**Submitted to *INFORMS Journal on Computing***
**manuscript (Please, provide the manuscript number!)**



# Hub Location with Protection under Inter-Hub Link Failures

Víctor Blanco
Institute of Mathematics (IMAG), Universidad de Granada, and
Dpt. Quant. Methods for Economics & Business, Universidad de Granada
vblanco@ugr.es

Elena Fernández
Dpt. Statistics & Operations Research, Universidad de Cádiz
elena.fernandez@uca.es

Yolanda Hinojosa
Dpt. Applied Economics I, Universidad de Sevilla, and Institute of Mathematics (IMUS), Universidad de Sevilla
yhinojos@us.es

This paper introduces the Hub Location Problem under Inter-Hub Link Failures, a hub location problem, in which activated inter-hub links may fail with a given probability. Two different optimization models are studied, which construct hub backbone networks protected under inter-hub link disruptions by imposing that for each commodity an additional routing path exists besides its original routing path. Both models consider the minimization of the fixed costs of the activated hubs and inter-hub links plus the expected value of the routing costs of the original and alternative paths. The first model builds explicitly the alternative routing paths, whereas the second model guarantees that for each commodity at least one alternative path exists using a large set of connectivity constraints, although the alternative paths are not built explicitly. The results of extensive computational testing allow to analyze the performance of the two proposed models and to evaluate the extra cost required to design a robust backbone network under inter-hub link failures. The obtained results support the validity of the proposal.

*Key words*: Hub Location; Integer Programming; Robust Network Design; Disruptions

## 1. Introduction

Hub location (HL) lies in the intersection of location analysis and network design, and produces challenging optimization problems with multiple applications, mostly in the fields of distribution and logistics (parcel delivery, air transportation, etc.) and telecommunications (Farahani et al. 2013). The increasing attention they have received in the last decades is thus not surprising (see e.g. Campbell and O'Kelly 2012, Erdoğan et al. 2022, Oliveira et al. 2022, Zheng et al. 2022). The interested reader is referred to the book chapter by Contreras and O'Kelly (2019) for a recent survey on HL, including terminology, modeling assumptions, etc.

One of the current trends in HL is the search of models suitable for dealing with different sources of uncertainty (Alumur et al. 2012). Whereas some models in the literature consider uncertainty in demand (Contreras et al. 2011, Zetina et al. 2017), other models are concerned with the robustness of solution hub networks, by associating uncertainty with the possibility (probability) of disruption of the elements involved in the solution networks and looking for solutions that are *robust* under disruptions.

Most hub networks are sensitive to failures in their links, being the impact on users particularly harmful in some cases. This is precisely the focus of this work, where we introduce the HL Problem under Inter-Hub Link Failures (HLPIHLF), a HL problem, in which activated inter-hub links may fail with a given probability. Our study can be very useful in typical HL applications, in which the total failure of a hub is highly unlikely, whereas partial failures occur only affecting some of the links incident with the hubs (certain air connections, train lines, etc.) We point out that by protecting inter-hub links under failure, we also partially protect hub nodes under failures. Examples of potential applications of the models that we study include the management of airlines and airport industries (Campbell et al. 2005), in which breakdowns in certain flight connections may occur, and passengers are directly affected, as well as rapid delivery packing systems (Çetiner et al. 2010), where users pay for fast services and failures in the hub network may cause large delays.

For dealing with the HLPIHLF we propose two alternative models, which guarantee that solution networks are protected under disruption of inter-hub links, in the sense that for each origin/destination demand pair (*commodity*) at least one alternative (backup) routing path exists. Both models can be seen as two-stage stochastic programming models, in which the *a priori* solution is determined by the strategic decisions associated with the selection of activated hub nodes and inter-hub links, together with an *original plan* given by a set of feasible routing paths, one for each commodity, containing exactly one inter-hub arc (directed link). The recourse action determines a *backup plan*, given by a set of alternative routing paths for the commodities, which can be used in case the inter-hub link of the original plan fails.

The main difference between the two models is how backup paths are enforced. The first model imposes that the alternative routing path of each commodity contains exactly one inter-hub arc (as in the original plan) and builds it explicitly. The second model is more flexible, in the sense that it allows for arbitrarily large sequences of inter-hub arcs to be used in the alternative routing paths, although such paths are not built explicitly. This is achieved with a set of exponentially many (on the number of nodes of the network) constraints, by imposing that the cut-set of the backbone network contains at least $\lambda$ links, for a given integer value of $\lambda \geq 2$. We study some properties of both models and propose a Mixed Integer Linear Programming (MILP) formulation for each case. For the second model, since it has exponentially many constraints, we also propose a branch-and-cut solution algorithm.

