailed in Section VI-C). It is possible to see from sub-figure (b), however, that increasing the size of the problem leads to a significant growth in the execution times of *Gurobi*, suggesting that our proposed heuristic methods represent better alternatives to cope with the larger problem sizes that appear in practice.

Fig. 13 also points out the high efficiency of our *GRE* method, which is able to produce solutions within times of the order of milliseconds even for the largest test cases. The figure also evidences that the use of the evolutionary approaches EA^{gre} and EA^{rnd} causes a significant increase in the execution time with respect to algorithm *GRE*. The execution times reported by approaches EA^{gre} and EA^{rnd} are however of the order of seconds in the majority of the cases. Thus, such observed increases in the execution times may be regarded as negligible compared to the meaningful advantages that the use of algorithm EA^{gre} can provide in terms of optimization performance, as seen in Section VI-A.

Finally, owing to the low computational cost of *GRE*, the repeated use of this algorithm within our specialized initialization has not caused a very noticeable impact on the running time of the EA^{gre} method. Only minor increases in

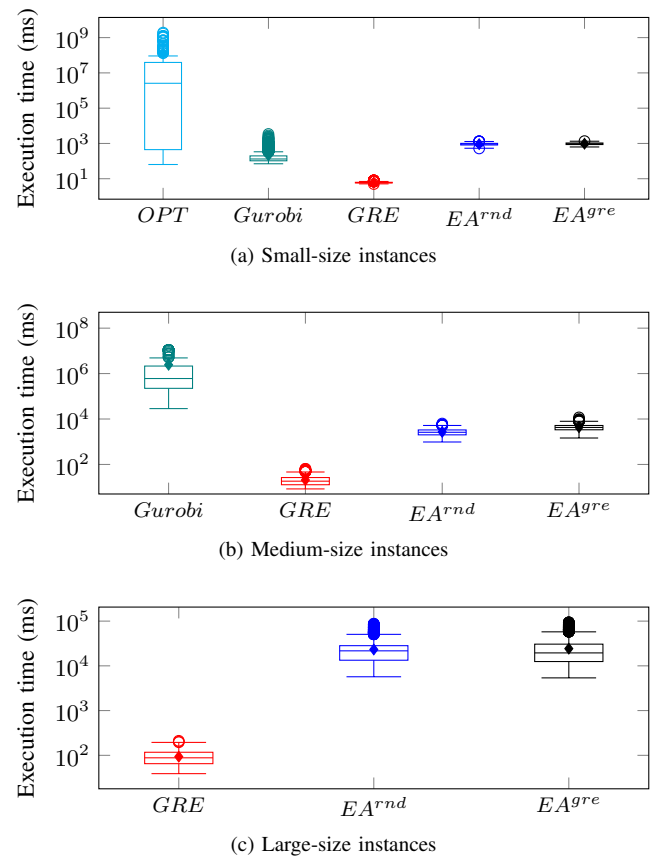


Fig. 13: Execution times of our experiments. Plots in (a), (b), and (c) present results for small, medium, and large instances, respectively. In (a), the times taken to find the optimum by exhaustive search (*OPT*) are included for a subset of instances (see Section VI-C). Results of solver *Gurobi* [49] for all small instances are included in (a), but only results for a subset of medium-size instances are reported in (b). Note that times are expressed in milliseconds and shown in logarithmic scale.

the execution time can be observed from Fig. 13 with respect to the use of a random initialization in approach EA^{rnd} (such increases are only statistically significant for the small-size and medium-size instances, see Appendix A).

The algorithmic solutions presented in this paper are the first solutions reported in the literature that have addressed the joint selection of route and split level holistically. At this point, the results obtained by our algorithmic solutions are encouraging to address the critical target of providing solutions to online versions of the problem addressed in this paper.

VII. CONCLUDING REMARKS AND FUTURE WORK

This paper addressed the critical problem of jointly selecting midhaul routes and determining appropriate functional splits over 5G C-RAN architectures, to support the quality of service requirements of vMNOs. We have shown that this is a highly complex problem, which was tackled by holistically considering the two following sub-problems: (1) the problem of deciding the most appropriate location for DUs and CUs composing the BS, as well as the optimal split distribution

between them; and (2) that of selecting the best routes for the traffic coming from each of the DUs. To our best knowledge, this is the first work that has addressed the aforementioned problems in a combined way. In fact, there exist few works that have considered either of them individually.

We have built a system model to address such a challenge, formulating a virtual embedding problem, which we have afterwards holistically tackled. Two algorithmic solutions have also been proposed: a greedy approach, and an evolutionary-based solution which implements an intelligent initialization procedure. After an extensive validation, which has been made over a diverse set of network topologies, we have shown that the proposed techniques provide appropriate trade-offs between the quality of the solutions, the acceptance rate of virtual requests, and execution time. We have also compared the solutions obtained by our methods with respect to the optimal solution (in those cases where this was possible), as well as with respect to a well-known commercial solver. The outcome of such an analysis was very satisfactory: our proposed heuristics were found to produce good approximations to the optimum, and better trade-offs regarding the above criteria when compared to the reference solver considered.

The model, algorithmic solutions, and test instances presented in this paper represent the first efforts reported to address the joint route and functional split selection problem over C-RAN architectures to fulfill vMNOs requests. At this point, the results obtained by our algorithmic solutions, in terms of quality of solutions, acceptance rates, and execution times, are encouraging to address the online version of this complex problem, which in turn will be part of our future work. We will also analyze the behavior of our approaches under special cases of the network topologies and other peculiarities with a potential impact over the aforementioned performance trade-offs. We will explore alternative optimization problem formulations and, in particular, definitions of the DoC that promote proportional fairness among requests, which would enable the implementation of different centralization policies.

