



UNIVERSITÀ
DEGLI STUDI
FIRENZE

FLORE

Repository istituzionale dell'Università degli Studi di Firenze

VNF placement for service chaining in a distributed cloud environment with multiple stakeholders

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:

VNF placement for service chaining in a distributed cloud environment with multiple stakeholders /
Cappanera, Paola; Paganelli, Federica; Paradiso, Francesca. - In: COMPUTER COMMUNICATIONS. - ISSN
0140-3664. - STAMPA. - 133:(2019), pp. 24-40. [10.1016/j.comcom.2018.10.008]

Availability:

This version is available at: 2158/1139828 since: 2019-03-19T13:04:01Z

Published version:

DOI: 10.1016/j.comcom.2018.10.008

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

Publisher copyright claim:

Conformità alle politiche dell'editore / Compliance to publisher's policies

Questa versione della pubblicazione è conforme a quanto richiesto dalle politiche dell'editore in materia di copyright.

This version of the publication conforms to the publisher's copyright policies.

(Article begins on next page)

VNF Placement for service chaining in a distributed cloud environment with multiple stakeholders

Paola Cappanera^a, Federica Paganelli^{b,*}, Francesca Paradiso^a

^a*Department of Information Engineering, University of Florence, Italy*

^b*CNIT, Research Unit at the University of Florence, Italy*

Abstract

The adoption of virtualization technologies in networking is promoting a radical innovation in the way network services are managed and delivered. Indeed, some network services may be provisioned to cope with complex and unpredictable traffic demands by dynamically creating a sequence of Virtual Network Functions (VNFs) and steering traffic flows through them. In this context, the optimized deployment of network services, composed of VNFs that may be instantiated in multiple Data Centers (DCs), is one of the most challenging orchestration target. VNF placement is the problem of choosing the set of optimal locations for a chain of VNFs according to the service request and the current characteristics of available computing resources and network links. With respect to the state of the art, our original contribution reflects a multi-stakeholder perspective (subscriber, service providers, infrastructure providers) in a multi-DC environment. We thus consider the problem of placing VNFs to maximize primarily the number of accepted requests from a set of incoming requests and secondarily the satisfaction of subscribers' preferences. Our model also allows to differentiate service requests in priority levels and guarantees that Quality of Service objectives for accepted service requests are fulfilled, including also a requirement on network service instantiation time. We provide an integer linear programming formulation of this problem that leverages a layered auxiliary graph built for each request in a set. Experimental evaluation is described in detail and

*Corresponding author

Email addresses: paola.cappanera@unifi.it (Paola Cappanera),
federica.paganelli@cnit.it (Federica Paganelli), francesca.paradiso@unifi.it
(Francesca Paradiso)

an assessment of the proposed placement approach is performed along three main directions: (i) service acceptance ratio in online and offline placement, (ii) preferences' satisfaction, and (iii) scalability expressed in terms of computational time. The performance of the approach is also compared to a greedy heuristic.

Keywords: Network Function Virtualization, Network Service, Service Chaining, VNF Placement, NFV Orchestration, Optimization Techniques

1. Introduction

Network Function Virtualization (NFV) is a paradigm proposed by the European Telecommunication Standardization Institute (ETSI)[1] to facilitate dynamic provisioning of network services through virtualization technologies. In this vision, network services can be implemented by chaining a set of functions, implemented either on dedicated hardware as Physical Network Functions (PNFs), or as software components on top of virtualized general-purpose hardware, i.e., Virtual Network Functions (VNFs). The adoption of virtualization allows flexible lifecycle management of network services as well as of their VNF components (e.g., creation, deletion, horizontal or vertical scaling operations). In this way, resource usage can be adapted to current demand and business targets, also avoiding the adoption of over-provisioning policies [2].

Software-Defined Networking (SDN) [3] complements NFV by offering programmatic access to abstracted network resources and full programmability of forwarding capabilities. Indeed, SDN control capabilities may be used to implement dynamic traffic steering policies so that flows are dynamically routed along a path traversing the VNF instances composing a given network service [4].

NFV and SDN technologies together introduce a level of flexibility in network service provisioning that is key for coping with requirements of complex and unpredictable traffic patterns in modern networking systems, such as Internet of Things, cloud networking and mobile data traffic toward new fifth generation (5G) networks [5, 6]. Indeed, NFV and SDN are jointly considered key technologies for supporting the degree of flexibility required by network slicing techniques in future 5G networks [7] as well as dynamic demand for low latency applications (Mobile Edge Computing [8]).

28 In this context, appropriate orchestration mechanisms are required to
29 support such operational flexibility and make services more responsive to cus-
30 tomer needs, while guaranteeing the achievement of target operating margins
31 [9]. Therefore, orchestration mechanisms should account for both business
32 value and customer experience, which can be represented as two conflicting
33 goals, respectively: i) cost-effective resource utilization, to achieve the tar-
34 get range of operating margins (business performance); and ii) fulfillment
35 of Quality of Service (QoS) objectives [10] specified in the Service Level
36 Agreement (SLA) between a customer and a service provider and typically
37 expressed as technical performance metrics.

38 In this scenario, the optimized deployment of network services, com-
39 posed of VNFs that may be instantiated in multiple distributed Data Centers
40 (DCs), is one of the most challenging orchestration target [11].

41 VNF placement is the problem of choosing the set of optimal locations
42 for chained VNF instances according to the current characteristics of avail-
43 able computing resources and network links. Optimality has been defined in
44 different ways in the literature (e.g., minimization of the overall delay or of
45 deployments costs, maximization of remaining bandwidth, etc.).

46 However, a broader perspective on VNF placement in a distributed multi-
47 DC environment, which also considers the needs of stakeholders, may help in
48 eliciting novel criteria to be taken into account. Indeed, network operators
49 are facing the problem of orchestrating resources so to profitably run VNFs,
50 i.e., efficiently managing capital and operational expenditures (CAPEX and
51 OPEX, respectively), while fulfilling SLAs agreed with subscribers [9]. The
52 industrial research community [12] is also arguing whether there is a real
53 benefit in minimizing SLA objectives, such as latency. Indeed, satisfiability
54 seems to be more important than optimization in this context and admis-
55 sion control techniques are usually employed to determine whether latency
56 targets can be met. As a consequence, the industrial community is looking
57 for more pragmatic approaches, such as decision policies aiming at maintain-
58 ing technical performance objectives within an acceptable range [9], while
59 maximizing request acceptance rate [11, 13].

60 Hence, while most recent works focus on optimizing technical performance
61 objectives (e.g., end-to-end delay and remaining bandwidth)[14–16] or cost
62 minimization [17–22] in a joint VNF placement and routing problem, in this
63 work we analyze the VNF placement problem and related orchestration scen-
64 ario from a business perspective. The aim is to derive stakeholders’ main
65 requirements and specify the problem statement and optimization objectives

66 accordingly.

67 We consider three types of stakeholders: *subscribers*, asking for the pro-
68 vision of network services, *service providers*, providing network services exe-
69 cuted on top of virtual and physical infrastructure resources, and *infrastructure*
70 *providers*, providing and managing virtual and physical infrastructures.
71 We elaborate on their needs and their mutual interactions within a refer-
72 ence NFV/SDN architecture based on current standards [23, 24]. We then
73 formulate our VNF Placement problem that reflects such multi-stakeholder
74 perspective, including possible constraints on the extent to which detailed
75 information about the infrastructure status is shared among stakeholders.
76 We thus consider the problem of placing VNFs to maximize the number of
77 accepted requests and subscribers’ preferences, while delegating the possible
78 optimization of technical performance objectives, such as latency or conges-
79 tion minimization, to intra-domain orchestration mechanisms (e.g. online
80 traffic engineering techniques implemented on top of SDN Controller North
81 Bound interfaces [25]). We formulate the problem by means of a 0-1 Inte-
82 ger Linear Programming model. Our model allows to differentiate service
83 requests in priority levels and guarantees that customized QoS objectives
84 for accepted service requests are fulfilled, including also a requirement on
85 network service instantiation time.

86 *Subscribers* can also express preferences and bans over infrastructure sites
87 (i.e., DCs), so that placement decisions may take into account personal or or-
88 ganization values and concerns (e.g., sustainability, ethics, reputation, etc.).
89 In order to cope with the elements discussed above, we model the infrastruc-
90 tural resource substrate by introducing features not considered in previous
91 works, including available virtualization technology, such as Virtual Machines
92 (VMs) vs containers, and DC’s carbon footprint. A preprocessing phase is
93 also provided that has a three-fold aim: (i) discard all those requests that
94 cannot be accomplished by the system for infeasibility reasons, (ii) define the
95 incompatibilities between a specific VNF of a given request and a DC (e.g.,
96 due to commercial or organization policies or DC’s insufficient capacity), and
97 (iii) process subscribers’ preferences so that they are taken into account in
98 the optimization model.

99 Summarizing, our work addresses the maximization of the request accep-
100 tance rate, while taking into account subscribers’ preferences, priority levels
101 and the fulfillment of QoS objectives. These issues have recently been iden-
102 tified in the literature [11, 13] as some of the more relevant criteria that need
103 to be taken into account by novel VNF placement approaches.

104 The remainder of this paper is organized as follows. Section 2 discusses
105 related work and highlights our contribution. In Section 3 we present the
106 reference scenario and state the problem. Section 4 discusses the computa-
107 tional complexity of the problem addressed and presents the optimization
108 model proposed. Section 5 describes the preprocessing phase in detail. Per-
109 formance evaluation results are reported in Section 6. Finally, Section 7
110 concludes the paper with insights for future work.

111 2. Related Work

112 The problem of how effectively deploying and managing network services
113 conceived as a chain of VNFs has raised a considerable interest in the research
114 community. The rest of this section is organized as follows. First, we briefly
115 analyze the literature on routing and placement for optimizing QoS metrics.
116 Then, we analyze works targeting minimization of costs. Finally, we focus
117 on stakeholders' perspectives, that is a crucial issue in our study. We then
118 conclude by discussing our contribution with respect to the state of the art.

119 Many works jointly address VNF placement and routing problems to opti-
120 mize specific QoS metrics, typically within a DC or in an operator's network.
121 Liu et al. [14] consider two performance metrics, i.e., end-to-end delay and
122 bandwidth consumption. They propose an integer linear program (ILP) and
123 design two heuristic algorithms, i.e., a greedy algorithm and a simulated an-
124 nealing approach. The work in [15] addresses both chain composition and
125 placement. Specifically, it proposes an under-specified structure of a com-
126 posed service that allows to dynamically modify the order of VNFs in a chain
127 and a heuristic algorithm that places service components along the shortest
128 paths. Bhamare et al. [16] formulate the problem of minimizing inter-cloud
129 traffic and response time in a multi-cloud scenario as an ILP problem and
130 propose an affinity-based allocation heuristic approach for solving it.

131 Several approaches have been proposed to minimize costs of running
132 VNFs on virtual infrastructures, while fulfilling SLAs. Bari et al. [17] pro-
133 pose an exact approach for small networks and a heuristic for larger networks
134 based on a multi-stage graph with the objective of minimizing total network
135 operational cost and resource fragmentation.

136 Mechtri et al. [18] propose both an approach based on the eigendecompo-
137 sition of adjacency matrices of the request and the infrastructure graphs, and
138 a heuristic algorithm for finding the maximum weight matching. Leivadreas

139 et al. [19] propose a set of algorithms that target minimization of provision-
140 ing costs as well as efficient resource usage. Gadre et al. [20] introduce a
141 divide-and-conquer algorithm and a heuristic aiming to minimize an overall
142 cost, assuming that the routes for the flows are a priori given and VNFs
143 in the request have instance and service costs associated. The solution in
144 Pham et al. [21] is based on a Markov approximation approach combined
145 with matching theory. A stable and efficient matching is searched for that
146 takes into account the service chain’s preference over nodes (nodes with the
147 greatest amount of available resources are preferred) as well as nodes’ prefer-
148 ences over VNFs (based on the adopted consolidation policy). More recently,
149 ASPER [22] is an automated approach for the joint scaling, placement and
150 routing of network services, whose objective is to find a minimal number of
151 constraint violations (i.e., CPU, memory and link capacity constraints) that
152 is Pareto optimal with respect to a set of secondary objectives (e.g., total
153 delay, total resource consumptions, etc.).

154 Recently, authors have begun explicitly contextualizing cost minimiza-
155 tion and efficient resource usage problems in a multi-DC setting. Liberati
156 et al. [26] propose a stochastic algorithm based on reinforcement learning
157 (RL) that maximizes an expected mapping reward, which may be configured
158 to target different objectives, such as costs minimization, load balancing or
159 maximization of the acceptance rate. Implementation cost minimization as
160 well as acceptance rate maximization are jointly addressed in [27] through
161 two approximation algorithms. Luizelli et al. [28] propose a novel fix-and-
162 optimize-based heuristic algorithm to minimize resource allocation, while
163 meeting network flow requirements and constraints and addressing scalabil-
164 ity. Wang et al. [29] address the cost-effective provision of VNF graphs in
165 inter-DC optical networks in a multidomain environment (i.e., private and
166 public domains). The problem is formulated as an ILP that models com-
167 pute and network bandwidth constraints, and minimize the cost of compute
168 resources and frequency slot usage on links. Gupta et al. [30] propose an
169 approach that aims to reduce network resource consumption for a WAN
170 interconnecting DCs by defining and placing multiple instances for each ser-
171 vice chain. Gupta et al. [31] formulate an ILP to minimize usage of network
172 resources, while evaluating four different deployment choices (e.g., hardware-
173 based middleboxes, DCs, NFV-capable network nodes, etc.). Ayoubi et al.
174 [32] consider both VNF placement and policy-aware traffic steering to max-
175 imize the number of served flows. The problem is decomposed into a master
176 problem (placement) and a subproblem (policy-aware routing of every flow

177 along the designated VNF instances). The model can be used to solve ei-
178 ther an online or an offline problem. In the former case, the set of input
179 requests is a batch of requests arrived within a time window, in the latter
180 case it represents all flow requests, known in advance. Finally, in [33] the op-
181 timal placement of VNF chains is addressed and shown to be NP-complete
182 but for very special cases. The authors also propose two polynomial time
183 algorithms that can be used to determine a feasible solution to a simplified
184 variant of the optimal VNF placement problem which occurs when the fol-
185 lowing two assumptions hold: (i) each VNF typology is hosted in one physical
186 server; (ii) each traffic flow is splittable. Both the two approaches, referred
187 to as the matrix-based algorithm and the multi-stage graph algorithm, use a
188 maximum flow algorithm as a subtool, guarantee capacity constraints at the
189 servers and bandwidth constraints on the links. Other kinds of constraints
190 on the request, such as for example those concerning latency, are disregarded.

191 Focusing on stakeholders' perspective and business requirements, Alt-
192 mann and Kashef [34] analyze cloud computing cost factors in federated
193 hybrid clouds and propose a cloud cost model. They also propose a service
194 placement optimization algorithm, which identifies the cost-minimizing ser-
195 vice placement option through exhaustive search. Recently, Naudts et al. [35]
196 consider the problem of service chain from an original perspective: indeed,
197 they aim at increasing the infrastructure providers revenue by proposing a
198 dynamic pricing algorithm where the requested substrate resources are priced
199 on the basis of historical data, current infrastructure utilization levels and
200 competitors' price.

