Noname manuscript No.
(will be inserted by the editor)

Data uncertainty in virtual network embedding

Robust optimization and protection levels

S. Coniglio · A. Koster · M. Tieves

Received: date / Accepted: date

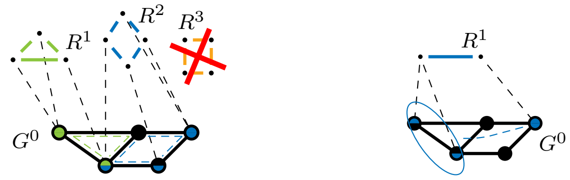
Abstract We address the Virtual Network Embedding problem (VNE) which, given a physical (substrate) network and a collection of virtual networks (VNs), calls for an embedding of the most profitable subset of VNs onto the physical substrate, subject to capacity constraints. In practical applications, node and link demands of the different VNs are, typically, uncertain and difficult to know *a priori*. To face this issue, we first model VNE as a chance-constrained Mixed-Integer Linear Program (MILP) where the uncertain demands are assumed to be random variables. We then propose a Γ-robust optimization approach to approximate the original chance-constrained formulation, capable of yielding solutions with a large profit that are feasible for almost all the possible realizations of the uncertain demands. To solve larger scale instances, for which the exact approach is computationally too demanding, we propose two MILP-based heuristics: a parametric one, which relies on a parameter setting chosen *a priori*, and an adaptive one, which does not. We conclude by reporting on extensive computational experiments where the different methods and approaches are compared.

Keywords network virtualization · data uncertainty · Γ-robustness · integer programming · math heuristics

The paper is an extended version of [1]. In its extended form, is has not been published or accepted for publication elsewhere, nor it has been submitted to any other journal.

S. Coniglio, A. Koster, M. Tieves
Lehrstuhl II für Mathematik, RWTH Aachen University
Tel.: +49 (0)241 80 94995
Fax.: +49 (0)241 80 92136
{coniglio,koster,tieves}@math2.rwth-aachen.de

Present address of S. Coniglio (corresponding author):
Department of Mathematical Sciences, University of Southampton
Tel.: +44 023 8059 3644
s.coniglio@soton.ac.uk



(a) (b)

Fig. 1: (a) Two VN requests, R^1 and R^2, embedded into the substrate G^0, while R^3 is rejected. (b) A (forbidden) one-to-many mapping of a virtual node.

1 Introduction

In the context of large scale networks, the paradigm of *network virtualization* has garnered a large attention, being advocated as one of the key technologies of the future of networking [2,3]. In its general form, the paradigm allows to decouple the physical (low level) management aspects of a networking environment from those (of higher level) involving service provisioning. This way, it benefits the two main actors of a networking context: the owner of the *physical* or *substrate network*, the so-called *infrastructure provider*—who, this way, can solely concentrate on the management aspects of the substrate—and the *service provider*—who, this way, can only focus on the provisioning aspects of his services. A prominent application scenario is that of the Internet which, due to only allowing for small and incremental updates to its structure as a consequence of its inherently plural nature, can largely benefit from virtualization techniques as a noninvasive way of upgrading itself, preventing *ossification* phenomena. For a more detailed treatment of the topic, we refer the reader to the surveys [2,3,4].

1.1 The VNE problem and previous work

When considering the infrastructure provider's perspective, we are faced with the so-called *Virtual Network Embedding* problem (VNE). It is the problem of, in the first place, deciding whether to *accept or reject* a subset of Virtual Network (VN) requests issued by the customers (the *admission control* aspect) and, then, of *embedding* the accepted VNs onto the substrate network, subject to capacity constraints. In this paper, we will focus on the offline version of VNE in which the set of VN requests is known beforehand. This suits the case where the VN requests are issued ahead of the time when their service will be activated, thus allowing for sufficient time for offline planning. Such requests are, typically, quite large in terms of resource requirements and, if accepted, sufficiently long lasting to assume that they will be embedded indefinitely.

In this work, we assume that each VN is composed of a set of *virtual nodes*, each of which endowed with an estimate of the node resources it requires (computing power), which we refer to as *node demands*, as well as a set of *traffic* or *link demands* between pairs of virtual nodes. An embedding thus consists of virtual-to-physical *node-to-node* and *link-to-path* mappings which do not exceed the node and link capacities of the substrate network. Note that, while we allow for the mapping of any number of virtual nodes belonging to the same VN request to the same physical node (so-called *co-location*), we forbid the mapping of a single virtual node to more than a single physical node[1]. As customary for VNE, see [3] and the references therein, we also consider *locality* constraints which restrict the set of physical nodes onto which a virtual node can be mapped. This allows for the encoding of technical specifications which are only met by certain physical nodes, as well as geographical restrictions (to prevent, for instance, the mapping of data intensive services too far away from the corresponding data centers). Throughout the paper, we will assume a single path unsplittable routing scheme, which is often preferred to a splittable one so to avoid packet reordering issues (see, e.g., [5]). Nevertheless, the generalization of the techniques that we will propose to the splittable case is straightforward, as we will better point out in the following. For an example of a VNE instance, together with a feasible solution, see Figure 1 (a).

As it is easy to observe, VNE is weakly $\mathcal{NP}$-hard by reduction from the 0-1 knapsack problem (KP) [3]. Strong $\mathcal{NP}$-hardness is established in [6] by reduction from the maximum stable set problem. The result also implies that VNE cannot be approximated in polynomial time within a factor of $|V^0|^{\frac{1}{2}-\epsilon}$ for any $\epsilon > 0$, unless $\mathcal{P} = \mathcal{NP}$ [6].

Most of the literature on VNE employs heuristics in which node and link mappings are carried out sequentially, usually in an online setting where the virtual network requests arrive over time. Examples can be found in, e.g., [7], which also accounts for reconfigurations of a given embedding, and [8], which employs both deterministic and randomized rounding techniques. For an extensive account on (mostly heuristic) algorithms for VNE, we refer the reader to the excellent survey [3]. For a rounding algorithm based on column generation, also encompassing admission control, see [9]. For a greedy algorithm based on the degree of utilization of the different physical nodes, see [10]. Also see [11] for an energy-efficient version of VNE subject to Gaussian traffic demands.

Among the few exact approaches, we mention that in [12], where the authors propose a Mixed-Integer Linear Programming (MILP) formulation to carry out each step of an algorithm for the online version of the problem, [13], which illustrates an MILP formulation for the offline version of VNE which also accounts for the installation (or rental) of network capacities (with a rent-at-bulk aspect), and [14,15], which extend [12] to the energy-aware and fault-tolerant cases.

[1] We assume that each task which can be parallelized on multiple physical nodes is described in a VN request via as many virtual nodes as the number of parallel threads it can use. This way, a task will be split only if, by splitting it, a more profitable embedding can be obtained. On the contrary, splitting a single virtual node would force us to split both the node *and* the traffic demands over the physical network in a not well defined way, see Figure 1 (b).

For the closely related problem of Network Functions Virtualization (NFV), we refer the reader to [16], which addresses network planning of a mixed physical-virtual CDN (Content Delivery Network), tackled from a stochastic programming point of view, and to [17], for the embedding of network functions with congestion minimization.

1.2 Contribution and outline of the work

To our knowledge, almost all the works on VNE assume that node and link demands of the different VNs are known deterministically[2]. In a realistic setting though, it is reasonable to assume that, for each VN, the actual demands of computing power and bandwidth (i.e., node and traffic demands) entail significant uncertainty (e.g., because of measurement errors or variability over time).

To cope with this issue, in this work we propose a robust optimization formulation (based on the model of Γ-robustness) for VNE which, after selecting an appropriate value for the parameter Γ, allows to achieve solutions which (under mild assumptions) satisfy each of the constraints where uncertain parameters are involved with any desired probability. For clarity, we first introduce a chance-constrained formulation for VNE where node and link demands are assumed to be random variables and the constraints of the problem are required to be satisfied with high probability, and show how this formulation can be approximated via the Γ-robust formulation which is, computationally, more tractable.

The paper is organized as follows[3]. In Section 2, we illustrate a deterministic MILP formulation for VNE and discuss on the $\mathcal{NP}$-hardness of its two natural subproblems. In Section 3, we describe the chance-constrained and MILP Γ-robust formulations. To tackle large-scale instances, we present, in Section 4, two MILP-based heuristics, both yielding Γ-robust solutions: a two-phase one, which relies on an *a priori* parameter setting, and an adaptive one which does not. Extensive computational experiments are reported and illustrated in Section 5. Concluding remarks are drawn in Section 6.

2 Deterministic MILP formulation and complexity

In this section, we provide a deterministic MILP formulation for the problem and address the $\mathcal{NP}$-hardness of its two (natural) subproblems.

---

[2] The works in [11,18] constitute the, to our knowledge, only notable exceptions where VNE is not tackled in a deterministic setting. Differently from the setting we assume in this paper though, both works tackle (heuristically) the online version of VNE where the VN requests arrive over time one by one (or, at most, in batches), whereas, in this work, we assume that the whole set of requests is known beforehand and that admission control is in place. Although our work can clearly be used for the case where a single VN request has to be embedded, we recall that, in this paper, embedding costs are not considered.

[3] A preliminary version of this work appeared in [1].

2.1 Deterministic MILP formulation

Let us assume, for now, a deterministic setting where all the parameters of the problem are precisely known beforehand. Let the directed graph $G^0 = (V^0, A^0)$ represent the physical network, with node capacities c_i^0 for all $i \in V^0$ and link capacities k_{ij}^0 for all $(i,j) \in A^0$. Let $R = 1, \ldots, |R|$ be the set of VN requests, with profits $p^r \geq 0$ for all $r \in R$. For each $r \in R$, let V^r be the set of virtual nodes, each of which endowed with a parameter ω_v^r denoting the corresponding node demand and a (possibly sparse) virtual traffic matrix $D^r \in \mathbb{R}_+^{|V^r| \times |V^r|}$, where each element $d_{vw}^r \geq 0$ accounts for the traffic demand between the two virtual nodes $v, w \in V^r$. We denote by $V^0(r,v) \subseteq V^0$ the set of physical nodes to which the virtual node $v \in V^r$, pertaining to request $r \in R$, can be mapped due to locality constraints.

The following MILP formulation of VNE is similar to the one proposed in [13], although it neglects some extra aspects (notably, rent-at-bulk) which are outside the scope of this paper. We employ three groups of decision variables: y_{ij}, x_{vi}^r, and $f_{ij}^{r,vw}$. Let $y^r \in \{0,1\}$ take value 1 if the request of index $r \in R$ is accepted and 0 otherwise. Let $x_{vi}^r \in \{0,1\}$ be equal to 1 if the virtual node $v \in V^r$, pertaining to request $r \in R$, is mapped onto the physical node $i \in V^0$, with $y^r = 0$ otherwise. Let $f_{ij}^{r,vw}$ take value 1 if the traffic between the two virtual nodes $v, w \in V^r$, for a request $r \in R$, is routed over the physical link $(i,j) \in A^0$, and 0 otherwise. Then, the VNE problem can be cast as the following MILP:

$$
\begin{aligned}
\max \quad & \sum_{r \in R} p^r y^r && && (1)\\
\text{s.t.} \quad & \sum_{i \in V^0(r,v)} x_{vi}^r = y^r && \forall r \in R, v \in V^r && (2)\\
& \sum_{r \in R} \sum_{\substack{v \in V^r: \\ i \in V^0(r,v)}} \omega_v^r x_{vi}^r \leq c_i^0 && \forall i \in V^0 && (3)\\
& \sum_{r \in R} \sum_{v,w \in V^r} d_{vw}^r f_{ij}^{vw,r} \leq k_{ij}^0 && \forall (i,j) \in A^0 && (4)\\
& \sum_{(i,j) \in \delta^+(i)} f_{ij}^{vw,r} - \sum_{(j,i) \in \delta^-(i)} f_{ji}^{vw,r} = x_{vi}^r - x_{wi}^r && \forall r \in R, v, w \in V^r, i \in V^0 && (5)\\
& y^r \in \{0,1\} && \forall r \in R && (6)\\
& x_{vi}^r \in \{0,1\} && \forall r \in R, v \in V^r, i \in V^0(r,i) && (7)\\
& f_{ij}^{vw,r} \in \{0,1\} && \forall r \in R, v, w \in V^r, (i,j) \in A^0. && (8)
\end{aligned}
$$

Constraints (2) enforce that each virtual node is mapped onto a (single) substrate node only if the corresponding request is accepted. Constraints (3) and (4) guarantee that the capacity of each physical node and link is not exceeded. Constraints (5) are flow balance constraints ensuring that the routing of the virtual traffic matrices (which is a function of the mapping of the virtual nodes $v, w \in V^r$, as encoded by $x_{vi}^r - x_{wi}^r$) takes place. Each physical node $i \in V^0$ acts as a source node if $x_{vi}^r = 1$ and $x_{wi}^r = 0$, as a sink node if

$x^r_{vi} = 0$ and $x^r_{wi} = 1$, and as a "regular" intermediate node (i.e., not a source nor a sink node) if $x^r_{vi} = x^r_{wi} = 0$. We remark that, if $x^r_{vi} = x^r_{wi} = 1$, then the two virtual nodes v, w are *co-located*, i.e., are mapped to the same physical node and, hence, their traffic demands (for both pairs v, w and w, v) vanish (that is, they do not need to be routed on any physical links). Constraints (6)–(8) denote the nature of the variables. Observe that, since $f^{vw,r}_{ij}$ is an integer variable, a single path unsplittable routing is enforced.

