Computer Networks 125 (2017) 64–75

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

On the energy cost of robustness for green virtual network function placement in 5G virtualized infrastructures

Antonio Marotta[a,*], Fabio D'Andreagiovanni[b], Andreas Kassler[a], Enrica Zola[c]

[a] Karlstad University, Universitetsgatan 2, Karlstad, 65188, Sweden
[b] Sorbonne Universités, Université de Technologie de Compiègne, CNRS, Heudiasyc UMR 7253, CS 60319, Compiègne 60203, France
[c] Universitat Politècnica de Catalunya, C. Jordi Girona, 1–3, Barcelona 08034, Spain

ARTICLE INFO

Article history:
Received 15 October 2016
Revised 3 March 2017
Accepted 18 April 2017
Available online 25 April 2017

Keywords:
Virtualization
Binary linear programming
Robust optimization
Network function virtualization (NFV)
EPC
5G

ABSTRACT

Next generation 5G networks will rely on virtualized Data Centers (vDC) to host virtualized network functions on commodity servers. Such Network Function Virtualization (NFV) will lead to significant savings in terms of infrastructure cost and reduced management complexity. However, green strategies for networking and computing inside data centers, such as server consolidation or energy aware routing, should not negatively impact the quality and service level agreements expected from network operators. In this paper, we study how robust strategies that place virtual network functions (VNF) inside vDC impact the energy savings and the protection level against resource demand uncertainty. We propose novel optimization models that allow the minimization of the energy of the computing and network infrastructure which is hosting a set of service chains that implement the VNFs. The model explicitly provides for robustness to unknown or imprecisely formulated resource demand variations, powers down unused routers, switch ports and servers, and calculates the energy optimal VNF placement and network embedding also considering latency constraints on the service chains. We propose both exact and heuristic methods. Our experiments were carried out using the virtualized Evolved Packet Core (vEPC), which allows us to quantitatively assess the trade-off between energy cost, robustness and the protection level of the solutions against demand uncertainty. Our heuristic is able to converge to a good solution in a very short time, in comparison to the exact solver, which is not able to output better results in a longer run as demonstrated by our numerical evaluation. We also study the degree of robustness of a solution for a given protection level and the cost of additional energy needed because of the usage of more computing and network elements.

1. Introduction

Telecom Service Providers are in the process of migrating vendor specific hardware and software that implement their network functions towards the Cloud. Virtualizing their infrastructure such as load-balancers, firewalls or the whole Evolved Packet Core (EPC), and deploying them in virtualized data centers (vDC) leads to the concept of Network Function Virtualization (NFV) [1], where Virtualized Network Functions (VNFs) run inside Virtual Machines (VMs) under the control of a hypervisor on commodity servers. This will dramatically reduce the cost of the infrastructure and simplify deployment of new services. By changing VM resources dynamically (e.g. by adding more computing or memory resources, adding more VMs), the VNFs may be scaled according to the load, which significantly simplifies the VNF operation and management and drastically reduces costs of operation. Virtualization enables resources consolidation, since more VMs may reside on the same physical server leading towards green strategies inside a data center. For example, server consolidation tries to migrate the VMs towards the fewest possible number of servers and consequently powers down unused ones to save energy. However, the more VMs are hosted by the same physical machine, the higher the potential for contention for e.g. CPU and, thus, the possibility of Service Level Agreement (SLA) violations. As VNFs are composed of a set of VNF Components (VNFC) that need to exchange data over the network under capacity and latency constraints, the networking plays also an important part. Deploying each VNFC on a different server may result in lower SLA violation due to CPU contention but will increase the energy cost due to more active resources and additional traffic exchanged, leading to higher router and link utilization, network contention and increased energy cost for the net-

* Corresponding author.
E-mail addresses: antonio.marotta@live.it, d.andreagiovann@hds.utc.fr, andreas.kassler@kau.se, enrica@entel.upc.edu (A. Marotta).

http://dx.doi.org/10.1016/j.comnet.2017.04.045
1389-1286/

work. By using Software Defined Networking, one can dynamically adjust the network topology and available capacity by powering down unused switch ports or routers that are not needed to carry a certain traffic volume [2] and re-route the flows to consume the least amount of energy at a potential expense of higher latency.

In order to save the most energy, reduce the electricity costs and the CO_2 footprint, it is evident to place the VNF components in the smallest number of servers and adjust the network topology and capacity to match the demands of the VNFCs. Such design of the VNF placement and network embedding can be formulated as a mathematical optimization problem, which pursues the optimization of an objective function expressing the aim of the data center administrator, while respecting a set of feasibility constraints that express the technical constraints of the computing and network infrastructure and the requirements of the users. Unfortunately, many parameters in such optimization problem are not known precisely when the problem is solved. For instance, it is hard to predict how much CPU a VNF will require or how much data a VNF v_i will send towards a VNF v_j during its execution time. The presence of uncertain data in an optimization problem can be very tricky: even small variations in the input parameters of an optimization problem may have very bad effects, turning optimal solutions into solutions of bad quality and even turning feasible solutions into infeasible ones that are thus useless in practice [3–5]. For example, if the CPU demands of a set of VNFCs allocated on the same server require more CPU than the expected amount, contention for CPU may occur which may result in SLA violation and service degradation for the customer.

The fundamental question that we address in this paper is whether it is possible to place a set of VNF Components in a robust way inside a virtualized data center while trying to minimize the energy consumption, given we do not know the input to the problem precisely. In particular, our main original contributions are the following. We propose an original robust optimization model that jointly optimizes VNF placement and routing in virtual networks and tackles variations in the resource demand of VNFCs. The model takes into account traffic demands and allows the specification of latency constraints for VNF service chains. Our model improves our recent work [6], proposing a new purely binary linear programming formulation which has reduced computational complexity. Moreover, we propose a fast variable fixing heuristic that exploits structural information coming from the linear relaxation of the problem. The solution of the heuristic can be used to warm-start the solution process of the solver, accelerating the convergence towards the optimum. We applied our heuristic to the vEPC deployment and our numerical results demonstrate that it is able to find a good solution in a very short time in comparison to the exact solver, which is not able to output better results even in a longer run, as demonstrated by our numerical evaluation. We also study the degree of robustness of a solution for a given protection level and the cost of additional energy needed because of the usage of more computing resources and network elements.

The remainder of the paper is organized as follows. In Section 2, we review the state of the art and point out the novelties of our work. Section 3 introduces our methodology. In Section 4, we present a robust optimization approach that is based on the theory of Γ-Robustness to cope with demand uncertainties for the green VNF placement and network embedding problem. Section 5 details our heuristic to solve the optimization problem fast. The computational results are presented in Section 6 and in Section 7 we derive conclusions and point out ideas for future work.

2. Related work

The need for adaptability and flexibility in the future network architectures (e.g., 5G) paves the way for Network Function Virtualization, a new concept that Telecom Service Providers are incrementally deploying to address their customers' demands. The gains in energy efficiency and flexibility enabled by virtualization have recently led the research community to study the VNF placement problem in depth. Authors in [7] define a generic VNF chain routing optimization problem and devise a MILP formulation. Larumbe and Sansò [8] proposes a dynamic optimization problem that can be used as a meta-scheduler to place and re-place VMs in the right cloud data centers in real-time, considering costs, QoS, energy consumption, and CO_2 emissions. Authors in [9] consider traffic flows with deadlines and formulate a mathematical problem for mapping and scheduling flows to VNFs in the most energy efficient way. The proposed heuristic generates good results in reasonable time. Zheng et al. [10] presents a novel solution to the VM consolidated placement problem that uses the biogeography-based optimization technique to optimize the VM placement, thus minimizing both the resource wastage and the power consumption at the same time.

In [11], the authors present an optimization model for the embedding of Virtual Mobile Core Networks. In their formulation latency is nicely modelled as a combination of processing, packet queuing and propagation delay, where the first two variables depend on the traffic utilization of the node the VNF is placed on, while the last one is a function of the path length. Authors show numerical results of the model on a real network topology. Bari et al. [12] presents an Integer Linear Programming (ILP) model for VNF orchestration. The problem consists in finding the number of necessary VNFs and allocating them in order to minimize the total network related cost and the resources fragmentation. In order to solve larger instances, the authors propose a dynamic programming-based heuristic.