From the algorithmic viewpoint, future work will be devoted to the design of novel genetic operators that exploit knowledge of the problem domain to enhance the generation of high-quality, feasible solutions. We will also analyze whether the consideration of alternative metaheuristic approaches could yield better performance trade-offs, providing a faster convergence while maintaining, or even improving, solution quality in comparison to our proposed evolutionary method.

In addition, we will also propose system model modifications to include practical requirements. First, we will exploit the developed framework and proposed methodology to analyze slicing techniques for Beyond 5G Networks, where appropriate resource management will become even more relevant. Furthermore, we will investigate additional restrictions on the split set, according to the coordination functionalities they provide and the required QoS. Finally, we will study the energy reduction brought by the proposed optimization problem, which comes from the increased centralization and the link usage minimization. We will also consider the addition of explicit energy metrics, to give more relevance to this particular aspect in the split decision process.

TABLE VI: Results of the statistical significance analysis. Algorithms GRE , EA^{rnd} , and EA^{gre} are compared pairwise, considering three different performance measures: average solution quality (Q), average acceptance rate (A), and execution time (T). The results are presented separately for small-, medium-, and large-size problem instances. Symbols \checkmark and \times indicate whether or not a statistically significant difference is observed between a specific pair of algorithms compared.

| Instances | $GRE - EA^{rnd}$ | | | $GRE - EA^{gre}$ | | | $EA^{rnd} - EA^{gre}$ | | |
|-----------|------------------|--------------|--------------|------------------|--------------|--------------|-----------------------|--------------|--------------|
| | Q | A | T | Q | A | T | Q | A | T |
| Small | \checkmark | \checkmark | \checkmark | \checkmark | \checkmark | \checkmark | \times | \checkmark | \checkmark |
| Medium | \checkmark | \checkmark | \checkmark | \checkmark | \checkmark | \checkmark | \checkmark | \checkmark | \checkmark |
| Large | \checkmark | \checkmark | \checkmark | \times | \times | \checkmark | \checkmark | \checkmark | \times |

APPENDIX A STATISTICAL SIGNIFICANCE ANALYSIS

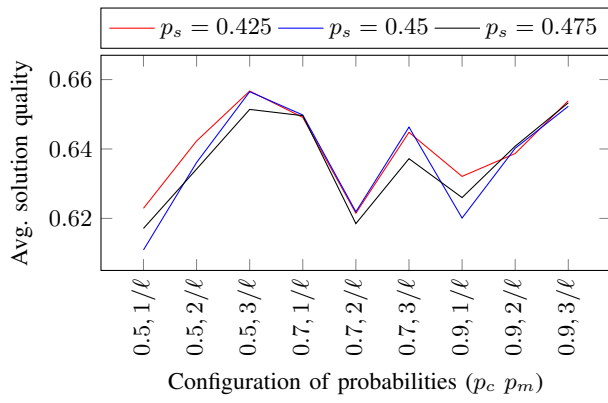
As described in Section V-C4, we conducted an analysis of statistical significance to determine how meaningful the performance differences between methods GRE , EA^{rnd} , and EA^{gre} are. This analysis focuses on a subset of relevant results, including: (i) average solution qualities and acceptance rates obtained, as reported in Section VI-A, Fig. 10; and (ii) execution times measured for these methods, as reported in Section VI-D, Fig. 13. Given that our initial analysis using the Kruskal-Wallis test revealed that in all the cases there is a significant difference within the group of methods analyzed, we therefore applied the Mann-Whitney U test (Bonferroni corrected) to the three possible pairs of methods. The findings of this post hoc analysis are summarized in Table VI.

The table indicates that, in the majority of the cases, a statistically significant difference is observed for every pair of algorithms compared, for the three performance measures under consideration. A few exceptions can be highlighted. For the large instances, the differences observed between our two proposed algorithms GRE and EA^{gre} are not statistically significant in terms of both solution quality and acceptance rate. Also, there is no statistically significant difference between the average quality of the solutions obtained by the two evolutionary methods EA^{rnd} and EA^{gre} for the small-size instances. Finally, the execution times reported by EA^{rnd} and EA^{gre} for the large instances are not significantly different.

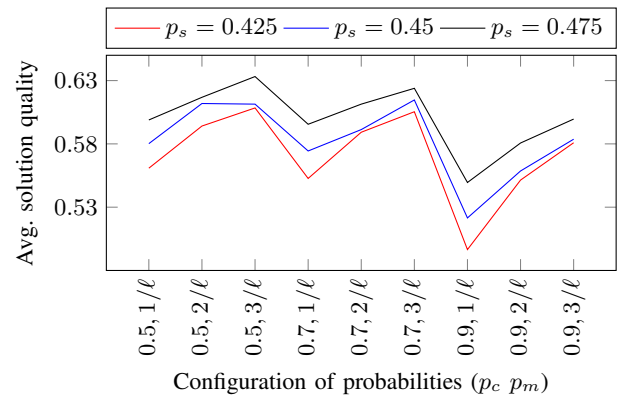
APPENDIX B PARAMETER TUNING OF EVOLUTIONARY ALGORITHM

In this appendix we discuss how the initial parameters for the proposed algorithms were selected.