201 While the main body of previous literature mainly addresses either the
202 optimization of performance objectives [14–16] or takes into consideration
203 the service providers' need of minimizing costs for service deployment and
204 operation [17–22], in this work we develop a new concept of VNF Placement
205 by moving from business requirements and considering the perspectives of
206 three types of stakeholders (subscribers, service providers and infrastruc-
207 ture providers) to different extents. Similarly to our work, in [35] the prob-
208 lem statement originates from the analysis of roles stakeholders play in an
209 NFV/SDN environment, but for a completely different problem. In addition,
210 our work can be seen as a complement of [35] in that it allows to represent
211 different pricing schemes for the requested substrate resources and manages
212 DC preferences on behalf of price-sensitive consumers. Analogously to [21],
213 we handle service chain's preferences over nodes, but in our case such pref-
214 erences are configurable and their weight can be customized for each service

215 request. Moreover, some works are explicitly contextualized in a multi-DC
216 setting (e.g., [16, 26–32, 34], but they do not take into account possible lim-
217 itations in information disclosure among different operators, as this work
218 does.

219 Summarizing, our work contributes to the literature in the following di-
220 rections: (i) it aims at jointly maximizing service providers’ profits in terms
221 of accepted requests and satisfaction rate of subscribers’ preferences, while
222 fulfilling SLA requirements and considering an abstracted multi-DC network
223 topology complying with possible information disclosure limitations among
224 operators; (ii) it allows taking into account different priority levels and ac-
225 commodate requests that need a fast deployment, as long as the substrate
226 network may support them depending on the virtualization technology of-
227 fered by nodes; (iii) it characterizes DC nodes in terms of their carbon foot-
228 print (we take Carbon Usage Effectiveness metric (CUE)[36] as reference
229 metric) and pricing schemes. This allows users to express optional prefer-
230 ences on DCs that implement sustainable energy policies (i.e., those showing
231 the lowest CUE values) and/or are more economically convenient. In regards
232 to sustainability, as far as we know, Khosravi et al. [37] consider a similar
233 parameter, i.e., Power Usage Effectiveness (PUE) but for a different purpose,
234 i.e., energy- and carbon-efficient placement of VMs in distributed DCs.

235 3. Problem Statement

236 We consider a reference scenario for NFV orchestration characterized by
237 the following three types of stakeholders: *Subscribers*, *Service providers*, *In-*
238 *frastructure providers*. Hereafter, we introduce the main concepts of our
239 reference scenario, in terms of stakeholders’ perspectives and reference archi-
240 tectural guidelines, and then formulate the problem.

241 3.1. Subscriber’s perspective

242 A Subscriber is an actor (also referred to as user or customer) that re-
243 quests the provisioning of a network service. We model the subscriber needs
244 in terms of both a set of QoS objectives that represent desired service per-
245 formance, and preferences regarding possible VNF deployment options (i.e.,
246 preferences over available infrastructure sites).

247 In this work we consider the following QoS parameters: maximum toler-
248 ated latency, minimum guaranteed bandwidth, and network service instanti-
249 ation time. The fulfillment of these objectives, when specified in the request,

250 is mandatory, otherwise the request cannot be satisfied. While the first two
251 objectives are quite common, the third objective concerns network service in-
252 stantiation time and becomes effective when subscriber requires that network
253 service is deployed and launched "as soon as possible". This requirement is
254 taken into account through a policy enforcing that the network service is
255 deployed on the appropriate infrastructure technology. In this work we take
256 two alternative virtualization technologies as reference, VM vs. containers.
257 Since container technologies may guarantee a shorter startup time with re-
258 spect to VMs [38], when the subscriber requests a fast service setup, the
259 orchestration maps such requirement into a specific constraint (i.e., deploy-
260 ing the network service components on containers). Although not yet widely
261 considered in the literature, the specification of a requirement on instantia-
262 tion time in VNF Placement is especially relevant if network service requests
263 have to be satisfied as soon as they arrive (such as for online service requests
264 [39]) to cope with dynamic user demands.

265 As regards preferences for VNF deployment, subscribers can specify pref-
266 erences to be taken into account by service providers in the selection of the
267 infrastructure site. Indeed, subscribers preferences typically regard pricing
268 and technical performance metrics, but can also include additional attributes,
269 such as provider reputation, ethicality and stability [40]. For instance, pref-
270 erences can also require that environmental objectives are taken into account
271 and services are provided with the smallest carbon footprint, as specified in
272 emerging green or energy-aware SLAs [41, 42].

273 3.2. *Service provider's perspective*

274 The role of service providers consists in handling network service requests.
275 They offer network services to subscribers and are therefore in charge of cor-
276 rect service provisioning and lifecycle management. Service providers can
277 buy/lease service components and infrastructure from other providers (i.e.,
278 service providers and infrastructure providers). In this case, which is intro-
279 duced by ETSI as NFVI as a Service (NFVIaaS) in [43], service providers
280 have control on services, while infrastructure operators control the infras-
281 tructure. Typically, a service provider can choose the provider infrastructure
282 domain and the site where VNFs should be placed.

283 Service providers aim at optimizing business value [9]. We mapped this
284 requirement into the maximization of accepted network service requests, with
285 respect to available infrastructure resources and a maximum accepted cost
286 for service operation. In accordance with subscribers' perspective, service

287 providers may also desire to minimize costs for service hosting on the physical
288 substrate to improve target operating margins.

289 Service providers may also assign different levels of priority to incom-
290 ing requests, depending on subscribers' profiles and application-based traffic
291 differentiation (e.g., Service Classes defined in DiffServ specifications [44]).
292 Requests which have a higher priority level will get preferential treatment
293 with respect to lower priority requests.

294 3.3. Infrastructure provider's perspective

295 These actors offer virtual and physical resource infrastructures (e.g., DC
296 providers and inter-DC Wide Area Network operators). Here, we consider
297 an infrastructure provider that manages a multi-DC infrastructure, offering
298 resources at a given price for capacity unit. Offered prices can vary from
299 DC to DC. Resource offers by infrastructure providers can also be enhanced
300 with information related to the carbon footprint of a DC. The infrastructure
301 providers' perspective is modeled in this work as the requirement of efficiently
302 using the infrastructure resources by balancing the load across multiple sites
303 to avoid overhead conditions. Within a DC, an infrastructure provider may
304 apply its own decision policies to orchestrate physical resources to optimize
305 a given utility function (e.g., minimize power consumption, maximize server
306 consolidation), but this problem is outside the scope of this work.

307 3.4. Reference architecture for network service provisioning

308 Hereafter, we briefly describe an NFV/SDN-based reference architecture
309 for network service provisioning, elaborated by taking into account standard
310 guidelines and architectural models promoted by the NFV ETSI Industry
311 Specification Group [1, 23, 24]. ETSI specifications define a set of *Manage-*
312 *ment and Orchestration (MANO) functions*, which include: i) a *Virtual In-*
313 *frastructure Manager (VIM)* responsible for managing physical, virtual and
314 software resources of related NFV Infrastructures (NFVI); ii) a *VNF Man-*
315 *ager* handling the lifecycle of VNFs; and iii) a *VNF Orchestrator (VNFO)*
316 managing the lifecycle of network services.

317 Fig. 1 shows a reference architecture for network service provisioning in a
318 multiple stakeholder and multi-DC environment. This architecture integrates
319 some ETSI functional blocks mentioned above with SDN network control ca-
320 pabilities within each DC domain and in the WAN segments interconnecting
321 the DCs. For the sake of clarity only two DCs and one WAN segment that
322 provides "on-demand connectivity services" are depicted in Fig. 1. The

323 WAN Infrastructure Manager (WIM) leverages the services provided by an
324 SDN Controller and offers a North-Bound application interface [24]. We
325 introduce two additional functional blocks: a *Service Portal* and a *Service*
326 *Orchestrator*. The Service Portal offers a Graphical User Interface (GUI) to
327 subscribers for selecting and requesting the provision of a network service
328 with a given SLA to a service provider. The Service Orchestrator is respon-
329 sible for the acceptance of service requests and for service deployment and
330 management operations. For the scope of this article, we outline two main
331 components of the Service Orchestrator: a *Service Request Manager* and
332 an ETSI-compliant NFVO. The former handles incoming service requests,
333 by mapping business-level service requests coming from the Service Portal
334 into network service instantiation requests to the NFVO. For this purpose
335 the Service Request Manager also performs decision making steps, including
336 VNF placement, which is actually the target of our work. The NFVO man-
337 ages such network service instantiation requests by handling the interaction
338 with the affected VIMs and WIM. In accordance with the placement decision
339 taken by the Service Request Manager, it generates appropriate requests for
340 instantiating the VNFs (to the VIMs) and for enforcing the appropriate for-
341 warding instructions (to VIMs and WIM) for steering traffic flows through
342 the deployed chains.

343 Fig. 1 shows how different stakeholders are involved in network service
344 provisioning. As also discussed in [11], VNF deployment and connectivity
345 decisions could be taken at a single point (Service Orchestrator), which,
346 to perform optimal decisions, needs to receive full NFVI information from NFVI
347 control and management systems. However, since service and infrastructure
348 providers can be different operators, this would require the full disclosure of
349 internal details across different administrative domains. On the contrary, the
350 NFVI provider could decide to expose only an abstracted view of resources
351 and topologies [45] and hide internal details. We therefore consider a scenario
352 where the responsibility of the Service Provider consists in deciding in which
353 DCs VNFs should be placed considering an abstracted view of the NFVI,
354 thus minimizing the type of monitoring and status information to be gath-
355 ered from VIMs and WIMs (although leading to a suboptimal decision with
356 respect to the previous case). This allows NFVI operators to hide internal
357 implementation and status details, and finetune deployment decisions within
358 their own organization domain boundaries.

359 In this work we consider a multi-domain NFV Infrastructure made by
360 a set of geographically distributed infrastructure sites of different size (e.g.,

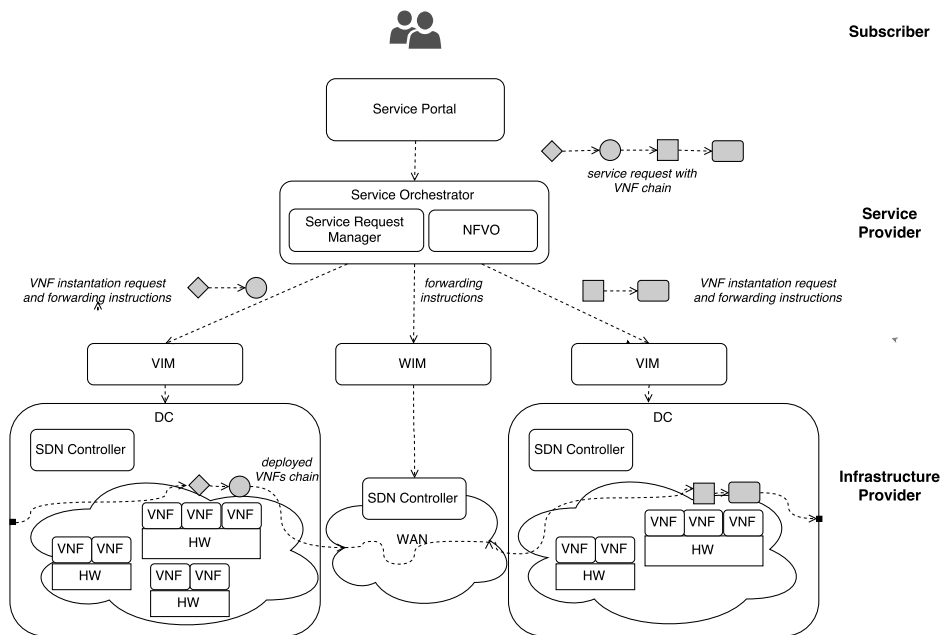


Figure 1: Reference NFV/SDN architecture

361 from micro to big DCs [46]). A DC is a container of physical hosts where
 362 one or more VNFs can be deployed. Each DC exposes its infrastructural re-
 363 sources at a given price per capacity unit and it is characterized by an energy
 364 efficiency and greenness metric (e.g., CUE) in order to promote sustainability
 365 assessments and comparisons among DCs. At a given instant in time, each
 366 DC is characterized by the amount of available resources (Capacity), such as
 367 CPU and memory. In this work, capacity is considered a multi-dimensional
 368 parameter in problem statement, while it is one-dimensional in the experi-
 369 mental testing as widely assumed in the literature ([47]). As discussed above,
 370 we also characterize DCs in terms of their technological infrastructure (e.g.,
 371 availability of container technology).

372 Incoming network service requests include the specification of a service
 373 function chain as an ordered sequence of VNFs, or more precisely, types of
 374 VNFs (e.g., NAT, firewall, etc.). We assume that the whole chain has to be
 375 instantiated preserving the order of the sequence. For each VNF type, the
 376 amount of requested resources is provided. The request is further character-
 377 ized by a *source node* (the source of the traffic flow) and a *destination node*
 378 (the destination of the traffic flow), priority levels, QoS parameters (max-

379 imum latency, minimum bandwidth and fast network service instantiation
 380 time) and maximum cost that can be afforded to deploy the service.

381 3.5. Problem formulation

382 Starting from a realistic network configuration where nodes correspond
 383 to forwarding elements or storage and compute elements in DCs, and links
 384 connect such nodes, we build an abstract network $G = (D, E)$ where nodes
 385 correspond to DCs and each arc $(i, j) \in E$ between DC i and DC j rep-
 386 resents a path in the original network between nodes i and j . Specifically,
 387 arc $(i, j) \in E$ corresponds to the path with minimum latency among all the
 388 paths connecting nodes i and j in the original network. All arcs belonging to
 389 E are bidirectional. In addition, we define T (indexed by t) as the set of pri-
 390 ority levels, R (indexed by r) as the set of service requests, and N (indexed
 391 by n) as the set of resources offered by the DCs service requests compete
 392 for. As an example, two types of requests can be considered: requests for
 393 premium services and requests coming for best effort services. In such a case,
 394 T would have cardinality two. In regards to the resources, typical resources
 395 considered in set N are CPU, RAM and storage, as an example. Sets used
 396 to state the problem formally are summarized in Table 1.

Table 1: Sets

D	set of nodes in the abstract network (each node corresponds to a DC)
E	set of arcs in the abstract network (arc (i, j) corresponds to a path from DC i to DC j)
T	set of priority levels
R	set of service requests
N	set of resources offered by DCs

397 In the following, a detailed description of network G in terms of nodes D
 398 and arcs E is given. We assume that each arc $(i, j) \in E$ is characterized by
 399 the parameters described in Table 2.

Table 2: Arc parameters

l_{ij}	latency of arc (i, j) expressed in ms
b_{ij}	available bandwidth of arc (i, j) expressed in Gbps

400 In this work latency refers to the propagation delay on the link which
 401 separates two nodes, thus it is directly dependent on the physical distance
 402 between them. Due to network abstraction, the bandwidth of an arc (i, j)

403 is the minimum bandwidth over all the links on the minimum latency path
 404 from i to j in the original network.