However, common to all these works is the assumption that input data is known precisely. As recently highlighted in [13], conventional optimization models hardly take into account uncertainties in the spatial distribution of demands, temporal variations of associated traffic flow properties, or the changes that arise in the underlying network topology. Consequently, ignoring uncertainty in input data can lead to solutions which are suboptimal or even infeasible [3,4,14]. Authors in [15] show how the emerging area of robust optimization can advance the network planning by a more accurate mathematical description of the demand uncertainty. This concept is applied in [16], where the VM consolidation problem is modeled as a robust MILP and the resource requirements of the VMs are allowed to variate between specific bounds. The price of the robustness is quantified in terms of energy saving against resource requirement violations. However, the robustness in the network and VNF service chains is not studied there.

Robust Optimization has been considered in [17] for virtual network embedding (VNE), namely the problem where a virtual network must be mapped to a physical network substrate. The objective is to maximize the revenue that comes from the embedding of virtual nodes and links with a constraint on the capacity budget. In order to solve large instances, the authors propose a two-phase heuristic based on Γ-robustness to deal with capacity requests variability. Again, they do not model service chains and resource demands for VNF components.

Regarding the joint robust VNF placement and network embedding, there has not been much work. Recently, Marotta and Kassler [6] proposes a joint robust placement and network embedding problem assuming that resource demands of VNF components are not known precisely. They model the problem using a set of service chains that are embedded into a network graph and consider the latency for service chains, link capacities of the network and CPU, memory and disc capacities of the computing infrastructure as constraints. They apply the theory of Γ-Robustness to cope with demand uncertainties for individual VNF components and study the exact solutions which may be computationally very expensive to obtain. In this paper, we improve [6] by proposing a new purely binary linear programming formulation which has reduced computational complexity. Moreover, we propose a fast variable fixing heuristic suitable for online optimization that exploits structural information coming from the linear relaxation of the problem.

3. Methodology

In this paper we investigate whether it is possible to place a set of service chains in a robust way inside a virtualized data center while trying to minimize the energy consumption. The proposed robust optimization model jointly optimizes VNF placement and routing in virtual networks and tackles variations in the resource demand of VNFCs. First, the purely binary linear optimization model is introduced in Section 4.1, where a set of VNFCs have to be allocated into the available servers, each of which is connected to different routers in the network. In this first model, we assume perfect knowledge of the amount of resources available at each server, of the amount of resources requested by each VNFC, of the power consumption of each router, of the traffic demands between the VNFCs, of the bandwidth of each link and of its maximum latency.

Motivated by the natural uncertainty of traffic conditions in telecommunication networks, we take a step further and propose a modification on the first model which takes into account the variability of the resource requests of VNFs (see Section 4.2). The robust version of the problem, which follows the theory of Γ-Robustness, is presented in Section 4.3. However, the solution to this problem may require significant computational resources and time, thus making it not suitable for online optimization, especially when the problem size grows (i.e., large network).

Thus, as a third step, a fast variable fixing heuristic that exploits structural information coming from the linear relaxation of the problem is also proposed and presented in Section 5. The solution of the heuristic can be used to warm-start the solution process of the solver, accelerating the convergence towards the optimum. Through the proposed heuristic, the VNFCs can be placed inside a virtualized data center in a robust way, thus guaranteeing that the solution remains feasible disregarding the variability in the resource demands.

4. Problem formulation

In this paper, we focus on an optimization problem that we call *Power Efficient VNF Placement and Flow Routing* (Eff-VNF), which is defined as follows. We consider a set S of servers, each of which characterized by a peculiar linear power profile and a maximum amount of available resources (e.g., individual CPU, memory and disc capacities, denoted as CPU, RAM, DISC - we also denote the set of such different type of resources by R). We model the network topology by a graph $G(N, L)$, where N is the set of network nodes and L is the set of links. Each link $\ell \in L$ corresponds to a pair (i, j) with $i, j \in N$: $i \neq j$. For each server $s \in S$, we denote by $n(s) \in N$ the network node to which s is connected to. V is the set of VNFCs we intend to place on the hardware resources of the VNI. $\mathcal{C}$ is a family of sets representing the set of service chains. Each $C \in \mathcal{C}$ is an ordered subset of $V \times V$ that represent the sequence of VNFCs included in a service chain. Every C contains couples (v_1, v_2) with $v_1, v_2 \in V$ and is associated with its own demands and latency bounds.

The objective of the problem Eff-VNF is to *find the optimal allocation of all the service chains on the physical servers and, consequently, the flow routing for all the traffic demands, so that the total power consumption is minimized, while satisfying the constraints on the server resources (CPU, RAM, DISC) and link capacities, as well as the latency bounds for each service chain.*

Table 1
Model parameters and decision variables.

Input parameters	
$G(N, L)$	network graph (N, L are the set of nodes and links, respectively)
S	set of servers
V	set of VNFCs
$\mathcal{C}$	set of service chains
R	set of resources
$n(s)$	is the network node to which server s is connected to
a_{rs}	is the amount of resource r available at server s
a_{vr}	is the amount of resource r requested by VNFC v
P_n	is the static power consumption of node n
P_{ij}	is the static power consumption of link (i, j)
P_s^{min}, P_s^{max}	are the idle and maximum power consumption of server s
b^{v_1,v_2}	is the traffic demand between v_1 and v_2
b_{ij}	is the bandwidth of the link (i, j)
l_{ij}	is the latency of the link (i, j)
$l_C^{v_1,v_2}$	is the maximum latency tolerable by (v_1, v_2) of service chain C
Decision variables:	
x_{vs}	is 1 if VNFC v is allocated to server s, 0 otherwise
y_s	is 1 if server s is active, 0 otherwise
z_n	is 1 if node n is active, 0 otherwise
$f_{ij}^{v_1,v_2}$	is 1 if the traffic demand b^{v_1,v_2} is forwarded on link (i, j)
g_{ij}	is 1 if the link (i, j) is used for transmitting any traffic

Fig. 1 illustrates the problem where we have in total seven servers (s_1 until s_7), each one with its own power profile (each server s has its own idle power P_s^{min} and maximum power consumption P_s^{max}) and individual CPU, memory and disc capacities. In the example given, server s_i has installed a_{1i} CPU, a_{2i} RAM and a_{3i} DISC. Each server is connected to an unique router (for example, s_1 is connected to n_1). Each link has a dedicated capacity and latency (for example, the latency for the link between n_1 and n_2 is denoted as l_{12} - we omit bandwidth from Fig. 1 to maintain readability). The servers, their capacities, together with the network nodes and links with their capacities form the NFV Infrastructure in terms of Computing Power, Storage and Network. In our example, we should embed into this NFV Infrastructure three service chains (denoted as c_1, c_2 and c_3), each one with their own latency bounds. In total, we have three different VNFCs (v_1, v_2 and v_3) and we assume that the traffic source for c_1 is the Sender S_1, which is connected to router n_2 and injects a certain volume of traffic into the service chain towards v_1. v_1 processes the packets (for which it needs CPU, memory and disc) and forwards the processed traffic (which may have a different volume than the one injected) towards VNFC v_2, which again processes it and forwards a certain volume to the destination D_1 that is connected to router n_2. Note that Fig. 1 assumes additional source/sink nodes where traffic for a service chain is created/terminated, which are not explicitly mentioned in our model but they could be introduced by adding network nodes. The figure depicts an exemplary VNF placement and network embedding into the physical substrate network. For example, the VNFC v_1 would be placed onto server s_3, v_3 onto server s_4 and so on. Servers hosting no VNFC would be powered down (e.g., s_1, s_2 or s_5) together with all the nodes not carrying any traffic (e.g., only n_1 in this case).

4.1. Binary optimization model

In Table 1 all the parameters and the decision variable of the optimization problem (Eff-VNF) are explained.