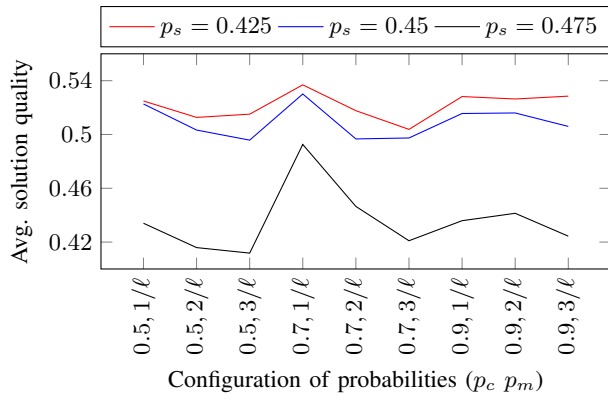
Initial experiments aimed at identifying suitable parameter settings for the proposed EA. We considered three possible values for the recombination, mutation, and selection parameters, whereas the population size and number of generations were fixed to 100 and 300, respectively. The values explored for the mutation probability are $p_m = \{1/\ell, 2/\ell, 3/\ell\}$, where ℓ refers to the total number of encoding positions in the genotype. The values used for the recombination probability



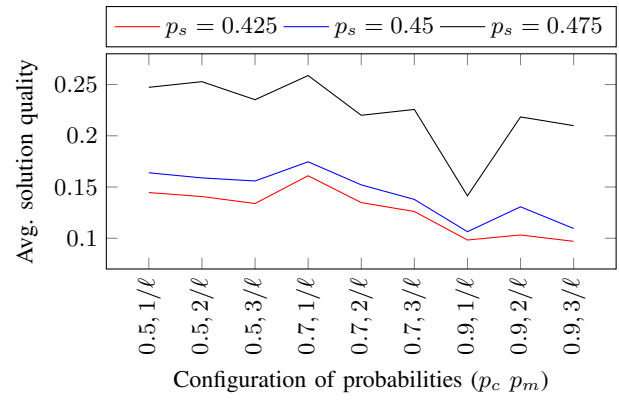
(a) Subset of small-size test instances



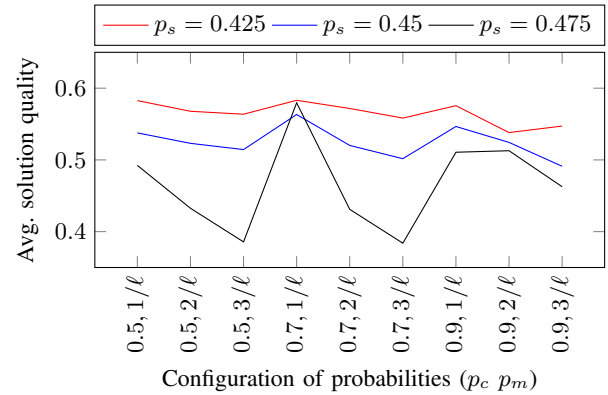
(a) Subset of small-size test instances



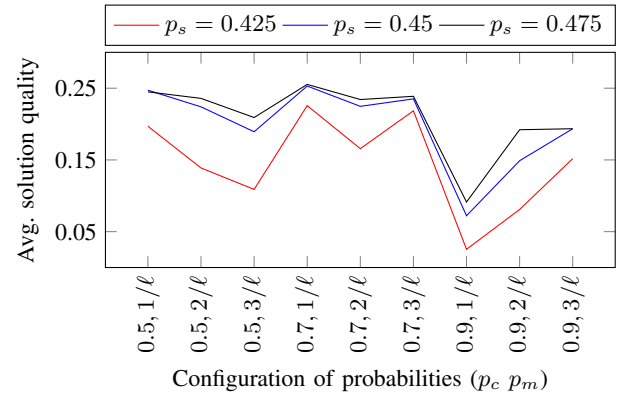
(b) Subset of medium-size test instances



(b) Subset of medium-size test instances



(c) Subset of large-size test instances



(c) Subset of large-size test instances

Fig. 14: Results of the parameter tuning of approach EA^{gre} .

Fig. 15: Results of the parameter tuning of approach EA^{rnd} .

are $p_c = \{0.5, 0.7, 0.9\}$. Finally, the settings considered for the selection mechanism based on stochastic ranking are $p_s = \{0.425, 0.45 \text{ and } 0.475\}$ corresponds to 0.425, 0.45 and 0.475. Thus, a total of 27 parameter combinations were evaluated using the genetic algorithms (EA^{rnd} and EA^{gre}). The 27 parameter combinations were evaluated with 300 test instances (100 instances randomly selected for each of the three categories described in Section V-A).

Fig. 14 summarizes the the average solution qualities obtained by EA^{gre} for each parameter configuration. It can be seen that the best result for the small-size instances is obtained using $p_c = 0.5$, $p_m = 3/l$, and $p_s = 0.425$. For

the medium-size instances, the best performance is obtained using $p_c = 0.7$, $p_m = 1/l$, and $p_s = 0.425$. Finally, the best result obtained by for the large-size instances is obtained using configuration $p_c = 0.7$, $p_m = 1/l$, and $p_s = 0.425$.

Similarly, Fig. 15 shows the results obtained by EA^{rnd} . From the figure, it is possible to identify the best parameter configurations as follows. For the small-size instances: $p_c = 0.5$, $p_m = 3/l$, and $p_s = 0.475$. For the medium-sized instances: $p_c = 0.7$, $p_m = 1/l$, $p_s = 0.475$. Finally, for the large-size instances: $p_c = 0.7$, $p_m = 1/l$, and $p_s = 0.475$.