405 Each node i in D , i.e., each DC, is characterized by the parameters de-
 scribed in Table 3.

Table 3: Node parameters

u_i^n	capacity of i in terms of resource n
\bar{p}_i^n	upper percentage utilization of DC i relative to resource n
s_i	equal to 1 if i provides container, 0 otherwise
c_i	price of i per capacity unit
f_i	carbon footprint of i

406

407 For each DC i , the resource capacity parameter u_i^n represents its capacity
 408 in terms of resource n and \bar{p}_i^n is a parameter which defines the maximum
 409 percentage utilization of i in terms of resource n . As mentioned above, the
 410 parameter s_i refers to the capability of DC i to instantiate VNFs in a con-
 411 tainer such as Docker [48], in order to allow a quicker service provision by
 412 avoiding setup time due to VM instantiation. Finally, c_i corresponds to the
 413 unitary price exposed by DC i and f_i refers to CUE as specified above.

414 The Orchestrator has to manage a set R of service requests characterized
 415 by different typologies. Specifically, for each priority level $t \in T$, R_t is the
 416 set of requests of typology t . Sets $R_t, \forall t$ define a partition of set R , i.e.,
 417 $\cup_t R_t = R, R_{t^i} \cap R_{t^j} = \emptyset \forall t^i, t^j \in T$. Each request r in R , is characterized by
 418 the parameters described in Table 4.

419 The proposed model can be used to solve either the online or offline VNF
 420 placement problem. In the offline case, R represents the whole set of service
 421 requests, to be known in advance, whereas in an online problem, R represents
 422 a batch of requests arrived within a time window.

423 In regards to instantiation time, we point out that when a certain request
 424 r requires the instantiation time to be as short as possible (i.e., $s^r = 1$), all
 425 of the VNFs of its chain H^r must be placed on DCs equipped with con-
 426 tainer technology (if available). Preferences and incompatibilities between
 427 the VNFs in a request and DCs are computed in a pre-processing phase as
 428 detailed in Section 5.

429 We conclude the section by recalling all those features that characterize
 430 the problem studied both in terms of objective function and constraints. The
 431 problem is then mathematically formulated in Section 4.2.

Table 4: Request parameters

t	priority level
o^r	origin node of traffic request r - ingress node
d^r	destination node of traffic request r - egress node
l^r	maximum end-to-end delay tolerated by r
b_{hh+1}^r	minimum data rate capacity (bandwidth) accepted by r from the h -th VNF to the $(h+1)$ -th VNF
c^r	maximum cost r is willing to pay to get service
s^r	equal to 1 when r requires short service instantiation time; 0 otherwise
f^r	equal to 1 if r is interested in environmental impact; 0 otherwise
$H^r = \{V_1^r, V_2^r, \dots, V_{ H^r }^r\}$	ordered sequence of VNFs composing r ($ H^r $ is the length of the chain)
$u_{V_h^r}^n \quad \forall h \in \{1, \dots, H^r \}$	quantity of resource n required by the h -th VNF of r
$p_{V_h^r}^i \quad \forall h \in \{1, \dots, H^r \}$	preference expressed by request r to place its h -th VNF on DC i

432 In regards to the objective function, it is defined so as to reflect stake-
433 holders' perspectives hierarchically: service provider perspective, first, and
434 subscriber perspective, second. The service provider is interested in maxi-
435 mizing its profit which is given by the weighted sum of the served requests.
436 Specifically, the weight associated with the accomplishment of a high pri-
437 ority request is bigger than the one associated with a low priority request.
438 According to the subscribers' perspective, the placement of VNFs should be
439 done to maximize their preferences. The secondary objective then consists
440 in maximizing the overall preferences coming from all the requests.

441 In regards to the constraints that feasible solutions have to satisfy, the
442 following are considered: (i) compatibility constraints; (ii) QoS constraints;
443 (iii) service cost constraints; (iv) energy efficiency constraints; and (v) band-
444 width constraints. Specifically, compatibility constraints assure that each of
445 the VNFs composing a certain request is assigned to a node which is able
446 to satisfy its requirements in terms of resource capacity and presence of a
447 container. In addition, the order in which VNFs of a certain request are
448 performed must respect the order specified in the request. QoS constraints
449 refer to the end-to-end delay and, for each request, they have to guarantee
450 that the delay of traffic flows traversing the service, once deployed over a
451 set of nodes, is not greater than the maximum tolerated end-to-end delay.
452 Service cost constraints guarantee that the cost paid by a request, given by
453 the sum of the costs spent for the deployment of its VNFs on nodes, does not

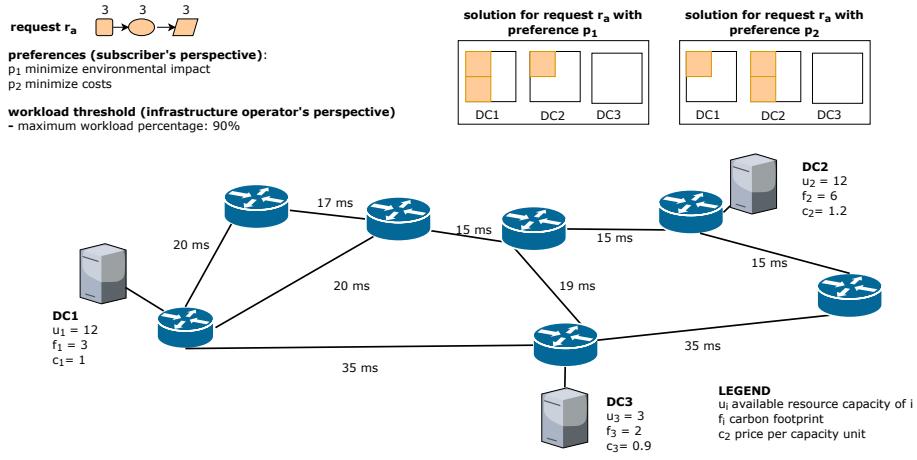


Figure 2: VNF Placement for service chaining problem: subscriber's and infrastructure provider's perspective

454 exceed the maximum cost request. A load distribution constraint guarantees
455 that the workload assigned to a DC does not exceed a given threshold, there-
456 fore allowing the infrastructure provider to enforce a load distribution policy
457 across managed sites. Finally, bandwidth constraints assure that, for each
458 link in the network, the overall bandwidth consumed by all requests using
459 that link does not exceed the bandwidth of the link.

460 Hereafter we provide two basic examples to clarify how stakeholder's per-
461 spectives are taken into account in the problem formulation. We consider
462 three DCs (DC1, DC2, DC3) geographically distributed and interconnected
463 via a WAN. The DCs offer a capacity of 20, 20 and 24 units, respectively.

464 Fig. 2 shows how subscribers' and infrastructure providers' perspectives
465 are taken into account in the placement decision process for a request r_a
466 made by three VNFs, each requiring 3 CPUs. The infrastructure provider
467 may define a threshold proportional to available capacity u_i to avoid overload
468 conditions (e.g., 90%). This implies that DC3 cannot be used. Subscribers
469 may express preferences for cost and/or carbon footprint reduction. If only
470 cost minimization is provided as preference, two VNFs will be placed in DC1
471 and one VNF in DC2. If carbon footprint is considered, one VNFs will be
472 placed in DC1 and two in DC2.

473 Finally, Fig. 3 shows an example of the abstracted network view that
474 the service provider has available for placement decisions. This view is built
475 using monitoring data provided by infrastructure operators willing to hide

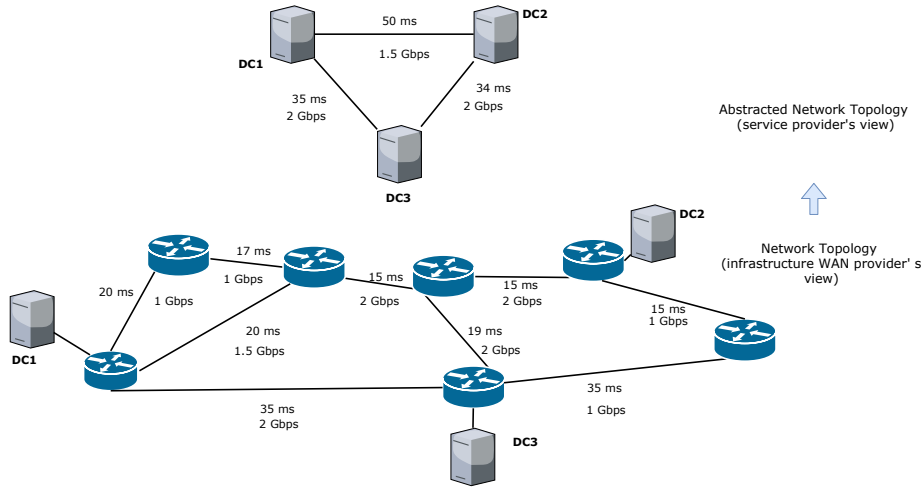


Figure 3: Example of abstracted network view: service provider's perspective

476 internal topology details. The example uses the bandwidth and latency defini-
 477 tions provided at the beginning of this section. An infrastructure operator
 478 can, of course, adopt different abstract latency and bandwidth definitions
 479 and is supposed to periodically provide the service provider with up-to-date
 480 monitoring data views.

481 4. Optimization Model

482 4.1. Computational complexity

483 This section analyzes the computational complexity of the problem ad-
 484 dressed. We start showing that even the special case of the problem studied
 485 in which requests consist of only one VNF, each request is compatible with
 486 every DC, and bandwidth is disregarded is strongly NP-hard. Indeed, this
 487 fact is due to a reduction from a knapsack-like problem, as described in the
 488 following.

489 **Theorem 1.** *Consider the special case of optimally deploying VNFs to serve*
 490 *a set of requests, each of which consisting of only one VNF, in a multi-DC*
 491 *NFV infrastructure where bandwidth is assumed to be sufficient to manage*
 492 *all of the requests at the same time and the capacity of each DC is a one-*
 493 *dimensional parameter. Let P denote this problem. Then, P is strongly*
 494 *NP-hard.*

Proof 1. Suppose that an instance of the 0-1 Multiple knapsack problem (MKP) is given. MKP is defined as follows. Given a set R of items with cardinality r , and a set D of knapsacks with cardinality d ($d \leq r$) with p_j equal to the profit of item j , w_j equal to the weight of item j , and c_i equal to the capacity of knapsack i , MKP consists in selecting d disjoint subsets of items so that the total profit of the selected items is a maximum, and each subset can be assigned to a different knapsack whose capacity is sufficient to contain the total weight of the items in the subset, computed as the sum of the weight of the items in the subset. More formally [49], MKP is:

$$\max \sum_{i=1}^d \sum_{j=1}^r p_j x_{ij} \quad (1)$$

$$\sum_{j=1}^r w_j x_{ij} \leq c_i \quad \forall i \in D \quad (2)$$

$$\sum_{i=1}^d x_{ij} \leq 1 \quad \forall j \in R \quad (3)$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in D, \forall j \in R \quad (4)$$

495 where x_{ij} is equal to one if item j is inserted in knapsack i and zero otherwise.
 496 As is usual in knapsack-related problems, it is assumed that (i) the coefficients
 497 w_j , p_j , and c_i are positive integers, (ii) $w_j \leq \max_{i \in D} c_i, \forall j \in R$, (iii) $c_i \geq$
 498 $\min_{j \in R} w_j \forall i \in D$, and (iv) $\sum_{j=1}^r w_j > c_i \forall i \in D$. Observe that non integer
 499 coefficients can be handled by multiplying them by a proper factor; all the
 500 items with a non positive profit or violating condition (ii) can be eliminated;
 501 all the knapsacks with a non positive capacity or violating condition (iii) can
 502 be eliminated. In addition, if there exists a knapsack with a capacity sufficient
 503 to contain all the items, i.e., a knapsack violating condition (iv), problem P
 504 admits the optimal trivial solution in which all the items are assigned to that
 505 knapsack. Finally, observe that if $d > r$ then the $(d - r)$ knapsacks with
 506 smallest capacity can be eliminated.

507 Now, suppose that an instance of MKP is given; we build an instance of
 508 P as follows. Each DC is associated with a knapsack, and each request is
 509 associated with an item. The priority level of a request is set to the profit of
 510 the corresponding item, the quantity of resource request j asks for is set to
 511 the weight of the corresponding item and the capacity of a DC is set to the
 512 capacity of the corresponding knapsack. From the optimal solution to P , we

513 *can obtain the optimal solution to MKP.*

514 Observe that when the weight (and/or the profit) of an item depends on the
515 knapsack in which it is inserted, MKP results in the *Generalized Assignment*
516 *Problem (GAP)* that is NP-hard in the strong sense too [50]. In our case,
517 the definition of a weight w_{ij} depending on the item j and on the knapsack i
518 allows to manage the compatibility between requests and DCs. If request j is
519 compatible with DC i , the weight reflects the quantity of resource required by
520 the unique VNF in request; otherwise, when there is incompatibility between
521 DC i and request j , the weight of item j is defined as greater than the capacity
522 of the knapsack i thus interdicting the assignment of j to i . In our problem,
523 a request j is compatible with DC i when (i) the latency of the path going
524 from the origin of the request to DC i and from DC i to the destination of
525 the request is not greater than the maximum end-to-end delay tolerated by
526 j ; (ii) the cost of assigning the VNF in the request to DC i is not smaller
527 than the maximum cost request is willing to pay to get service, and (iii)
528 DC j is able to satisfy the requirement of request i in terms of container
529 virtualization technology. All these constraints can be managed by properly
530 defining weight coefficients.

531 In summary, problem P is a special case of the problem addressed in this
532 study and it is a MKP when each request can be accommodated by every DC
533 or a GAP when incompatibility constraints between requests and DCs exist.
534 Both MKP and GAP are NP-hard in the strong sense and, according to [51],
535 this facts excludes the existence of a fully polynomial-time approximation
536 scheme for them.

537 In the more general setting, the problem of optimally deploying VNFs
538 on DCs to serve a set of requests in a multi-DC NFV infrastructure, con-
539 sists in selecting the subsets of requests providing the maximum profit that
540 can be accomplished by network resources. For each accepted request, the
541 problem asks to find a (constrained) path connecting the origin node of the
542 request with its destination node while satisfying global capacity constraints
543 at DC nodes. The problem is thus a Maximum Integral k-multicommodity
544 flow problem which is shown [52] to be APX-complete when the underlying
545 network is a tree and paths are not constrained.