We remark that this formulation, as well as the methods that we will introduce in the remainder of the paper, can be directly adapted to the case of splittable routing by just relaxing Constraints (8) into $f^{vw,r}_{ij} \in [0, 1]$.

2.2 VNE and subproblems: computational complexity

VNE entails the solution of two natural subproblems: node mapping and link mapping. Here, we address the case where both of them allow for admission control. The first subproblem is obtained after dropping Constraints (4) and (5) and the variables reported in (8). The second one is obtained after fixing x^r_{vi} for all $r \in R, v \in V^r$, and $i \in V^0(r, v)$. This way, the first subproblem is a relaxation of VNE, whereas the second one is a restriction.

From a combinatorial point of view, we refer to the first subproblem as a Multi-Knapsack Problem with Grouped items (MKP-G). It is a Multi-Knapsack Problem (MKP), i.e., an extension of the classical knapsack problem where more simultaneous knapsacks are present, in which the items are grouped so that, if an item is put into one of the knapsacks, then all the other items in the same group have to be put in some knapsacks as well. From a VNE perspective, each group corresponds to the set of virtual nodes belonging to a virtual network request, the items are virtual nodes, and the knapsacks correspond to physical nodes.

We refer to the second subproblem as to an Unsplittable Multi-Commodity Flow problem with Admission Control (UMCF-AC). In it, the demands of the different commodities can be neglected and a linear function which associates a profit to each accepted flow is maximized. The problem is also subject to "grouping" constraints by which, if the demand for a pair of virtual nodes is routed, then all the demands between pairs belonging to the same group must be routed as well.

In the following, we discuss on the strong $\mathcal{NP}$-hardness of the two subproblems. The first result is also pointed out in [6]:

Proposition 1 (See [6]) *MKP-G is strongly NP-hard.*

Proof It suffices to assume that each group contains a single item. Then, MKP-G is equivalent to MKP, whose strong $\mathcal{NP}$-hardness is shown in [19]. □

The second result is, to the best of our knowledge, new:

Proposition 2 *UMCF-AC is strongly NP-hard.*

Proof Consider the Edge Disjoint Path Problem (EDPP), which calls for k edge disjoint paths in a directed graph between k pairs of nodes. Construct an UMCF-AC instance with the same graph, unit link capacities, unit profits, unit demands, and a group per demand composed of a single element. Then, UMCF-AC admits a solution of value k if and only if the graph admits k edge disjoint paths. □

For an sketch of the reduction for a given instance, see Figure 2.

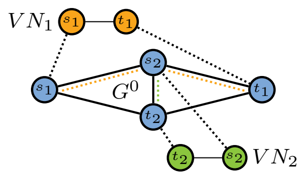


Fig. 2: Reduction from EDPP with $k = 2$ VNs corresponding to paths being embedded without crossing edges due to the presence of unit link demands and capacities.

Note that, when assuming $|V^0(r, v)| = 1$ for all $r \in R$ and $v \in V^r$, VNE becomes an instance of UMCF-AC. Therefore, the proposition also implies that VNE is strongly $\mathcal{NP}$-hard by reduction from EDPP. The two propositions not only illustrate that VNE is a hard problem, but also that both of its naturally occurring subproblems are difficult to solve, at least from a theoretical standpoint.

3 Addressing the case of data uncertainty

In many practical applications, it is reasonable to assume that, for each VN, the actual demand of computing resources and traffic may vary, often substantially, over time. For instance, an online gaming or a movie streaming service may have more or less customers, and, therefore, a different resource consumption, depending on its popularity, which clearly changes over time (with, e.g., peaks for new content releases, after an advertising campaign, and so on). This poses a problem from a network reliability point of view, as it can lead to traffic congestion, quality of service degradations, or, even, service disruptions.

Classical approaches to circumvent data uncertainty typically consider a so-called *worst case* setting, so to guarantee that the network will be operational even for peak values of traffic. Although guaranteeing feasibility, this practice comes at an often unnecessary cost as, in many cases, it is very unlikely for every demand in every VN to simultaneously be at its peak. Indeed, in a number of practical cases, it is reasonable to assume that the probability that *all* demands simultaneously reach their peak values is fairly small. This is reasonable, in our example, when assuming that new content releases and advertising campaign do not take place for *all* services at the same time.

The idea is to look for a solution where the different VNs are provisioned for demands which are smaller than their peak values, thus guaranteeing that the substrate network has sufficient capacity for *almost all* the traffic configurations, only neglecting a few unlikely cases. This way, we are likely to obtain more profitable solutions where more VN requests are embedded, thus avoiding costly issues of overprovisioning.

3.1 A chance-constrained MILP formulation

One natural way of taking demand uncertainties into account is of interpreting ω_v^r and d_{vw}^r not as constants, but as (bounded) random variables. In the reminder of this work, we will assume that, for any $r \in R$, each uncertain node demand ω_v^r is an independent random variable taking value in the symmetric interval $[\bar{\omega}_v^r - \hat{\omega}_v^r, \bar{\omega}_v^r + \hat{\omega}_v^r]$, with *nominal (expected) value* $\bar{\omega}_v^r$ and *maximum deviation* $\hat{\omega}_v^r$. Similarly, we assume that each uncertain link demand $d_{vw}^r \in D^r$ takes values in the symmetric interval $[\bar{d}_{vw}^r - \hat{d}_{vw}^r, \bar{d}_{vw}^r + \hat{d}_{vw}^r]$, centered around the nominal (expected) value $\bar{d}_{vw}^r$, with a maximum deviation $\hat{d}_{vw}^r$.

Let $\epsilon \in [0, 1]$ be the probability with which each constraint is required to be satisfied. When requiring the satisfaction of Constraints (3) and (4) with, at least, a probability of ϵ, we obtain the following chance-constrained MILP formulation of VNE:

$$\max \ (1) \tag{9}$$

$$\text{s.t.} \ \Pr\left(\sum_{r \in R} \sum_{v \in V^r : i \in V^0(r,v)} \omega_v^r x_{vi}^r \leq c_i^0\right) \geq \epsilon \qquad \forall i \in V^0 \tag{10}$$

$$\Pr\left(\sum_{r \in R} \sum_{v,w \in V^r} d_{vw}^r f_{ij}^{vw,r} \leq k_{ij}^0\right) \geq \epsilon \qquad \forall (i,j) \in A^0 \tag{11}$$

$$(2), (5), (6), (7), (8). \tag{12}$$

Note that, if the deterministic Formulation (1)–(8) is solved with *worst case* data, i.e., by setting each random variable to its maximum value, we obtain an embedding which is also a feasible solution to the chance-constrained Formulation (9)–(12) when solved with any ϵ, and (neglecting zero-measure events) an optimal one for $\epsilon = 1$. As we will see with our computational experiments, which we report in Section 5, the objective function value of such solutions is typically very poor.

Clearly, Formulation (1)–(8) can also be solved with *nominal* data, substituting for each random variable its expected value. Although typically yielding much larger objective function values, this choice corresponds to setting $\epsilon = 0$, thus asking for solutions that are feasible only for the zero-measure event where all the uncertain data take a single set of values. As such, these solutions are (theoretically and, often, also in practice) infeasible with probability 1.

3.2 A Γ-robustness approach

For most nontrivial probability distributions of the random variables, chance-constrained problems are, in general, very hard to solve. This is because, to rely on mathematical programming tools, they require a closed-form solution to the integrals corresponding to each probabilistic constraint the derivation of which is, usually, not known. An attractive way to circumvent this drawback, also successfully applied to a number of networking problems, see, e.g., [20], is of recurring to robust optimization and, specifically, to a Γ-robust approach.

Roughly speaking, the Γ-robustness model [21,22] assumes that, in any possible realization of the uncertain data (i.e., of the random variables of the chance-constrained model), at most Γ coefficients will simultaneously deviate from their nominal value. This model naturally meets the features of VNE if we assume that the number of demands simultaneously reaching their peak values is bounded by Γ, as Γ-robust solutions are guaranteed to be feasible for any realization of the uncertain coefficients with at most Γ deviations. What is more, Γ-robustness also establishes, under the sole assumption of independence of the random variables and of symmetry of their intervals, that, if more than Γ deviations occur, feasibility will still be retained with a probability corresponding to a monotone increasing function of Γ. Thus, the model allows to approximate our chance-constrained formulation for an arbitrary ϵ by selecting a suitable value for Γ (see [22] for more details). Most interestingly, the model leads to problems which are computationally much more tractable than those involving chance constraints, as we will show in the following.

In the remainder of the paper, we will refer to the probability that a solution is feasible as *protection level* (and to its estimation, as observed via computational experiments, as *empirical protection level*). The adoption of higher values for Γ, which guarantee a higher protection level, comes at a cost, the so-called *price of robustness*, as a consequence of the set of feasible solutions becoming smaller for larger values of Γ.

3.3 A robust MILP formulation for VNE

Let us now show how to derive an MILP Γ-robustness formulation for VNE. We will address the case where, for each node and link, at most Γ demands deviate from their nominal value. Equivalently, this corresponds to the case where *all but* Γ demands deviate from their worst case value. We recall that, for $\Gamma = 0$, the robust problem corresponds to the original problem with nominal data while, for $\Gamma = \infty$, it corresponds to the original problem with worst case data where all the coefficients simultaneously deviate to their maximum value.

3.3.1 Γ-robust MILP formulation for VNE *I: node demands*

For convenience, define $\mathcal{V}_i^N := \{(r,v) \in R \times \cup_{r\in R} V^r : i \in V^0(r,v)\}$. For any physical node $i \in V^0$, the set $\mathcal{V}_i^N$ corresponds to all the request-node pairs r, v

where the virtual node $v \in V^r$ can be mapped to the physical node i. Letting $\Gamma \in \mathbb{Z}_+$, the (nonlinear) robust counterpart to Constraints (10) is:

$$\underbrace{\sum_{r\in R} \sum_{\substack{v\in V^r:\\ i\in V^0(r,v)}} \bar{\omega}_v^r x_{vi}^r}_{\text{nominal LHS}} + \underbrace{\max_{\substack{T\subseteq \mathcal{V}_i^N\\ |T|\leq \Gamma}} \sum_{(r,v)\in T} \hat{\omega}_v^r x_{vi}^r}_{\text{maximum deviation}} \leq c_i^0 \qquad \forall\, i \in V^0.$$

The constraint accounts for the scenario where the Γ coefficients with the largest value of $\hat{\omega}_v^r x_{vi}^r$ (those in the set T) simultaneously deviate. If the constraint is satisfied for the maximum (total) deviation, the nominal constraint will then be satisfied for any realization in the uncertainty set. As originally shown in [22], this nonlinear constraint can be recast in a linear way (i.e., without the need for the internal max operator) with the introduction of a set of auxiliary variables and constraints. We briefly illustrate this in the following.

Let $z^{rv} = 1$ if $(r,v) \in T$ and 0 otherwise. Assuming that x_{vi}^r is fixed, the inner maximization problem can be cast as the following Linear Program (LP), whose dual variables are reported in brackets:

$$\max \quad \sum_{r\in R} \sum_{\substack{v\in V^r:\\ i\in V^0(r,v)}} (\hat{\omega}_v^r x_{vi}^r)\, z_i^{rv} \tag{13}$$

$$\text{s.t.} \quad \sum_{r\in R} \sum_{\substack{v\in V^r:\\ i\in V^0(r,v)}} z_i^{rv} \leq \Gamma \qquad [\pi_i] \tag{14}$$

$$z_i^{rv} \in [0,1] \qquad \forall\, r \in R, v \in V^r : i \in V^0(r,v) \qquad [\rho_i^{rv}]. \tag{15}$$

Note that, since the problem has a totally unimodular constraint matrix, z_i^{rv} will be integer in any optimal solution. The LP dual reads:

$$\min \quad \Gamma \pi_i + \sum_{r\in R} \sum_{\substack{v\in V^r:\\ i\in V^0(r,v)}} \rho_i^{rv} \tag{16}$$

$$\text{s.t.} \quad \pi_i + \rho_i^{rv} \geq \hat{\omega}_v^r x_{vi}^r \qquad \forall\, r \in R, v \in V^r : i \in V^0(r,v). \tag{17}$$

$$\pi_i \geq 0, \quad \rho_i^{rv} \geq 0 \qquad \forall\, r \in R, v \in V^r : i \in V^0(r,v). \tag{18}$$

By LP duality, any feasible solution to (16)–(18) has an objective function value at least as large as that of an optimal solution to (13)–(15). Therefore, for each $i \in V^0$, the robust counterpart to Constraint (10) corresponds to:

$$\sum_{r\in R} \sum_{v\in V^r : i\in V^0(r,v)} (\bar{\omega}_v^r x_{vi}^r + \rho_i^{rv}) + \Gamma \pi_i \leq c_i^0, \tag{19}$$

together with Constraints (17) and the variables in (18).