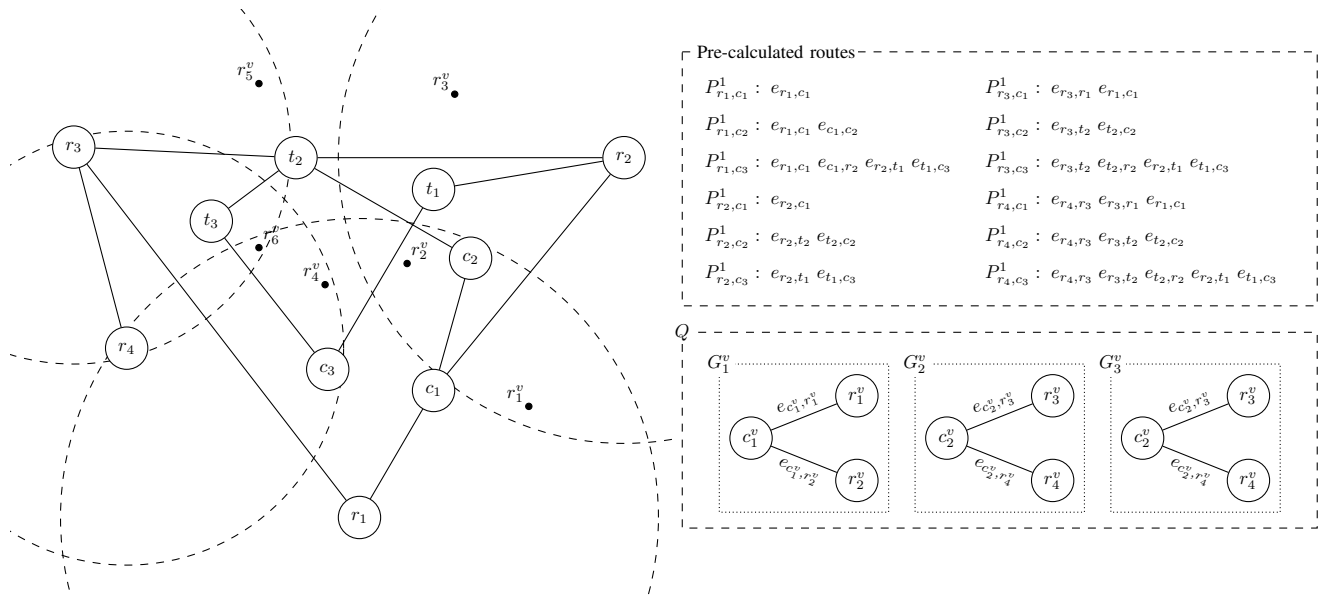


Fig. 16: Example of a small-sized test instance. The instance is made up of a physical network (left graph), a set of resource requests (Q bottom-right part) and a set of pre-calculated routes (P_{r_i, c_j}^1 upper-right part.)

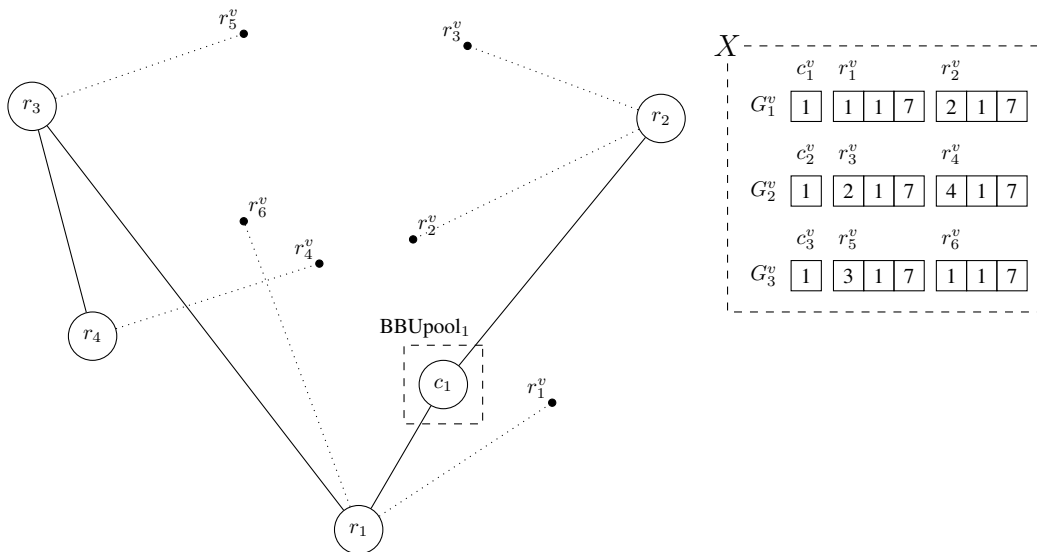


Fig. 17: The best solution found by the EA^{gre} algorithm for the small-size test instance. The unassigned elements have been removed to better highlight the assignments that were made (X). Each virtual DU has been assigned to a physical DU. All virtual CUs were assigned to a single physical CU, this being the resource pool (BBU_{pool_1}). Each virtual DU implements the PHY II functional split (f_7).

APPENDIX C

EXAMPLE TEST INSTANCE AND SOLUTION FOUND

This appendix presents an example of a test instance and a solution of our validation simulations. We use a small-size test instance, which includes the set of resources requests, a substrate network with 10 nodes, and a set of pre-defined routes. Finally, a solution found by algorithm EA^{gre} is presented.

Fig. 16 illustrates the corresponding small-sized instance. It contains a substrate network with 10 nodes and 12 links, from which four nodes were chosen as DUs, and three nodes were selected as CUs. The set of resource requests (Q) contains three resource requests (G_1^v , G_2^v , and G_3^v). Each G_i^v requires a

virtual CU and two virtual DUs. For each CU-DU pair, a route was generated ($k = 1$). A total of 12 routes were obtained for this instance. The figure shows the desired positions for all virtual DUs, as well as the coverage of each physical DU. As can be seen, virtual DUs r_1^v , r_2^v , r_4^v , and r_6^v can be assigned to more than one physical DU, while virtual DUs r_5^v and r_3^v have only one possible DU assignment. Fig. 17 shows the best solution that was obtained by algorithm EA^{gre} for this instance. The unused elements were removed from the graph to clearly observe the assignments made. It can be seen that the algorithm places all virtual CUs in a single physical CU (this CU being the only one that is part of the pool of resources).

Virtual DUs r_1^v and r_6^v were assigned to physical DU r_1 , virtual links $e_{r_1^v, c_1^v}$ and $e_{r_6^v, c_3^v}$ were assigned to path P_{r_1, c_1}^1 . Similarly, virtual DUs r_2^v and r_3^v were assigned to physical DU r_2 , and virtual links $e_{r_2^v, c_1^v}$ and $e_{r_3^v, c_2^v}$ were assigned to path P_{r_2, c_1}^1 . Virtual DUs r_4^v and r_5^v were assigned to DUs r_4 and r_3 , respectively. Virtual links $e_{r_4^v, c_2^v}$ and $e_{r_5^v, c_3^v}$ were assigned to routes P_{r_4, c_1}^1 and P_{r_3, c_1}^1 respectively. Finally, all virtual DUs were assigned the functional division PHY II (f_7).