In both cases we consider fixed costs for both activated hubs and activated inter-hub links. Thus, we do not fix the number of inter-hub links to activate, allowing for incomplete backbone networks. As usual in HL, the routing costs apply a discount factor $\alpha$ to inter-hub arcs.

Extensive computational experiments have been carried out on a large set of benchmark instances based on the well-known CAB (O'Kelly 1987), AP (Ernst and Krishnamoorthy 1996), and TR (Tan and Kara 2007) datasets, for varying settings of the failure probabilities and other cost parameters. The obtained results are summarized and analyzed, comparing the computational performance of each of the models and the effect of the different parameters. Managerial insights are derived from the analysis of the characteristics of the solutions produced by each of the models and their comparison. In particular, we analyze the distribution of the costs among the different elements considered (routing costs and fixed costs for hubs and links), the number of activated hub and links, and the density of the obtained backbone networks. Finally, we have carried out an empirical analysis of the different models in terms of their efficiency and robustness. Specifically, for each of the proposed models, we analyze their *a posteriori* capability to re-route the commodities under multiple failure scenarios. The obtained results assess the validity of the proposal.

The remainder of this paper is structured as follows. In Section 2 we review the related literature and in Section 3 we introduce the notation that we will use and formally define the HLPIHLF from a general perspective. In that section we also specify and motivate the modeling assumptions that we make. Most of them are usual in classical models in the HL literature. This allows us to better focus on the novel aspects of our proposal. Section 4 is devoted to the first HLPIHLF model that we analyze, in which it is assumed that the alternative paths for the commodities contain exactly one inter-hub arc. We study some of its properties and propose a MILP formulation for it. The model in which we impose that the backbone network is $\lambda$-connected is detailed in Section 5 where we also present a MILP formulation for it. Section 6 describes the computational experiments we have carried out and summarizes the obtained results. Some managerial insights from the analysis of the structure of the solution networks produced by each model are also derived in this section. Finally, Section 7 describes the empirical analysis that has been performed considering multiple failure scenarios. The paper closes in Section 8 with some conclusions.

## 2. Related literature

Several works have studied models, in which it is assumed that activated hub nodes may fail (totally or partially) with a certain probability. An et al. (2015) propose a model, in which two backup hub nodes are determined for each commodity. Rostami et al. (2019) assume that a finite set of hub breakdown scenarios is known and provide a two-stage formulation for the single allocation HL problem with possible hub breakdown. Cui et al. (2010) provide a stochastic model to determine a subset of the activated hub nodes of given size through which each commodity can be routed, in such a way that if the cheapest route fails, the commodity can be routed through the second cheapest, and so on, or through an emergency facility. The authors provide a MILP formulation for the problem as well as an approximate Lagrangean relaxation scheme for its solution based on the ideas in Snyder and Daskin (2005) for the $p$-median problem. The planar version of this model is also analyzed there.

Kim and O'Kelly (2009) propose the reliable $p$-hub location problem (PHMR) and the $p$-hub mandatory dispersion (PHMD). In the PHMR the goal is to determine the location of $p$ nodes based on the level of reliability to maximize the flows sent through the selected hubs. The PHMD imposes a certain minimum separation between the $p$ selected hub nodes, and maximizes the reliability of the network. MILP formulations and heuristic approaches are provided for the problems, both in the single and the multiple-allocation framework. The reliability of hub backbone networks has been also studied in Zeng et al. (2010), Korani and Eydi (2021) and Li et al. (2022).

Parvaresh et al. (2013) consider the multiple allocation $p$-hub median problem under intentional disruptions, in which the goal is to identify the optimal strategy for the location of $p$ hub nodes by minimizing the expected transportation cost, taking into account the worst-case situation when an interdictor attacks the backbone network. A bilevel mixed integer formulation is provided as well as a simulated annealing heuristic for its solution.

The problem of designing robust networks under link failures has also been studied in the literature under different settings. Aneja et al. (2001) study the single-commodity maximum flow problem when link failures occur by means of the maximum residual flow problem, whose goal is to determine the maximal flow, in which the largest link flow is as small as possible. This problem is closely related to the network interdiction problem that consists of determining a certain number of links whose removal from the network minimizes the maximum amount of flow that one can send through the network (see e.g. Altner et al. 2010, Cormican et al. 1998, Royset and Wood 2007, Wood 1993). Ma et al. (2016) propose the Conditional Value-at-Risk Constrained Minimum Spanning $k$-Core Problem where the possibility of link disruptions is prevented when trying to construct minimum-cost subgraphs of a network with a minimum number of incident links at each node. Andreas and Smith (2008) study shortest path problems under link failures by imposing that the probability that all links that successfully operate in at least one path be greater than certain threshold value.