3.3.2 Γ-robust MILP formulation for VNE *II: traffic (link) demands*

Let $\mathcal{V}^L := \{(r,v,w) : r \in R, v,w \in V^r\}$. Letting $\Gamma \in \mathbb{Z}_+$, the (nonlinear) robust counterpart to Constraint (11), for all $(i,j) \in A^0$, reads:

$$\underbrace{\sum_{r\in R}\sum_{v,w\in V^r} \bar{d}^r_{vw} f^{vw,r}_{ij}}_{\text{nominal LHS}} + \underbrace{\max_{\substack{T\subseteq\mathcal{V}^L\\|T|\leq\Gamma}} \sum_{(r,v,w)\in T} \hat{d}^r_{vw} f^{vw,r}_{ij}}_{\text{maximum deviation}} \leq k^0_{ij} \qquad \begin{array}{l}\forall r \in R, v,w \in V^r,\\ \forall (i,j) \in A^0.\end{array} \tag{20}$$

Similarly to the node case, a linear reformulation can be obtained after introducing the variables $\pi_{ij}, \rho^{vw,r}_{ij} \geq 0$ for all $r \in R, v,w \in V^r$ and, for all $(i,j) \in A^0$, the constraints:

$$\sum_{r\in R}\sum_{v,w\in V^r} (\bar{d}^r_{vw} f^{vw,r}_{ij} + \rho^{vw,r}_{ij}) + \Gamma\pi_{ij} \leq k^0_{ij} \tag{21}$$

$$\pi_{ij} + \rho^{vw,r}_{ij} \geq \hat{d}^r_{vw} f^{vw,r}_{ij} \qquad \forall r \in R, v,w \in V^r \tag{22}$$

$$\pi_{ij} \geq 0, \quad \rho^{vw,r}_{ij} \geq 0 \qquad \forall r \in R, v,w \in V^r. \tag{23}$$

The complete Γ-robust formulation for VNE is obtained from that in (1)–(8) after substituting for Constraints (3) and (4) their robust counterparts.

4 Heuristics for the *Γ*-robust VNE problem

Although much more tractable than its original chance-constrained counterpart (as it "only" requires the solution of an MILP), the Γ-robust version of VNE is still, as we will see in Section 5, very hard to solve for large instances within a reasonable computing time. Hence, in this section we propose two heuristics approaches to produce good-quality robust solutions at a smaller computational effort. Both approaches rely on splitting the VNE problem into the robust counterparts to the two subproblems which we mentioned before, which are then solved sequentially within a given time limit.

4.1 Two-phase heuristic

First, let us outline our two-phase method. In the first phase, we carry out admission control and Γ-robust node embedding, but neglect link mapping and link capacities. In the second phase, we complete the partial solution found in phase one by looking for a Γ-robust link mapping for the accepted VNs (assuming that their node mapping is fixed, as found in the first phase), while still allowing for VN rejections. Similar ideas (node mapping in the first phase, routing in the second one) have already been applied to VNE, see for instance [8], but, differently from other methods, in our case we allow for VN rejections in both phases, include the robustness aspect in each phase, and solve each subproblem as an MILP. In this sense, due to entailing the solution of two MILPs at each iteration, our algorithm could be classified as a *math heuristic.*

4.1.1 Phase one subproblem

In the first phase, we restrict VNE to the Γ-robust node embedding subproblem with admission control. Formally, this amounts to the Γ-robust MILP:

$$\max \sum_{r \in R} p^r y^r \tag{24}$$

$$\text{s.t.} \sum_{i \in V^0(r,v)} x^r_{vi} = y^r \qquad \forall r \in R, v \in V^r \tag{25}$$

$$\sum_{r \in R} \sum_{v \in V^r : i \in V^0(r,v)} (\bar{\omega}^r_v x^r_{vi} + \rho^{rv}_i) + \Gamma \pi_i \leq c^0_i \qquad \forall i \in V^0 \tag{26}$$

$$\pi_i + \rho^{rv}_i \geq \hat{\omega}^r_v x^r_{vi} \qquad \forall r \in R, v \in V^r, i \in V^0(r,v) \tag{27}$$

$$y^r \in \{0,1\} \qquad \forall r \in R \tag{28}$$

$$x^r_{vi} \in \{0,1\} \qquad \forall r \in R, v \in V^r, i \in V^0(r,v) \tag{29}$$

$$\pi_i \geq 0, \rho^{rv}_i \geq 0 \qquad \forall r \in R, v \in V^r, i \in V^0(r,v). \tag{30}$$

This subproblem is the Γ-robust counterpart to the MKP-G problem that we introduced in Section 2. While, as we have shown, MKP-G is strongly $\mathcal{NP}$-hard, it is fairly easier to solve than the whole Γ-robust VNE problem, as we will see in Section 5. We remark that, by construction, optimal solutions to this subproblem provide upper bounds to the Γ-robust version of VNE.

4.1.2 Phase two subproblem

Let $(\tilde{y}, \tilde{x})$ be a solution to the first phase problem and let $\tilde{R} := \{r \in R : \tilde{y}^r \neq 0\}$ be the subset of requests that have been accepted in the first phase. Also define $\tilde{\Delta}^r \in \{0,1\}^{|V^r| \times |V^r|}$ so that $\tilde{\delta}^r_{vw} = 0$ if and only if $\tilde{x}^r_{vi} = \tilde{x}^r_{wi} = 1$ for some $i \in V^0$ (recall that the corresponding flow vanishes due to co-location). Then, the second phase problem corresponds to the following Γ-robust MILP:

$$\max \sum_{r \in R'} p^r y^r \tag{31}$$

$$\text{s.t.} \sum_{(i,j) \in \delta^+(i)} f^{vw,r}_{ij} - \sum_{(j,i) \in \delta^-(i)} f^{vw,r}_{ji} = \begin{cases} y^r & \text{if } \tilde{x}^r_{vi} = 1 \\ -y^r & \text{if } \tilde{x}^r_{wi} = 1 \\ 0 & \text{else} \end{cases} \qquad \begin{array}{l} \forall i \in V^0, r \in R', \\ \forall v, w \in V^r : \tilde{\delta}^r_{vw} = 1 \end{array} \tag{32}$$

$$\sum_{r \in R'} \sum_{\substack{v,w \in V^r: \\ \tilde{\delta}^r_{vw} = 1}} (\bar{d}^r_{vw} f^{vw,r}_{ij} + \rho^{vw,r}_{ij}) + \Gamma \pi_{ij} \leq k^0_{ij} \qquad \forall (i,j) \in A^0 \tag{33}$$

$$\pi_{ij} + \rho^{vw,r}_{ij} \geq \hat{d}^r_{vw} f^{vw,r}_{ij} \qquad \forall r \in R', v, w \in V^r : \tilde{\delta}^r_{vw} = 1, (i,j) \in A^0 \tag{34}$$

$$y^r \in \{0,1\} \qquad \forall r \in R' \tag{35}$$

$$f^{vw,r}_{ij} \in \{0,1\} \qquad \forall r \in R', v, w \in V^r : \tilde{\delta}^r_{vw} = 1, (i,j) \in A^0 \tag{36}$$

$$\pi_{ij} \geq 0, \rho^{vw,r}_{ij} \geq 0 \qquad \forall r \in R', v, w \in V^r : \tilde{\delta}^r_{vw} = 1, (i,j) \in A^0. \tag{37}$$

The subproblem is the Γ-robust counterpart to the UMCF-AC problem that we introduced in Section 2. In spite of its strong $\mathcal{NP}$-hardness, as we will see in Section 5, this subproblem will be much easier to solve, for all instances that we will consider, than MKP-G.

We note that, by solving the phase one and phase two subproblems in sequence with the same value of Γ, the heuristic that we introduced always provides a lower bound (i.e., a feasible solution) to the Γ-robust VNE problem.

4.1.3 Revised phase one subproblem

Preliminary experiments have shown that, in many cases, more than 50% of the requests accepted in the first phase are then discarded in the second phase. This is a consequence of the fact that the phase one subproblem is oblivious of the routing aspect. Among its feasible solutions, we would indeed prefer one where pairs of virtual nodes sharing a traffic demand are mapped to physical nodes which are as close as possible. This is because, the more links are used in the routing, the higher the consumption of link capacity in the substrate network will be. A sketch of this simple observation can be found in Figure 3.

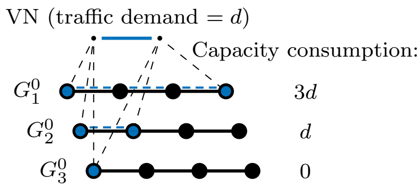


Fig. 3: Three embeddings of a single VN requiring a different amount of physical link capacity. Note how, in case of co-location, the flow between source and sink nodes vanishes.

To circumvent this drawback, we restrict the feasible region of the first phase subproblem to solutions where pairs of virtual nodes sharing a traffic demand are mapped to physical nodes that are not too far away from each other, thus, hopefully, reducing the number of rejections in phase two. For the purpose, we cluster the virtual node pairs $v, w \in V^r$ of any VN request of index $r \in R$ into a set $\mathcal{C}$ of categories, depending on their traffic demand values d^r_{vw}. For each category $C \in \mathcal{C}$, we introduce a parameter z_C which describes the maximum distance, in terms of number of links, that we allow between the physical nodes onto which v and w can be mapped.

More formally, for any two physical nodes $i, j \in V^0$, let $\sigma(i, j)$ be the length of a shortest path (in terms of number of links) in G^0 between them. We partition the different pairs of virtual nodes into three categories, based on the magnitude of their traffic demands: L (for *low*), M (for *medium*), and H (for *high*). Given the three corresponding parameters $z_L, z_M, z_H \in \mathbb{Z}_+$, we introduce the following *distance-bounding* constraints to the phase one problem:

$$x^r_{vi} + x^r_{wj} \leq 1 \quad \forall\, r \in R, v, w \in V^r, i, j \in V^0 : \begin{cases} \sigma(i,j) > z_L \wedge \{v,w\} \in L \\ \sigma(i,j) > z_M \wedge \{v,w\} \in M \\ \sigma(i,j) > z_H \wedge \{v,w\} \in H. \end{cases} \tag{38}$$

The results obtained with this heuristic will be discussed in Section 5.

4.2 Adaptive heuristic

While the two-phase heuristic very effectively provides good quality solutions in a short amount of computing time (see Section 5), its success heavily depends on a good choice of its input parameters z_L, z_M, z_H. To avoid the need for finding a suitable choice of such parameters *a priori*, we now propose an adaptive algorithm in which a suitable parameter setting is chosen automatically.

The method solves a sequence of Γ-robust phase one and phase two subproblems. If, at any iteration, the phase two subproblem terminates accepting all the requests that were accepted in phase one, it halts. If not, it looks for a pair of virtual nodes which, if their VN were accepted, would consume the largest quantity of link capacity, adds a constraint similar to Constraint (38) to the phase one subproblem to reduce the corresponding amount of physical link capacity consumption, and iterates until a time or iteration limit is met.

For any virtual node v, denote by $i(v)$ the physical node to which v has been mapped in the last iteration. At each iteration and for each request $r \in R$, we associate to each pair of virtual nodes $\{v, w\}$, with $v, w \in V^r$, a value equal to the product between their traffic demand and the distance $\sigma(i(v), i(w))$ between the corresponding physical nodes $i(v), i(w)$ in terms of number of links. Then, for each request of index $r \in R$, we identify the following virtual node pair:

$$(v', w') := \underset{v,w \in V^r}{\operatorname{argmax}} \big\{ \max\{d^r_{vw}, d^r_{wv}\} \cdot \sigma(i(v), i(w)) \big\}.$$

Note that, due to employing the shortest path measure in term of number of links, the expression which is maximized corresponds to the *minimum* physical resource consumption that would correspond to any embedded virtual pair.

Then, to impose a mapping to a closer pair of physical nodes, we add to the first phase problem, for the current triple (r, v', w'), the constraints:

$$x^r_{v'i} + x^r_{w'j} \leq 1 \quad \forall i, j \in V^0 : \sigma(i,j) > \left\lceil \frac{\sigma(i(v'), i(w'))}{2} \right\rceil \tag{39}$$

if $\sigma(i(v'), i(w')) > 4$, and the constraints:

$$x^r_{v'i} + x^r_{w'j} \leq 1 \quad \forall i, j \in V^0 : \sigma(i,j) > \sigma(i(v'), i(w')) - 1 \tag{40}$$

if $\sigma(i(v'), i(w')) \leq 4$.

The pseudocode for the adaptive heuristic is reported in Algorithm 1.

Algorithm 1 Adaptive Heuristic

```
while time limit not reached do
    Solve Phase I subproblem via the MILP (24)–(30)
    Let R̃ := {r ∈ R : y^r = 1}
    Solve Phase II subproblem via the MILP (31)–(37)
    if y^r = 1 ∀r ∈ R̃ then
        terminate
    end if
    Let S := {}
    for r ∈ R do
        (v', w') := argmax_{v,w∈V^r} { max {d^r_vw, d^r_wv} · σ(i(v), i(w)) }
        for i, j ∈ V^0 do
            if σ(i(v'), i(w')) > 4 then
                Add Constraint (39) to S
            end if
            if σ(i(v'), i(w')) ≤ 4 then
                Add Constraint (40) to S
            end if
        end for
    end for
    Add all constraints in S to Phase I problem
end while
```

5 Computational results

Our computations are carried out on an Intel(R) Core(TM) i7-3770 CPU @ 3.40 GHz with 32 GB RAM. We employ the state-of-the-art MILP solver CPLEX 12.4, relying on AMPL as modeling language. We set a time limit of 3600 seconds for the exact Γ-robust MILP formulations, adopting a much shorter time limit of 300 seconds per subproblem in both the two-phase and the adaptive heuristics. The latter is run for, at most, 12 iterations.

As physical networks, we consider four instances of similar size and density, all taken from the SNDlib [23] and transformed into directed graphs with antiparallel links: Abilene (12 nodes, 30 links), Atlanta (15, 44), Nobelus (14, 42), and Polska (12, 36). The physical node capacities are randomly drawn from the tuple $(10, 50, 100, 500)$, with a probability of $(0.1, 0.4, 0.4, 0.1)$. Physical link capacities are set to 500 for all the edges.