APPENDIX D

DETAILS ABOUT SUBSTRATE NETWORK TOPOLOGY AND ITS INFLUENCE ON ALGORITHMS PERFORMANCE

This appendix gives details about the substrate network topologies and the performance of the *GRE*, *EA^{rnd}*, and *EA^{gre}* approaches with each topology. As described earlier in Table III, for the small-size instances we generated 42 substrate networks, and 56 substrate networks for the medium-size and the large-size instances. The process to generate the substrate networks with the BRITE toolkit is as follows:

- 1) A number of m possible DU nodes were selected from the convex hull. We used Graham's method to obtain the nodes that are part of the convex hull of the graph generated with BRITE. The value of m is chosen from the minimum and maximum values described in Table III.
- 2) We randomly selected n possible CU nodes that are not part of the convex hull of the graph. The value of n is chosen from the minimum and maximum values described in Table III.
- 3) Finally, we assigned parameters for bandwidth, types of links, capacities of the DU and CU nodes, etc, with the values from Table III. The parameters used for the links are described in Section V-A. The types of link can be: wired, optical, or wireless.

Using BRITE we generated four types of substrate network topologies: Barabasi-Albert, Waxman, Top-Down hierarchical, and Bottom-Up hierarchical, with the following parameters:

- Barabasi-Albert [50]: this model generates scale-free graphs. The parameter used in BRITE was $m = 2$.
- Waxman [51]: this model generates graphs with short communication links since long links tend to be more expensive in the real world. The parameters used in BRITE were $\alpha = 0.2$ and $\beta = 0.3$.
- Top-Down hierarchical: this model generates a two-level hierarchy model using Waxman or Barabasi-Albert at either of the two levels.
- Bottom-Up hierarchical: this model generates a two-level hierarchy where the second level is generated using Waxman or Barabasi-Albert and the first level is generated by randomly grouping nodes. These groups are formed with a constant number of nodes ($total_nodes/total_groups$). The total number of nodes is described in Table III for each instance size. The total number of groups was ranged from 10 to 100 for large instances, while for medium instances 10 groups were used. For small instances, this model was not used.

For small instances, we generated 17 substrate networks using the hierarchical Top-Down model, using Waxman at level 1 and Barabasi-Albert at level 2. Also, we generated 9 substrate networks using the Top-Down model using Barabasi-Albert at the two levels. A total of 10 substrate networks were also generated using the Barabasi-Albert model and 6 substrate networks using the Waxman model.

For the medium-size instances we generated 33 substrate networks using the hierarchical Bottom-Up model, using Barabasi-Albert. Also, we generated 11 substrate networks using the hierarchical Bottom-Up model using Waxman and 12 substrate networks using the Waxman model.

For large instances, we generated 27 substrate networks with the hierarchical Bottom-Up model using Barabasi-Albert. We also generated 20 substrate networks using the hierarchical Bottom-Up model using Waxman and 9 substrate networks using the Barabasi-Albert model.

The performance of the *GRE*, *EA^{rnd}*, and *EA^{gre}* approaches in terms of (a) average solution quality, (b) best solution quality, (c) average acceptance rate, and (d) execution time, is shown in Fig. 18. For the small-size instances (Fig. 18, left column) the Waxman topologies are more suitable for the average solution quality, whereas the best solution quality is achieved with the hierarchical Top-Down topology for the *EA^{gre}* algorithm. The highest acceptance rates are obtained with the hierarchical Top-Down topologies. Finally, the execution time seems to be agnostic to substrate network topology for the small-size instances.

For the medium-size instances (Fig. 18, center column), the highest values for average solution quality are obtained with the Waxman model. For best solution quality the best results are achieved with the algorithm *EA^{gre}*. The higher average acceptance rates are achieved with the hierarchical Bottom-Up model using Waxman. The execution time of the algorithms is similar for all topologies of the medium-size instances.

For the large-size instances (Fig. 18, right column), the results for average solution quality, best solution quality, and execution time are similar for every topology considered. The algorithm *EA^{rnd}* obtains the best solution quality, and the average acceptance rate is higher in the hierarchical Bottom-Up model using Waxman topologies.

Overall, the highest values for the average solution quality, the best solution quality, and the average acceptance rates are obtained in hierarchical topologies. Regarding the similarity in the results of the execution time by each algorithm in the different topologies, this behavior is achieved because we have pre-calculated the k possible routes between the CU and DU nodes, with the intention to let the algorithms focus on exploring the search space to identify the choices that must be made to satisfy the requests of the vMNOs.

ACKNOWLEDGEMENTS

This paper has been partially supported by the Secretary of Public Education of Mexico (SEP) and Cinvestav through research grant 262, and the National Council of Research and Technology (CONACYT) through grant ERANetLAC-FONCICYT No. 272278. Luis Diez and Ramón Agüero