We are only aware of one work (Mohammadi et al. 2019), dealing with potential failure of inter-hub links, which is the specific focus of our work. Nevertheless, in contrast to Mohammadi et al. (2019), who consider a $p$-hub single allocation pattern, and complete hub networks with no set-up costs for inter-hub links, we adopt a multiple allocation policy on a non-complete backbone network where the number of hubs is not a fixed parameter, and assume fixed costs for inter-hub links. Furthermore, Mohammadi et al. (2019) apply meta-heuristics to solve the models whereas we solve them to proven optimality.

## 3. Notation and definition of the problem

Consider a graph $N = (V, E)$, where the node set $V = \{1, 2, \dots, n\}$ represents a given set of users and the edge set $E$ the existing undirected connections or links between pairs of users. We assume that $N$ is a complete graph and that $E$ contains loops, i.e. for all $i \in V$, edge $\{i, i\} \in E$. We further assume that potential locations for hubs are placed at the nodes of the graph and the set of potential locations coincides with $V$. For each potential location $k \in V$, we denote by $f_k$ the fixed cost for activating a hub at node $k$. Any pair of hub nodes can be connected by means of an *inter-hub* edge, provided that both end nodes $k$ and $l$ are activated as hub nodes as well.

The set $E$ will be referred to as the set of potential inter-hub edges or just as set of potential hub edges. Activated hub edges incur fixed costs; let $h_{kl} \geq 0$ be the fixed cost for activating hub edge $\{k, l\} \in E$. A set of activated hubs will be denoted by $H \subseteq V$, and a set of activated hub edges for $H$ by $E_H \subseteq E[H]$, where $E[H]$ is the set of edges with both end nodes in $H$. Note that the assumption that $N$ is a complete graph implies no loss of generality, since $(i)$ arbitrarily large fixed costs can be associated with nodes that are not potential hubs; and, $(ii)$ arbitrarily large activation costs can be associated with non-existing hub edges. Activating a hub edge allows to send flows through it in either direction. Let $A = \{(i, j) \cup (j, i) : \{i, j\} \in E,\, i, j \in V\}$ be the (directed) arc set. Arcs in the form $(i, i)$, $i \in V$, are allowed and will also be called *loops*. We will use $A_H \subseteq A$ to denote the set of inter-hub arcs induced by $E_H$.

Service demand is given by a set of commodities defined over pairs of users, indexed in a set $R$. Let $\mathcal{D} = \{(o_r, d_r, w_r) : r \in R\}$ denote the set of commodities, where the triplet $(o_r, d_r, w_r)$ indicates that an amount of flow $w_r \geq 0$ must be routed from origin $o_r \in V$ to destination $d_r \in V$. The origin/destination pair associated with a given commodity will also be referred to as its OD pair. Commodities must be routed via paths of the form $\pi = (o_r, k_1, \dots, k_s, d_r)$ with $k_i \in H$, $1 \leq i \leq s$. Similarly to most HL problems, a routing path $\pi = (o_r, k_1, \dots, k_s, d_r)$ is *feasible* if (1) it includes at least two hub nodes, i.e. $s \geq 2$ (being possible that $k_i = k_{i+1}$ for some $i = 1, \cdots, s-1$), and (2) the underlying edges of all traversed arcs other than the *access* and *delivery* arcs, $(o_r, k_1)$ and $(k_s, d_r)$, respectively, are activated inter-hub edges, i.e., $\{k_i, k_{i+1}\} \in E_H$, $1 \leq i \leq s-1$. Note that this implies that in any feasible path all intermediate nodes are activated as hubs as well, i.e. $k_i \in H$, $1 \leq i \leq s$. In the following, the set of feasible paths for a given commodity $r \in R$ will be denoted by $\Pi_{(H,E_H)}(r)$.

Routing flows through the arcs of a hub-and-spoke network incurs different types of costs. These costs, which may depend on the type of arc, account for transportation costs as well as for some additional collection/handling/distribution costs at the end nodes of the arcs. As usual in the literature, we assume that transportation costs of flows routed through inter-hub arcs are subjected to a discount factor $0 \leq \alpha \leq 1$. In this work we will denote by $c_{ij} \geq 0$ the unit routing cost through inter-hub arc $(i,j) \in A_H$, which includes discounted transportation costs and handling costs, and we will denote by $\bar{c}_{ij} \geq 0$ the unit routing cost for an access or a delivery arc $(i,j) \in A \setminus A_H$, which may also incorporate different discounted access or delivery costs.