We consider VN requests with 12 virtual nodes, a profit chosen uniformly at random between 20 and 100, and a random topology with a link density of 0.5. As to the *locality* aspect, we construct each set $V^0(r, v)$ by first sampling uniformly at random a cardinality factor α_v^r from the interval $[\frac{1}{2}, 1]$ and then adding node $i \in V^0$ to $V^0(r, v)$ with a probability α_v^r.

For the virtual node and traffic demands, we mimic a case where historical data are available, creating a *historical* sequence of 100 data sets. First, for each uncertain coefficient ω_v^r or d_{vw}^r, we sample a value from the tuple $(10, 50, 100, 500)$ (scaled by 0.04 for nodes and 0.06 for links) uniformly at random, with a probability of $(0.1, 0.4, 0.4, 0.1)$. The historical sequence for that coefficient is then constructed by adding to the previously sampled value a Gaussian error with zero mean and a standard deviation equal to three times the original value, for each of the 100 snapshots in the sequence, forcing any demand value thus obtained to 0 if negative. Finally, the nominal node and traffic demands $\bar{w}_v^r$ and $\bar{d}_{vw}^r$ are computed as the arithmetic average over the 100 snapshots, computing $\hat{\omega}_v^r$ and $\hat{d}_{vw}^r$ as the largest deviations w.r.t. $\bar{w}_v^r$ and $\bar{d}_{vw}^r$ over the historical sequence.

We generate the instances with an increasing number of requests, that is $|R| \in \{5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 24, 28, 32\}$. They are constructed incrementally for each topology, so that every instance of a given topology with r requests contains the same requests as an instance with the same topology and $r' < r$ requests, plus $r - r'$ additional ones. As a consequence, the value of an optimal solution for any given topology is a nondecreasing function of $|R|$. During the generation of the instances, we choose the random seeds so that the sequence of VN requests is different for each physical topology. The data set thus constructed is composed of 56 instances.

In the following, we will compare the solutions obtained via the different methods w.r.t. their objective function value and their *empirical* protection level. The latter is defined as the number of snapshots, in the historical sequence of each instance, for which no node or link capacity constraint is violated by the robust solution that we have found.

5.1 Exact nominal and worst case solutions

To better motivate the relevance of data uncertainty for VNE, as well as the profitability of a robust optimization approach, we first evaluate the solutions obtained for worst case demands (i.e., for $\Gamma = \infty$ and $\epsilon = 1$) and average demands (for $\Gamma = 0$ and $\epsilon = 0$), as mentioned in Section 3.

Complete results for the full data set are reported in Table 1. The table shows that the instances with the worst case data are quite harder to solve than those with average data (with, overall, an average gap of 32% versus one of 0%, and an average computing time 3.5 times larger), thus showing that the problem gets more difficult for a higher load. More precisely, out of 56 instances, while only 8 instances cannot be solved to optimality with average data (with an average gap of 2.75%), this number increases to 33 instances with worst case data (with an average gap of 53%).

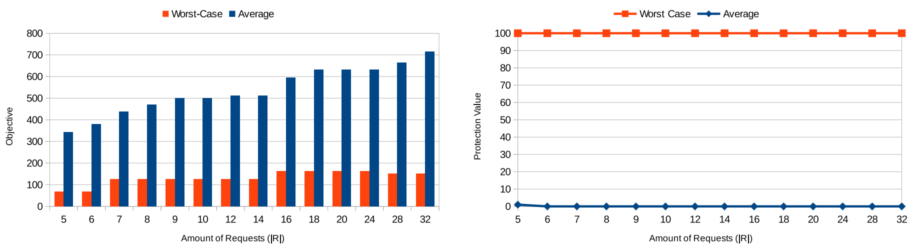
As we anticipated in Section 3, although the objective function values with average data are much larger than those for the worst case, for the majority of instances the corresponding solutions are infeasible in almost all the snapshots of the historical sequence. On the contrary, the solutions with worst case data are always feasible, but at the cost of a very poor objective function value. This is illustrated, for the ABILENE instances, in Figure 4.

5.2 Exact solutions via the Γ-robust MILP formulation

We now illustrate the results obtained with the Γ-robust MILP formulation, first focusing on the ABILENE instances. We assume $\Gamma \in \{1, 2, 3\}$, adopting the

Table 1: Results for the deterministic MILP formulation obtained within 3600 seconds with nominal (average) and worst case data. Entries are rounded to the nearest integer.

		$\|R\|$	5	6	7	8	9	10	12	14	16	18	20	24	28	32	**Avg**
Objective Fct.	Nominal	Abil	342	381	438	471	500	500	512	512	595	632	632	632	664	715	**538**
		Atl	402	444	523	559	623	695	853	970	1061	1197	1273	1324	1443	1447	**915**
		Nob	346	393	441	532	570	629	734	810	894	917	1011	1165	1211	1282	**781**
		Pol	265	312	398	452	550	644	782	875	968	1082	1082	1065	1219	1284	**784**
		Avg	**339**	**383**	**450**	**504**	**561**	**617**	**720**	**792**	**880**	**957**	**1000**	**1047**	**1134**	**1182**	755
	W.-Case	Abil	68	68	125	126	126	126	126	126	163	163	163	163	151	151	**132**
		Atl	251	332	344	210	301	301	194	263	184	174	336	220	227	348	**263**
		Nob	225	225	180	98	189	228	130	146	228	193	228	215	185	237	**193**
		Pol	169	169	217	245	277	277	277	301	301	252	238	238	348	263	**255**
		Avg	**178**	**199**	**217**	**170**	**223**	**233**	**182**	**209**	**219**	**196**	**241**	**209**	**228**	**250**	211
Protection Level	Nominal	Abil	1	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Atl	27	19	1	3	0	0	0	0	0	0	0	0	0	0	**4**
		Nob	54	19	1	0	0	0	0	0	0	0	0	0	0	0	**5**
		Pol	38	29	22	14	7	2	0	0	0	0	0	0	0	0	**8**
		Avg	**30**	**17**	**6**	**4**	**2**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	4
	W.-Case	Abil	100	100	100	100	100	100	100	100	100	100	100	100	100	100	**100**
		Atl	100	100	100	100	100	100	100	100	100	100	100	100	100	100	**100**
		Nob	100	100	100	100	100	100	100	100	100	100	100	100	100	100	**100**
		Pol	100	100	100	100	100	100	100	100	100	100	100	100	100	100	**100**
		Avg	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	100
Optimality Gap	Nominal	Abil	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Atl	0	0	0	0	0	0	0	0	0	0	0	0	0	2	**0**
		Nob	0	0	0	0	0	0	0	4	0	0	0	1	2	4	**1**
		Pol	0	0	0	0	0	0	0	0	0	0	0	5	3	1	**1**
		Avg	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**2**	**1**	**2**	0
	W.-Case	Abil	0	0	0	0	0	0	0	0	0	0	0	0	21	21	**3**
		Atl	32	0	0	81	26	26	96	58	127	145	25	97	100	31	**60**
		Nob	0	0	52	179	44	20	110	87	20	41	20	42	74	35	**52**
		Pol	0	22	13	0	0	0	0	0	0	29	36	36	0	33	**12**
		Avg	**8**	**5**	**16**	**65**	**18**	**11**	**51**	**36**	**37**	**54**	**20**	**44**	**49**	**30**	32
Computing Time	Nominal	Abil	3	14	7	13	168	853	507	364	244	78	16	71	1394	896	**331**
		Atl	1	1	1	4	5	5	10	35	47	15	518	177	3600	3600	**573**
		Nob	1	1	2	6	4	8	12	3600	33	502	3112	3600	3600	3600	**1291**
		Pol	0	1	1	1	2	4	5	8	143	24	1484	3600	3600	3600	**891**
		Avg	**1**	**4**	**3**	**6**	**44**	**218**	**133**	**1002**	**117**	**155**	**1283**	**1862**	**3049**	**2924**	771
	W.-Case	Abil	1	1	226	688	692	1659	1249	1357	31	29	44	34	3600	3600	**944**
		Atl	3600	242	105	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	**3111**
		Nob	30	176	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	**3100**
		Pol	14	3600	3600	3207	2738	2149	2109	255	1890	3600	3600	3600	3298	3600	**2661**
		Avg	**911**	**1005**	**1883**	**2774**	**2657**	**2752**	**2640**	**2203**	**2280**	**2707**	**2711**	**2708**	**3525**	**3600**	**2454**

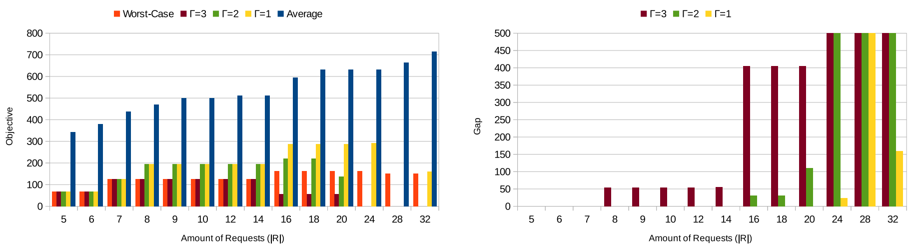


(a) (b)

Fig. 4: (a) Objective function values and (b) protection values for the worst case and average case, reported as a function of $|R|$, for the Abilene instances.

same value for both the node and link capacity constraints. We do not report further results for larger values of Γ as, in our experiments, we achieve a very high empirical protection level already for $\Gamma = 3$.

Figure 5 (a) reports the objective function value for different values of Γ, as a function of $|R|$. Note that the value of an optimal Γ-robust solution should be between the average and worst case ones and that such value should be larger for smaller values of Γ. As we can see, this is not always the case. E.g., we observe that, for $|R| = 16$ and $\Gamma = 3$, as a consequence of prematurely reaching the time limit, we achieve a strictly smaller objective function value than for the worst case. In general, it seems that, with many requests ($|R| > 14$) and for increasing values of Γ, the Γ-robust formulations are more and more difficult to solve. As an example, observe that, for $|R| = 28$, no solution is found for $\Gamma \in \{1, 2, 3\}$, while the solution for $|R| = 32$ and $\Gamma = 1$ has a smaller value than that for the same Γ and $|R| = 24$. This is better shown in Figure 5 (b), which reports the optimality gap of the solutions (truncated to 500 for illustration purposes), showing that, for instances with a large $|R|$ and for larger values of Γ, the exact approach based on the Γ-robust formulation does not scale well.



(a) (b)

Fig. 5: (a) Objective function values for the ABILENE instances, obtained with the different models and reported as a function of $|R|$. (b) Gaps for the ABILENE instances.

We obtain qualitatively comparable results also for the other topologies, as reported in Table 2. Indeed, when considering the full data set (56 instances) with three values of Γ (168 VNE problems in total), in 119 cases we cannot find an optimal solution within the time limit, registering an average gap of 76% (only considering the instances where the gap is finite). In 101 cases, not even a nontrivial solution (i.e., one where at least a single VN is accepted) is found (thus, the gap is infinite). This shows that, when compared to the average and worst case data, with Γ-robustness we obtain much harder problems.

5.3 Two-phase and adaptive heuristics

Let us now consider the two heuristic methods. Differently from the exact case, where we consider $\Gamma \in \{0, 1, 2, 3\}$ for both node and link capacity constraints,

Table 2: Results for the exact MILP Γ-robust formulation obtained within 3600 seconds. Entries are rounded to the nearest integer.