The power consumption of each server $s \in S$ is linearly increasing according to the CPU utilization (given by the sum of the

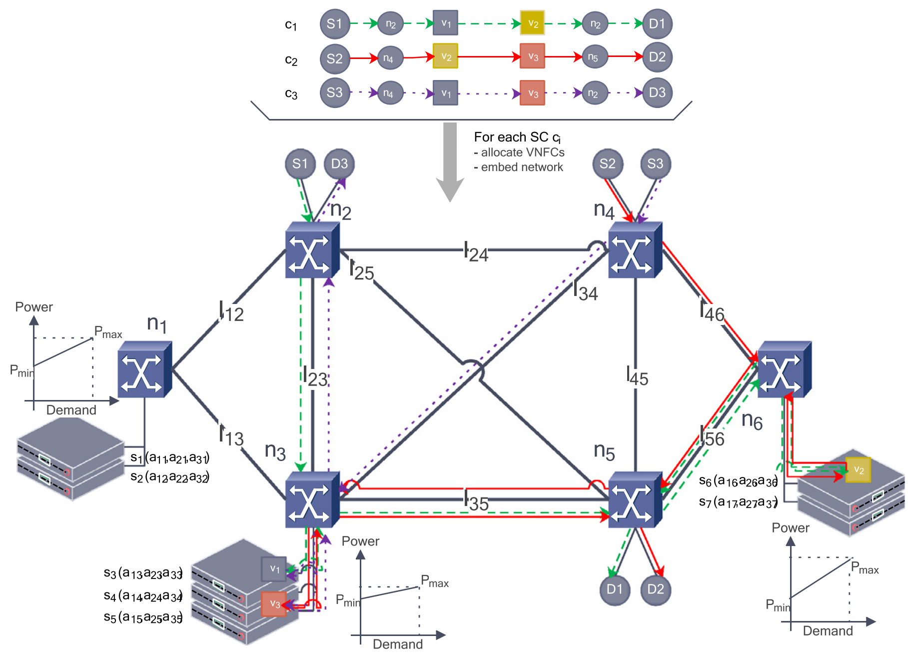


Fig. 1. The joint VNF placement and network embedding problem.

CPU demands of the VNFCs allocated to the server) in the range $[P_s^{\min}, P_s^{\max}]$. The idle power consumption of each activated node $n \in N$ is P_n, whereas the power consumption of an activated link $(i, j) \in L$ is P_{ij}. Each server s has an amount a_{rs} of available resource r; instead a_{vr} is the amount of resource r requested by VNFC v. The bandwidth requested for the data transfer of VNFC couple (v_1, v_2) is b^{v_1,v_2}. b_{ij} is the bandwidth of link (i, j) and l_{ij} is the latency of link (i, j). We denote by $l_C^{v_1,v_2}$ the maximum latency allowed for each service chain $C \in \mathcal{C}$ and couple $(v_1, v_2) \in C$.

The complete Binary Linear Programming problem that we define to model the problem (Eff-VNF) and that we denote by the acronym BLP is presented in Table 2.

The VNFC-server allocation variables $x_{vs} \in \{0, 1\}, \forall v \in V, s \in S$ are equal to 1 if VNFC v is allocated to server s and 0 otherwise. The server activation variable $y_s \in \{0, 1\}, \forall s \in S$ is 1 if server s is active and is 0 otherwise (then the server is powered off). The activation of a network node is represented through the decision variables $z_n \in \{0, 1\}, \forall n \in N$. If all ports of a network node are not carrying traffic, then the node is powered down. If a single port is carrying traffic through a given link, then the node is activated and powered on. A link activation variable $g_{ij} \in \{0, 1\}, \forall (i, j) \in L$ is equal to 1 if link (i, j) is used for carrying traffic and 0 otherwise. In the proposed model, we consider single-path transmissions (i.e., traffic exchanged between two network entities cannot be sent on multiple parallel paths) modelled through an unsplittable flow problem (see [5,18] for an introduction to splittable and unsplittable flow concepts): for this reason, the variables $f_{ij}^{v_1,v_2} \in \{0, 1\}, \forall (i, j) \in L, (v_1, v_2) \in \bigcup_{C \in \{\mathcal{C}\}} C$ are binary and a generic variable $f_{ij}^{v_1,v_2}$ equals 1 if the *entire* traffic sent from v_1 to v_2 is routed on link (i, j) and is 0 otherwise.

The objective of the model, expressed in (1), is to minimize the total power consumption in the VNI. This latter can be expressed as the sum of three terms: the first summation is the power consumption due to the usage of resources in all servers in S, obtained as the sum of the minimum power associated with the activation of a server plus the linearly increasing power consumption due to the usage of the CPU of a server, induced by the demands of the VNFCs allocated to that server; the second summation takes into account the power consumption of the activated network nodes; the last summation expresses the power consumption of the activated links.

Constraints (2) express that each VNFC v must be allocated into exactly one server. Constraints (3) link the activation of a server and the allocation of a VNFC to it: if no VNFC is allocated to a server, then the server is not activated. Constraints (4) introduce a further linking between the activation of a server and the allocation variables: if some VNFC is allocated to a server, then the server must be activated. In (5), the resource capacity of a server is defined: given all the VNFCs allocated on the server, the total used resources must not exceed the available ones. The flow model taken into account does not use the continuous flow variables: instead the flow conservation constraint (6) relies on binary variables expressing the unsplittable nature of flows. The left-hand-side includes two summations that express the flow balance of a node n for the data sent for a couple (v_1, v_2) of a service chain, considering the incoming flow over links (n, i) and the outgoing flow over links (i, n). The right-hand-side includes a summation over all the servers that are connected to node n. Its value depends on the allocation of the VNFCs v_1, v_2 to servers: if v_1, v_2 are not allocated to any of the servers connected to n, then the summation is equal to 0 and the node is just a transition node with null flow conservation balance for (v_1, v_2); if only one of v_1, v_2 is allocated to a server connected to n, then the summation is either equal to b^{v_1,v_2} or $-b^{v_1,v_2}$ and the node n is either a source or a sink for couple (v_1, v_2), respectively; finally, if both v_1, v_2 are allocated to servers connected to n, then n is again associated with a null flow balance. We then need the capacity constraints for the bandwidth, including the flow conservation variables (7). These constraints also model the fact that if any $f_{ij}^{v_1,v_2}$ is equal to 1 and thus some traffic is sent over (i, j), then the link activation variable w_{ij} must be equal to 1. The constraints (8) and (9) link the boolean status of link activation variables to the status of the node activation variables: if a link is used, then its end-nodes must be activated; if a node is not activated, then a link ending in it cannot be used. Furthermore, the constraints (10) and (11) link the boolean status of flow variables to the status of the node activation variables: if a flow variables is equal to 1, then the end-nodes of the corresponding link must ac-

Table 2
The binary liner programming model BLP for problem (Eff-VNF).

$$\min \quad \sum_{s\in S}\left[P_s^{\min}\cdot y_s + (P_s^{\max}-P_s^{\min})\cdot \frac{1}{a_{rs}}\cdot \sum_{v\in V} a_{vr}\cdot x_{vs}\right] \tag{1}$$

$$+\sum_{n\in N} P_n\cdot z_n + \sum_{(i,j)\in L} P_{ij}\cdot g_{ij} \qquad r = CPU$$

$$\sum_{s\in S} x_{vs} = 1 \quad v\in V \tag{2}$$

$$y_s \le \sum_{v\in V} x_{vs} \quad s\in S \tag{3}$$

$$x_{vs} \le y_s \quad s\in S, v\in V \tag{4}$$

$$\sum_{v\in V} a_{vr}\cdot x_{vs} \le a_{rs}\cdot y_s \quad s\in S, r\in R \tag{5}$$

$$\sum_{(n,i)\in L} b^{v_1,v_2}\cdot f_{ni}^{v_1,v_2} - \sum_{(i,n)\in L} b^{v_1,v_2}\cdot f_{in}^{v_1,v_2} = \sum_{s\in S: n(s)=n} b^{v_1,v_2}\cdot (x_{v_1 s} - x_{v_2 s}) \quad n\in N, (v_1,v_2)\in \bigcup_{C\in\mathcal{C}} C \tag{6}$$

$$\sum_{(v_1,v_2)\in \bigcup_{C\in\mathcal{C}} C} b^{v_1,v_2}\cdot f_{ij}^{v_1,v_2} \le b_{ij}\cdot g_{ij} \quad (i,j)\in L \tag{7}$$

$$g_{ij} \le z_i \quad (i,j)\in L \tag{8}$$

$$g_{ij} \le z_j \quad (i,j)\in L \tag{9}$$

$$f_{ij}^{v_1,v_2} \le z_i \quad (i,j)\in L, (v_1,v_2)\in \bigcup_{C\in\mathcal{C}} \tag{10}$$

$$f_{ij}^{v_1,v_2} \le z_j \quad (i,j)\in L, (v_1,v_2)\in \bigcup_{C\in\mathcal{C}} \tag{11}$$

$$\sum_{(i,j)\in L} l_{ij}\cdot f_{ij}^{v_1,v_2} \le l_C^{v_1,v_2} \quad C\in\mathcal{C}, (v_1,v_2)\in C \tag{12}$$

$$x_{vs}\in\{0,1\}, y_s\in\{0,1\}, z_n\in\{0,1\}, f_{i,j}^{v_1,v_2}\in\{0,1\}, g_{i,j}\in\{0,1\}$$

tivated; if a node is not activated, then a link ending in it cannot be used and thus the corresponding flow variables are forced to be equal to 0. Finally, constraints (12) express the latency requirement for a service chain: for each chain C and couple (v_1, v_2) of C, (12) impose that the summation of the latency over links used for sending data from v_1 to v_2 must respect the latency limit $l_C^{v_1,v_2}$.