With the above notation, the routing cost of commodity $r \in R$ through a feasible path $\pi = (o_r, k_1, \ldots, k_s, d_r) \in \Pi_{(H,E_H)}(r)$ is:

$$C_\pi^r = w_r \left( \overline{c}_{o_r k_1} + \sum_{i=1}^{s-1} c_{k_i k_{i+1}} + \overline{c}_{k_s d_r} \right), \tag{1}$$

where the first and last addends correspond to the access and delivery arcs, respectively, and the intermediate ones are the service costs through the backbone network $(H, E_H)$.

Broadly speaking, under the above assumptions, the goal of a HL problem is to decide the location of the hub nodes $H$ and to select a suitable subset of hub edges $E_H$, to *optimally* route the commodities through the backbone network $(H, E_H)$ induced by the activated hub nodes and hub edges, so as to minimize the sum of the overall fixed costs for activating hub nodes and hub edges, plus the commodities routing costs. With the above notation, this problem can be stated as:

$$\min_{H \subseteq V, E_H \subseteq E[H]} \sum_{k \in H} f_k + \sum_{e \in E_H} h_e + \sum_{r \in R} \min_{\pi \in \Pi_{(H,E_H)}(r)} C_\pi^r. \tag{HLP}$$

In the reminder of this paper we make the following assumptions:

- All our models assume multiple allocation of OD nodes to open hubs. With the multiple allocation policy, in case two commodities share the same origin, each of them is allowed to route its demand using a different access arc. Similarly, two commodities with the same destination may be routed using different delivery arcs. This strategy provides greater flexibility to hub backbone networks than the single allocation pattern, in which OD nodes are assigned to a single hub node such that all the demand originated at (or arriving to) such a node is routed initially (or finally) through the same access (or delivery) arc. The multiple allocation policy is assumed in several so-called *fundamental* HL models (see, e.g., Contreras and O'Kelly 2019), and is commonly applied by delivery companies that allocate origins to hub centers based on the destination of the deliveries, or by airlines that route passengers departing from the same city through different intermediate airports (hubs) based, for instance, on the continent of their final destination.

- Loops (*self-arcs*) of the form $(k,k)$ are considered as hub arcs. This is a slightly more general setting than in most HL problems studied in the literature, which only consider *proper* arcs $(k,l)$ with $k \neq l$ as potential hub arcs. In this work, if a loop $(k,k)$ is used in a routing path, it is required not only that $k$ is activated as a hub node, but also that the loop $\{k,k\}$ is activated as a hub edge as well. The rationale for this assumption is that loops allow one to model the different operations that are required when flows enter a hub network, as the preparation of the machinery, workforce, etc. As far as we know, loops have never been explicitly considered in the literature of HL, although loops are present in other types of network design problems (see, e.g., Wei et al. 2021, Huang and Tian 2017).

- Activated hub edges in $E_H$ may fail. We are given the failure probabilities for each potential inter-hub edge, $p_{kl}$, $\{k,l\} \in E$. When $k \neq l$ failure of edge $e$ arises not only when the edge $\{k,l\}$ can no longer be used, but also when, for any reason, the collection and distribution services at any end node of edge $\{k,l\}$ cannot be carried out. Thus $p_{kl}$ represents the probability that any of these events happen. In case edge $e$ is a loop, i.e., $e=\{k,k\}$, with $k \in H$, then $p_{kk}$ represents the probability that the handling process carried out when $k$ is used as the unique intermediate hub fails.

- In case hub edge $\{k,l\} \in E_H$ fails, then the inter-hub arcs $(k,l),(l,k) \in A_H$ can no longer be used for routing the commodities. In order to protect solution networks from failure we adopt a policy that does not alter the strategic decisions on the activated hubs and inter-hub edges and focuses solely on the operational decisions concerning the re-routing of affected commodities. Accordingly, we impose that, for each commodity, the backbone network $(H,E_H)$ contains a routing path and, in addition, a substitute path connecting $o_r$ and $d_r$, for each $r \in R$. Such a substitute path will be referred to as *backup* or *alternative* path whereas, the routing path used in case the edges do not fail will be called *original* or *initial* path. The allocation pattern may be different in the original and backup paths, so they may use different access/delivery arcs.