		$\|R\|$	5	6	7	8	9	10	12	14	16	18	20	24	28	32	Avg
Objective Fct.	$\Gamma = 1$	Abil	68	68	125	194	194	194	194	194	288	288	288	292	-	161	196
		Atl	332	374	453	489	553	-	-	-	-	-	-	-	-	-	440
		Nob	294	341	389	480	518	518	-	-	-	-	-	-	-	-	423
		Pol	265	312	398	452	550	550	-	-	-	-	-	-	-	-	421
		Avg	240	274	341	404	454	421	194	194	288	288	288	292	-	161	295
	$\Gamma = 2$	Abil	68	68	125	194	194	194	194	194	220	220	137	-	-	-	164
		Atl	332	374	-	-	-	-	-	-	-	-	-	-	-	-	353
		Nob	294	294	294	98	-	-	-	-	-	-	-	-	-	-	245
		Pol	265	312	321	-	104	76	-	-	-	-	-	-	-	-	180
		Avg	240	262	247	97	149	135	194	194	220	220	137	-	-	-	190
	$\Gamma = 3$	Abil	68	68	125	126	126	126	126	125	57	57	57	-	-	-	96
		Atl	332	-	-	-	-	-	-	-	-	-	-	-	-	-	332
		Nob	225	225	222	-	-	-	-	-	-	-	-	-	-	-	224
		Pol	197	-	-	-	-	-	-	-	-	-	-	-	-	-	197
		Avg	206	147	174	126	126	126	126	125	57	57	57	-	-	-	121
Protection Level	$\Gamma = 1$	Abil	96	99	94	83	89	88	83	97	81	60	80	85	-	88	86
		Atl	98	93	86	63	22	-	-	-	-	-	-	-	-	-	84
		Nob	92	86	94	72	64	78	-	-	-	-	-	-	-	-	81
		Pol	95	85	66	76	33	42	-	-	-	-	-	-	-	-	73
		Avg	95	91	85	33	77	69	83	97	81	60	80	85	-	88	82
	$\Gamma = 2$	Abil	100	100	99	100	98	99	100	99	100	100	100	-	-	-	100
		Atl	100	100	-	-	-	-	-	-	-	-	-	-	-	-	100
		Nob	100	99	100	100	-	-	-	-	-	-	-	-	-	-	100
		Pol	100	96	100	-	100	100	-	-	-	-	-	-	-	-	99
		Avg	100	99	100	100	99	99	100	99	100	100	100	-	-	-	100
	$\Gamma = 3$	Abil	100	100	100	100	100	100	100	100	100	100	100	-	-	-	100
		Atl	100	-	-	-	-	-	-	-	-	-	-	-	-	-	100
		Nob	100	100	100	-	-	-	-	-	-	-	-	-	-	-	100
		Pol	100	-	-	-	-	-	-	-	-	-	-	-	-	-	100
		Avg	100	100	100	100	100	100	100	100	100	100	100	-	-	-	100
Optimality Gap	$\Gamma = 1$	Abil	0	0	0	0	0	0	0	0	0	0	0	23	∞	159	14
		Atl	0	0	0	0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
		Nob	0	0	0	0	0	0	∞	∞	∞	∞	∞	∞	∞	∞	0
		Pol	0	0	0	0	0	0	∞	∞	∞	∞	∞	∞	∞	∞	0
		Avg	0	0	0	0	0	0	0	0	0	0	0	23	-	159	14
	$\Gamma = 2$	Abil	0	0	0	0	0	0	0	0	31	31	110	∞	∞	∞	16
		Atl	0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
		Nob	0	0	16	319	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	84
		Pol	0	0	24	∞	429	624	∞	∞	∞	∞	∞	∞	∞	∞	215
		Avg	0	0	13	160	214	312	0	0	31	31	110	-	-	-	79
	$\Gamma = 3$	Abil	0	0	0	54	54	54	54	55	405	405	405	∞	∞	∞	135
		Atl	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	-
		Nob	0	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
		Pol	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	0
		Avg	0	0	0	54	54	54	54	55	405	405	405	-	-	-	135
Computing Time	$\Gamma = 1$	Abil	2	1	2	61	3	33	5	49	6	671	146	3600	3600	3600	841
		Atl	1	3	6	3600	10	3600	3600	3600	3600	3600	3600	3600	3600	3600	2573
		Nob	2	3	3	2068	13	19	3600	3600	3600	3600	3600	3600	3600	3600	2208
		Pol	3	270	338	11	1975	2890	3600	3600	3600	3600	3600	3600	3600	3600	2449
		Avg	2	69	87	1435	500	1635	2701	2712	2702	2868	2737	3600	3600	3600	2018
	$\Gamma = 2$	Abil	4	5	36	425	784	156	399	441	3601	3600	3600	3600	3600	3600	1704
		Atl	186	2192	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3256
		Nob	8	433	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3117
		Pol	144	2502	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3275
		Avg	86	1283	2709	2806	2896	2739	2800	2810	3600	3600	3600	3600	3600	3600	2838
	$\Gamma = 3$	Abil	12	31	37	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	2834
		Atl	876	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3405
		Nob	97	751	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3146
		Pol	180	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3356
		Avg	291	1996	2709	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3600	3185

in the two heuristics methods we always set $\Gamma = 0$ for the second subproblem (which carries out the link mapping). This is because, even without explicitly accounting for robustness in it, we still obtain solutions with a very high empirical protection level, as we will better illustrate in the following.

Let us first focus on the two-phase heuristic. As described in Section 4, we cluster the virtual nodes pairs into the three categories L, M, H (low, medium, and high). Each pair $v, w \in V^r$ belongs to H if $d^r_{vw} \geq 50$, to M if $10 \leq d^r_{vw} < 50$, and to L otherwise. To assess the sensitivity of the w.r.t. the parameters z_L, z_M, and z_H, we illustrate the results obtained with the method for all the combinations of $z_L, z_M, z_H \in \{|V^0|, \frac{|V^0|}{2}, \frac{|V^0|}{4}, 2, 1\}$, with $z_L \leq z_M \leq z_H$. Table 3 reports a comparison over all parameter settings, aggregated over all substrate networks and all Γ values. Let $SOL(\Gamma, s, r, p)$ be the solution value of the two-phase heuristic for a given $\Gamma \in G := \{0, 1, 2, 3\}$, substrate network $s \in S :=$ {abilene, atlanta, nobel-us, polska}, number of requests $r = |R| \in \{5, 6, 7, 8, 9, 10, 12, 14, 16, 18, 20, 24, 28, 32\}$, and parameter setting $p \in P$. Then, for a fixed r and parameter setting p_0, each entry of Table 3 is computed as $\frac{1}{|G|\,|S|\,|P|} \sum_{\Gamma \in G} \sum_{s \in S} \sum_{p \in P} \frac{SOL(\Gamma,s,r,p)}{SOL(\Gamma,s,r,p_0)}$. Thus, each entry describes the relative quality of the solutions obtained with parameter setting p_0, compared to the solutions obtained with all other parameter settings. The lower the value, the better p_0 performs w.r.t. the others (the results for the best setting, on average, over all the instances are underlined). As we can see, the unique "winner" is $z_L = |V^0|$, $z_M = 2$, and $z_H = 1$.

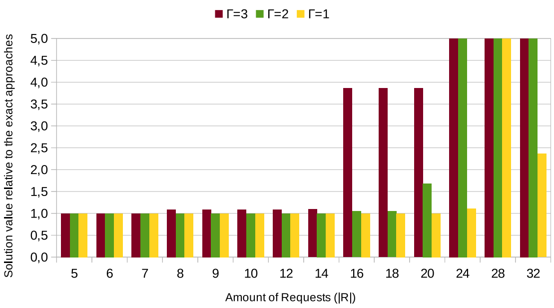


Fig. 6: Objective function value ratio between the best solution found via the two-phase heuristic ($z_L = |V^0|$, $z_M = 2$, and $z_H = 1$) and the exact formulation, reported as a function of $|R|$, for the Abilene instances.

We now compare the results for the "winning" parameter setting to the exact approach. We first illustrate the results for the abilene instances. In Figure 6, we report the ratio between the solution values found heuristically and the best ones obtained via the exact formulation (in case no nonzero solution is found with the exact method, we set this ratio to 5.0). Interestingly, for $\Gamma = 1$, the heuristic provides competing solutions to those obtained via the exact approach for all the abilene instances. For the first half of the instances (those with $|R| \leq 14$), the heuristic achieves a comparable objective

Table 3: Performance ratio of each parameter setting $z_L, z_M, z_H \in \{|V^0|, \frac{|V^0|}{2}, \frac{|V^0|}{4}, 2, 1\}$ with $z_L \leq z_M \leq z_H$, for each $|R|$. The results for the, on average, best setting over all the instances are underlined.

z_L	z_M	z_H	$\|R\|$ 5	6	7	8	9	10	12	14	16	18	20	24	28	32	**Avg**
1	1	1	1.41	1.31	1.41	1.56	1.60	1.39	1.43	1.44	1.48	1.36	1.48	1.34	1.10	1.01	**1.38**
2	1	1	1.16	1.09	1.15	1.20	1.21	1.17	1.17	1.15	1.00	0.98	1.01	1.00	0.93	0.89	**1.08**
2	2	1	0.97	0.96	1.02	0.99	0.98	0.98	0.95	0.94	0.96	0.93	0.91	0.91	0.93	0.89	**0.95**
2	2	2	0.97	0.97	1.01	0.98	0.98	0.99	0.97	0.96	0.97	0.94	0.92	0.94	0.94	0.99	**0.97**
$\|V^0\|$	1	1	1.16	1.08	1.15	1.19	1.20	1.17	1.16	1.14	1.01	0.99	0.99	0.97	0.92	0.89	**1.07**
$\|V^0\|$	2	1	0.96	0.96	0.99	0.96	0.95	0.96	0.93	0.93	0.93	0.93	0.93	0.89	0.94	0.91	**0.94**
$\|V^0\|$	2	2	0.96	0.96	1.01	0.97	0.98	0.99	0.97	0.95	0.96	0.97	0.99	0.97	0.98	0.98	**0.97**
$\|V^0\|$	$\|V^0\|$	1	0.96	0.96	0.98	0.97	0.96	0.96	1.05	0.96	0.99	0.98	0.98	0.99	0.96	1.01	**0.98**
$\|V^0\|$	$\|V^0\|$	2	0.96	0.96	0.96	0.97	0.98	0.98	1.04	1.02	1.03	1.07	1.05	1.05	1.08	1.04	**1.01**
$\|V^0\|$	$\|V^0\|$	$\frac{\|V^0\|}{2}$	0.96	0.98	1.00	1.07	1.06	1.05	1.12	1.17	1.13	1.14	1.14	1.15	1.16	1.10	**1.09**
$\|V^0\|$	$\|V^0\|$	$\frac{\|V^0\|}{4}$	0.96	0.98	0.99	0.99	1.00	0.99	1.03	1.05	1.09	1.13	1.10	1.11	1.12	1.08	**1.05**
$\|V^0\|$	$\|V^0\|$	$\|V^0\|$	0.96	0.98	1.00	1.07	1.06	1.05	1.12	1.17	1.13	1.14	1.14	1.15	1.16	1.10	**1.09**
$\|V^0\|$	$\frac{\|V^0\|}{2}$	1	0.96	0.96	0.98	0.96	0.96	0.96	0.96	0.96	0.98	0.98	0.98	0.99	0.97	1.00	**0.97**
$\|V^0\|$	$\frac{\|V^0\|}{2}$	2	0.96	0.96	0.96	0.97	0.98	1.00	1.03	1.02	1.03	1.07	1.05	1.06	1.10	1.04	**1.02**
$\|V^0\|$	$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{2}$	0.96	0.98	1.00	1.07	1.06	1.05	1.11	1.16	1.12	1.14	1.15	1.15	1.14	1.11	**1.08**
$\|V^0\|$	$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{4}$	0.96	0.98	0.99	0.99	1.00	0.99	1.01	1.05	1.07	1.13	1.10	1.12	1.13	1.08	**1.04**
$\|V^0\|$	$\frac{\|V^0\|}{4}$	1	0.96	0.96	0.96	0.97	0.94	0.97	0.95	0.95	0.96	0.95	0.97	0.93	0.95	0.94	**0.95**
$\|V^0\|$	$\frac{\|V^0\|}{4}$	2	0.97	0.96	0.97	0.97	0.98	1.01	1.01	0.99	0.99	0.99	1.02	1.06	1.02	1.05	**1.00**
$\|V^0\|$	$\frac{\|V^0\|}{4}$	$\frac{\|V^0\|}{4}$	0.96	0.97	0.98	0.99	0.99	1.00	0.99	1.03	1.04	1.05	1.02	1.05	1.06	1.67	**1.06**
$\frac{\|V^0\|}{2}$	1	1	1.16	1.08	1.15	1.19	1.20	1.18	1.16	1.14	1.01	0.99	0.99	0.95	0.92	0.88	**1.07**
$\frac{\|V^0\|}{2}$	2	1	0.96	0.96	0.99	0.96	0.95	0.96	0.93	0.92	0.94	0.96	0.93	0.91	0.95	0.90	**0.95**
$\frac{\|V^0\|}{2}$	2	2	0.96	0.96	1.01	0.97	0.98	1.00	0.97	0.95	0.96	0.97	0.99	0.98	0.97	0.99	**0.98**
$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{2}$	1	0.96	0.96	0.98	0.96	0.96	0.96	0.96	0.96	0.98	0.98	0.98	0.99	0.97	1.02	**0.97**
$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{2}$	2	0.96	0.96	0.96	0.97	0.98	0.98	1.02	1.02	1.04	1.07	1.04	1.06	1.08	1.02	**1.01**
$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{2}$	0.96	0.98	1.00	1.07	1.12	1.05	1.11	1.14	1.13	1.14	1.13	1.16	1.15	1.11	**1.09**
$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{4}$	0.96	0.98	0.99	0.99	1.00	0.99	1.01	1.05	1.10	1.13	1.10	1.12	1.11	1.09	**1.05**
$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{4}$	1	0.96	0.96	0.96	0.97	0.94	0.97	0.95	0.95	0.96	0.95	0.96	0.94	0.96	0.94	**0.96**
$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{4}$	2	0.96	0.96	0.97	0.97	0.98	1.03	1.01	0.99	0.99	0.99	1.02	1.06	1.02	1.05	**1.00**
$\frac{\|V^0\|}{2}$	$\frac{\|V^0\|}{4}$	$\frac{\|V^0\|}{4}$	0.96	0.96	0.98	0.99	0.99	1.00	0.99	1.02	1.04	1.05	1.02	1.04	1.05	1.65	**1.05**
$\frac{\|V^0\|}{4}$	1	1	1.16	1.08	1.15	1.19	1.20	1.18	1.17	1.13	1.02	1.00	0.99	0.96	0.94	0.90	**1.08**
$\frac{\|V^0\|}{4}$	2	1	0.97	0.97	1.02	0.96	0.95	0.96	0.93	0.92	0.94	0.94	0.92	0.95	0.98	0.93	**0.95**
$\frac{\|V^0\|}{4}$	2	2	0.97	0.97	1.01	0.96	0.96	0.97	0.95	0.95	0.95	0.95	0.98	0.95	1.03	1.03	**0.97**
$\frac{\|V^0\|}{4}$	$\frac{\|V^0\|}{4}$	1	0.96	0.96	0.96	0.96	0.96	0.96	0.94	0.94	0.93	0.94	0.95	0.95	0.96	0.92	**0.95**
$\frac{\|V^0\|}{4}$	$\frac{\|V^0\|}{4}$	2	0.96	0.97	0.96	0.97	0.97	0.97	0.99	0.98	0.97	0.94	1.00	1.00	1.02	1.03	**0.98**
$\frac{\|V^0\|}{4}$	$\frac{\|V^0\|}{4}$	$\frac{\|V^0\|}{4}$	0.96	0.96	0.97	0.99	0.98	1.00	0.95	0.97	1.01	1.01	0.99	1.06	1.35	1.04	**1.02**

Table 4: Detailed results for the two-phase heuristic with $z_L = |V^0|$, $z_M = 2$, and $z_H = 1$.