546 These results motivate us to formulate the problem as an ILP.

547 *4.2. The mathematical model*

548 This section describes the mathematical model used to formulate the
549 problem of optimally deploying VNFs on DCs to serve a set of requests in a

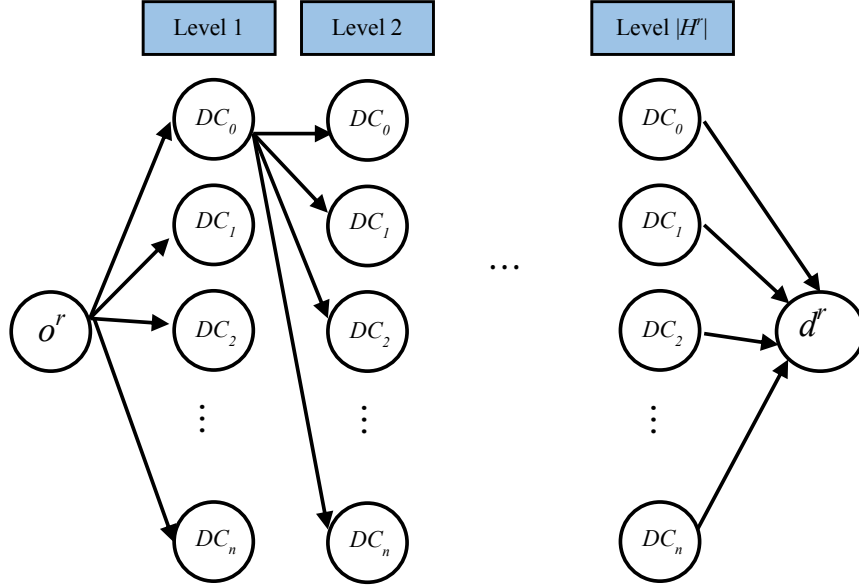


Figure 4: Auxiliary multi-layer graph for a request r

550 multi-DC NFV infrastructure. As already done in [53] for the optimal VNFs
551 selection problem for service chaining, we make use of an auxiliary graph
552 $G^r = (N^r, A^r)$ for each request $r \in R$. Specifically, G^r is a layered graph
553 with a level for each of the $|H^r|$ VNFs appearing in request r (numbered
554 from 1 to $|H^r|$), in addition to two extra levels: the first level, namely level
555 0, containing the origin node o^r of the request and the last level, namely level
556 $|H^r| + 1$ containing the destination node d^r of the request. Each intermediate
557 level $h \in H^r$ is composed by all the DCs. The arc set A^r is organized in three
558 groups: (i) arcs connecting the source node in level 0 to each node in level
559 1; (ii) arcs connecting each DC in level $|H^r|$ to the destination node in last
560 level; and (iii) arcs linking each DC i in level h with each DC j in level $h + 1$
561 for each intermediate level ($h \in \{1, \dots, |H^r| - 1\}$). In this latter group, arc
562 from DC i to DC j is characterized by the propagation latency l_{ij} and the
563 bandwidth b_{ij} . DCs and arcs between any couple of DCs are shared among
564 requests. A graphical representation of the auxiliary graph G^r is given in
565 Figure 4.

566 Servicing request r corresponds to determine a path in G^r from o^r to d^r .

567 By construction, such a path visits exactly a node in each level. Specifically,
 568 the node visited in intermediate level h corresponds to the DC where the
 569 h -VNF of the request r , namely V_h^r , is deployed. The layered structure of
 570 the graph thus ensures that the order of VNFs specified in the request is
 571 preserved.

572 This work does not lose generality if, for the sake of clarity, it is assumed
 573 that all the requests are characterized by VNF chains of the same length
 574 $|H| = \max_r \{|H^r|\}$. In that case, for each request, last level corresponds to
 575 level $|H| + 1$ and it contains the destination of each request. Indeed, every
 576 time a request is characterized by a chain shorter than $|H|$, then all the nodes
 577 in level $|H^r|$ are connected directly to the destination node in $|H| + 1$ and
 578 all the levels comprised between $|H^r| + 1$ and $|H|$ are consequently neglected
 579 (hop across levels). In other words, when $|H^r| < |H|$, with a short abuse of
 580 notation, level $|H^r| + 1$ identifies the last level, i.e., $|H| + 1$.

581 In order to model the problem, two groups of decision variables are consid-
 582 ered corresponding respectively to path design variables (allocation of VNFs
 583 to DCs), and requests' satisfaction (maximal covering). Specifically, they are
 584 defined as follows:

$$585 \quad x_{ihjh+1}^r = \begin{cases} 1 & \text{if the arc linking node } i \text{ in level } h \text{ and node } j \text{ in level} \\ & (h + 1) \text{ belongs to the path relative to } r \in R \\ 0 & \text{otherwise} \end{cases}$$

$$586 \quad \forall r \in R, i \in D \cup \{o^r\}, j \in D \cup \{d^r\}, h \in H^r \cup \{0, |H| + 1\}$$

$$587 \quad z^r = \begin{cases} 1 & \text{if request } r \text{ is served} \\ 0 & \text{otherwise} \end{cases} \quad r \in R.$$

588 Besides the notation introduced in Tables 1, 2, 3 and 4, the model makes
 589 use of the additional parameters defined in Table 5. All the sets and the
 590 parameters contained in these tables define the input of the optimization
 model.

Table 5: Additional model parameters

w_t	weight associated with a request of priority level t
W	weight used in the hierarchical objective function
L_h	nodes belonging to level h in the auxiliary graph
I	incompatibilities set

591

592 By using the above-defined variables and notation, the problem can be
 593 stated formally, as follows.

$$\max \quad W \sum_{t \in T} \sum_{r \in R_t} w_t \cdot z^r + \sum_{r \in R} \sum_{h=0}^{|H^r|-1} \sum_{i \in L_h} \sum_{j \in L_{h+1}} p_{V_{h+1}^r}^i \cdot x_{jih+1} \quad (5)$$

$$\sum_{j \in L_1} x_{or_0j1}^r = z^r, \quad \forall r \in R \quad (6)$$

$$\sum_{j \in L_{|H^r|}} x_{j|H^r|d^r|H|+1}^r = z^r, \quad \forall r \in R \quad (7)$$

$$\sum_{j \in L_{h-1}} x_{jh-1ih}^r - \sum_{j \in L_{h+1}} x_{ihjh+1}^r = 0, \quad \forall r \in R, \forall h \in \{1, \dots, |H^r|\}, \forall i \in L_h \quad (8)$$

$$\sum_{r \in R} \sum_{h=0}^{|H^r|-1} \sum_{j \in L_h} u_{V_{h+1}^r}^n x_{jih+1}^r \leq \bar{p}_i^n \cdot u_i^n, \quad \forall i \in D, \forall n \in N \quad (9)$$

$$\sum_{h=0}^{|H^r|-1} \sum_{i \in L_h} \sum_{j \in L_{h+1}} c_j \sum_{n \in N} u_{V_{h+1}^r}^n x_{ihjh+1}^r \leq c^r, \quad \forall r \in R \quad (10)$$

$$\sum_{h=0}^{|H^r|} \sum_{i \in L_h} \sum_{j \in L_{h+1}} l_{ij} x_{ihjh+1}^r \leq l^r, \quad \forall r \in R \quad (11)$$

$$\sum_{r \in R} \sum_{h=0}^{|H^r|} b_{hh+1}^r x_{ihjh+1}^r \leq b_{ij}, \quad \forall (i, j) \in E \quad (12)$$

$$\sum_{j \in L_{h-1}} x_{jh-1ih}^r = 0, \quad \forall r \in R, \forall (V_h^r, i) \in I \quad (13)$$

$$x_{ihjh+1}^r \in \{0, 1\}, \quad \forall r \in R, \forall i \in D \cup \{o^r\}, \forall j \in D \cup \{d^r\}, \forall h \in \{0, \dots, |H^r|\} \cup \{0, |H| + 1\} \quad (14)$$

$$z^r \in \{0, 1\}, \quad \forall r \in R \quad (15)$$

594 The hierarchical objective function is defined in (5) and it consists in the
 595 maximization of the weighted sum of the two criteria introduced in Section
 596 3, namely provider utility and user utility. Weight W is used to give more
 597 relevance to the first criterion. In addition, weights w_t are set so as to give
 598 more privileges to requests with higher priority level. The second criterion

599 accounts for preferences satisfaction: $p_{V_h^r i}^r$ expresses the preference grade of
 600 request r for placing the h -th VNF on DC i . Constraints (6), (7) and (8)
 601 are, for each $r \in R$, the flow conservation constraints defining the path from
 602 o^r to d^r . Specifically, for each r , constraint (6) assures that exactly one unit
 603 of flow leaves the source node o^r when request r is accepted ($z^r = 1$); in that
 604 case, since by definition, the path design decision variables are 0-1 variables,
 605 exactly one of the arcs outgoing from o^r will be selected. The ending node
 606 of such an arc belongs to level L_1 and it identifies the DC that hosts the
 607 first virtual function in request r . Conversely, when request r is not served
 608 ($z^r = 0$), no unit of flow will leave the source node. Symmetrically, for
 609 each accepted request $r \in R$, exactly one unit of flow enters the destination
 610 node d^r as imposed by constraint (7). Constraints (8) assure that for each
 611 request $r \in R$, for each intermediate level h and for each node $i \in L_h$, the
 612 quantity of flow entering node i is exactly the same as the one leaving node
 613 i . Constraints (9) are the workload constraints and they are defined for each
 614 DC and for each resource n . Specifically, they guarantee that the actual
 615 workload of each DC, which is given by the the sum of resources of a given
 616 typology required to execute VNFs deployed on it, must not exceed a given
 617 threshold which is proportional to its capacity u_i^n . Percentage \bar{p}_i^n is used
 618 to define the maximum ($\bar{p}_i^n \cdot u_i^n$) workload of DC i relevant to resource n ,
 619 thus avoiding overhead. Constraints (10) guarantee that for each request,
 620 the total cost spent for all the resources and the VNFs of its chain does not
 621 exceed the cost c^r request r is willing to pay to get the service. Constraints
 622 (11), for each request r , assure that the end-to-end delay experienced to
 623 accomplish the service must not exceed the maximum tolerated latency l^r .
 624 Constraints (12), for each arc (i, j) in the abstract network, guarantee that
 625 the total bandwidth required to accomplish all of the service requests using
 626 (i, j) must not exceed the maximum available bandwidth b_{ij} . Observe that,
 627 the inner summation in constraints (12) considers all the copies of arc (i, j)
 628 between any two consecutive layers. Constraints (13) are the incompatibility
 629 constraints and they guarantee that if the h -VNF of request r is incompatible
 630 with DC i , then none of the arcs ingoing node i in level h can be used by
 631 request r or equivalently, the corresponding path design variable is set to 0.
 632 Finally constraints (14) and (15) define variable domain.

633 **5. Pre-Processing phase**

634 The system designed to solve the VNF Placement problem is equipped
 635 with a pre-processing phase which has a three-fold aim: (i) discard all those
 636 requests that cannot be accomplished by the system for infeasibility reasons,
 637 (ii) define the potential incompatibility between a specific virtual function
 638 of a given request and a DC, and (iii) define user preferences that are then
 639 used in the secondary objective function of the optimization model. In the
 640 following three sections the three features of the pre-processing phase are
 641 described in detail.

642 *5.1. Infeasibility check*

643 In regards to infeasibility check, three conditions are controlled concerning
 644 respectively latency, bandwidth and cost. Specifically,

- 645 1. *Latency check*: for each request $r \in R$, the maximum tolerated end-to-
 646 end delay l^r is compared with the minimum possible achievable prop-
 647 agation latency from o^r to d^r , i.e., $l_{o^r d^r}$. If, for a given r ,

$$l^r < l_{o^r d^r}, \quad (16)$$

648 then request r is rejected.

- 649 2. *Bandwidth check*: for each request $r \in R$, the maximum bandwidth
 650 consumption of r is compared with the maximum possible achievable
 651 bandwidth for all the paths connecting o^r to d^r in the abstract network,
 652 namely P^r . If, for a given r ,

$$\max_{p \in P^r} \{ \min_{(i,j) \in p} b_{ij} \} < \max_{h=1, \dots, |H^r|-1} b_{hh+1}^r, \quad (17)$$

653 then request r is rejected.

- 654 3. *Cost check*: for each request $r \in R$, the maximum cost r is willing to
 655 pay for the service, i.e., c^r is compared with the minimum cost achiev-
 656 able on the network which occurs when the total capacity required by
 657 the request, namely u^r is provided by the DC with minimum cost per
 658 capacity unit. If, for a given r ,

$$u^r \cdot \min_{i \in D} \{ c_i \} > c^r \quad (18)$$

659 where

$$u^r = \sum_{h=1}^{|H^r|} \sum_{n \in N} u_{V_h^n}^r \quad (19)$$

660 then request r is rejected. In equation (19), we assume that the total
661 quantity of resources required by a request is given by the sum of the
662 quantities required by all the resources and all the VNFs in the chain.
663 We also assume that the cost of a request depends on the aggregated
664 use of resources. However, in order to define the cost of a request,
665 other linear combinations of the parameters involved can be managed
666 as well; as an example, in [54], the price exposed by a DC depends on
667 the DC itself and on the resource considered.

668 These three controls must all be satisfied to allow the request be given in
669 input to the optimization solver; this does not guarantee that it will definitely
670 be served, but only that it is compatible with the system supply.

671 5.2. Incompatibility definition

672 In regards to the definition of incompatibility between a specific VNF of
673 a request and a DC, two conditions are controlled concerning respectively
674 capacity and setup time. Specifically,

- 675 1. *Capacity check*: if a resource $n \in N$ exists for which the corresponding
676 resource capacity of the i_{th} DC, namely u_i^n is smaller than the capacity
677 required by the h -th virtual function of request r in terms of resource
678 n , namely $u_{V_h^r}^n$, then the assignment between V_h^r and i is forbidden and
679 the couple (V_h^r, i) is inserted in the incompatibility set I , i.e.,

$$\text{if there exists } n \text{ s.t. } u_i^n < u_{V_h^r}^n \text{ then } (V_h^r, i) \in I. \quad (20)$$

- 680 2. *Instantiation time check*: the availability of a container-based virtual-
681 ization technology at the i_{th} DC, namely the binary parameter s_i , is
682 compared with the instantiation time requirement of request r , namely
683 the binary parameter s^r . Specifically, request r can be served by DC
684 i when $s_i \geq s^r$ that means that if request r needs to be deployed on a
685 container to minimize the instantiation time ($s^r = 1$), then DC i has
686 to provide a container ($s_i = 1$). If the condition does not hold, then
687 none of the virtual functions of r can be deployed on DC i , i.e.,

$$\text{if } s_i < s^r \text{ then } (V_h^r, i) \in I \quad \forall h \in H^r. \quad (21)$$

688 This preprocessing phase can be easily extended by managing further
689 conditions, such as commercial alliances and conflicts of interest (e.g., DCs
690 managed by competitors are banned).

691 5.3. Definition of user preferences

User preferences are built upon a ranking algorithm that provides, for each request r , an ordered preference list of DCs to be used in the placement. Specifically, a set M of preference criteria are considered to define the global vote q_i^r request r attributes to DC i , i.e.,

$$q_i^r = \sum_{m \in M} w_m^r q_{mi} \quad \text{with} \quad \sum_{m \in M} w_m^r = 1 \quad \forall r \in R, \forall i \in D, \quad (22)$$

692 where weight w_m^r reflects the importance request r gives to criterion m and
 693 q_{mi} expresses the vote to DC i with respect to criterion m . Indeed, users
 694 assign weights to the criteria according to their business and/or private goals.
 695 Votes q_{mi} assume values in the range $[0,1]$, thus also the global vote q_i^r is in the
 696 range $[0,1]$. Then, for each request r , DCs are ranked according to decreasing
 697 values of q_i^r .

698 Starting from q_i^r , user preference grades $p_{V_h^r}^r$ can be assigned according to
 699 different policies and range of values which contribute to design a flexible tool
 700 capable of copying with general preference schemes. In this study, we consider
 701 two preference criteria: i) cost minimization (C) and ii) environmental impact
 702 minimization (F) (e.g., carbon dioxide emissions).

703 In regards to costs, we assume that each request competes with the others
 704 to place its constituent VNFs in the DCs that are more economically conve-
 705 nient. For each DC i , the vote with respect to cost minimization ($m = C$) is
 706 given by

$$q_{Ci} = \frac{c_{min}}{c_i} \quad (23)$$

707 where

$$c_{min} = \min_{i \in D} \{c_i\} \quad (24)$$

708 is the minimum service price exposed over the whole set of DCs.