4.2. Resource request uncertainty and robust optimization

Uncertainty of traffic conditions is naturally present in telecommunications network design, since the future behaviour of users is generally not known precisely in advance [15]. In the case of our optimization problem (Eff-VNF), we address in particular the uncertainty of resource requests of VNFs: the amount of resources requested by each VNF can just be estimated and these estimates can (deeply) differ from the actual amount requested in the future. We thus assume that the amount a_{vr} is *uncertain* for each VNFC v and resource r, i.e. the value of a_{vr} is not known exactly when (Eff-VNF) is solved. To better clarify the concept of resource request uncertainty, we model data uncertainty through Γ-Robustness [4], a *cardinality-constrained interval deviation model*. According to this model, we assume that for each uncertain a_{vr} we know a so-called *nominal value* $\bar{a}_{vr}$ and the *maximum deviation* $\Delta a_{vr} \ge 0$, from it. We therefore assume that the (unknown) *actual value* a_{vr} lies in the interval: $a_{vr} \in [\bar{a}_{vr} - \Delta a_{vr}, \bar{a}_{vr} + \Delta a_{vr}]$.

In our direct experience with several real-world problems related to the design of telecommunication networks (e.g., [5,15,19]), we have observed that professionals often identify the nominal values of uncertain quantities with the value of forecast networks conditions (e.g., an expected value derived from historical data), whereas the deviation Δa_{vr} is identified as the maximum deviation from the forecast considered relevant by the network designer, again using historical data as reference.

As we sketched in the introduction, dealing with data uncertainty in optimization problems is a very delicate issue: as it is well-known from sensitivity analysis, also small variations of the input data may fully compromise the optimality and feasibility of produced solutions. The feasibility issue is particularly dangerous, because, due to uncertainty, we risk to produce solutions that will be completely useless in practice. For a detailed discussion on the issues associated with data uncertainty in optimization, we refer the reader to [3,14]. As a consequence, we cannot afford to neglect resource request uncertainty and thus risk that our design solution will turn out to be infeasible or of bad quality when implemented. We have therefore decided to tackle data uncertainty by adopting a Robust Optimization (RO) approach. RO is a methodology for dealing with data uncertainty that has received a lot of attention and has been highly appreciated in recent time w.r.t. more traditional methodologies like Stochastic Programming, especially thanks to its accessibility and computational tractability. We refer the reader to [3,14] for an exhaustive introduction to RO and for a discussion about its determinant advantages over Stochastic Programming.

RO is based on two major facts: 1) the decision maker must define an *uncertainty set*, which identifies the deviations in the nominal value of data against which the decision maker wants to get protection; 2) protection against deviations specified by the uncertainty set is guaranteed under the form of hard constraints that cut off all the feasible solutions that may become infeasible for some deviations included in the uncertainty set. More formally, we suppose that we are given a generic binary linear program:

$$v = \min c'x \quad \text{with} \quad x\in\mathcal{F} = \{Ax \ge b, x\in\{0,1\}^n\}$$

and that the coefficient matrix A is uncertain, i.e. we do not know the exact value of its entries. However, we are able to identify a family $\mathcal{A}$ of coefficient matrices that represent possible realizations of the uncertain matrix A, i.e. $A \in \mathcal{A}$. This family represents the uncertainty set of the robust problem. Then we can produce a *robust optimal solution*, i.e. a solution that is protected against data deviations, by considering the *robust counterpart* of the original problem:

$$v^{\mathcal{R}} = \min c'x \quad \text{with} \quad x \in \mathcal{R} = \{\tilde{A}x \geq b, \forall \tilde{A} \in \mathcal{A}, x \in \{0, 1\}^n\}$$

A solution in the feasible set $\mathcal{R}$ of the robust counterpart is feasible *for all* the coefficient matrices in the uncertainty set $\mathcal{A}$. As a consequence, $\mathcal{R}$ is a subset of the feasible set of the original problem, i.e. $\mathcal{R} \subseteq \mathcal{F}$. The choice of the coefficient matrices included in $\mathcal{A}$ should reflect the risk aversion of the decision maker. We note that such definition of robust counterpart can be extended to any mixed-integer linear program that involves continuous and integer decision variables. Imposing protection according to an RO paradigm leads to the so-called *price of robustness* [4,19]: this is a deterioration in the optimal value of the robust counterpart with respect to the optimal value of the original deterministic problem (i.e., $v^{\mathcal{R}} \leq v$), which is caused by the presence of the additional hard constraints imposing robustness. The price of robustness is a consequence of restricting the feasible set to the (*in general smaller*) set of robust solutions. Such price reflects the characteristics of the uncertainty set: uncertainty sets associated with higher risk aversion consider more severe and unlikely deviations and lead to higher protection but also higher price of robustness; in contrast, uncertainty sets expressing risky attitudes tend to not consider unlikely deviations, offering less protection and a reduced price of robustness.

We note that in practice it is really unlikely that all coefficients deviate to their worst possible value at the same time, so one of the aims of "smart" RO models is to define appropriate uncertainty sets that result not too conservative, while guaranteeing a reasonable protection. In the next paragraph, we describe the model of uncertainty that we adopt.

4.3. Adopting Γ-robust optimization

In problem (Eff-VNF), the constraints containing the uncertain data are those expressing the capacity of a server $s \in S$ for each type of resource $r \in R$:

$$\sum_{v \in V} \bar{a}_{vr} \cdot x_{vs} \leq a_{rs} \cdot y_s \tag{13}$$

This is a deterministic version of the constraint that takes into account only the nominal value of each uncertain coefficient a_{vr}. For each VNFC v and resource r, we can write the uncertain version of the constraint taking into account resource request uncertainty as:

$$\sum_{v \in V} \bar{a}_{vr} \cdot x_{vs} + DEV_{rs}(\Gamma, x) \leq a_{rs} \cdot y_s \tag{14}$$

which is the constraint (13) with the additional term $DEV_{rs}(\Gamma, x)$, which represents the worst deviation that the left-hand-side of the constraint may experience under Γ-ROB for an allocation vector x, when at most Γ coefficients deviate from their nominal value $\bar{a}_{vr}$.

Before giving a precise characterization of $DEV_{rs}(\Gamma, x)$ as the optimal value of a suitable optimization problem, we notice that the worst deviation that the nominal value $\bar{a}_{vr}$ may experience is $+\Delta a_{vr}$: the most positive deviation indeed entails the highest increase in a resource request of a VNFC v and thus brings towards the violation of the capacity constraint (13). Under these premises, for a fixed allocation vector x, the value $DEV_{rs}(\Gamma, x)$ corresponds to the optimal value of the following binary linear programming problem:

$$\begin{aligned} DEV_{rs}(\Gamma, x) = \max & \sum_{v \in V} (\Delta a_{vr} \cdot x_{vs}) \cdot y_{rsv} \\ & \sum_{v \in V} y_{rsv} \leq \Gamma \\ & y_{rsv} \in \{0, 1\} \qquad v \in V \ . \end{aligned}$$

In this problem, 1) a binary variable y_{rsv} is equal to 1 if, in the capacity constraint corresponding to the resource-server couple (r, s), the resource request coefficient deviates from its nominal value and experiences the worst deviation $\Delta a_{vr} \cdot x_{vs}$, whereas it is equal to 0 otherwise; 2) the single constraint imposes an upper bound $0 \leq \Gamma \leq |V|$ on the number of fading coefficients which may deviate in the considered constraint; 3) the objective function maximizes the deviation from the nominal value for the allocation vector x. The parameter Γ controls the robustness of the model: for $\Gamma = 0$ no coefficient is allowed to deviate and the model equals the deterministic one neglecting data uncertainty. As the value of Γ increases, the total deviation increases, until for $\Gamma = |V|$ we reach the highest possible total deviation, when all coefficients are allowed to deviate simultaneously and the solution protects against this fact.