		$\|R\|$	5	6	7	8	9	10	12	14	16	18	20	24	28	32	**Avg**
Objective Function	$\Gamma = 0$	Abi	342	381	438	471	500	500	512	512	595	632	632	632	664	715	**538**
		Atl	402	444	523	559	623	695	853	900	988	1001	1119	1091	1179	1090	**819**
		Nob	346	393	441	532	570	629	696	771	782	841	851	997	1105	1128	**720**
		Pol	265	312	398	452	522	644	702	718	726	714	795	873	774	866	**626**
		Avg	**339**	**383**	**450**	**504**	**554**	**617**	**691**	**725**	**773**	**797**	**849**	**898**	**931**	**950**	**676**
	$\Gamma = 1$	Abi	68	68	125	194	194	194	194	194	288	288	288	360	360	381	**228**
		Atl	332	374	453	489	553	625	783	900	912	960	949	925	892	903	**718**
		Nob	294	341	389	480	518	518	555	631	621	636	606	724	650	637	**543**
		Pol	265	312	398	452	522	550	570	606	642	633	602	573	651	732	**536**
		Avg	**240**	**274**	**341**	**404**	**447**	**472**	**526**	**583**	**616**	**629**	**611**	**646**	**638**	**663**	**506**
	$\Gamma = 2$	Abi	68	68	125	194	194	194	194	194	220	220	220	256	277	277	**193**
		Atl	332	374	453	489	511	519	605	655	686	697	703	683	737	730	**584**
		Nob	294	294	342	366	402	402	402	395	401	360	404	467	411	410	**382**
		Pol	265	312	368	394	454	406	478	500	491	526	554	573	601	606	**466**
		Avg	**240**	**262**	**322**	**361**	**390**	**380**	**420**	**436**	**450**	**451**	**470**	**495**	**507**	**506**	**406**
	$\Gamma = 3$	Abi	68	68	68	126	126	126	126	126	220	220	220	220	220	220	**154**
		Atl	332	332	411	447	447	447	459	497	535	560	497	585	604	582	**481**
		Nob	225	225	273	282	320	320	320	319	320	320	330	457	366	354	**317**
		Pol	197	244	322	346	397	406	456	500	500	526	526	475	500	585	**427**
		Avg	**206**	**217**	**269**	**300**	**323**	**325**	**340**	**361**	**394**	**407**	**393**	**434**	**423**	**435**	**345**
Protection Level	$\Gamma = 0$	Abi	0	1	1	0	1	2	0	0	0	0	1	1	0	0	**1**
		Atl	3	11	6	8	1	1	2	0	1	0	0	0	0	0	**2**
		Nob	37	39	28	9	6	3	1	0	0	0	0	0	0	0	**9**
		Pol	18	4	7	2	2	0	1	0	0	0	0	0	0	0	**2**
		Avg	**15**	**14**	**11**	**5**	**3**	**2**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**4**
	$\Gamma = 1$	Abi	100	99	97	92	92	92	96	94	82	94	88	78	77	74	**90**
		Atl	99	86	86	76	70	64	54	49	57	45	43	49	40	60	**63**
		Nob	88	91	84	81	74	75	77	52	62	56	64	59	59	81	**72**
		Pol	98	75	82	75	67	64	57	56	69	66	64	71	68	60	**69**
		Avg	**96**	**88**	**87**	**81**	**76**	**74**	**71**	**63**	**68**	**65**	**65**	**64**	**61**	**69**	**73**
	$\Gamma = 2$	Abi	100	100	100	99	100	100	100	100	100	100	100	100	100	100	**100**
		Atl	100	99	97	99	98	99	99	99	95	97	97	99	97	100	**98**
		Nob	100	100	99	99	99	100	98	99	98	100	99	99	99	99	**99**
		Pol	99	98	99	99	97	100	97	100	91	98	98	96	94	98	**97**
		Avg	**100**	**99**	**99**	**99**	**99**	**100**	**99**	**100**	**96**	**99**	**99**	**99**	**98**	**99**	**99**
	$\Gamma = 3$	Abi	100	100	100	100	100	100	100	100	100	100	100	100	100	100	**100**
		Atl	100	100	100	100	100	100	100	100	100	100	100	100	100	100	**100**
		Nob	100	100	100	100	100	100	100	100	100	100	100	100	100	100	**100**
		Pol	100	100	100	100	100	100	100	100	100	100	100	100	100	99	**100**
		Avg	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**

function value (with a ratio close to 1.0), while it clearly outperforms the exact approaches on the harder instances ($|R| \geq 16$) where, for $\Gamma = 3$, the heuristic solutions are better by a factor larger than 3.5.

The results for the complete data set can be found in Table 4. Times are reported in Table 5. We remark that the heuristic method finds nonzero solutions for all the instances and for all values of Γ, whereas, with the exact method, we find nonzero solutions for only 87 cases out of 168. For the cases where a solution to the exact formulation is known, we find solutions which are substantially better than the best ones found via the exact method within the time limit. On average, the heuristic methods yields solutions with an objective function value that is better by a factor of 1.42. If we restrict to $\Gamma > 0$, this factor goes up to 2.04.

We remark that, for all the instances that can be solved to optimality with the exact approach, equivalent solution values are obtained with the heuristic approaches. This indicates that, at least on the smaller instances for which the optimal solution value is known, the heuristics are definitely competitive

Table 5: Detailed computing times for the two-phase heuristic with $z_L = |V^0|, z_M = 2$, and $z_H = 1$.

		$\|R\|$	5	6	7	8	9	10	12	14	16	18	20	24	28	32	**Avg**
Computing Time Phase I	$\Gamma = 0$	Abilene	0	0	0	0	1	5	6	7	1	2	2	2	1	4	**2**
		Atlanta	0	0	0	0	0	0	0	0	0	0	1	1	14	15	**2**
		Nobel-us	0	0	0	0	0	0	0	300	0	0	9	13	14	95	**31**
		Polska	0	0	0	0	0	0	0	0	1	1	1	8	300	59	**26**
		Avg	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**77**	**1**	**1**	**3**	**6**	**82**	**43**	**15**
	$\Gamma = 1$	Abilene	0	0	0	0	0	0	0	0	0	0	0	1	300	300	**43**
		Atlanta	0	0	0	0	0	0	1	2	300	300	300	300	300	300	**129**
		Nobel-us	0	0	0	0	0	1	1	8	300	300	300	300	300	300	**129**
		Polska	0	0	0	0	0	0	0	300	300	300	300	300	300	300	**150**
		Avg	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**77**	**225**	**225**	**225**	**225**	**300**	**300**	**113**
	$\Gamma = 2$	Abilene	0	0	0	1	0	1	0	1	13	13	22	47	300	272	**48**
		Atlanta	0	0	0	1	2	3	300	300	300	300	300	300	300	300	**172**
		Nobel-us	0	0	0	300	300	300	300	300	300	300	300	300	300	300	**236**
		Polska	0	1	300	300	300	300	300	300	300	300	300	300	300	300	**257**
		Avg	**0**	**0**	**75**	**150**	**151**	**151**	**225**	**225**	**228**	**228**	**230**	**237**	**300**	**293**	**178**
	$\Gamma = 3$	Abilene	0	0	0	1	0	1	1	1	2	2	2	34	68	80	**14**
		Atlanta	0	1	1	5	13	7	6	18	300	300	300	300	300	300	**132**
		Nobel-us	0	0	0	1	3	3	300	300	300	300	300	300	300	300	**172**
		Polska	0	1	300	139	300	300	300	300	300	300	300	300	300	300	**246**
		Avg	**0**	**0**	**76**	**37**	**79**	**78**	**152**	**155**	**226**	**226**	**226**	**233**	**242**	**245**	141
Computing Time Phase II	$\Gamma = 0$	Abilene	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Atlanta	0	0	0	0	0	0	1	2	3	9	5	13	107	17	**11**
		Nobel-us	0	0	0	0	0	0	1	1	8	1	1	2	2	2	**1**
		Polska	0	0	0	0	1	0	1	1	1	1	1	0	1	1	**1**
		Avg	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**1**	**3**	**3**	**2**	**4**	**28**	**5**	**3**
	$\Gamma = 1$	Abilene	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Atlanta	0	0	0	0	0	0	0	0	0	0	0	9	0	0	**1**
		Nobel-us	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Polska	0	0	0	0	0	0	0	0	0	0	0	1	0	0	**0**
		Avg	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**2**	**0**	**0**	**0**
	$\Gamma = 2$	Abilene	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Atlanta	0	0	0	0	1	0	0	0	0	0	0	0	0	0	**0**
		Nobel-us	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Polska	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Avg	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
	$\Gamma = 3$	Abilene	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Atlanta	0	0	0	0	0	0	0	1	0	0	0	0	0	0	**0**
		Nobel-us	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Polska	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Avg	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Computing Time Phase I+II	$\Gamma = 0$	Abilene	0	0	0	0	1	5	6	7	1	2	2	2	1	4	2
		Atlanta	0	0	0	0	0	0	1	2	3	9	6	14	121	32	13
		Nobel-us	0	0	0	0	0	0	1	301	9	1	10	15	16	97	32
		Polska	0	0	0	0	1	0	1	1	2	1	2	8	301	61	27
		Avg	**0**	**0**	**0**	**0**	**0**	**1**	**2**	**78**	**3**	**3**	**5**	**10**	**110**	**48**	19
	$\Gamma = 1$	Abilene	0	0	0	0	0	0	0	0	0	0	0	1	300	300	**43**
		Atlanta	0	0	0	0	0	0	2	2	300	300	300	309	300	300	**130**
		Nobel-us	0	0	0	0	0	1	1	8	300	300	300	300	300	300	**129**
		Polska	0	0	0	0	1	1	1	300	300	300	300	301	300	300	**150**
		Avg	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**78**	**225**	**225**	**225**	**228**	**300**	**300**	**113**
	$\Gamma = 2$	Abilene	0	0	0	1	0	1	0	1	13	13	22	47	300	272	**48**
		Atlanta	0	0	0	1	3	4	300	300	300	300	300	300	300	300	**172**
		Nobel-us	0	0	0	300	300	300	300	300	300	300	300	300	300	300	**236**
		Polska	0	1	300	300	300	300	300	300	300	300	300	300	300	300	**257**
		Avg	**0**	**0**	**75**	**151**	**151**	**151**	**225**	**225**	**228**	**228**	**231**	**237**	**300**	**293**	**178**
	$\Gamma = 3$	Abilene	0	0	0	1	0	1	1	1	2	2	2	34	68	80	**14**
		Atlanta	0	1	1	5	13	7	6	20	300	300	300	300	300	300	**132**
		Nobel-us	0	0	0	1	4	3	300	300	300	300	300	300	300	300	**172**
		Polska	0	1	300	139	300	300	300	300	300	300	300	300	300	300	**246**
		Avg	**0**	**0**	**76**	**37**	**79**	**78**	**152**	**155**	**226**	**226**	**226**	**234**	**242**	**245**	**141**

in terms of solution quality. Table 5 also illustrates how easier the second phase problem is w.r.t. the first one in terms of computing times.

It is worth mentioning though that, if one is not able to find a good parameter setting beforehand, the variability of the two-phase heuristic in terms of solution quality can be quite high. For a visual depiction, see Figure 7, which reports the solution values for all parameter settings for the ABILENE instance. This phenomenon better shown in Table 6, which reports the minimum and maximum objective function value achieved with different parameter settings for each instance. Remarkably, the differences in solution quality are quite significant. As an example, for $\Gamma = 2$ we have solutions between, on average, 427 and 261.

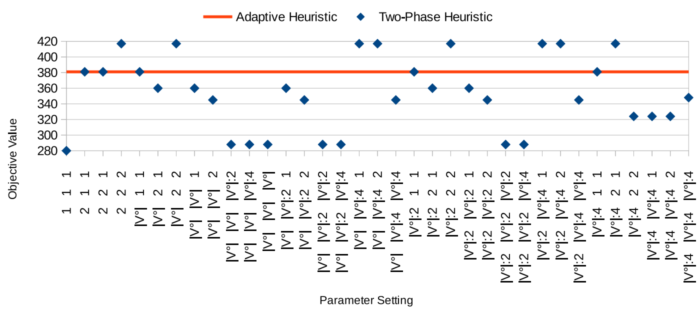


Fig. 7: Objective function values (blue) for the two-phase heuristic on the ABILENE instances, with $|R| = 28$ and $\Gamma = 1$ with $z_L, z_M, z_H \in \left\{|V^0|, \frac{|V^0|}{2}, \frac{|V^0|}{4}, 2, 1\right\}$ and $z_L \leq z_M \leq z_H$. The corresponding solution obtained via the adaptive heuristic is shown in red.