709 In regards to environmental impact, we assume that the users who have
 710 expressed their interest in reducing the environmental impact ($f^r = 1$), favor
 711 DCs characterized by the lowest possible CUE. Thus, for each DC i , the vote
 712 with respect to environmental impact minimization ($m = F$) is given by:

$$q_{Fi} = \frac{f_{min}}{f_i} \quad (25)$$

713 where

$$f_{min} = \min_{i \in D} \{f_i\} \quad (26)$$

714 is the minimum CUE value over the whole set of DCs.

715 Different strategies can be adopted to exploit the DC ranking based on
716 the above described global vote calculation procedure to assign appropriate
717 values to preferences in the user utility part of the hierarchical objective
718 function defined in (5) in Section 4.2. In practice, strategies can differ on
719 what is considered full or partial satisfaction, taking into account that, due
720 to resource capacity constraints, not all VNFs can be placed on the respec-
721 tive first ranked DCs. In this work we consider a strategy considering that
722 only the assignment to the first and second positioned DCs can be respec-
723 tively considered as full and partial satisfaction, while the remaining options
724 are considered dissatisfaction. This strategy (called 2LevelSat strategy) is
725 implemented as follows. The global vote formula in (22) is used for creating
726 a rank of DCs for each VNF in a request, then the first positioned DC is as-
727 signed a preference value equal to 1, the second positioned DC a value equal
728 to 0.5 and 0 otherwise. We also formulate an alternative strategy that use
729 more granular preferences respect to the previous strategy. More specifically,
730 preferences assume exactly the same value of the global vote, in the range
731 $[0,1]$, as defined in (22). This means that VNFs assigned to DCs that are not
732 in the first two positions howsoever contribute to the global satisfaction level
733 and are consequently considered as partially satisfied. The 2LevSat strat-
734 egy adopts a more restrictive definition of partial satisfaction with respect
735 to GradSat. In Section 6.4 (Performance Evaluation) we evaluate how far
736 preferences are satisfied by these two strategies.

737 6. Performance Evaluation

738 In this section we describe the activities carried out to evaluate the pro-
739 posed VNF placement solution. First, we briefly describe the experimental
740 settings and the adopted metrics, then we describe the tests and discuss
741 obtained results.

742 To evaluate the proposed solution, we have developed a testing tool based
743 on CPLEX and MATLAB. The preprocessing steps are performed by Matlab
744 scripts while the VNF Placement problem is solved using CPLEX 12.8.

745 6.1. Benchmark instances

746 We considered three different network topologies, namely a hypothetical
747 German backbone network (17 nodes), a Pan-European network (28 nodes)

748 and a US Network (14 nodes). Topological parameters have been gathered
 749 from the literature [55].

750 We adopted the betweenness centrality metric to select the nodes that
 751 can host VNFs (the so called DC nodes). Betweenness centrality of a node is
 752 calculated as the number of shortest all-to-all paths that pass through that
 753 node and is thus a good indicator of the importance of a node in the network
 754 [56]. The sum of the resources available in all DC nodes is called overall
 755 capacity and assumed to be equal to 100 units.

756 We generated request data sets by mirroring realistic traffic using the traf-
 757 fic distribution used in [56] and elaborated from the global IP Traffic Forecast
 758 by Cisco [57]. We considered three types of service requests, similarly to the
 759 settings in [30, 56, 58, 59]. Each service request type contains a sequence of
 760 VNFs and requires a specific amount of bandwidth and a maximum end-to-
 761 end latency (see Table 6). Within each service request set, service request
 762 types are distributed according to percentages derived from realistic traffic
 763 distribution [57].

Table 6: Service chains that have been considered to compose each request set [58]

Service	Chain	Latency	Bandwidth	percentage
Web Service (WS)	NAT-FW-TM-WOC-IDPS	500 ms	100 kbit/s	18.2 %
VoIP	NAT-FW-TM-FW-NAT	100 ms	64 kbit/s	11.8 %
Video Streaming (VC)	NAT-FW-TM-VOC-IDPS	80 ms	4 Mbit/s	70.0 %

764 At each iteration, a set of requests is generated that stresses the net-
 765 work with a given overall request load, defined as the ratio between the total
 766 amount of resources required by the requests in the set and the overall ca-
 767 pacity offered by the multi-DC network. For instance, given a target request
 768 load of 80%, the amount of required resources by all requests in the set is
 769 calculated as a percentage of the actual overall capacity (i.e., 80 over 100
 770 units), and is equally distributed among all requests in the request set. We
 771 consider two priority levels, premium and best effort, where premium’s pri-
 772 ority level is higher than best effort’s one. Since each type of chain contains
 773 5 VNFs and we assume that all VNFs require the same amount of resources
 774 (1 unit), the target request load is thus achieved by varying the number of
 775 requests in the set.

776 Each request of the set is generated by varying its characteristics at each
 777 iteration. Source and destination nodes are randomly selected among DC
 778 nodes. Configuration of further attributes (e.g., priority level, service cost,

779 setup time, and carbon footprint preference) is described hereafter for each
780 test case. Finally, the weights of the hierarchical objective function have
781 been defined in order to give more relevance to the acceptance rate criterion
782 (weight $W=1000$) with respect to preference satisfaction and to preferably
783 accept premium requests than best effort ones ($w_p=3$ and $w_b=1$).

784 6.2. Evaluation metrics

785 We define a test case for each of the following metrics:

- 786 • *Acceptance Rate*: the ratio between the number of accepted requests
787 (i.e., requests that have been deployed in the optimal solution), also
788 differentiated per priority level, and the total number of requests in a
789 request set. The request set is generated so that all requests pass the
790 feasibility check.
- 791 • *Preference satisfaction*: it provides a measure of how much the prefer-
792 ences expressed in a request have been satisfied.
- 793 • *Execution time*: time required by the solver to process a set of requests
794 and return the optimal solution.
- 795 • *DC utilization factor*: percentage of used resources against maximum
796 resource capacity for each DC.
- 797 • *Request Load spread across DCs*: percentage of the overall resource
798 demand of a request set assigned to each DC.
- 799 • *Request latency vs maximum tolerated latency*: it is the ratio between
800 the computed latency of an accepted request vs its corresponding max-
801 imum tolerated latency.

802 6.3. Acceptance Rate

803 This test case has the goal of assessing to which extent the service provider
804 profit is maximized in terms of acceptance rate. Tests have been run over
805 the three network topologies where 60% of nodes have been modeled as DC
806 nodes. We consider three different combinations of premium (P) and best
807 effort (BE) priority levels in the request set, as follows:

- 808 1. P=70% and BE=30%;
- 809 2. P=50% and BE=50%;

810 3. P=30% and BE=70%.

811 We vary the request load from 70% to 120% with an increment step of
812 10% in order to increasingly stress the network.

813 We run 50 test iterations for each combination of priority level distribu-
814 tion and request load. In each iteration we slightly vary some parameters
815 characterizing the request set and the substrate. As regards the request set,
816 source and destination nodes of the service request are randomly mapped to
817 the subset of compute nodes in the network and the maximum cost allowed
818 for each request is calculated by multiplying the amount of resources required
819 by the request with a maximum cost for unit capacity that randomly varies
820 in the range [0.9,1.1]. A 25% of requests (randomly selected) requires a fast
821 instantiation time (i.e., s^r set to 1). As regards preference criteria, 75% of
822 requests in each set has cost reduction as unique preference criterion and the
823 remaining 25% of requests has both cost and carbon footprint preference cri-
824 teria (see Section 5.3). As regards topology settings, the price offered by each
825 node per capacity unit randomly varies in the range [0.7,1.2], while the CUE
826 randomly varies in a discretized range [1,7] and 50% randomly selected nodes
827 offer a container virtualization technology, i.e., they can satisfy requests with
828 s^r set to 1.

829 Fig. 5 shows the average percentage of accepted requests for each combi-
830 nation of P and BE requests, without differentiating results per classes,
831 for the Pan-European topology. For request loads lower than 100%, almost
832 all requests are accepted, with negligible difference with respect to the three
833 combinations of P and BE requests. When the request load is more challeng-
834 ing (i.e., greater than 100%), the overall acceptance rate slightly decreases,
835 but such decrease is mainly caused by the reduction in the number of accepted
836 BE requests in favor of premium ones, as more clearly shown in Fig. 6. This
837 was expected, since in our tests premium and best effort requests require
838 the same amount of resources and when resources offered by the substrate
839 are getting scarce for high request loads, preference is given to premium re-
840 quests. Tests conducted with the German and US network topologies show
841 analogous trends thus confirming the expected behavior of the algorithm.

842 6.4. Preference satisfaction

843 This test case aims at evaluating how far preferences are satisfied in the
844 placement decision, considering both 2LevelSat and GradSat preference as-
845 signment strategies.

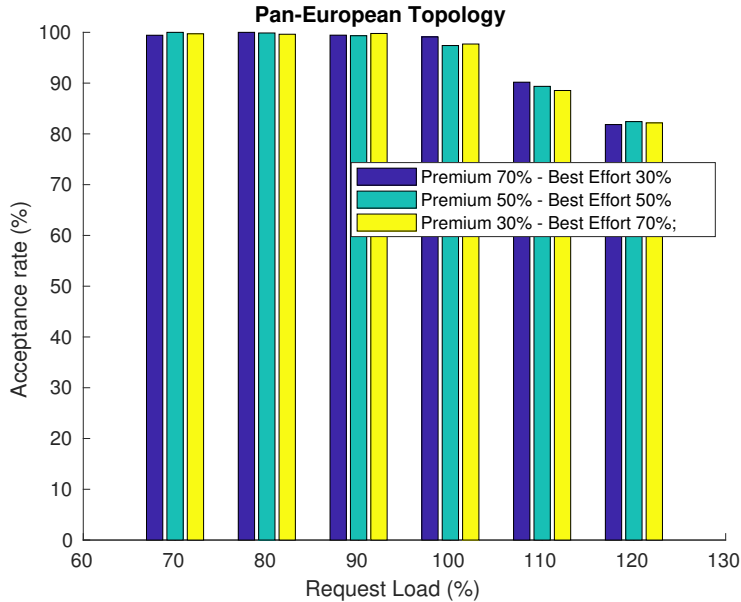


Figure 5: Overall Acceptance Rate vs Request load for different combinations of premium and best effort requests - Pan-European topology

846 Based on the global votes calculated in the preprocessing phase (Section
 847 5.3), an ordered list of DCs is created for each VNF in the request set,
 848 expressing a descending order of preference for placement.

849 In order to measure how far preferences are satisfied, we count how many
 850 preferences expressed in the request set have been satisfied. More specifically,
 851 we count how many VNFs of the request set have been placed in the first-
 852 ranked DCs and how many VNFs in the second-ranked DCs.

853 Tests have been carried out on the German and Pan-European topologies
 854 with the same settings of the substrate network as in the previous test. As
 855 regards the request set, we considered three different combinations of pre-
 856 mium (P) and best effort (BE) priority levels as in the previous test (i.e.,
 857 P=70% and BE=30%, P=50% and BE=50%, P=30% and BE=70%). We
 858 considered increasing load values (70%, 80%, 90%, 100%), maximum cost in
 859 the range [0.9,1.1] and two different preference settings, described hereafter.

860 First, we evaluate results obtained with the adoption of 2LevelSat strat-
 861 egy. Table 7 shows the results obtained with the first preference settings
 862 (called Settings A) where 25% of requests equally take into account cost
 863 and environmental impact as guiding criteria ($w_C^r = w_F^r = 0.5$ in equa-

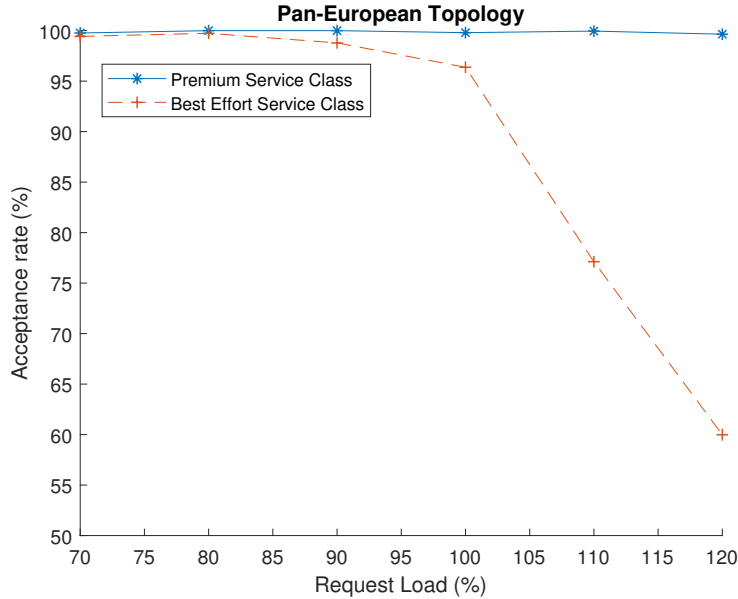


Figure 6: Acceptance Rate per priority level vs Request load - Pan-European topology

864 tion (22)), while 75% of requests take into account only the cost criterion
 865 ($w_C^r = 1, w_F^r = 0$).

866 As the request load (and thus the overall number of VNFs to be placed)
 867 increases, the percentage of VNFs placed in DCs ranked in the 1st and 2nd
 868 position clearly decreases. This is due to the fact that more VNFs compete to
 869 be placed in the preferred DCs and this effect is exacerbated by the fact that
 870 with these preference settings a large percentage (75%) of requests compete
 871 to be placed in the most economically convenient nodes. As the percentage
 872 of DC nodes increases, preference satisfaction decreases since, as explained
 873 before, the overall resource capacity is fixed to 100 and the resource quota
 874 assigned to each node diminishes as the number of DCs increases. Therefore,
 875 as the number of DC nodes increases, first and second positioned DCs can
 876 accommodate fewer requests. However, Table 7 shows that, even with high
 877 request loads and number of DCs, the percentage of VNFs placed in first or
 878 second position is quite high.

879 Table 8 shows the results obtained with the second preference settings
 880 (called Settings B) where 50% of requests consider only the cost criterion
 881 ($w_C^r = 1, w_F^r = 0$) and the remaining 50% considers only the environmental

882 impact ($w_C^r = 0, w_F^r = 1$). As shown in Table 8, preference satisfaction
 883 improves with respect to the previous configuration. This is due to the fact
 884 that we divided the request set in two disjoint subsets (one targeting cost
 885 effective DCs, and the other one targeting DCs minimizing the environmental
 886 impact) and thus the competition on the resource substrate decreases.