- The original routing path of each commodity $r=(o_r,d_r,w_r)$ contains exactly one inter-hub arc, which can be a loop. Most HL problems studied in the literature (see, e.g., Contreras and O'Kelly 2019), which do not consider loops explicitly, restrict the set of potential paths for routing the commodities to those using at most one hub arc. When loops are also considered as potential hub arcs as we do, potential routing paths are of the form $(o_r,k,m,d_r)$, where $k$ and $m$ are activated hubs, and $(k,m)$ is an inter-hub arc, which reduces to a loop when $k=m$.

Thus, for $H$ and $E_H$ given, the set of potential original paths for routing commodity $r \in R$ is given by $\Pi_{(H,E_H)}(r) = \{\pi = (o_r, k, l, d_r) : k, l \in H, \{k,l\} \in E_H\}$. In such a case the routing cost of commodity $r$ through path $\pi = (o_r, k, l, d_r)$ reduces to

$$C^r_{kl} := C^r_\pi = w_r(\overline{c}_{o_r k} + c_{kl} + \overline{c}_{l d_r}).$$

- For each edge $e = \{k, l\} \in E$, let $X_{kl}$ denote the random variable modeling whether $e$ fails. We assume that the random variable $X_{kl}$ follows a Bernoulli distribution with probability $p_{kl}$, for each $\{k, l\} \in E$. Observe that when edges may fail with a given probability, the costs of feasible routing paths are also random variables, which will be denoted by $\mathcal{C}_r$, $r \in R$. Furthermore, the probability distribution of $\mathcal{C}_r$, $r \in R$, is dictated by the failure probability distribution of the involved inter-hub edges. In particular, when $\pi_0 = (o_r, k, l, d_r)$ is the original routing path of a given commodity $r \in R$, the expected routing cost of commodity $r$ can be calculated as:

$$E[\mathcal{C}_r|\pi_0] = (1 - p_{kl})C^r_{kl} + p_{kl}C^r_{\overline{\pi}_0}. \tag{2}$$

where $\overline{\pi}_0 \in \Pi_{(H,E_H)}(r)\backslash\{\pi_0\}$ is the backup path in case $\{k, l\}$ fails.

- The objective is to minimize the sum of the fixed costs of the activated hubs and inter-hub edges, plus the expected routing costs.

In the following we deal with the problem of finding hub backbone networks protected against inter-hub edge failures under the above assumptions.

That is, the HLPIHLF can be stated as:

$$\min_{H \subseteq V, E_H \subseteq E[H]} \sum_{k \in H} f_k + \sum_{e \in E_H} h_e + \sum_{r \in R} \min_{\pi_0 \in \Pi_{(H,E_H)}(r)} E[\mathcal{C}_r|\pi_0]. \tag{HLPIHLF}$$

Indeed, multiple choices fall within the above generic framework which differ from each other in how the alternative routing paths are obtained. In the following sections we propose two models for determining such backup paths, based on different assumptions, and provide mathematical programming formulations for each of them.

## 4. HLPIHLF with single inter-hub arc backup paths

The HLPIHLF we address in this section enforces that the alternative paths for routing the commodities have the same structure as the original ones. That is, we assume that backup paths contain exactly one inter-hub arc (possibly a loop). This avoids having to use many transshipment points in case of failure. This model will be referred to as (HLPIHLF-1BP).

Given a commodity $r \in R$, the backbone network $(H, E_H)$ and the original routing path $\pi_0 = (o_r, k, l, d_r)$, we assume that the backup path is in the form $\overline{\pi}_0 = (o_r, \bar{k}, \bar{l}, d_r)$, with $\{k, l\} \neq \{\bar{k}, \bar{l}\}$.

Thus, the expected routing cost of commodity $r$ is:

$$E[\mathcal{C}_r|\pi_0] = (1-p_{kl})C^r_{kl} + p_{kl}C^r_{\bar{k}\bar{l}}. \tag{3}$$

Figure 1 illustrates the different situations that may arise in case a hub edge fails. Figure 1a shows a backbone network with four hub nodes ($k$, $l$, $m$ and $q$) and five hub edges, two of them corresponding to loops, $\{m,m\}$ and $\{l,l\}$, and the other ones corresponding to edges $\{k,q\}$, $\{k,l\}$ and $\{q,m\}$. The figure also depicts the origin ($o_r$) and destination ($d_r$) of a given commodity $r \in R$ and a possible path for this commodity through hub arc $(k,l)$. We assume that this is the *original path* for the commodity $r \in R$. Access/distribution arcs are depicted as dashed lines and hub edges as solid lines.