Let us now consider the adaptive heuristic which, by design and differently from the two-phase heuristic, does not depend on a user-supplied parameter initialization. The corresponding results are shown in Table 7. As we can see, the adaptive heuristic provides results which are comparable to those obtained with the two-phase heuristic employing the "winning" parameter setting—although, on average, the results for the former are slightly worse (although by a mere 3%) than those for the latter. Nevertheless, there are cases where the adaptive heuristic improves over the two-phase heuristic with the "winning" parameter setting $z_L = |V^0|$, $z_M = 2$, and $z_H = 1$, such as for ABILENE with $|R| = 28$, as can be observed in Figure 7.

Overall, both heuristic methods dramatically outperform the exact approach, with the adaptive one being able to do so even without an *a priori* knowledge of a "good" parameter setting (although at the cost of a higher computing time when compared to the two-phase heuristic).

Table 6: Best (max) and worst (min) solution values obtained via the two-phase heuristic over all parameter settings, i.e., $z_L, z_M, z_H \in \left\{|V^0|, \frac{|V^0|}{2}, \frac{|V^0|}{4}, 2, 1\right\}$, with $z_L \leq z_M \leq z_H$. Entries are rounded to the nearest integer.

	$\|R\|$	5	6	7	8	9	10	12	14	16	18	20	24	28	32	∅
$\Gamma = 0$	Abilene	342	381	438	471	500	500	512	512	595	632	632	632	664	715	**538**
		224	301	339	304	332	332	304	318	342	379	438	396	413	369	**342**
	Atlanta	402	444	523	559	623	695	853	970	1061	1118	1206	1262	1416	1387	**894**
		332	363	453	489	517	536	582	622	631	712	665	664	633	701	**564**
	Nobel-us	346	393	441	532	570	629	734	846	894	917	1011	1176	1223	1313	**788**
		346	393	441	487	522	499	603	617	705	712	660	715	725	786	**587**
	Polska	265	312	398	452	550	644	782	837	968	1054	961	1011	1082	1141	**747**
		265	312	370	424	473	539	614	585	580	504	506	473	535	601	**484**
	Avg (max)	**339**	**383**	**450**	**504**	**561**	**617**	**720**	**791**	**880**	**930**	**953**	**1020**	**1096**	**1139**	**742**
	Avg (min)	**292**	**342**	**401**	**426**	**461**	**477**	**526**	**536**	**565**	**577**	**567**	**562**	**577**	**614**	**494**
$\Gamma = 1$	Abilene	68	68	125	194	194	194	194	194	288	288	288	360	417	417	**235**
		0	0	57	57	57	57	57	57	151	151	151	223	280	280	**113**
	Atlanta	332	374	453	489	553	625	783	900	912	976	955	996	1070	1115	**752**
		251	251	330	366	366	438	596	588	552	644	574	622	0	0	**398**
	Nobel-us	294	341	389	480	518	518	623	652	631	636	649	744	776	775	**573**
		196	243	291	382	420	420	420	464	464	464	519	644	649	539	**437**
	Polska	265	312	398	452	550	550	600	644	644	687	687	698	793	856	**581**
		235	282	360	360	458	435	480	537	557	532	530	525	501	574	**455**
	Avg (max)	**240**	**274**	**341**	**404**	**454**	**472**	**550**	**598**	**619**	**647**	**645**	**700**	**764**	**791**	**535**
	Avg (min)	**171**	**194**	**260**	**291**	**325**	**338**	**388**	**412**	**431**	**448**	**444**	**504**	**358**	**348**	**351**
$\Gamma = 2$	Abilene	68	68	125	194	194	194	194	194	288	288	288	256	288	288	**209**
		0	0	57	57	57	57	57	57	57	57	57	57	114	114	**57**
	Atlanta	332	374	453	489	511	519	610	722	709	714	739	759	784	740	**604**
		251	251	330	366	366	438	524	559	548	499	508	449	89	0	**370**
	Nobel-us	294	294	342	366	402	402	402	402	402	433	419	512	486	478	**402**
		196	196	196	196	234	304	341	278	278	278	278	328	366	0	**248**
	Polska	265	312	370	422	454	466	494	528	538	573	573	573	624	704	**493**
		235	282	340	384	397	397	381	399	367	334	359	404	447	417	**367**
	Avg (max)	**240**	**262**	**323**	**368**	**390**	**395**	**425**	**462**	**484**	**502**	**505**	**525**	**546**	**553**	**427**
	Avg (min)	**171**	**182**	**231**	**251**	**264**	**299**	**326**	**323**	**313**	**292**	**301**	**310**	**254**	**133**	**261**
$\Gamma = 3$	Abilene	68	68	125	137	137	137	137	137	220	220	220	220	277	277	**170**
		0	0	57	57	57	57	57	57	57	57	57	57	114	114	**57**
	Atlanta	332	332	411	447	447	447	459	560	560	560	560	585	648	644	**499**
		251	251	330	366	366	366	459	449	451	445	451	499	544	0	**373**
	Nobel-us	225	225	273	282	320	320	320	320	320	320	337	457	453	468	**331**
		82	129	129	129	129	266	157	158	158	266	158	242	275	327	**186**
	Polska	197	244	322	346	406	406	456	500	500	554	526	526	571	638	**442**
		169	169	283	271	271	271	321	305	305	324	324	356	399	428	**300**
	Avg (max)	**206**	**217**	**283**	**303**	**328**	**328**	**343**	**379**	**400**	**414**	**411**	**447**	**487**	**507**	**361**
	Avg (min)	**126**	**137**	**200**	**206**	**206**	**240**	**249**	**242**	**243**	**273**	**248**	**289**	**333**	**217**	**229**

5.4 Considerations on the protection level

We now focus on the empirical protection level achieved by the different approaches. Recall that, when adopting worst case or average data, the protection level is equal to 0 for the former whereas, as it can be observed computationally, it is almost always equal to 100% for the latter.

Table 7: Results for the adaptive heursitic.

		|R|	5	6	7	8	9	10	12	14	16	18	20	24	28	32	Avg
Objective Fct. Values	Γ = 0	Abi	322	381	438	450	480	480	472	472	566	507	507	564	513	658	486
		Atl	402	444	523	559	623	695	783	906	866	929	956	937	990	920	752
		Nob	346	393	441	532	570	629	734	770	879	842	836	1046	989	1106	722
		Pol	265	312	398	452	550	644	752	739	750	740	597	626	724	798	596
		Avg	334	383	450	498	556	612	685	722	765	755	724	793	804	871	639
	Γ = 1	Abi	68	68	125	194	194	194	194	194	288	288	288	360	381	381	230
		Atl	332	374	453	453	553	625	783	762	833	832	850	750	605	602	629
		Nob	294	341	389	480	480	473	586	607	636	630	609	728	750	748	554
		Pol	265	312	398	452	522	522	553	604	595	640	623	649	572	706	530
		Avg	240	274	341	395	437	454	529	542	588	598	593	622	577	609	486
	Γ = 2	Abi	68	68	125	194	194	194	194	194	288	288	288	256	277	277	208
		Atl	332	374	453	447	511	519	610	686	722	660	662	659	552	550	553
		Nob	294	294	342	366	402	402	402	395	395	405	401	459	397	467	387
		Pol	265	284	368	394	454	454	478	475	491	498	492	548	583	606	456
		Avg	240	255	322	350	390	392	421	438	474	463	461	481	452	475	401
	Γ = 3	Abi	68	68	125	126	126	126	126	126	220	220	220	220	220	220	158
		Atl	332	332	411	447	411	447	459	560	560	560	560	574	580	554	485
		Nob	225	225	260	282	320	320	320	312	320	320	337	420	419	456	324
		Pol	189	244	294	308	369	369	412	437	481	472	472	472	517	585	402
		Avg	204	217	273	291	307	316	329	359	395	393	397	422	434	454	342
Protection Values	Γ = 0	Abi	1	0	1	2	1	1	3	2	0	1	0	1	0	0	1
		Atl	75	72	75	1	1	0	2	0	0	0	1	0	0	0	16
		Nob	59	49	1	1	0	0	0	0	0	0	0	0	0	0	8
		Pol	96	86	23	2	3	0	0	1	0	0	1	1	0	0	15
		Avg	58	52	25	2	1	0	1	1	0	0	1	1	0	0	10
	Γ = 1	Abi	98	98	98	97	99	99	99	94	89	94	87	71	62	80	90
		Atl	91	84	91	90	81	68	54	63	39	42	28	42	50	31	61
		Nob	88	84	81	82	75	85	60	56	42	52	54	57	49	61	66
		Pol	90	81	75	72	68	68	59	66	69	61	62	59	63	57	68
		Avg	92	87	86	85	81	80	68	70	60	62	58	58	56	57	71
	Γ = 2	Abi	100	100	100	100	100	100	100	100	98	98	98	99	100	100	100
		Atl	99	98	99	100	99	99	99	96	96	100	98	98	98	99	99
		Nob	100	100	99	99	99	99	99	99	100	97	99	100	98	99	99
		Pol	99	99	97	95	100	100	95	98	95	96	99	98	94	96	97
		Avg	100	99	99	99	100	100	98	98	97	98	99	99	98	99	99
	Γ = 3	Abi	100	100	100	100	100	100	100	100	100	100	100	100	99	99	100
		Atl	100	100	100	100	100	100	100	100	100	100	100	100	99	100	100
		Nob	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
		Pol	100	100	100	100	100	100	100	100	99	100	100	100	100	99	100
		Avg	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
Computing Times	Γ = 0	Abi	6	14	12	37	29	33	13	34	42	45	49	56	62	67	36
		Atl	9	9	11	36	59	67	99	207	267	334	350	355	464	464	195
		Nob	8	10	8	8	8	9	44	128	137	211	206	465	486	614	167
		Pol	6	7	16	12	27	51	125	97	110	116	90	97	94	119	69
		Avg	7	10	12	23	31	40	70	116	139	177	174	244	276	316	117
	Γ = 1	Abi	6	6	6	8	7	8	8	9	7	7	9	41	38	36	14
		Atl	11	9	10	10	32	17	127	180	522	2059	3180	3484	623	1231	821
		Nob	8	9	9	9	8	11	13	45	310	309	611	613	1518	613	292
		Pol	6	6	12	30	34	62	68	213	308	551	569	2798	1891	1973	609
		Avg	8	7	9	14	20	24	54	112	287	732	1093	1734	1017	963	434
	Γ = 2	Abi	7	11	9	15	10	11	11	11	31	33	31	911	912	910	208
		Atl	10	9	10	10	29	18	1664	3098	3445	2137	3042	3361	622	613	1291
		Nob	8	8	8	308	308	309	309	308	309	311	308	308	308	308	244
		Pol	6	6	6	367	307	307	2114	3320	3318	3357	3600	2716	3466	3320	1872
		Avg	8	9	8	175	164	161	1024	1684	1776	1459	1745	1824	1327	1288	904
	Γ = 3	Abi	6	6	6	306	307	307	307	308	612	607	607	607	1808	1209	500
		Atl	9	9	10	10	10	15	15	21	1222	1521	1519	1819	2136	1820	724
		Nob	8	8	8	8	8	8	307	308	308	308	308	612	1214	1211	330
		Pol	6	6	308	607	2173	1528	2020	3019	2819	3316	3314	3315	3389	2717	2039
		Avg	7	7	83	233	625	465	662	914	1240	1438	1437	1588	2137	1739	898

Let us first consider the exact Γ-robust approach for $\Gamma \in \{1, 2, 3\}$, as reported in Table 2. We remark that the empirical protection level is induced by the feasibility of a solution and not by its optimality. Therefore, it is reasonable to measure the former for all the solutions provided by the method, regardless of them being optimal or not. Quite interestingly, we observe that, although the empirical protection level for $\Gamma = 1$ is not very high (being equal to, on average, 82%), it already reaches a value of, on average, 100% for $\Gamma = 2$.

Similar observations can be drawn with both of our heuristics, as reported in Table 4 and in Table 7. For $\Gamma = 0$ (the case with average data), the empirical protection level is, as expected, very small (3% on average for the two-phase heuristic and 10% for the adaptive one). For $\Gamma = 1$, it is still not very high for both methods, being close to 71% on average for both. Differently, for $\Gamma = 2, 3$ we obtain solutions with very few violations and a higher empirical protection level equal to, respectively, 99% and 100% on average (again for both heuristics).

5.5 Recommendations

As a consequence of the results that we observed in our experiments, we would advise to resort to the exact MILP formulation for the nominal case of VNE only with no more than $|R| = 20$ requests and to the exact Γ-robust formulation only for $\Gamma = 1$ and with up to $|R| = 10$ requests. For all the other cases, we would suggest the adoption of our two-phase heuristic (which, among the two proposed algorithms, is definitely the faster one) with parameters $z_L = |V^0|, z_M = 2$, and $z_H = 1$. In case substantial differences in solution quality can be observed by experimenting with other parameter values (such differences could be substantial, as illustrated in Table 6) and a fine parameter tuning cannot be carried out in a preprocessing step, we advise the adoption of the computationally more demanding but, without a good guess on a suitable parameter choice, more stable, adaptive heuristic.

As to the choice of Γ, when aiming for a very high, i.e., $> 95\%$, protection level, we would advise, based on our experiments, to select $\Gamma = 2$ for the node capacity constraints, while (possibly) letting $\Gamma = 0$ for the link capacity ones.