Table 7: Preference satisfaction with 2LevelSat preference assignment strategy - percentage of VNFs placed in 1st and 2nd ranked DCs for sets with 75% requests with $w_C^r = 1$ and $w_F^r = 0$ and 25% requests with $w_C^r = w_F^r = 0.5$ (Settings A)

	German Topology				Pan-European Topology			
	11 DC nodes (~60%)		14 DC nodes (~80%)		16 DC nodes (~60%)		21 DC nodes (~80%)	
Request Load	1st ranked	2nd ranked	1st ranked	2nd ranked	1st ranked	2nd ranked	1st ranked	2nd ranked
70%	27.0	19.7	22.7	17.3	19.3	17	18.3	15
80%	25.3	18.3	21.0	16.0	18.3	15.0	15.7	13.3
90%	22.7	16.7	19.7	14	17.3	13.0	14.0	11.7
100%	21.3	16.3	17.7	14.0	15.7	13.0	12.7	11.0

Table 8: Preference satisfaction with 2LevelSat preference assignment strategy - percentage of VNFs placed in 1st and 2nd ranked DCs, 50% requests with $w_C^r = 1$ and $w_F^r = 0$ and 50% requests with $w_C^r = 0$ and $w_F^r = 1$ (Settings B)

	German Topology				Pan-European Topology			
	8 DC nodes (50%)		14 DC nodes (80%)		17 DC nodes (60%)		22 DC nodes (80%)	
Request Load	1st ranked	2nd ranked	1st ranked	2nd ranked	1st ranked	2nd ranked	1st ranked	2nd ranked
70%	29.7	22.3	24.3	20.3	22.3	18.3	17.7	16.3
80%	27.0	21.3	22.3	18.7	20.7	17.0	16.7	14.3
90%	24.0	19.0	21.3	16.0	18.0	15	14.7	13.3
100%	23.0	18.0	18.7	16.0	17.0	13.7	14.0	12.3

887 The remaining part of this section is dedicated to show the results ob-
 888 tained with the alternative GradSat strategy to assign preferences. We re-
 889 peated the same tests (i.e., reusing the same request sets, preference and
 890 topology configurations) and report the results in Tables 9 and 10.

891 The resulting behaviour is quite similar to the one obtained with the
 892 previous preference assignment strategy, i.e., the percentage of VNFs placed

Table 9: Preference satisfaction with GradSat preference assignment strategy - percentage of VNFs placed in 1st and 2nd ranked DCs for sets with 75% requests with $w_C^r = 1$ and $w_F^r = 0$ and 25% requests with $w_C^r = w_F^r = 0.5$ (Settings A)

	German Topology				Pan-European Topology			
	11 DC nodes (~60%)		14 DC nodes (~80%)		16 DC nodes (~60%)		21 DC nodes (~80%)	
Request Load	1st ranked	2nd ranked	1st ranked	2nd ranked	1st ranked	2nd ranked	1st ranked	2nd ranked
70%	24.3	20.0	20.0	17.0	17.3	14.7	14.7	14.0
80%	22.0	17.3	18.0	15.0	16.7	14.0	13.3	11.7
90%	20.0	15.0	17.0	12.3	14.3	12.0	12.3	10.3
100%	19.0	15.0	15.3	13.0	13.7	11.3	11.0	9.7

Table 10: Preference satisfaction with GradSat preference assignment strategy - percentage of VNFs placed in 1st and 2nd ranked DCs for sets with 50% requests with $w_C^r = 1$ and $w_F^r = 0$ and 50% requests with $w_C^r = 0$ and $w_F^r = 1$ (Settings B)

	German Topology				Pan-European Topology			
	8 DC nodes (50%)		14 DC nodes (80%)		17 DC nodes (60%)		22 DC nodes (80%)	
Request Load	1st ranked	2nd ranked	1st ranked	2nd ranked	1st ranked	2nd ranked	1st ranked	2nd ranked
70%	26.7	23.3	21.0	20.3	19.3	17.7	15.0	15.7
80%	24.0	21.3	19.0	18.3	17.3	15.7	13.7	13.7
90%	20.0	19.7	18.0	14.7	14.7	13.7	12.3	12.0
100%	18.7	17.7	15.3	14.3	14.0	12.7	11.7	10.3

893 in the DCs ranked in the 1st and 2nd position decreases with the request
894 load both in Table 9 and Table 10. Also in this case preference satisfaction
895 in Table 10 is higher than in Table 9.

896 Comparing these two strategies, it is evident that the first strategy (2Lev-
897 elSat) succeeds in allocating a greater percentage of VNFs in the first and
898 second ranked DCs. In addition, different preference assignment strategies
899 may also impact the computational time required to solve the optimization
900 problem. Although the evaluation on computational time is discussed in the
901 following section, it is worth highlighting here how the first strategy leads to
902 generally shorter execution time than the second one does, with an average
903 computational time over all iterations of 1636 ms versus 4821 ms, respec-
904 tively. However, the comparison of the user utility objective value achieved
905 shown in Fig. 7 shows that the GradSat strategy obtains higher objective

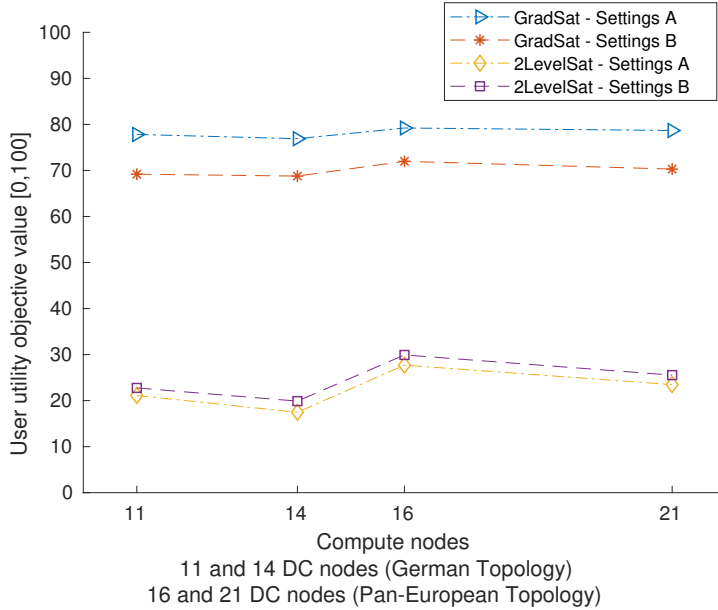


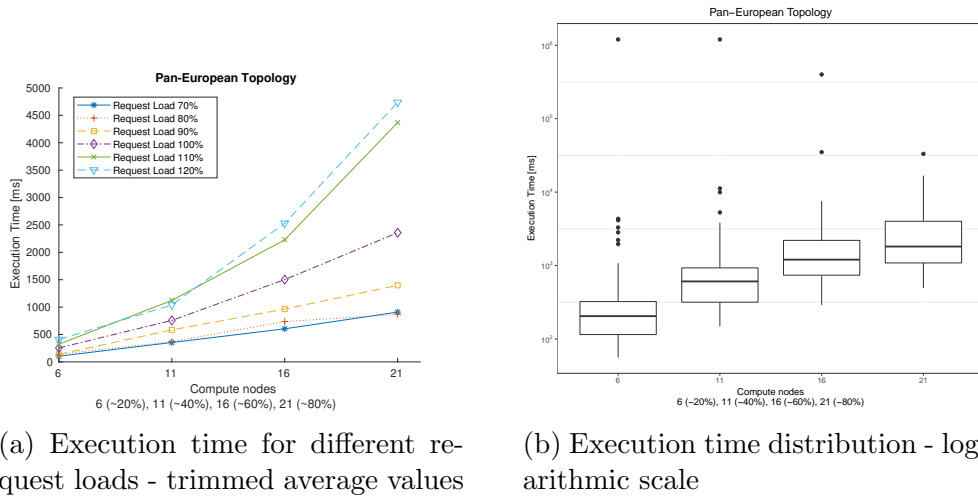
Figure 7: Comparison of Preference assignment strategies in terms of user utility objective value (Request Load=100%)

906 function values than 2LevelSat’s ones in both types of tests (i.e., Settings A
 907 and B). Fig.7 shows the average user utility objective value obtained with
 908 Request Load equal to 100%.

909 6.5. Execution Time

910 This test aims at evaluating the computational time required by the op-
 911 timization algorithm to solve the VNF placement problem. Tests have been
 912 performed on all topologies by varying the percentage of nodes selected as
 913 DCs (approximately 20%, 40%, 60%, 80%). We have varied the number of
 914 requests in the input request set (from 14 to 24 requests) to correspondingly
 915 vary the overall request load (from 70% to 120% of the overall capacity with
 916 an increasing step of 10%).

917 For each combination of request load and DC nodes percentage, we run
 918 50 iterations, varying some parameters’ settings. Analogously to previous
 919 test settings, at each iteration we vary the following parameters: source and
 920 destination nodes of the service request are randomly mapped to the subset
 921 of compute nodes in the network and the maximum cost allowed for each
 922 request is determined by multiplying the amount of resource required by the



(a) Execution time for different request loads - trimmed average values

(b) Execution time distribution - logarithmic scale

Figure 8: Execution time vs % of compute nodes - Pan-European topology

923 request with a maximum cost for unit capacity that is made randomly vary in
 924 the range $[0.9, 1.2]$. A 25% percentage of requests (randomly selected) require
 925 a fast instantiation time (i.e., s^r set to 1). Moreover, 75% of requests in each
 926 set have cost containment as unique preference criterion and the remaining
 927 25% of requests express both cost and carbon footprint preference criteria.
 928 The settings of the topology substrate is the same as in previous tests.

929 Figure 8 shows results obtained for the Pan-European topology. Graphic
 930 (a) on the left, shows how the ten percent trimmed value of execution time
 931 varies against the percentage of nodes considered as possible VNF locations
 932 for different request loads. Conversely, the distribution of the execution
 933 time, including also outliers, is shown in graphic (b) on the right. Analogous
 934 results have been obtained for the German and US topology, which are not
 935 reported here for the sake of conciseness, thus corroborating the validity of
 936 the approach. As expected, the results confirm that the time needed to
 937 find the optimal solution is influenced by the request load more than by the
 938 number of DCs. Specifically, we observe that the number of nodes has an
 939 almost linear impact on the computational time.

940 It is worth noticing that all tests have been run with a time limit for the
 941 solver set to 1200 seconds. As shown in Fig. 8b, in most cases the compu-
 942 tational time stays well under this limit, while some outliers are highlighted
 943 with values well above 3 secs.

944 In order to evaluate the tradeoff between solution quality and efficiency

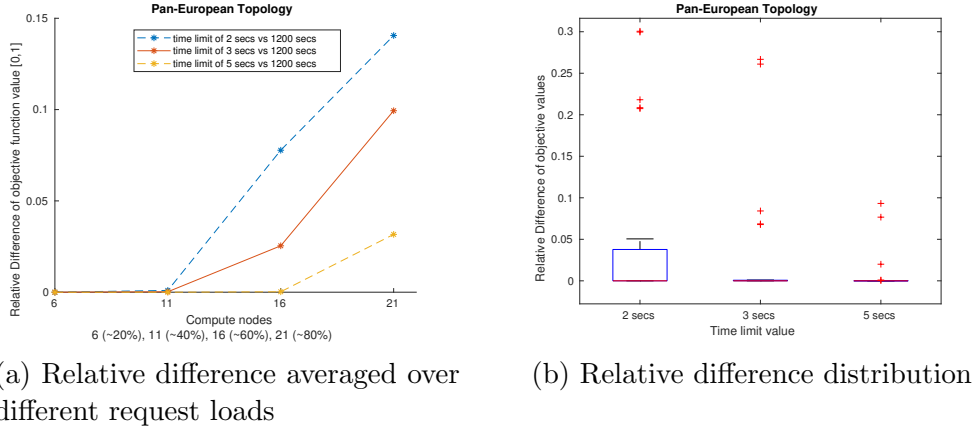


Figure 9: Relative difference of the obtained objective function values with respect to the algorithm configured with a time limit of 1200 seconds

945 (computational time), we repeated the same test case by imposing a time
 946 limit of 5, 3 and 2 seconds, respectively. This further experiment is done
 947 only for the Pan-European topology which is the one with the highest com-
 948 putational times.

949 Figure 9a shows the relative difference of the obtained objective function
 950 values with respect to the algorithm configured with a time limit of 1200
 951 seconds. Results shows that the relative difference (averaged over different
 952 request loads) is almost zero in most cases and increases with the number
 953 of DC nodes, but it is lower than 14%. Fig 9b shows the distribution of the
 954 objective function relative difference, highlighting outliers and median values
 955 (close to zero).

956 6.6. DC Utilization factor and request load spread across DCs

957 We analyzed results of the tests conducted on a 11 DC network in the
 958 Pan-European topology to evaluate how DC resources are used for request
 959 sets demanding 70% of overall resources (i.e. request load). Fig. 10 shows the
 960 percentage of resource usage for each DC. Considering the above mentioned
 961 request load and the fact that no upper thresholds on DC resource usage have
 962 been set, it is worth noticing that the average utilization factor of each DC
 963 is above 60%, thus demonstrating a good balance of resource usage across
 964 DCs.

965 Figure 11 shows how the request load is spread across DCs, highlighting
 966 a quite fair distribution of request loads across DCs.

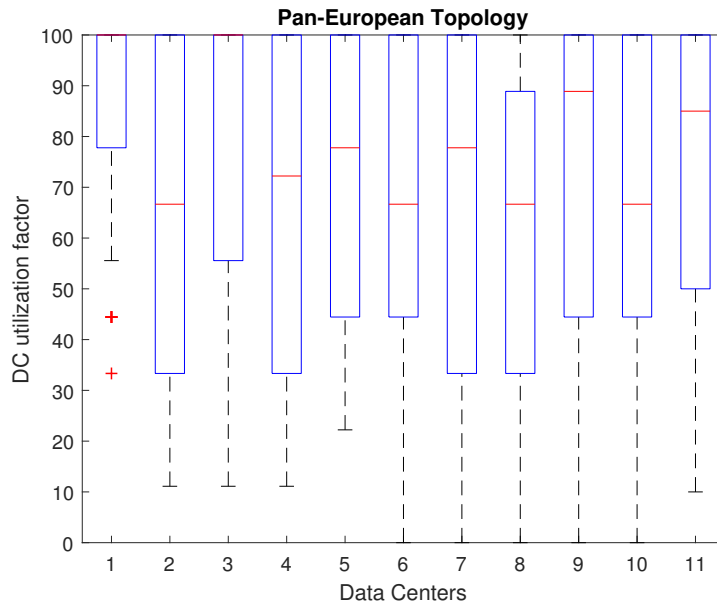


Figure 10: DC utilization factor - Pan-European topology, 11 DCs, 70% request load

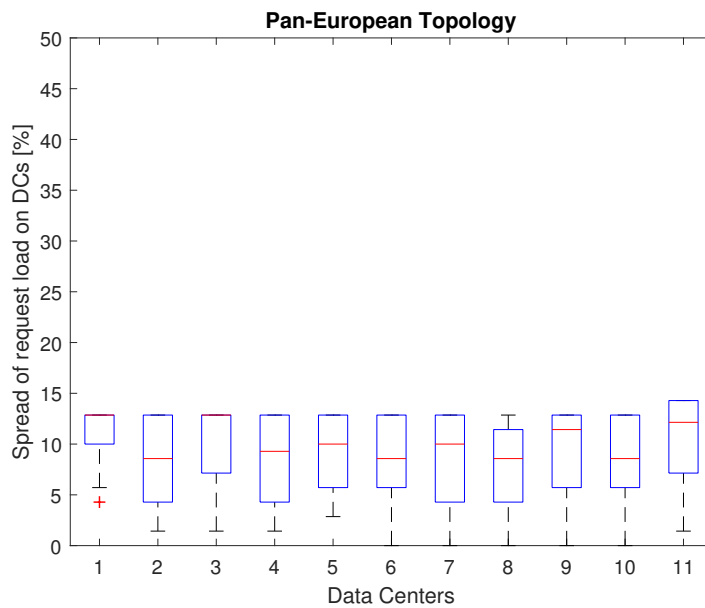


Figure 11: Spread of request load across DCs - Pan-European topology, 11 DCs, 70% request load

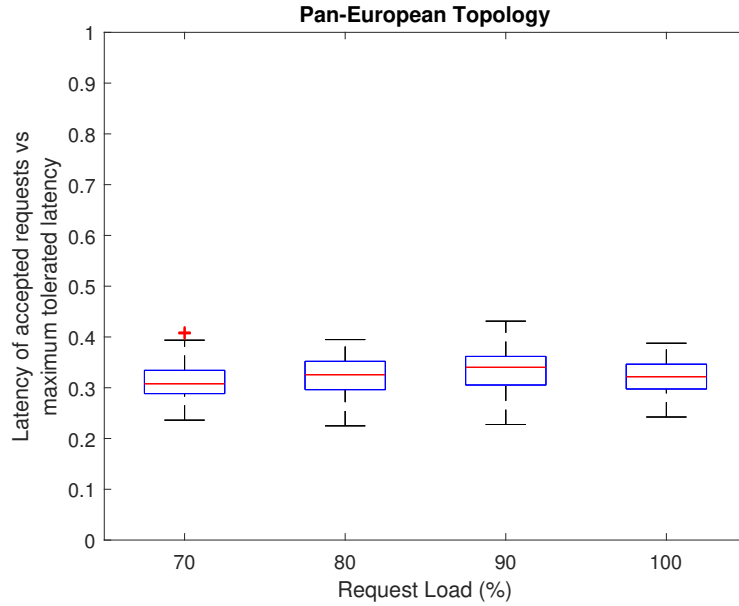


Figure 12: Latency variability across requests - Pan-European topology, 11 DCs

967 *6.7. Request latency vs maximum tolerated latency*

968 We also evaluated the ratio between the latency of accepted requests and
 969 the corresponding maximum tolerated latency at increasing request loads.
 970 Fig. 12 shows that for all request loads the latency of accepted requests is
 971 well below the maximum tolerated one.