We note that the robust version of the constraints (13) including the terms $DEV_{rs}(\Gamma, x)$ actually includes inner maximization problems which in turn contain the products of variables $x_{vs} \cdot y_{rsv}$. Constraints (14) are thus non-linear. However, as proved in [4], such non-linearities can be linearized according to the following procedure. First, we note that for a fixed vector x, the value $DEV_{rs}(\Gamma, x)$ is equal to the optimal value of its *linear relaxation*, where the integrality requirements on variables y_{rsv} are dropped:

$$DEV_{rs}(\Gamma, x) = \max \sum_{v \in V} (\Delta a_{vr} \cdot x_{vs}) \cdot y_{rsv} \qquad \text{(DEV-primal)} \tag{15}$$

$$\sum_{v \in V} y_{rsv} \leq \Gamma \tag{16}$$

$$0 \leq y_{rsv} \leq 1 \qquad v \in V \ . \tag{17}$$

We can then define the *dual problem* of the previous linear program, introducing the dual variables v_{rs}, w_{rsv} for $v \in V$ corresponding to the constraints (16) and (17), respectively:

$$\begin{aligned} & \min \Gamma \cdot v_{rs} + \sum_{v \in V} w_{rsv} \qquad \text{(DEV-dual)} \\ & \quad v_{rs} + w_{rsv} \geq \Delta a_{vr} \cdot x_{vs} \qquad v \in V \\ & \quad v_{rs} \geq 0 \\ & \quad w_{rsv} \geq 0 \qquad v \in V \ . \end{aligned}$$

Since the problem DEV-primal is feasible and bounded, on the basis of strong duality we can conclude that also its dual problem DEV-dual is feasible and bounded and their optimal values are equal. We can then substitute each (non-linear) uncertain version of (14) with the following family of linear constraints and decision variables obtained from DEV-dual [4]:

$$\sum_{v \in V} \bar{a}_{vr} \cdot x_{vs} + \left(\Gamma \cdot v_{rs} + \sum_{v \in V} w_{rsv} \right) \leq a_{rs} \cdot y_s \tag{18}$$

$$v_{rs} + w_{rsv} \geq \Delta a_{vr} \cdot x_{vs} \qquad v \in V \tag{19}$$

$$v_{rs} \geq 0 \tag{20}$$

$$w_{rsv} \geq 0 \qquad v \in V \ . \tag{21}$$

The robust version of the optimization problem BLP, which we denote by ROB-BLP, is thus obtained by replacing the non-robust capacity constraints (5) of BLP with the robust constraints and variables (18)–(21).

We remark that the increase in the dimension of the problem caused by the additional variables and constraints used in the dualization approach is not excessive: the linear robust formulation is indeed *compact*, i.e. its size is polynomial in the size of the input.

5. A fast fixing heuristic

The robust version of problem (Eff-VNF) is a binary linear programming model and, at least in principle, can be solved by using any commercial optimization solver, such as IBM ILOG CPLEX. However, the problem can be very hard to solve even for an advanced state-of-the-art solver like CPLEX when the size of the instances increase: the solver may have difficulties in identifying feasible solutions of good quality in a reasonable amount of time and can show a really slow convergence to an optimum. In this case, in order to enhance the performance of the solver, we can profit from integrating the solver with an efficient warm-start heuristic, which provides an initial feasible solution of good quality used to "warm-start" the solver and accelerate the convergence to an optimal solution.

The warm-start heuristic that we propose to adopt in this paper is based on two major phases:

- the execution of a *deterministic variable fixing procedure*, which exploits information coming from the linear relaxation of the problem. Variable fixing is a procedure according to which a subset of decision variables of the problem has their value fixed a-priori on the basis of some criteria: given all the decision variables $VAR_i \in \{0, 1\}$, $i \in I$ of a problem, variable fixing identifies two disjoint subset of variables with indices I^{FIXto0}, $I^{FIXto1} \subseteq I$: $I^{FIXto0} \cap I^{FIXto1} = \emptyset$ and the value of decision variables is fixed as follows: $VAR_i = 0$ for $i \in I^{FIXto0}$ and $VAR_i = 1$ for $i \in I^{FIXto1}$. The fixed variables are thus not anymore part of the decision process and we face a subproblem of the original optimization problem that is in general easier to solve;
- the solution of a smaller version of the original binary linear program, including the fixing of variables operated in the first phase. This phase exploits the power of a state-of-the-art MIP solver that, though not being able to solve the entire original problem efficiently and quickly, can instead fast provide solution of high quality to appropriate subproblems.

The complete algorithm of the heuristic is presented in Algorithm 1. Here, we rely on the following notation: 1) ROB-BLP is the robust problem containing only binary variables; 2) ROB-BLP^{rel} is the *linear relaxation* of the robust problem, i.e. the problem where the binary variables become continuous and can assume any value in the interval [0, 1]); 3) ROB-BLP^{FIX} is a subproblem of the robust problem that includes additional constraints fixing the value of a subset of variables (we must not decide anymore the value of these variables).

The heuristic first provides for solving the linear relaxation ROB-BLP^{rel}, obtaining an optimal solution denoted by $(\bar{x}, \bar{y}, \bar{z}, \bar{f}, \bar{g}, \bar{v}, \bar{w})$ (we remark that this solution may have fractional values). The optimal solution is used as basis for fixing the values of a subset of decision variables in the original binary problem ROB-BLP, thus obtaining the problem ROB-BLP^{FIX}. Our fixing strategy essentially consists in defining ROB-BLP^{FIX} by a-priori setting to 1 the value of variables whose value in $(\bar{x}, \bar{y}, \bar{z}, \bar{f}, \bar{g}, \bar{v}, \bar{w})$ is sufficiently close to 1. The rationale of this strategy is that if the value

Algorithm 1 - Warm-start heuristic.

FIXED := $\emptyset$ *//initialization of the subset of fixed variables*
Solve ROB-BLP^{rel} and get its optimal solution $(\bar{x}, \bar{y}, \bar{z}, \bar{f}, \bar{g}, \bar{v}, \bar{w})$
for $s = 1$ to $|S|$ **do**
  $NF := 0$ *//number of fixed variables*
  Sort the values $\bar{x}$ non-increasingly and let $\ell = 1, \ldots, |V|$ be the corresponding sorted indices $v \in V$
  for $\ell = 1$ to $|V|$ **do**
    if $x_{\ell s}^{RELAX} \geq 1 - \epsilon$ **and** $NF \leq UB$ **then**
      set $x_{vs} = 1$
      FIXED := FIXED $\cup$ (v, s)
      $NF := NF + 1$
    else
      break
    end if
  end for
end for
Define ROB-BLP^{FIX} by adding the fixing constraints $x_{vs} = 1$, $\forall (v, s) \in FIXED$ to ROB-BLP
Solve ROB-BLP^{FIX} (with time limit)
Use the best solution found for ROB-BLP^{FIX} as warm-start solution for solving ROB-BLP with CPLEX

of a variable is sufficiently close to 1 in the optimal solution of the linear relaxation, we have a pretty good indication that in a good feasible solution of the original problem ROB-BLP we should fix the decision variable to that value. Note that, in contrast to the general fixing rule previously presented, we do not consider the fixing of variables to the value 0.

More formally, we focus on the following fixing rule, which only involve the VNFC-server allocation decision variables x_{vs}. Let $\bar{x}$ be the value of the VNFC-server allocation decision variables in the optimal solution $(\bar{x}, \bar{y}, \bar{z}, \bar{f}, \bar{g}, \bar{v}, \bar{w})$ of the linear relaxation ROB-BLP^{rel}, then the rule is:

$$\text{if } \bar{x}_{vs} \geq 1 - \epsilon \text{ then set } \bar{x}_{vs} = 1$$

where $0 < \epsilon < 1$ is a parameter that must be chosen.

Let FIXED be the set of couples (v, s) that satisfy the previous fixing rule. After having established the set FIXED, we define and solve the subproblem ROB-BLP^{FIX} obtained by adding to ROB-BLP the constraints:

$$x_{vs} = 1 \qquad (v, s) \in FIXED$$

ROB-BLP^{FIX} is a more-constrained version of the original robust problem, where the value of the variables x_{vs} with $(v, s) \in FIXED$ is set and is not anymore part of the decision process. ROB-BLP^{FIX} thus actually constitutes a subproblem that can be solved faster to optimality (smaller feasible solution set to explore for the solver). It is solved by means of the solver CPLEX. We stress that a feasible solution for the subproblem ROB-BLP^{FIX} is also feasible for the complete problem ROB-BLP. We use the best solution found for ROB-BLP^{FIX} within the time limit by CPLEX as starting solution for solving the original problem ROB-BLP, thus supporting a warm-start for CPLEX.