Figures 1b and 1c show different backup paths with a single inter-hub arc for commodity $r \in R$ in case the inter-hub link $\{k,l\}$ of the original path fails. In Figure 1b, the backup path uses the inter-hub arc $(q,m)$ to re-route the commodity, whereas in Figure 1c the backup path uses the loop arc $(l,l)$.

Next we develop a mathematical programming formulation for the above problem, first introducing the decision variables associated with the design decisions on the elements of the network that are activated, hubs nodes and hub edges:

$$z_k = \begin{cases} 1 \text{ if a hub is opened at the potential hub node } k, \\ 0 \text{ otherwise} \end{cases} \quad \text{for } k \in V,$$

$$y_{kl} = \begin{cases} 1 \text{ if hub edge } \{k,l\} \text{ is activated,} \\ 0 \text{ otherwise} \end{cases} \quad \text{for } \{k,l\} \in E.$$

The formulation uses two additional sets of variables, which represent the original and alternative routing path for each commodity, respectively. In particular, for $r \in R$ and $(k,l) \in A$:

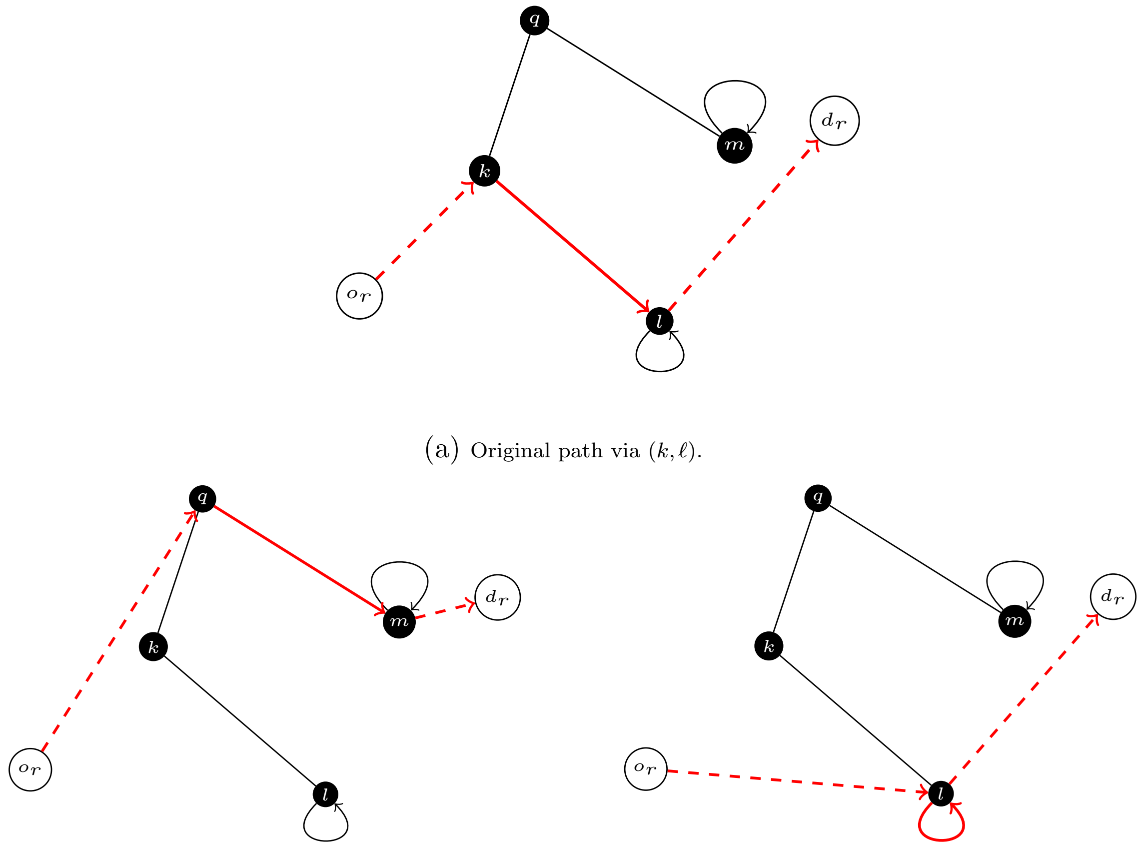
$$x^r_{kl} = \begin{cases} 1 \text{ if the original routing path for commodity } r \text{ is } (o_r, k, l, d_r), \\ 0 \text{ otherwise,} \end{cases}$$

$$\bar{x}^r_{kl} = \begin{cases} 1 \text{ if the alternative path for commodity } r \in R \text{ is } (o_r, k, l, d_r), \\ 0 \text{ otherwise.} \end{cases}$$