972 *6.8. Greedy heuristic*

973 In this section we present the results obtained with a simple greedy heuristic
 974 which works as follows. Requests are considered according to their priority
 975 level so as to manage first the premium ones. Then, for each request, VNFs
 976 are considered in the order they appear in the chain and placed on the first
 977 DC in the ordered list of DCs if the assignment is feasible. DCs are ordered
 978 according to the price they offer so as to consider first the most convenient
 979 DCs. More specifically, for a given request, VNFs are considered one by one
 980 and the placement of a VNF on a DC is feasible only if (i) the DC and the
 981 VNF are compatible, (ii) the DC has enough capacity, (iii) the cost and the
 982 latency of the VNFs currently placed do not exceed their maximum allowed
 983 values, (iv) bandwidth on links is not exceeded. If the assignment is feasible,
 984 network resources are updated accordingly; otherwise, the next DC is

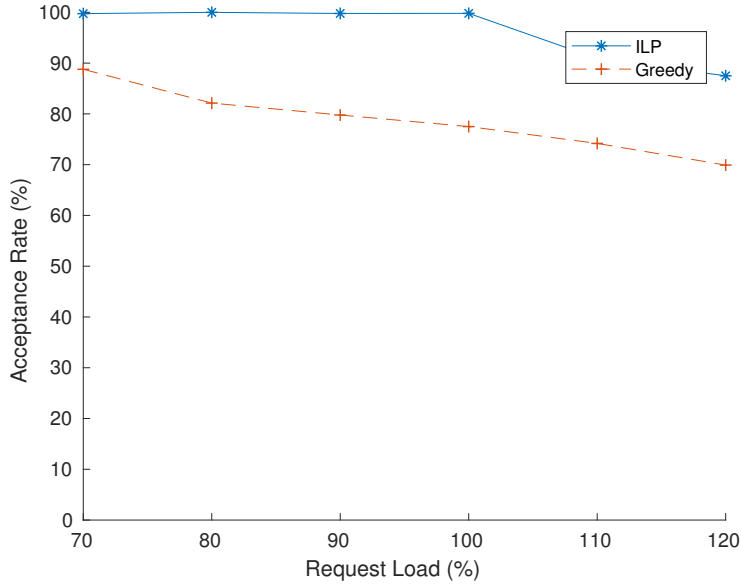


Figure 13: Overall acceptance rate - Pan-European topology, 11 DCs

985 considered until a DC is found or the list of DCs is exhausted. When, for
 986 the considered request, the assignment of a VNF to any DC is infeasible,
 987 the request is discarded and the resources potentially allocated to the previ-
 988 ous VNFs in the chain are restored. We compare our ILP approach with the
 989 greedy heuristic in terms of acceptance rate at increasing request loads. Tests
 990 are performed on a Pan-European network topology of 11 nodes, requests are
 991 generated to vary the request load from 70% to 120% with each VNF in a
 992 chain requiring an amount of resources in the set $\{0.5, 1, 1.5, 2\}$. Our ap-
 993 proach outperforms the greedy one in the acceptance rate of both classes of
 994 requests. Since the heuristic prioritizes placement of premium requests, the
 995 gap between the two approaches (ILP vs greedy) in the acceptance rate for
 996 premium requests is lower than for best effort ones. The execution time of
 997 the greedy heuristic is around 20 ms and remains almost stable, as opposed
 998 to the performance of the ILP approach, characterized by an execution time
 999 that increases with the request load as discussed in Section 6.5 and with an
 1000 average value of 900 ms.

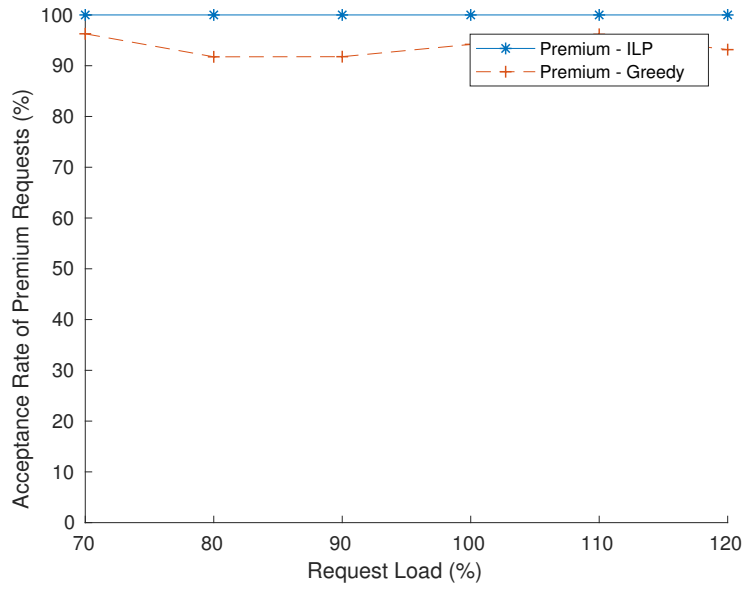


Figure 14: Acceptance rate for Premium requests - Pan-European topology, 11 DCs

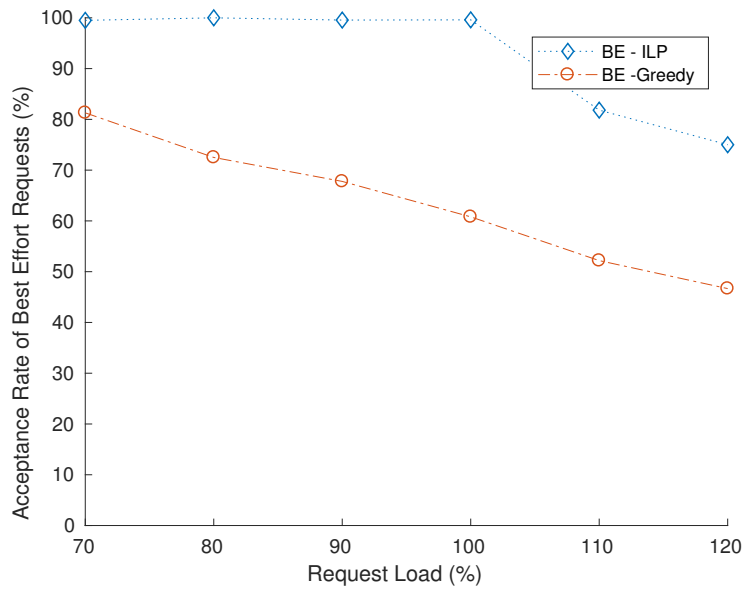


Figure 15: Acceptance rate for Best Effort requests - Pan-European topology, 11 DCs

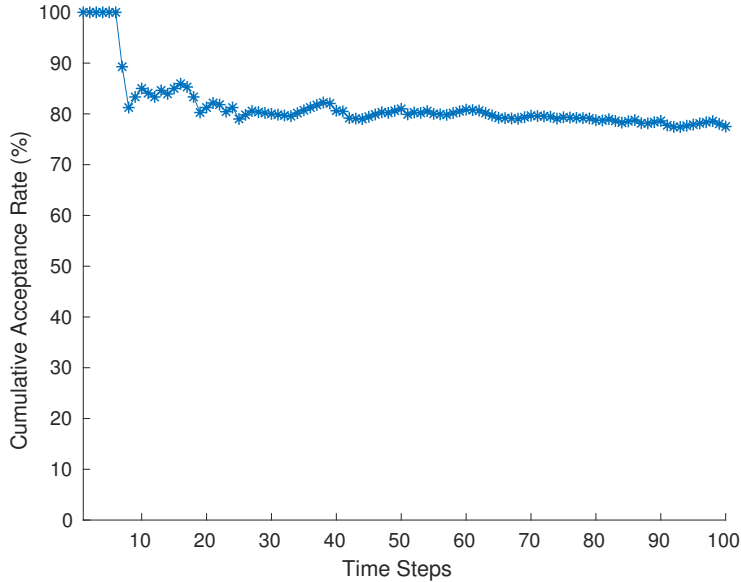


Figure 16: Cumulative acceptance rate - Pan-European topology, 11 DCs

1001 *6.9. Evaluation in an online placement scenario*

1002 In the online placement scenario, at each time step, the algorithm eval-
 1003 uates a batch of b incoming requests. Each VNF in a chain may request an
 1004 amount of resource capacity in the set $\{0.5, 1, 1.5, 2\}$. For each request in the
 1005 set, the service duration time is randomly set in the range of $[1,10]$ timesteps.
 1006 At the end of each time step, the status of the network is updated according
 1007 to deployment choices and the amount of resources of terminating services
 1008 to be released. The tests have been performed considering a Pan-European
 1009 network topology with 11 DCs. The batch size b is set to 4 and requests are
 1010 generated so that the overall request load of the batch is 20 units. Simula-
 1011 tions are run for 100 time steps. The curve of cumulative acceptance rate
 1012 in Fig. 16 shows a trend that, after a few iterations, becomes stable around
 1013 80%. Fig. 17 shows the execution time at each time step, corresponding to
 1014 an average execution time of 61 ms. We consider this value acceptable in
 1015 comparison with network service deployment time (e.g. 40-50 secs ca. [13]).

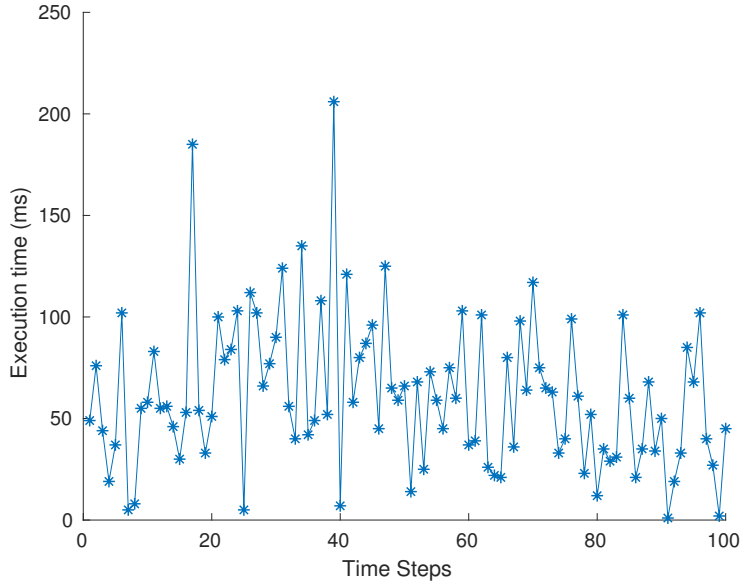


Figure 17: Execution time - Pan-European topology, 11 DCs

1016 **7. Conclusions**

1017 In this paper we presented a novel VNF placement algorithm for embed-
 1018 ding a set of network service requests in a multi-DC physical substrate that
 1019 accounts for multiple stakeholders’ perspective. More specifically, we formu-
 1020 late an ILP-based optimization problem aiming at maximizing primarily ser-
 1021 vice acceptance rate and, secondarily, satisfaction of subscribers preferences,
 1022 while handling different priority levels and guaranteeing QoS objectives’ ful-
 1023 fillment. The problem formulation leverages a layered auxiliary graph built
 1024 considering the characteristics of the physical substrate topology. The layered
 1025 structure of the graph ensures that the order of virtual functions specified
 1026 in the request is preserved. Additional constraints (e.g., maximum allowed
 1027 network latency on the whole path, minimum bandwidth) are taken into
 1028 account during the graph construction phase. Our optimization algorithm
 1029 solves the placement of a batch of requests assumed to be arrived within a
 1030 given time window, however it allows to differentiate services that need a fast
 1031 setup from standard ones.

1032 Experimental evaluation has been carried out through extensive testings.
 1033 We showed that the proposed algorithm is effective in maximizing the service

1034 acceptance rate for offline and online placement problems and we compared
1035 two different subscribers' preference assignment strategies. In regards to ef-
1036 ficiency, we evaluated how the computational time varies with the request
1037 load and topology size, demonstrating that computational time limits of 2,
1038 3 and 5 seconds lead to solutions that are very close to the optimal one.
1039 Test results also show that the proposed approach fairly distributes the over-
1040 all request load across available DCs. Finally, we compared our ILP-based
1041 approach with a greedy heuristic, which shows a faster execution time but
1042 penalizes best effort requests.

1043 We plan to extend this work in a number of ways. We plan to further
1044 study the layered graph building step on top of the physical network topology
1045 to more robustly handle the dynamic change of topology characteristics (e.g.,
1046 available bandwidth). We also plan to improve the formulation of a request's
1047 expected latency by extending the model to consider link transmission delays
1048 as well as delay introduced by VNFs (i.e., VNF processing delay). We also
1049 plan to evaluate our placement approach in a multi-DC testbed. To this pur-
1050 pose, we are developing a Service Request Manager component that manages
1051 the deployment of network services on top of a multi-DC environment lever-
1052 aging the proposed placement algorithm. The placement decision is used to
1053 appropriately compose a Network Service Description file that is sent to a
1054 NFV Orchestrator for actual network service deployment, in compliance with
1055 ETSI standard specifications. In order to accomplish service deployment in
1056 the physical infrastructure, the Service Request Manager will interface with
1057 some existing implementations of NFV Orchestrator (e.g., OpenBaton [60])
1058 and Virtual Infrastructure Management components (e.g., OpenStack [61]).