5.6 Results on larger instances

To better assess how our algorithms scale on instances of larger size, we conclude the section by reporting on a set of experiments carried out on 8 physical networks taken from the Internet Topology Zoo database [24], `Fatman`, `Digex`, `Cernet`, `Bellsouth`, `Intellifiber`, `RedBestel`, `Deltacom`, and `Cogentco`, with, resp., $|V^0| = 20$, 30, 40, 60, 72, 83, 112, and 199. Due to their size (larger than those used in the previous experiments), we consider up to $|R| = 50$ VN requests.

We experiment with our two-phase heuristic, adopting the parameter setting $z_L = |V^0|, z_M = 2$, $z_H = 1$ which we have found to perform better in the previous experiments, with $\Gamma = 1$ in phase one and $\Gamma = 0$ in phase two. The results are illustrated in Table 8. Note that, in the table, the computing time accounts for the total time spent in the two phases, neglecting the preprocessing time invested to compute the all pairs shortest paths which are needed for the distance-bounding constraints in phase one.

Overall, the table shows that our two-phase algorithm can solve reasonably well VNE instances with large physical networks (`Cogentco` contains $|V^0| = 199$ nodes) with up to $|R| = 40$ simultaneous VN requests, while the method starts to fail for $|R| = 45, 50$ due to the solver failing to find a nonzero solution to the phase one subproblem in the time limit.

The table also illustrates an interesting phenomenon. Indeed, it clearly shows that, although both subproblems get harder, as one would expect, when $|R|$ increases, the phase one subproblem gets substantially easier when the size of the physical network $|V^0|$ grows. This is, most likely, a feature of the underlying multi-knapsack structure, due to which the introduction of more physical nodes only makes node capacity a more abundant resource, without complicating the structure of the problem too much. Interestingly, the situation is reversed for the phase two subproblem, which gets harder for physical networks of larger size. This is, possibly, a consequence of the network topologies still playing a large role in it, so that, having a physical network of increased size does not directly translate into a problem which is easier to solve.

We remark that these two opposite behaviors are, quite interestingly, somewhat complementary in that, by increasing the value of $|V^0|$, while the phase one subproblem gets harder, the phase two subproblem gets easier. Overall, we end up with a situation where, if one of the two subproblems is not solved to optimality, then the other one (in most of the cases) is, thus obtaining an, overall, still effective algorithm.

6 Concluding remarks

Based on a chance-constrained formulation for the Virtual Network Embedding problem where node and traffic demands of the virtual networks are assumed to be random variables, we have proposed an exact Γ-robust Mixed-Integer Linear Programming (MILP) formulation which allows to find solutions with large profits that are guaranteed to be feasible with a high probability. Based on this formulation, which is suitable to solve small size instances in a reasonable amount of computing time but which scales poorly for larger networks, we have introduced two MILP-based Γ-robust heuristics: a two-phase heuristic and an adaptive one.

Computational experiments indicate that, while the exact approaches become computationally challenging for instances with an increasing number of virtual network requests, both heuristics provide high quality solutions even for larger problems. We advise to adopt the first heuristic for the case where its input parameters can be determined beforehand whereas, if this is not the

Table 8: Results on 8 Internet Topology Zoo instances of increasing size $|V^0|$. Entries are rounded to the nearest integer.

	$\|V^0\|$	$\|R\|$	8	9	10	12	14	16	18	20	24	28	32	35	40	45	50	**Avg**
Objective Fct.	20	Fatman	328	302	331	315	341	450	417	429	370	413	523	468	434	441	391	**397**
	30	Digex	405	409	439	422	422	424	487	476	450	445	422	235	284	305	0	**375**
	40	Cernet	396	477	477	664	679	820	704	692	724	811	724	911	682	0	0	**584**
	50	Bellsouth	366	379	478	459	497	544	582	734	571	763	827	789	864	758	0	**574**
	72	Intellifiber	385	428	442	441	538	493	496	579	583	620	524	576	599	0	290	**466**
	83	RedBestel	280	379	349	380	407	416	443	458	505	543	565	522	540	0	509	**420**
	112	Deltacom	343	408	472	541	687	547	619	632	806	803	695	617	833	996	0	**600**
	199	Cogentco	365	459	496	534	631	566	684	740	702	778	793	779	837	0	0	**558**
		Avg	**359**	**405**	**436**	**470**	**525**	**533**	**554**	**593**	**589**	**647**	**634**	**612**	**634**	**313**	**149**	**497**
Prot. Level	20	Fatman	99	100	98	98	95	92	98	96	91	97	94	96	94	90	97	**96**
	30	Digex	100	100	100	100	99	98	93	97	93	92	100	100	100	100	100	**98**
	40	Cernet	97	100	100	100	100	97	92	90	96	99	94	90	92	100	100	**96**
	50	Bellsouth	100	99	100	99	97	100	98	96	95	88	91	95	92	98	100	**97**
	72	Intellifiber	100	99	99	99	99	99	99	93	98	96	84	93	98	100	100	**97**
	83	RedBestel	100	100	99	98	99	99	100	97	97	95	93	97	85	100	99	**97**
	112	Deltacom	98	99	99	99	98	99	99	98	97	98	97	95	95	95	93	**97**
	199	Cogentco	100	99	100	98	100	100	100	100	99	99	99	98	95	99	97	**99**
		Avg	**99**	**100**	**99**	**99**	**98**	**98**	**97**	**96**	**96**	**96**	**94**	**96**	**94**	**98**	**98**	**97**
O.Gap (Ph.I)	20	Fatman	0	0	0	0	4	3	10	15	23	50	44	48	78	78	78	**29**
	30	Digex	0	0	0	0	0	0	3	6	19	64	184	821	639	472	∞	**158**
	40	Cernet	0	0	0	0	0	0	0	12	17	38	53	50	68	∞	∞	**18**
	50	Bellsouth	0	0	0	0	0	0	0	0	0	0	5	6	8	71	∞	**6**
	72	Intellifiber	0	0	0	0	0	0	0	0	0	0	0	8	18	∞	684	**51**
	83	RedBestel	0	0	0	0	0	0	0	0	0	0	0	0	0	∞	78	**6**
	112	Deltacom	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	**0**
	199	Cogentco	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
		Avg	**0**	**0**	**0**	**0**	**1**	**0**	**2**	**4**	**7**	**19**	**36**	**117**	**101**	**124**	**168**	**33**
O.Gap (Ph.II)	20	Fatman	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	30	Digex	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	40	Cernet	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	50	Bellsouth	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	**0**
	72	Intellifiber	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	**1**
	83	RedBestel	0	0	0	0	0	0	0	0	0	6	0	11	2	0	0	**1**
	112	Deltacom	0	0	0	0	0	0	0	0	0	8	3	0	10	5	∞	**2**
	199	Cogentco	0	0	0	0	0	5	0	0	10	8	7	11	11	∞	∞	**4**
		Avg	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**	**3**	**2**	**3**	**3**	**1**	**0**	**1**
Comput. Time	20	Fatman	0	1	10	300	300	301	301	300	301	301	301	301	301	2	2	**201**
	30	Digex	2	6	3	9	11	30	328	338	317	330	307	300	300	2	0	**152**
	40	Cernet	2	2	2	2	6	9	72	315	318	327	321	326	327	299	299	**175**
	50	Bellsouth	1	1	1	3	5	4	7	9	140	256	373	331	372	315	299	**141**
	72	Intellifiber	8	9	10	28	21	44	107	66	124	265	600	458	404	0	3	**143**
	83	RedBestel	6	6	20	12	33	56	86	114	236	333	181	600	600	0	202	**166**
	112	Deltacom	8	15	18	42	78	118	115	96	163	309	312	295	359	553	600	**205**
	199	Cogentco	11	84	51	78	160	304	84	164	307	304	310	311	321	342	313	**210**
		Avg	**5**	**16**	**14**	**59**	**77**	**108**	**137**	**175**	**238**	**303**	**338**	**365**	**373**	**189**	**215**	**174**

case, we suggest to employ the adaptive heuristic, which provides competitive solutions, albeit at the cost of a larger investment in computing time. Experimenting with different values of Γ, so to establish a trade-off between the objective function value and the probability of being feasible for all realizations of the uncertain data, we have observed that, in our setting, $\Gamma = 2$ provides the most favorable option, yielding at the same time solutions with a high objective function value which are, empirically, feasible with a very high probability.

Acknowledgements This work is supported by the

This work is supported by the BMWi grant 03ET7528B.

German Federal Ministry of Education and Research (BMBF grant 05M13PAA, joint project 05M2013 - VINO: Virtual Network Optimization).

References

1. S. Coniglio, A. Koster, and M. Tieves, “Virtual network embedding under uncertainty: Exact and heuristic approaches,” in *Design of Reliable Communication Networks (DRCN), 2015 11th International Conference on the*, pp. 1–8, 2015.
2. N. Chowdhury and R. Boutaba, “A survey of network virtualization,” *Computer Networks*, vol. 54, no. 5, pp. 862–876, 2010.
3. A. Fischer, J. Botero, M. Beck, H. D. Meer, and X. Hesselbach, “Virtual network embedding: A survey,” *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 4, 2013.
4. N. Chowdhury and R. Boutaba, “Network virtualization: state of the art and research challenges,” *Communications Magazine, IEEE*, vol. 47, pp. 20–26, July 2009.
5. M. Yu, Y. Yi, J. Rexford, and M. Chiang, “Rethinking virtual network embedding: substrate support for path splitting and migration,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.
6. E. Amaldi, S. Coniglio, A. Koster, and M. Tieves, “On the computational complexity of the virtual network embedding problem,” *Electronic Notes Discrete Mathematics*, 2015. To appear.
7. Y. Zhu and M. Ammar, “Algorithms for assigning substrate network resources to virtual network components,” in *INFOCOM 2006, IEEE*, pp. 1–12, IEEE, 2006.
8. M. Chowdhury, M. R. Rahman, and R. Boutaba, “Vineyard: Virtual network embedding algorithms with coordinated node and link mapping,” *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 1, pp. 206–219, 2012.
9. A. Jarray and A. Karmouch, “Decomposition approaches for virtual network embedding with one-shot node and link mapping,” *Networking, IEEE/ACM Transactions on*, vol. 23, pp. 1012–1025, June 2015.
10. X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, “Virtual network embedding through topology-aware node ranking,” *SIGCOMM Computer Communication Reviews*, vol. 41, pp. 38–47, Apr. 2011.
11. Z. Zhang, S. Su, J. Zhang, K. Shuang, and P. Xu, “Energy aware virtual network embedding with dynamic demands: Online and offline,” *Computer Networks*, vol. 93, pp. 448–459, 2015.
12. I. Houidi, W. Louati, W. Ben-Ameur, and D. Zeghlache, “Virtual network provisioning across multiple substrate networks,” *Computer Networks*, vol. 55, no. 4, pp. 1011–1023, 2011.
13. S. Coniglio, B. Grimm, A. Koster, M. Tieves, and A. Werner, “Optimal offline virtual network embedding with rent-at-bulk aspects,” *arXiv preprint*, 2015. arXiv:1501.07887.
14. J. F. Botero, X. Hesselbach, M. Duelli, D. Schlosser, A. Fischer, and H. De Meer, “Energy efficient virtual network embedding,” *IEEE Commununication Letters*, vol. 16, no. 5, pp. 756–759, 2012.
15. I. Houidi, W. Louati, and D. Zeghlache, “Exact multi-objective virtual network embedding in cloud environments,” *The Computer Journal*, vol. 58, no. 3, pp. 403–415, 2015.
16. M. Mangili, F. Martignon, and A. Capone, “Stochastic planning for content delivery: Unveiling the benefits of network functions virtualization,” in *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pp. 344–349, Oct 2014.
17. J. Elias, F. Martignon, S. Paris, and J. Wang, “Optimization models for congestion mitigation in virtual networks,” in *Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, pp. 471–476, Oct 2014.
18. G. Sun, H. Yu, L. Li, V. Anand, Y. Cai, and H. Di, “Exploring online virtual networks mapping with stochastic bandwidth demand in multi-datacenter,” *Photonic Network Communications*, vol. 23, no. 2, pp. 109–122, 2012.
19. H. Kellerer, U. Pferschy, and D. Pisinger, “Introduction to NP-Completeness of Knapsack Problems,” in *Knapsack Problems*, pp. 483–493, Springer Berlin Heidelberg, 2004.
20. A. Koster, M. Kutschka, and C. Raack, “Robust network design: formulations, valid inequalities, and computations,” *Networks*, vol. 61, no. 2, pp. 128–149, 2013.
21. D. Bertsimas and M. Sim, “Robust discrete optimization and network flows,” in *Math. Prog.*, vol. 98, pp. 49 – 71, 2003.

22. D. Bertsimas and M. Sim, "The price of robustness," *Operations Research*, vol. 52, no. 1, pp. 35 – 53, 2004.
23. "SNDlib." http://sndlib.zib.de.
24. S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *Selected Areas in Communications, IEEE Journal on*, vol. 29, pp. 1765–1775, October 2011.

Stefano Coniglio is a Lecturer in Operational Research at the University of Southampton. He received his Ph.D. in Information Technology in 2011 from Politecnico di Milano, Italy. His research focuses on mathematical programming and combinatorial optimization.

Arie M.C.A. Koster is a Full Professor of Discrete Optimization at RWTH Aachen University, Germany. He received his Ph.D. degree in Mathematics of Operations Research (1999) from Maastricht University, The Netherlands. His research interests are in the development and application of discrete optimization methods, including algorithmic graph theory and robust optimization.

Martin Tieves is a Ph.D. student in Mathematics at RWTH Aachen University, Germany. He holds a Master of Science in Mathematics (minor: Business Administration) from the same university (2011). He is interested in the optimization of robust and dynamic communication networks.