We note that we just consider the fixing of the VNFC-server allocation decision variables x_{vs} since they are particularly important in the decision process and when we impose $x_{vs} = 1$ for some couple (v, s), from constraint (2) we know that we can impose at the same time $x_{v\sigma} = 0$ for any server $\sigma \in S$ such that $\sigma \neq s$, thus immediately determining the value of many other relevant variables.

A very important thing to remark is that we should not fix the value of too many variables x_{vs} to 1, since this may reduce the possibility of finding good quality solutions when solving ROB-BLP^{FIX} (the problem would be too constrained). So we impose an upper bound $UB > 0$ on the number of variables x_{vs} that can be fixed to

1 for each server s. The aim of this is to not assign too many VNFCs to the same server s, leading to a potential overbooking of that server. Specifically, for each $s \in S$, we sort the variables $\bar{x}_{vs}$ from the highest to the lowest value and then, we fix to 1 the $UB > 0$ variables with highest value $\bar{x}_{vs} \geq 1 - \epsilon$.

6. Numerical evaluation

We performed a numerical evaluation focusing on an important use-case for VNF, namely the Evolved Packet Core (EPC), which represents the cornerstone of next generation mobile networks. Each component belonging to this VNF has a particular task and can be run on a stand-alone VM. The EPC architecture distinguishes between user data - user plane (UP), and signalling traffic - control plane (CP). Typically, both have different latency constraints. We considered different configurations for the EPC, which are determined by the actual load. The traffic which the virtualized EPC is able to process can be expressed in terms of the number of events generated by the users attached to the base stations during a time frame of one hour (ev/h). This metric was used to dimension the VNF and, therefore, the number of each component type (Base Station, Mobility Management Entity, etc.) belonging to the EPC, by applying the dimensioning rules from [20].

In our evaluation, we considered uncertainty on the CPU demands requested by each VNFC. Typically, such maximum demand deviation can be obtained from workload traces by analyzing historical data or by workload prediction mechanisms. For example, using collaborative filtering modeling and prediction, authors in [21] were able to predict diverse workload throughput values with low training overhead and within approximately 30% of the correct figure. Consequently, we assume that the components may have a CPU utilization varying at maximum 30% from the nominal demands in the worst case. In the evaluation, we consider the protection against the deviation of a given number of VNFCs, by using a protection factor (Γ). The solution is protected from the deviation of a maximum number Γ of uncertain parameters, each one specifying the CPU demand of a given VNFC. The service chains are composed of VNFCs that belong to a particular communication path both for the CP and the UP.

6.1. Comparison between full model and fast fixing (FF) heuristic

First, we were interested to compare our heuristic against the optimal solution provided by CPLEX through the standard branch-and-cut algorithm. The problem we are facing is very hard to solve in the exact way, even by considering very small instances, as also shown in our previous work [6]. Therefore, the evaluation was conducted by considering three different hard time limits (short, medium and large), with the aim of finding out if the heuristic is able to output comparable or even better results, in comparison to the optimal solver for a given time limit. The choice of the intervals was based on the fact that such problems need to be solved in a very short time, when dealing with TOs' decision making processes:

- short - 200 s;
- medium - 600 s;
- large - 2500 s.

For the heuristic, we used both the short and the medium time intervals, while CPLEX original model was run with the medium and large time limits. Since the heuristic is composed of different phases, we split the available time between the phase where we solve the problem ROB-BLP^{FIX}, that includes the additional fixing of variables, and the phase where we solve ROB-BLP with warm-start, where we try to improve the solution found solving ROB-BLP^{FIX} (specifically, for the short interval we set 150 s for solving ROB-BLP^{FIX} and 50 s for solving ROB-BLP with warm-start, whereas and for the medium interval we set 400 s and 200 s). This is because most of our experiments showed that the first phase stage of the heuristic finds a very good solution that is hard to improve even in longer runs by the warm start stage. As shown in Fig. 2, we compare the energy efficiency of the FF heuristic with the CPLEX solver for increasing problem sizes, defined in number of events per hour, ranging from 1.3 millions up to 3.1 millions, with a step size of 300,000 events. We compare the objective function (total power consumption of both network and computing infrastructure) of the resulting VNF placement and network embedding. For the sake of brevity, we only show the results for three different protection factors (Γ= 0, 2, 6). As displayed in Fig. 2, we consider two different runs of the heuristic (short and medium) and two for the original CPLEX model (medium and long). In the first three cases (up to 1.6M events) the results are almost comparable for all the Γ values, while for the other configurations there are some differences. In particular when $\Gamma = 0$, meaning that we are considering no protection at all, the heuristic with the medium hard time limit (600 s) shows very similar results to the original model solved in the long run (2500 s), and in two configurations it is able to achieve even better results. This is because CPLEX was not able to find the optimal solution within the given time limit, but our FF heuristic found a better one due to the fixing rules that limit the problem size.

Starting from the configuration characterized by a load of 2.2M events, the total power consumption considerably increases. This is due to the activation of several links and network nodes that are needed to accommodate the traffic and the higher number of components needed to implement the service chains. If at maximum two components ($\Gamma = 2$) are allowed to deviate from their nominal demand, the results show the same trend and the heuristic with the medium time limit is performing similarly to the CPLEX model solved in the long run. What is interesting to observe is that, when Γ is increasing (e.g equal to 6), the heuristic with the short and medium time limit is achieving almost the same results and they output even better results in around 75% of the considered configurations, especially when the number of events is considerably high. Despite the significant larger amount of time allowed for the optimal solver, the heuristic still provides excellent solution qualities as depicted, even in the short run. These results are encouraging and show that our heuristic is able to achieve very good results in short time for scenarios with high number of events if the allocation needs to be protected more.

6.2. Solution quality

Finally, we investigated the solution quality of our heuristic and the original CPLEX model, in terms of robustness and additional cost for protecting against uncertainty for a given Γ. To this end, we solved the problem for a given Γ using our heuristic (short run), and the original CPLEX model (medium run). By considering the output of the VNFC allocation to the physical servers and the routing path, we created 10.000 different instances of our problem in the following. For each instance, if a VNFC requires a_{vr} units of CPU, we allowed to deviate randomly its demand in the range $[\bar{a}_{vr} - \Delta a_{vr}, \bar{a}_{vr} + \Delta a_{vr}]$. After updating the CPU utilization on each server according to the random values calculated within the given bounds, we checked the resource budget constraint and computed the number of constraint violations due to the uncertainty. Two performance indicators are considered: the *robustness degree* and the *price of robustness*. The former is computed as:

$$robustness = 1 - \frac{\#violations}{\#runs} \tag{22}$$

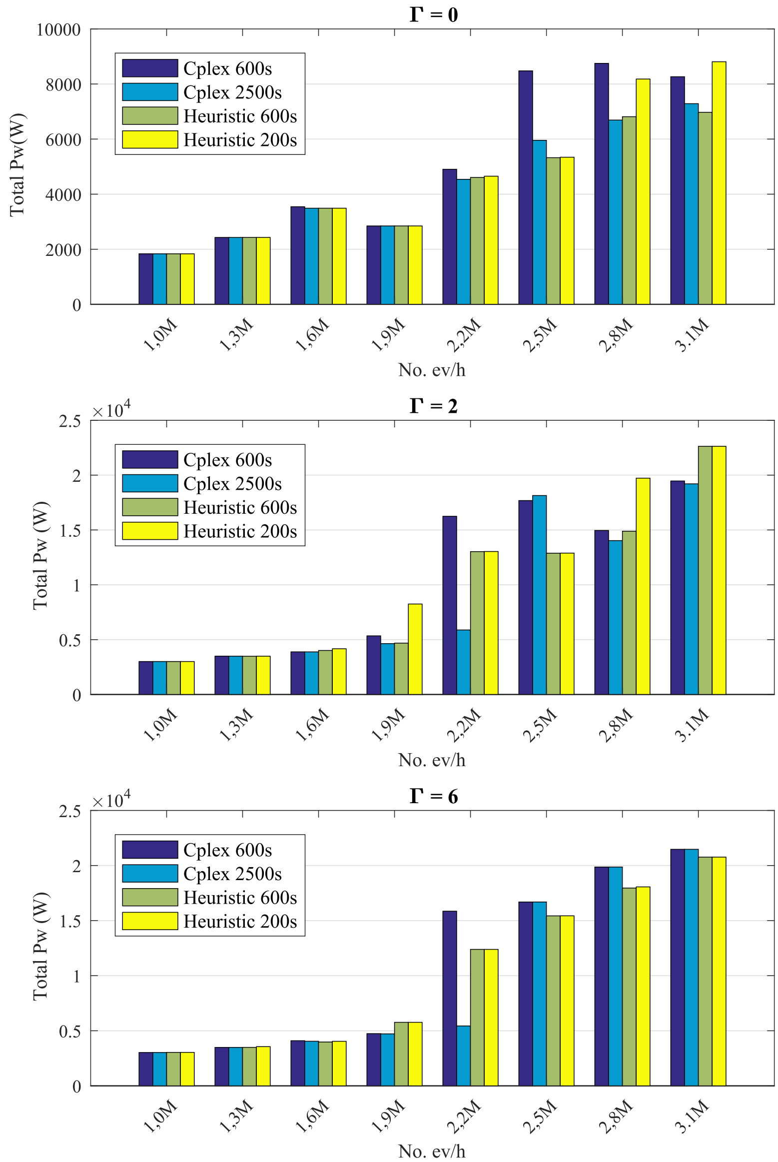


Fig. 2. CPLEX - FF heuristic comparison.

The price of robustness is computed, for a given Γ, as the increase in the objective function (i.e., the total power consumption) compared to the best value achieved when no protection is applied ($\Gamma = 0$):

$$price_{(\Gamma=x)} = \frac{total\_power_{(\Gamma=x)} - total\_power_{(\Gamma=0)}}{total\_power_{(\Gamma=0)}} \quad (23)$$

Fig. 3 shows the robustness degree (in blue) and the price of robustness (in red) as the protection factor increases for three different configurations of the vEPC. In the case of $\Gamma = 0$, we do not protect against uncertainty and thus no additional resources are needed. Consequently, the degree and price of robustness are zero as the objective function (i.e., the total power consumption) is the minimum possible. When Γ increases, the objective function increases because the solution requires more energy due to the activation of more resources needed to protect the allocation from the demand deviations. What is interesting to observe is that the short run of our heuristic offers the same or even better robustness (e.g. $\Gamma = 3$ and ev/h=2.8M) in comparison to the original CPLEX model by showing a lower price, in almost all the cases. Our experiments show that the heuristic converges in a very short time to solutions characterized by a high quality in terms of additional price for a given degree of protection. Selecting a proper Γ is up to the decision maker because it allows the trade-off between the additional price to pay and the desired level of robustness. An upper bound for constraint violation probability can be calculated as in [4]. If a given NFVI operator wants to protect its VNF more from demand deviations, it would select a larger Γ at the expense of higher costs to run the infrastructure. A more opportunistic operator would select a lower value leading to a potential higher constraint violation probability, which may lead to increased resource contention and ultimately also to SLA violations at the benefit of significant cost savings.

7. Conclusions and future work

Network Function Virtualization will be a key cornerstone for 5G network infrastructure. In Network Function Virtualization, a set of network functions are virtualized and run on commodity servers inside virtual datacenters. In such a setup, it is crucial to optimize the deployment and operation of the Virtual Network Functions to be both energy efficient for controlling the opera-

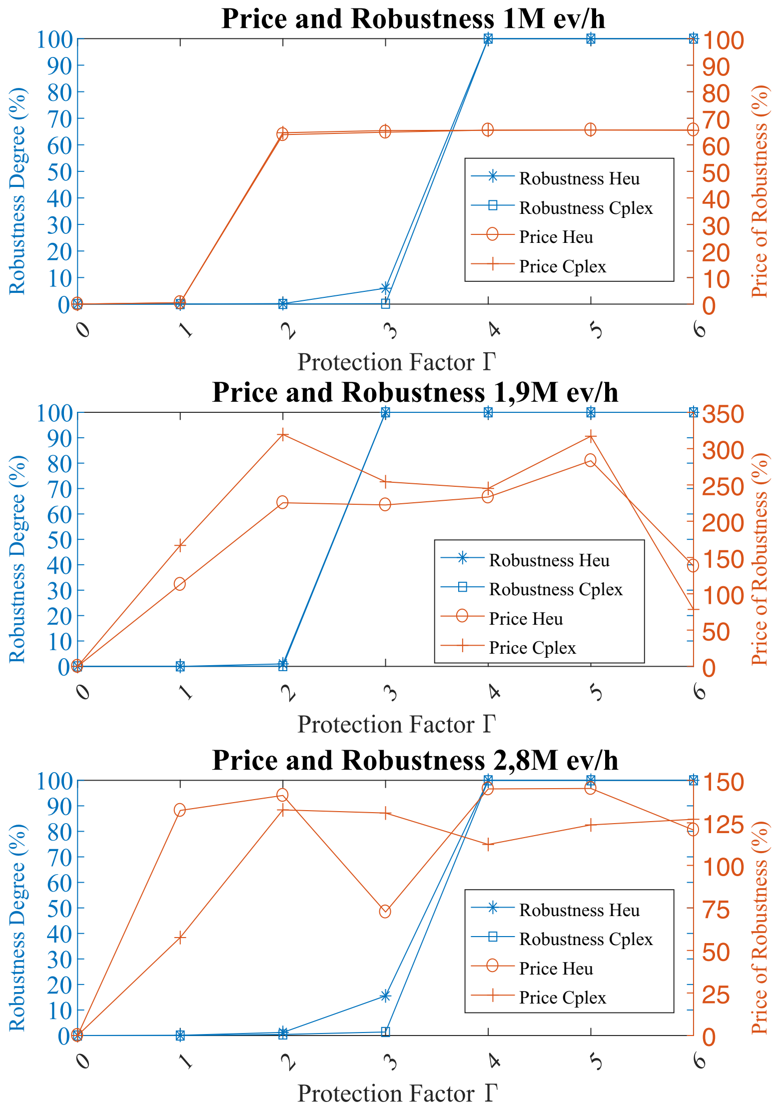


Fig. 3. Degree and price of robustness. (For interpretation of the references to colour in the text, the reader is referred to the web version of this article.)

tional costs as well as robust to cope with fluctuations or imprecise knowledge in resource demands for VNFCs.

In this paper, we tackled the problem of designing a power efficient Virtual Network Function placement and network embedding. The methodology followed here is made up of three steps. First, an exact formulation using binary programming has been developed, which places a set of VNF Components inside a virtualized data center while trying to minimize the energy consumption. Second, the theory of Γ-Robustness has been applied and a robust version of the problem has been proposed, where input to the problem is not known precisely but rather resource demands are allowed to deviate within bounds; the robust algorithm has reduced computational complexity compared to our previous work [6]. Third, a fast variable fixing heuristic that exploits structural information coming from the linear relaxation of the problem has been developed, aiming at solving the robust model faster. Our robust model and heuristic can tradeoff energy efficiency and robustness under uncertainty constraints.

We compared the heuristic against the optimal solution provided by CPLEX, by imposing a hard time limit for solving in both approaches. We showed that our heuristic achieves better results with respect to the state-of-the-art branch-and-bound algorithm performed by CPLEX in reasonable time and is therefore suited for online optimization. Also, we investigated the solution qualities of our heuristic in terms of robustness and additional cost for protecting against uncertainty for a given Γ. We found that the cost for achieving a given robustness degree has a stable trend for all $\Gamma \neq 0$, while the degree of robustness increases with Γ, as expected.

There are several interesting aspects to be tackled for future work. First, having better knowledge of the distribution of the uncertainty in the form of a more accurate description would allow us to calculate more precise solutions for the given input parameters using the theory of Multiband Robust Optimization (e.g., [19]). Also, different heuristic solutions could be explored that would allow faster computation of solutions using e.g. global first fit based approaches that need to be modified to cope with the uncertainty of the input data. Finally, we intend to integrate our online algorithm into open source cloud platforms such as OpenStack with the Watcher framework or NFV platforms such as OpenBaton.

Acknowledgements

Part of this work has been funded by the Knowledge Foundation of Sweden through the Profile HITS (20140037), by the Spanish Government and ERDF through CICYT project

TEC2013-48099-C2-1-P, and by the German Federal Ministry of Education and Research (BMBF grant 05M2013 - VINO: Virtual Network Optimization).

References

[1] ETSI, European telecommunications standards institute, network functions virtualisation - introductory white paper, 2015, https://portal.etsi.org/nfv/nfv_white_paper.pdf.