1059 8. References

- 1060 [1] ETSI, Network Function Virtualization - Introductory White Paper,
1061 http://portal.etsi.org/NFV/NFV_White_Paper.pdf, 2012.
- 1062 [2] B. Martini, F. Paganelli, A Service-Oriented Approach for Dynamic
1063 Chaining of Virtual Network Functions over Multi-Provider Software-
1064 Defined Networks, *Future Internet* 8 (2016).
- 1065 [3] D. Kreutz, F. M. Ramos, P. E. Verissimo, C. E. Rothenberg, S. Azodol-
1066 molky, S. Uhlig, Software-defined networking: A comprehensive survey,
1067 *Proceedings of the IEEE* 103 (2015) 14–76.

- 1068 [4] F. Paganelli, M. Ulema, B. Martini, Context-aware service composition
1069 and delivery in NGSONs over SDN, *IEEE Communications Magazine*
1070 52 (2014) 97–105.
- 1071 [5] W. Stallings, *Foundations of Modern Networking: SDN, NFV, QoE,*
1072 *IoT, and Cloud*, Pearson Education, 2015.
- 1073 [6] M. K. Weldon, *The future X network: a Bell Labs perspective*, CRC
1074 press, 2016.
- 1075 [7] S. Vassilaras, L. Gkatzikis, N. Liakopoulos, I. N. Stiakogiannakis, M. Qi,
1076 L. Shi, L. Liu, M. Debbah, G. S. Paschos, The algorithmic aspects of
1077 network slicing, *IEEE Communications Magazine* 55 (2017) 112–119.
- 1078 [8] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, V. Young, Mobile
1079 edge computing A key technology towards 5G, ETSI white pa-
1080 per, [http://www.etsi.org/images/files/ETSIWhitePapers/etsi_](http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf)
1081 [wp11_mec_a_key_technology_towards_5g.pdf](http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf), 2015.
- 1082 [9] Catalyst project: Maximizing profitability with network functions
1083 virtualization, TeleManagement Forum White Paper, [https://www.](https://www.viavisolutions.com/fr-fr/node/50121)
1084 [viavisolutions.com/fr-fr/node/50121](https://www.viavisolutions.com/fr-fr/node/50121), 2014.
- 1085 [10] International Telecommunication Union, Quality of telecommunica-
1086 tion services: concepts, models, objectives and dependability plan-
1087 ning Use of quality of service objectives for planning of telecom-
1088 munication networks, E.860, [https://www.itu.int/rec/T-REC-E.](https://www.itu.int/rec/T-REC-E.860-200206-I/en)
1089 [860-200206-I/en](https://www.itu.int/rec/T-REC-E.860-200206-I/en), 06 2002.
- 1090 [11] A. Veitch, Use Cases and Analysis on Integrated NFV and Network
1091 Optimization, IETF Internet-Draft, [https://tools.ietf.org/html/](https://tools.ietf.org/html/draft-veitch-nfvrg-nfv-nw-optimization-00)
1092 [draft-veitch-nfvrg-nfv-nw-optimization-00](https://tools.ietf.org/html/draft-veitch-nfvrg-nfv-nw-optimization-00), 2017.
- 1093 [12] J. Evans, A. Afrakteh, R. Xiu, Demand Engineering: An new approach
1094 to SDN-based traffic management for IP and MPLS networks, CoRR
1095 abs/1606.04720 (2016).
- 1096 [13] M. Mechtri, C. Ghribi, O. Soualah, D. Zeghlache, NFV Orchestration
1097 Framework addressing SFC Challenges, *IEEE Communications Maga-*
1098 *zine* 55 (2017) 16–23.

- 1099 [14] J. Liu, Y. Li, Y. Zhang, L. Su, D. Jin, Improve Service Chaining Per-
1100 formance with Optimized Middlebox Placement, *IEEE Transactions on*
1101 *Services Computing* 10 (2017) 560–573.
- 1102 [15] S. Draxler, H. Karl, Specification, composition, and placement of net-
1103 work services with flexible structures, *International Journal of Network*
1104 *Management* 27 (2017).
- 1105 [16] D. Bhamare, M. Samaka, A. Erbad, R. Jain, L. Gupta, H. A. Chan, Op-
1106 timal virtual network function placement in multi-cloud service function
1107 chaining architecture, *Computer Communications* 102 (2017) 1 – 16.
- 1108 [17] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, O. C. M. B. Duarte,
1109 Orchestrating virtualized network functions, *IEEE Transactions on Net-*
1110 *work and Service Management* 13 (2016) 725–739.
- 1111 [18] M. Mechtri, C. Ghribi, D. Zeghlache, A scalable algorithm for the place-
1112 ment of service function chains, *IEEE Transactions on Network and*
1113 *Service Management* 13 (2016) 533–546.
- 1114 [19] A. Leivadreas, M. Falkner, I. Lambadaris, G. Kesidis, Optimal virtual-
1115 ized network function allocation for an SDN enabled cloud, *Computer*
1116 *Standards & Interfaces* 54 (2017) 266–278.
- 1117 [20] A. Gadre, A. Anbiah, K. M. Sivalingam, A customizable agile approach
1118 to Network Function Placement, in: 2017 European Conference on
1119 *Networks and Communications*, EuCNC 2017, pp. 1–6.
- 1120 [21] C. Pham, N. H. Tran, S. Ren, W. Saad, C. S. Hong, Traffic-aware and
1121 Energy-efficient vNF Placement for Service Chaining: Joint Sampling
1122 and Matching Approach, *IEEE Transactions on Services Computing*
1123 (2017).
- 1124 [22] S. Dräxler, H. Karl, Z. . Mann, Jasper: Joint optimization of scaling,
1125 placement, and routing of virtual network services, *IEEE Transactions*
1126 *on Network and Service Management* (2018) 1–1.
- 1127 [23] ETSI, Network Function Virtualization Management and Orchestration,
1128 GS NFV-MAN 001 V1.1.1, [http://www.etsi.org/deliver/etsi_gs/
1129 NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_NFV-MAN001v010101p.pdf),
1130 2014.

- 1131 [24] ETSI, Network function virtualization Ecosystem - Report on
1132 SDN Usage in NFV Architectural Framework, GS NFV-EVE 005
1133 V1.1.1, [http://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/
1134 005/01.01.01_60/gs_nfv-eve005v010101p.pdf](http://www.etsi.org/deliver/etsi_gs/NFV-EVE/001_099/005/01.01.01_60/gs_nfv-eve005v010101p.pdf), 2015.
- 1135 [25] B. Martini, F. Paganelli, A. A. Mohammed, M. Gharbaoui, A. Sgam-
1136 belluri, P. Castoldi, SDN controller for context-aware data delivery in
1137 dynamic service chaining, in: Proceedings of the 2015 1st IEEE Con-
1138 ference on Network Softwarization (NetSoft), pp. 1–5.
- 1139 [26] F. Liberati, A. Giuseppe, A. Pietrabissa, V. Suraci, A. Di Giorgio,
1140 M. Trubian, D. Dietrich, P. Papadimitriou, F. Delli Priscoli, Stochastic
1141 and exact methods for service mapping in virtualized network infras-
1142 tructures, *International Journal of Network Management* 27 (2017).
- 1143 [27] Z. Xu, W. Liang, A. Galis, Y. Ma, Throughput maximization and
1144 resource optimization in NFV-enabled networks, in: 2017 IEEE Inter-
1145 national Conference on Communications (ICC), pp. 1–7.
- 1146 [28] M. C. Luizelli, W. L. da Costa Cordeiro, L. S. Buriol, L. P. Gasparry,
1147 A fix-and-optimize approach for efficient and large scale virtual network
1148 function placement and chaining, *Computer Communications* 102 (2017)
1149 67 – 77.
- 1150 [29] Y. Wang, P. Lu, W. Lu, Z. Zhu, Cost-efficient virtual network func-
1151 tion graph (vnfg) provisioning in multidomain elastic optical networks,
1152 *Journal of Lightwave Technology* 35 (2017) 2712–2723.
- 1153 [30] A. Gupta, B. Jaumard, M. Tornatore, B. Mukherjee, A Scalable Ap-
1154 proach for Service Chain Mapping With Multiple SC Instances in a
1155 Wide-Area Network, *IEEE Journal on Selected Areas in Communica-*
1156 *tions* 36 (2018) 529–541.
- 1157 [31] A. Gupta, M. F. Habib, U. Mandal, P. Chowdhury, M. Tornatore,
1158 B. Mukherjee, On service-chaining strategies using virtual network func-
1159 tions in operator networks, *Computer Networks* 133 (2018) 1 – 16.
- 1160 [32] S. Ayoubi, S. Sebbah, C. Assi, A cut-and-solve based approach for
1161 the vnf assignment problem, *IEEE Transactions on Cloud Computing*
1162 (2017) 1–1.

- 1163 [33] S. Khebbache, M. Hadji, D. Zeglache, Virtualized network functions
1164 chaining and routing algorithms, *Computer Networks* 114 (2017) 95 –
1165 110.
- 1166 [34] J. Altmann, M. M. Kashef, Cost model based service placement in
1167 federated hybrid clouds, *Future Gener. Comput. Syst.* 41 (2014) 79–90.
- 1168 [35] B. Naudts, M. Flores, R. Mijumbi, S. Verbrugge, J. Serrat, D. Colle, A
1169 dynamic pricing algorithm for a network of virtual resources, *International Journal of Network Management* 27 (2017).
- 1171 [36] D. Azevedo, M. Patterson, J. Pouchet, R. Tipley, Carbon usage ef-
1172 fectiveness (CUE): a green grid data center sustainability metric,
1173 *The green grid*, White Paper, [http://www.thegreengrid.org/en/
1174 library-andtools.aspx](http://www.thegreengrid.org/en/library-andtools.aspx), 2010.
- 1175 [37] A. Khosravi, S. K. Garg, R. Buyya, Energy and carbon-efficient place-
1176 ment of virtual machines in distributed cloud data centers, in: *European
1177 Conference on Parallel Processing*, Springer, pp. 317–328.
- 1178 [38] K. Kaur, T. Dhand, N. Kumar, S. Zeadally, Container-as-a-service at
1179 the edge: Trade-off between energy efficiency and service availability at
1180 fog nano data centers, *IEEE Wireless Communications* 24 (2017) 48–56.
- 1181 [39] B. Zhang, J. Hwang, T. Wood, Toward online virtual network function
1182 placement in software defined networks, in: *2016 IEEE/ACM 24th
1183 International Symposium on Quality of Service (IWQoS)*, IEEE, pp.
1184 1–6.
- 1185 [40] I. Patiniotakis, Y. Verginadis, G. Mentzas, Pulsar: preference-based
1186 cloud service selection for cloud service brokers, *Journal of Internet
1187 Services and Applications* 6 (2015) 26.
- 1188 [41] A. Amokrane, M. F. Zhani, Q. Zhang, R. Langar, R. Boutaba, G. Pu-
1189 jolle, On satisfying green slas in distributed clouds, in: *10th International
1190 Conference on Network and Service Management (CNSM) and
1191 Workshop*, pp. 64–72.
- 1192 [42] N. Joy, K. Chandrasekaran, A. Binu, Energy aware sla and green cloud
1193 federations, in: *2016 IEEE Distributed Computing, VLSI, Electrical
1194 Circuits and Robotics (DISCOVER)*, pp. 7–11.

- 1195 [43] ETSI, Network Functions Virtualisation (NFV); Use Cases, ETSI GS
1196 NFV 001 (V1.1.1), http://www.etsi.org/deliver/etsi_gs/nfv/001_099/001/01.01.01_60/gs_nfv001v010101p.pdf, 2013.
- 1198 [44] J. Babiarz, K. Chan, F. Baker, Configuration guidelines for diffserv
1199 service classes, Internet Requests for Comments, RFC 4594, 2006.
- 1200 [45] ETSI, Network Functions Virtualisation (NFV) Release 3; Manage-
1201 ment and Orchestration; Report on architecture options to sup-
1202 port multiple administrative domains, ETSI GR NFV-IFA 028
1203 V3.1.1, http://www.etsi.org/deliver/etsi_gr/NFV-IFA/001_099/028/03.01.01_60/gr_NFV-IFA028v030101p.pdf, 2018.
- 1205 [46] S. Clayman, E. Maini, A. Galis, A. Manzalini, N. Mazzocca, The dy-
1206 namic placement of virtual network functions, in: 2014 IEEE Network
1207 Operations and Management Symposium (NOMS), pp. 1–9.
- 1208 [47] S. Zhang, Z. Qian, Z. Luo, J. Wu, S. Lu, Burstiness-aware resource reser-
1209 vation for server consolidation in computing clouds, *IEEE Transactions*
1210 *on Parallel and Distributed Systems* 27 (2016) 964–977.
- 1211 [48] Docker web site, <https://www.docker.com>, 2018. Accessed: 2018-03-
1212 20.
- 1213 [49] S. Martello, P. Toth, Knapsack Problems, Wiley & Sons, Chichester,
1214 1990.
- 1215 [50] M. R. Garey, D. S. Johnson, Computers and intractability: a Guide
1216 to the Theory of NP-Completeness, W.H. Freeman and Company, San
1217 Francisco, 1979.
- 1218 [51] M. R. Garey, D. S. Johnson, “strong” np-completeness results: Moti-
1219 vation, examples, and implications, *J. ACM* 25 (1978) 499–508.
- 1220 [52] N. Garg, V. V. Vazirani, M. Yannakakis, Primal-dual approximation
1221 algorithms for integral flow and multicut in trees, *Algorithmica* 18 (1997)
1222 3–20.
- 1223 [53] B. Martini, F. Paganelli, P. Cappanera, S. Turchi, P. Castoldi, Latency-
1224 aware composition of Virtual Functions in 5G, in: 2015 1st IEEE Con-
1225 ference on Network Softwarization (NetSoft), IEEE, pp. 1–6.

- 1226 [54] C. Li, L. Y. Li, Optimal resource provisioning for cloud computing
1227 environment, *The Journal of Supercomputing* 62 (2012) 989–1022.
- 1228 [55] A. Betker, C. Gerlach, R. Hülsermann, M. Jäger, M. Barry, S. Bodamer,
1229 J. Späth, C. Gauger, M. Köhn, Reference transport network scenarios,
1230 MultiTeraNet Report (2003).
- 1231 [56] N. Huin, B. Jaumard, F. Giroire, Optimization of Network Service Chain
1232 Provisioning, in: *IEEE International Conference on Communications*
1233 2017, Paris, France.
- 1234 [57] CISCO, Cisco visual networking index: Forecast and methodology, 2015.
- 1235 [58] M. Savi, M. Tornatore, G. Verticale, Impact of processing costs on ser-
1236 vice chain placement in network functions virtualization, in: *Network*
1237 *Function Virtualization and Software Defined Network (NFV-SDN)*,
1238 2015 IEEE Conference on, IEEE, pp. 191–197.
- 1239 [59] L. Askari, A. Hmaity, F. Musumeci, M. Tornatore, Virtual-network-
1240 function placement for dynamic service chaining in metro-area networks,
1241 in: *2018 International Conference on Optical Network Design and Mod-*
1242 *eling (ONDM)*, IEEE.
- 1243 [60] G. A. Carella, M. Pauls, T. Magedanz, M. Cilloni, P. Bellavista, L. Fos-
1244 chini, Prototyping nfv-based multi-access edge computing in 5G ready
1245 networks with Open Baton, in: *2017 IEEE Conference on Network*
1246 *Softwarization (NetSoft)*, pp. 1–4.
- 1247 [61] Openstack: Open source cloud computing software, [https://www.](https://www.openstack.org/)
1248 [openstack.org/](https://www.openstack.org/), 2018. Accessed: 2018-03-20.