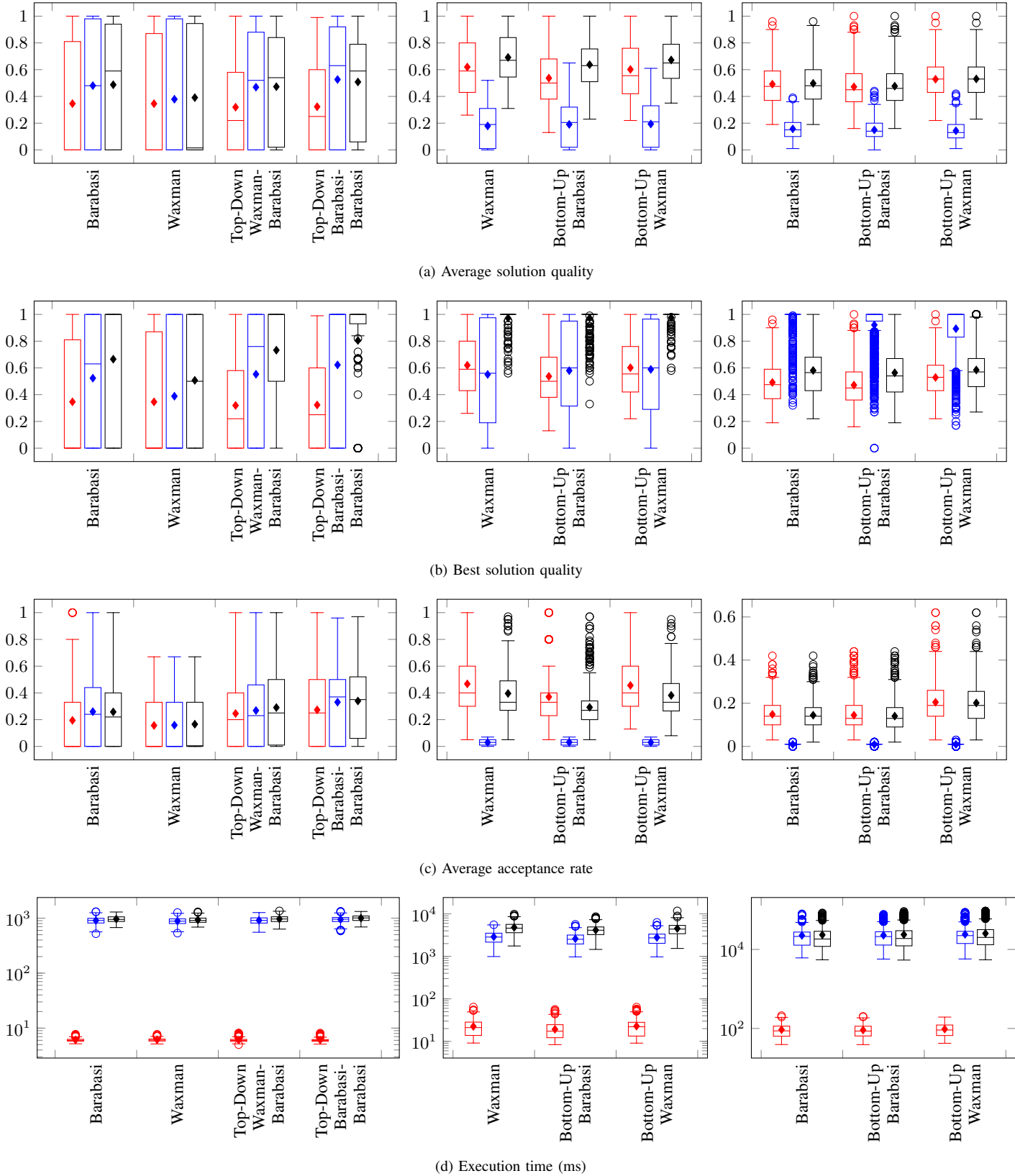


Fig. 18: Summary of the results obtained by approaches GRE (red), EA^{rnd} (blue), and EA^{gre} (black) grouped by topology. Results are shown for small-size (left), medium-size (center), and large-size instances (right) and are evaluated in terms of average quality of the solution (a), best solution quality (b), average acceptance rate (c), and execution time (d).

acknowledge the funding by the Spanish Government (Ministerio de Economía y Competitividad, Fondo Europeo de Desarrollo Regional, MINECO-FEDER) by means of the project FIERCE: Future Internet Enabled Resilient smart CitiEs (RTI2018-093475-AI00).