[2] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, N. McKeown, ElasticTree: saving energy in data center networks, in: Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation, NSDI'10, USENIX Association, Berkeley, USA, April, 2010, p. 17.

[3] D. Bertsimas, D.B. Brown, C. Caramanis, Theory and applications of robust optimization, SIAM Rev. 27 (2008) 295–308, doi:10.1137/080734510.

[4] D. Bertsimas, M. Sim, The price of robustness, Oper. Res. 52 (2004) 35–53, doi:10.1287/opre.1030.0065.

[5] F. D'Andreagiovanni, J. Krolikowski, J. Pulaj, A fast hybrid primal heuristic for multiband robust capacitated network design with multiple time periods, App. Soft Comp. 26 (2015) 497–507, doi:10.1016/j.asoc.2014.10.016.

[6] A. Marotta, A. Kassler, A power efficient and robust virtual network functions placement problem, in: Proceedings of the 28th International Teletraffic Congress (ITC28), Würzburg, Germany, September, 2016.

[7] B. Addis, D. Belabed, M. Bouet, S. Secci, Virtual network functions placement and routing optimization, in: 4th IEEE International Conference on Cloud Networking, Niagara Falls, Canada, October, 2015, pp. 171–177, doi:10.1109/CloudNet.2015.7335301.

[8] F. Larumbe, B. Sansò, Green cloud broker: on-line dynamic virtual machine placement across multiple cloud providers, in: 5th IEEE International Conference on Cloud Networking, Pisa, Italy, October, 2016, pp. 119–125, doi:10.1109/CloudNet.2016.41.

[9] N.E. Khoury, S. Ayoubi, C. Assi, Energy-aware placement and scheduling of network traffic flows with deadlines on virtual network functions, in: 5th IEEE International Conference on Cloud Networking, Pisa, Italy, October, 2016, pp. 89–94, doi:10.1109/CloudNet.2016.40.

[10] Q. Zheng, R. Li, X. Li, N. Shah, J. Zhang, F. Tian, K.-M. Chao, J. Li, Virtual machine consolidated placement based on multi-objective biogeography-based optimization, Future Gener. Comput. Syst. 54 (C) (2016) 95–122, doi:10.1016/j.future.2015.02.010.

[11] A. Baumgartner, V.S. Reddy, T. Bauschert, Combined virtual mobile core network function placement and topology optimization with latency bounds, in: Fourth European Workshop on Software Defined Networks (EWSDN), Bilbao, Spain, October, 2015, pp. 97–102, doi:10.1109/EWSDN.2015.68.

[12] F. Bari, S.R. Chowdhury, R. Ahmed, R. Boutaba, On orchestrating virtual network functions in NFV, CoRR, abs/1503.06377 (2015). http://arxiv.org/abs/1503.06377.

[13] D. Papadimitriou, New challenges in network optimization, in: 17th International Conference on High Performance Switching and Routing (IEEE HPSR), Yokohama Japan, June, 2016, pp. 1–7, doi:10.1109/HPSR.2016.7525631.

[14] A. Ben-Tal, L. El Ghaoui, A.S. Nemirovski, Robust Optimization, Princeton Series in Applied Mathematics, Princeton University Press, 2009.

[15] T. Bauschert, C. Büsing, F. D'Andreagiovanni, A.M.C.A. Koster, M. Kutschka, U. Steglich, Network planning under demand uncertainty with robust optimization, IEEE Commun. Mag. 52 (2) (2014) 178–185, doi:10.1109/MCOM.2014.6736760.

[16] E. Zola, A.J. Kassler, Optimising for energy or robustness? Trade-offs for VM consolidation in virtualized datacenters under uncertainty, Optim. Lett. (2016) 1–22, doi:10.1007/s11590-016-1065-x.

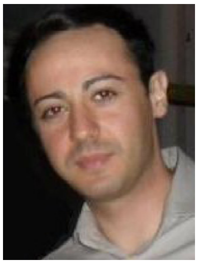
[17] S. Coniglio, A.M.C.A. Koster, M. Tieves, Virtual network embedding under uncertainty: exact and heuristic approaches, in: 11th International Conference on the Design of Reliable Communication Networks (IEEE DRCN), Kansas City, USA, March, 2015, pp. 1–8, doi:10.1109/DRCN.2015.7148978.

[18] R.K. Ahuja, T. Magnanti, J. Orlin, Network Flows: Theory, Algorithms, and Applications, Prentice Hall, Upper Saddle River, 1993.

[19] C. Büsing, F. D'Andreagiovanni, New results about multi-band uncertainty in robust optimization, in: R. Klasing (Ed.), Experimental Algorithms, LNCS, 7276, Springer, Heidelberg, 2012, pp. 63–74, doi:10.1007/978-3-642-30850-5_7.

[20] F.Z. Yousaf, P. Loureiro, F. Zdarsky, T. Taleb, M. Liebsch, Cost analysis of initial deployment strategies for virtualized mobile core network functions, IEEE Comm. Mag. 53 (12) (2015) 60–66, doi:10.1109/MCOM.2015.7355586.

[21] J.D. et. al., Packing light: portable workload performance prediction for the cloud, in: 29th IEEE International Conference on Data Engineering Workshops (ICDEW), Brisbane, Australia, March, 2013, pp. 258–265, doi:10.1109/ICDEW.2013.6547460.



Antonio Marotta received the M.Sc. and Ph.D. degrees from the University of Napoli "Federico II" in 2010 and 2014, respectively. He is currently a post-doc researcher at the Karlstad University. His research interests include cloud computing, critical infrastructure protection, Software Defined Networks: Besides he is on energy optimization models in cloud environment with focus on the uncertainty of the model parameters.

Fabio D'Andreagiovanni has been a First Class Research Scientist at the French National Center for Scientific Research (CNRS) and at the Laboratory "Heudiasyc" of Sorbonne University - University of Technology of Compiegne since October 2016. Until September 2016, he was Head of Research Group at the Department of Mathematical Optimization of Zuse Institute Berlin and Lecturer at the Department of Mathematics and Computer Science of Freie Universität Berlin and at the Faculty of Engineering of Technische Universität Berlin. He received his M.Sc. in Industrial Engineering (2006) and Ph.D. in Operations Research (2010) from Sapienza Università di Roma and he was a Research Scholar in the Department of Industrial Engineering and Operations Research at Columbia University in the City of New York (2008–2009). His research has been focused on theory and applications of Robust Optimization and Mixed Integer Prsogramming and has received several awards, such as the Accenture M.Sc. Prize 2006, the INFORMS Telecom Doctoral Dissertation Award 2010 and the INFORMS Telecom Best Paper Award 2014. He has worked as consultant for several major European telecommunications and electric utility companies.

Dr. Andreas Kassler received his MSc degree in Mathematics/Computer Science from Augsburg University, Germany in 1995 and his PhD degree in Computer Science from University of Ulm, Germany, in 2002. Currently, he is employed as Full Professor with the Department of Mathematics and Computer Science at Karlstad University in Sweden. Before joining Karlstad University, he was Assistant Professor at the School of Computer Engineering, Nanyang Technological University, Singapore, between 2003 and 2004. Dr. Kassler is (co-)author of more than 100 peer reviewed books, journal and conference articles. He served as a guest editor of a feature topic in EURASIP Wireless Communications and Networking Journal, and is on the editorial boards of several international journals. Dr. Andreas J. Kassler is a senior member of IEEE Computer Society and IEEE Communications.

Enrica Zola received the double M.Sc. degree in Telecommunications Engineering from both Politecnico di Torino (Italy) and Universitat Politècnica de Catalunya (UPC, Spain), in 2002 and 2003, respectively. In 2011, she earned a Ph.D. from the UPC. From September 2001 to August 2002, she collaborated with the Radio Department of the Spanish teleoperator Amena. From March 2003 to February 2006, she has been working at UPC as a full-time Lecturer. From March 2006, she serves as an Assistant Professor at the Department of Telematics Engineering at UPC. She has been teaching design and planning of communication networks and wireless networks. Dr. Zola has been involved in a number of research projects supported by the Spanish Government and the European Commission on performance modeling of wireless systems and networks (IST Emily, RUBI, IST Liaison, COST Winemo, COST290). Her research interest areas encompass wireless networking in general, with special attention to mobility management and radio resource management. Recently, her interest has focused on performance optimization modeling and robust optimization techniques, and on the design of 5G networks.