With these sets of decision variables the expected routing cost of commodity $r \in R$ can be expressed as:

$$\sum_{(k,l)\in A} x^r_{kl}\Big[C^r_{kl}(1-p_{kl}) + p_{kl} \sum_{(k',l')\in A\setminus\{(k,l)\cup(l,k)\}} C^r_{k'l'}\bar{x}^r_{k'l'}\Big], \tag{4}$$

where the two addends in each term of the above expression correspond to the expected routing cost of the original and backup plan of commodity $r$, respectively. Observe that both terms only apply if, in the original plan, the commodity is routed through the inter-hub arc corresponding




(a) Original path via $(k,\ell)$.

(b) Backup path via $(q,m)$.

(c) Backup path via $(l,l)$.

**Figure 1** **A network with four nodes and one commodity $r=(o_r,d_r,w_r)$: Original path, $\pi_0=(o_r,k,l,d_r)$, and different backup paths in case edge $\{k,l\}$ fails.**

to the term. In particular, the first addend gives the overall routing cost for commodity $r$ in case the arc of the backbone network used for routing $r$ in the original plan does not fail (multiplied by the probability of not failing). The second term computes the cost of the alternative routing path, multiplied by the probability of failure of the inter-hub arc of the original plan. Observe that in case $(k,l)$ is the arc used initially by commodity $r\in R$ and $(k',l')$ is the backup arc for $r$, one obtains the cost $(1-p_{kl})C^r_{kl}+p_{kl}C^r_{k'l'}$.

Rearranging the terms and after some algebra we get:

$$\sum_{(k,l)\in A} x^r_{kl}p_{kl} \sum_{(k',l')\in A\setminus\{(k,l)\cup(l,k)\}} C^r_{k'l'}\bar{x}^r_{k'\,l'} = \sum_{(k,l)\in A} \sum_{(k',l')\in A\setminus\{(k,l)\cup(l,k)\}} C^r_{k'l'}\bar{x}^r_{k'l'}p_{kl}x^r_{k\,l}$$
$$= \sum_{(k,l)\in A} C^r_{kl}\bar{x}^r_{kl} \sum_{(k',l')\in A\setminus\{(k,l)\cup(l,k)\}} p_{k'\,l'}x^r_{k'\,l'}.$$

Thus, one can rewrite the overall routing cost for commodity $r$ as:

$$\sum_{(k,l)\in A} C^r_{kl}\Big[(1-p_{kl})x^r_{kl} + \bar{x}^r_{kl}\sum_{(k',l')\in A\setminus\{(k,l)\cup(l,k)\}} p_{k'\,l'}x^r_{k'\,l'}\Big]. \quad (5)$$

Note that in the above expression, the term $\bar{x}^r_{kl}\sum_{(k',l')\in A\setminus\{(k,l)\cup(l,k)\}} p_{k'\,l'}x^r_{k'\,l'}$ indicates the probability of using inter-hub arc $(k,l)$ in the alternative path of commodity $r$. Thus, the impact of a given arc $(k,l)\in A$ in the routing cost of commodity $r$ is either 0 (if it is not used neither in the original nor the alternative path); $(1-p_{kl})C^r_{kl}$ if it is used in the original path; or $C^r_{kl}p_{k',l'}$ in case arc $(k,l)$ is used in the alternative path and arc $(k',l')$ in the original one.

The above decision variables and routing cost function lead to the following integer nonlinear programming formulation for (HLPIHLF-1BP):

$$\min \sum_{k\in V} f_k z_k + \sum_{\{k,l\}\in E} h_{kl}y_{kl} + \sum_{(k,l)\in A} C^r_{kl}\Big[(1-p_{kl})x^r_{kl} + \bar{x}^r_{kl}\sum_{(k',l')\in A\setminus\{(k,l)\cup(l,k)\}} p_{k'\,l'}x^r_{k'\,l'}\Big]$$

$$\text{s.t.} \sum_{(k,l)\in A} x^r_{kl} = 1 \qquad \forall r\in R \qquad (1.1)$$

$$\sum_{(k,l)\in A} \bar{x}^r_{kl} = 1 \qquad \forall r\in R \qquad (1.2)$$

$$x^r_{kl} + x^r_{lk} + \bar{x}^r_{kl} + \bar{x}^r_{lk} \le y_{kl} \qquad \forall r\in R, \{k,l\}\in E, k\neq l \qquad (1.3)$$