REFERENCES

- [1] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud ran for mobile networks—a technology overview," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, pp. 405–426, Firstquarter 2015.
- [2] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, "Cloud radio access network (c-ran): a primer," *IEEE Network*, vol. 29, no. 1, pp. 35–41, Jan 2015.
- [3] G. O. Pérez, J. A. Hernández, and D. Larrabeiti, "Fronthaul network modeling and dimensioning meeting ultra-low latency requirements for 5g," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 6, pp. 573–581, June 2018.
- [4] C. I. Y. Yuan, J. Huang, S. Ma, C. Cui, and R. Duan, "Rethink fronthaul for soft ran," *IEEE Communications Magazine*, vol. 53, no. 9, pp. 82–88, Sep. 2015.
- [5] L. M. P. Larsen, A. Checko, and H. L. Christiansen, "A survey of the functional splits proposed for 5g mobile crosshaul networks," *IEEE Communications Surveys Tutorials*, vol. 21, no. 1, pp. 146–172, Firstquarter 2019.
- [6] P. Arnold, N. Bayer, J. Belschner, and G. Zimmermann, "5g radio access network architecture based on flexible functional control / user plane splits," in *2017 European Conference on Networks and Communications (EuCNC)*, June 2017, pp. 1–5.
- [7] D. Harutyunyan and R. Riggio, "Flex5g: Flexible functional split in 5g networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 961–975, Sep. 2018.
- [8] —, "Flexible functional split in 5g networksflexible functional split in 5g networks," in *2017 13th International Conference on Network and Service Management (CNSM)*, Nov 2017, pp. 1–9.
- [9] 3rd Generation Partnership Project (3GPP), "Study on new radio access technology: Radio access architecture and interfaces," 2017.
- [10] —, "Technical specification group radio access network; study on cdu lower layer split for nr," 2017.
- [11] I. W. Group, "Next generation fronthaul interface." [Online]. Available: <http://sites.ieee.org/sagroups-1914/>
- [12] A. Garcia-Saavedra, J. X. Salvat, X. Li, and X. Costa-Perez, "Wizhaul: On the centralization degree of cloud ran next generation fronthaul," *IEEE Transactions on Mobile Computing*, vol. 17, no. 10, pp. 2452–2466, 2018.
- [13] A. Martínez Alba and W. Kellerer, "A dynamic functional split in 5g radio access networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.
- [14] L. Wang and S. Zhou, "Flexible functional split in c-ran with renewable energy powered remote radio units," in *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2018, pp. 1–6.
- [15] D. A. Temesgene, M. Miozzo, and P. Dini, "Dynamic functional split selection in energy harvesting virtual small cells using temporal difference learning," in *2018 IEEE 29th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, Sep. 2018, pp. 1813–1819.
- [16] L. Wang and S. Zhou, "Energy-efficient uav deployment with flexible functional split selection," in *2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, 2018, pp. 1–5.
- [17] A. Sriram, M. Masoudi, A. Alabbasi, and C. Cavdar, "Joint functional splitting and content placement for green hybrid cran," in *2019 IEEE 30th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2019, pp. 1–7.
- [18] A. Marotta, D. Cassioli, K. Kondepu, C. Antonelli, and L. Valcarenghi, "Exploiting split in converged software defined access networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 11, no. 11, pp. 536–546, 2019.
- [19] —, "Efficient management of flexible functional split through software defined 5g converged access," in *2018 IEEE International Conference on Communications (ICC)*, May 2018, pp. 1–6.
- [20] Y. Li, J. Mårtensson, M. Fiorani, B. Skubic, Z. Ghebretensae, Y. Zhao, J. Zhang, L. Wosinska, and P. Monti, "Flexible ran: A radio access network concept with flexible functional splits and a programmable optical transport," in *2017 European Conference on Optical Communication (ECOC)*, Sep. 2017, pp. 1–3.
- [21] B. Ojaghi, F. Adelantado, E. Kartsakli, A. Antonopoulos, and C. Verikoukis, "Sliced-ran: Joint slicing and functional split in future 5g radio access networks," in *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.
- [22] A. Alabbasi, M. Berg, and C. Cavdar, "Delay constrained hybrid cran: A functional split optimization framework," in *2018 IEEE Globecom Workshops (GC Wkshps)*, 2018, pp. 1–7.
- [23] Y. Zhou, J. Li, Y. Shi, and V. W. S. Wong, "Flexible functional split design for downlink c-ran with capacity-constrained fronthaul," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 6050–6063, 2019.
- [24] C. Chang, N. Nikaiein, R. Knopp, T. Spyropoulos, and S. S. Kumar, "Flexcran: A flexible functional split framework over ethernet fronthaul in cloud-ran," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–7.
- [25] H. Rastegarfar, T. Svensson, and N. Peyghambarian, "Optical layer routing influence on software-defined c-ran survivability," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 11, pp. 866–877, 2018.
- [26] Y. Nakayama, D. Hisano, T. Kubo, Y. Fukada, J. Terada, and A. Otaka, "Low-latency routing scheme for a fronthaul bridged network," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 1, pp. 14–23, 2018.
- [27] Y. Nakayama, D. Hisano, T. Kubo, T. Shimizu, H. Nakamura, J. Terada, and A. Otaka, "Low-latency routing for fronthaul network: A monte carlo machine learning approach," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
- [28] N. Patriciello, J. Núñez-Martínez, J. Baranda, M. Casoni, and J. Mangués-Bafalluy, "Towards backpressure routing in wireless mesh backhalls for dense lte deployments," in *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2019, pp. 282–287.
- [29] J. Núñez-Martínez, J. Baranda, and J. Mangués-Bafalluy, "Backpressure routing for the backhaul in sparse small cell deployments," in *2013 IEEE 32nd International Performance Computing and Communications Conference (IPCCC)*, 2013, pp. 1–2.
- [30] H. Liu, S. Hao, and J. Li, "Routing and heuristic scheduling algorithm for millimeter wave wireless backhaul networks," in *2017 IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2017, pp. 300–304.
- [31] Honglei Miao and M. Faerber, "Self-organized multi-hop millimeter-wave backhaul network: Beam alignment and dynamic routing," in *2015 European Conference on Networks and Communications (EuCNC)*, 2015, pp. 275–279.
- [32] O. I. Abdullaziz, M. Capitani, C. E. Casetti, C. F. Chiasserini, S. B. Chundrigar, G. Landi, X. Li, F. Moscatelli, K. Sakaguchi, and S. T. Talat, "Energy monitoring and management in 5g integrated fronthaul and backhaul," in *2017 European Conference on Networks and Communications (EuCNC)*, 2017, pp. 1–6.
- [33] S. S. Tadesse, C. Casetti, C. F. Chiasserini, and G. Landi, "Energy-efficient traffic allocation in cdn-basec backhaul networks: Theory and implementation," in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2017, pp. 209–215.
- [34] X. Li, R. Ferdous, C. F. Chiasserini, C. E. Casetti, F. Moscatelli, G. Landi, R. Casellas, K. Sakaguchi, S. B. Chundrigar, R. Vilalta, J. Mangués, A. Garcia-Saavedra, X. Costa-Pérez, L. Goratti, and D. Siracusa, "Novel resource and energy management for 5g integrated backhaul/fronthaul (5g-crosshaul)," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, 2017, pp. 778–784.
- [35] A. Alameer and A. Sezgin, "Optimization framework for baseband functionality splitting in c-ran," in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2017, pp. 1–5.
- [36] I. Koutsopoulos, "Optimal functional split selection and scheduling policies in 5g radio access networks," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*, May 2017, pp. 993–998.
- [37] A. Garcia-Saavedra, G. Iosifidis, X. Costa-Perez, and D. J. Leith, "Joint optimization of edge computing architectures and radio access networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 11, pp. 2433–2443, 2018.

- [38] B. M. Khorsandi, F. Tonini, E. Amato, and C. Raffaelli, "Dedicated path protection for reliable network slice embedding based on functional splitting," in *2019 21st International Conference on Transparent Optical Networks (ICTON)*, 2019, pp. 1–4.
- [39] K. A. De Jong, *Evolutionary Computation: A Unified Approach*. Cambridge, MA, USA: MIT Press, 2016.
- [40] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*, 2nd ed. Berlin, Heidelberg: Springer, 2015.
- [41] T. Bartz-Beielstein, J. Branke, J. Mehnen, and O. Mersmann, "Evolutionary algorithms," *WIREs Data Mining and Knowledge Discovery*, vol. 4, no. 3, pp. 178–195, 2014. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1124>
- [42] F. Rothlauf, *Representations for Genetic and Evolutionary Algorithms*, 2nd ed. Berlin, Heidelberg: Springer, 2006.
- [43] P. A. Whigham, G. Dick, and J. Maclaurin, "On the Mapping of Genotype to Phenotype in Evolutionary Algorithms," *Genetic Programming and Evolvable Machines*, vol. 18, no. 3, pp. 353–361, Sep. 2017. [Online]. Available: <https://doi.org/10.1007/s10710-017-9288-x>
- [44] R. J. Collins and D. R. Jefferson, "Selection in Massively Parallel Genetic Algorithms," in *Proceedings of the 4th International Conference on Genetic Algorithms, San Diego, CA, USA, July 1991*, R. K. Belew and L. B. Booker, Eds. Morgan Kaufmann, 1991, pp. 249–256.
- [45] M. Črepinšek, S.-H. Liu, and M. Mernik, "Exploration and Exploitation in Evolutionary Algorithms: A Survey," *ACM Computing Surveys*, vol. 45, no. 3, pp. 35:1–35:33, Jul. 2013.
- [46] G. Syswerda, "Uniform Crossover in Genetic Algorithms," in *International Conference on Genetic Algorithms*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1989, pp. 2–9.
- [47] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, 1st ed. USA: Addison-Wesley Longman Publishing Co., Inc., 1989.
- [48] T. Runarsson and X. Yao, "Stochastic Ranking for Constrained Evolutionary Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 3, pp. 284–294, 2000.
- [49] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2021. [Online]. Available: <http://www.gurobi.com>
- [50] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.
- [51] B. M. Waxman, "Routing of multipoint connections," *IEEE journal on selected areas in communications*, vol. 6, no. 9, pp. 1617–1622, 1988.



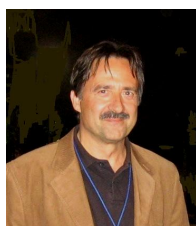
Ramón Agüero Calvo (S'07, M'09, SM'15) received his MSc in Telecommunications Engineering (1st class honors) from the University of Cantabria in 2001 and the PhD (Hons) in 2008. He is currently an Associate Professor at the Communications Engineering Department in that university. His research focuses on future network architectures, especially regarding the (wireless) access part of the network and its management. He is also interested on multi-hop (mesh) networks, and Network Coding. He has published more than 200 scientific papers in such areas and he is a regular TPC member and reviewer on various related conferences and journals. Ramon Agüero serves in the Editorial Board of IEEE Communication Letters (Senior Editor since 2019), IEEE Open Access Journal of the Communications Society, Wireless Networks (Springer), Mobile Information Systems (Hindawi). Dr. Agüero has supervised 5 PhDs and more than 70 BSc and MSc thesis. He is the main instructor in courses dealing with Networks, and Traffic Modeling, both at BSc and MSc levels. Since 2016, he is the Head of the IT Area (deputy CIO) at the University of Cantabria.



Luis Diez received his M.Sc. and Ph.D. from University of Cantabria in 2013 and 2018 respectively. He is currently Assistant Professor at the Communications Engineering Department in that University. He has been involved in different international and industrial research projects. His research focuses on future network architectures, resource management in wireless heterogeneous networks, and IoT solutions and services. He has published more than 40 scientific and technical papers in those areas, and he has served as TPC member and reviewer in a number of international conferences and journals. As for teaching, Dr. Diez has supervised 15 BSc and MSc Thesis, and he teaches in courses related to cellular networks, network dimensioning and service management.



Cristian C. Erazo-Agrede was born in 1990 in Colombia, he received a degree in systems engineering from the Universidad del Cauca, Popayán, in 2017. He received a M.Sc. degree at the Center for Research and Advanced Studies of the National Polytechnic Institute of Mexico, where he is currently pursuing the Ph.D. in engineering and computational technologies. His research interests include computational intelligence, multi-objective optimization and network virtualization.



Joan Serrat received the degree of Telecommunication Engineer in 1977, and the Doctor degree in Telecommunication Engineering in 1983, both from Universitat Politècnica de Catalunya -UPC-. Currently he is Full Professor at UPC where he has been involved in several collaborative projects with different European research groups, both through bilateral agreements or through participation in European funded projects. His topics of interest are in the field of autonomic networking and service and network management.



Mario Garza-Fabre received the M.Sc. and Ph.D. degrees in Computer Science from the Center for Research and Advanced Studies (Cinvestav), Mexico, in 2009 and 2014, respectively. From 2015 to 2018, he worked as a Research Associate/Fellow at the University of Manchester and Liverpool John Moores University, United Kingdom. Since 2018, he joined Cinvestav, where he is currently an Associate Professor in Optimization and Computational Intelligence. His main research interests involve the analysis and design of (meta-)heuristic optimization techniques, as well as their application to problems from areas such as bioinformatics, data mining/machine learning, and transportation.



Javier Rubio-Loyola received the Engineering degree in communications and electronics and the M.Sc. degree in digital systems from the National Polytechnic Institute of Mexico and the Ph.D. degree in telecommunications from the Universitat Politècnica de Catalunya, Barcelona. He is a Research Scientist with the Center for Research and Advanced Studies, Mexico. He has participated in a number of Spanish and IST-European research projects mainly in the network management area. His research interests focus on network management, network functions virtualization, and software-defined networks.