$$x^r_{kk} + \bar{x}^r_{kk} \le y_{kk} \qquad \forall r\in R, \{k,k\}\in E \qquad (1.4)$$

$$y_{kl} \le z_k \qquad \forall \{k,l\}\in E \qquad (1.5)$$

$$y_{kl} \le z_l \qquad \forall \{k,l\}\in E, k\neq l \qquad (1.6)$$

$$x^r_{kl}, \bar{x}^r_{kl} \in \{0,1\} \qquad \forall r\in R, (k,l)\in A \qquad (1.7)$$

$$z_k \in \{0,1\} \qquad \forall k\in V \qquad (1.8)$$

$$y_{kl} \in \{0,1\} \qquad \forall \{k,l\}\in E, \qquad (1.9)$$

where Constraints (1.1) and (1.2) enforce that each commodity uses exactly one inter-hub arc both in the original and the backup path. Constraints (1.3) and (1.4) impose that the original and the backup path do not coincide. These constraints also guarantee that any used inter-hub edge is activated. Constraints (1.5) and (1.6) ensure that the end nodes of activated inter-hub edges are activated as hub nodes. Finally, (1.7)–(1.9) are the domains of the decision variables.

To reinforce the relationship between the routing variables and the hub activation variables, we incorporate the following valid inequalities:

$$x^r_{kk} + \sum_{\substack{l\in V\\ l\neq k}} (x^r_{kl} + x^r_{lk}) \le z_k \qquad \forall\, r\in R,\, k\in V \qquad (1.10)$$

$$\bar{x}^r_{kk} + \sum_{\substack{l\in V\\ l\neq k}} (\bar{x}^r_{kl} + \bar{x}^r_{lk}) \le z_k \qquad \forall\, r\in R,\, k\in V. \qquad (1.11)$$

These well-known valid inequalities were introduced in Marín et al. (2006) and have now become standard in different HL problems (see, for instance, Cánovas et al. 2007, de Camargo et al. 2009, among others).

### 4.1. Linearization of the objective function

The reader may have observed the non-linearity of the objective function term corresponding to the expected routing cost. As we explain below this term can be suitably linearized by introducing a new auxiliary variable $P^r_{kl} \in \mathbb{R}_+$ associated with each commodity $r \in R$ and each arc $(k,l) \in A$.

Let $P^r_{kl} = \bar{x}^r_{kl} \sum_{(k',l') \in A \setminus \{(k,l) \cup (l,k)\}} p_{k'l'} x^r_{k'l'}$ denote the probability of using inter-hub arc $(k,l)$ in the alternative path of commodity $r$ that appears in the objective function term (5). Observe that by Constraints (1.3) and (1.4), $P^r_{kl}$ can be written as $P^r_{kl} = \bar{x}^r_{kl} \sum_{(k',l') \in A} p_{k'l'} x^r_{k'l'}$. Thus, because of the minimization criterion, the non-negativity of the routing costs, and Constraints (1.1), the value of $P^r_{kl}$ can be determined by the following set of constraints:

$$P^r_{kl} \geq \sum_{(k',l') \in A} p_{k'l'} x^r_{k'l'} + (\bar{x}^r_{kl} - 1), \qquad \forall r \in R, (k,l) \in A, \tag{1.12}$$

$$P^r_{kl} \geq 0, \qquad \forall r \in R, \ (k,l) \in A. \tag{1.13}$$

The above constraints linearize the expression that defines $P^r_{kr}$. Observe that in case $\bar{x}^r_{kl}$ takes value 1, since $P^r_{kl}$ is being minimized, its associated Constraint (1.12) becomes active and we will get that $P^r_{kl} = \sum_{(k',l') \in A} p_{k'l'} x^r_{k'l'}$. Otherwise, in case $\bar{x}^r_{kl}$ takes value 0, Constraints (1.1) and (1.13) imply that (1.12) is redundant. Therefore, by the minimization criterion, $P^r_{kl}$ will take the value 0.

Thus, by introducing Constraints (1.12) and (1.13) in our model, the objective function can be rewritten as:

$$\sum_{k \in V} f_k z_k + \sum_{\{k,l\} \in E} h_{kl} y_{kl} + \sum_{r \in R} \sum_{(k,l) \in A} C^r_{kl} \Big( (1 - p_{kl}) x^r_{kl} + P^r_{kl} \Big). \tag{7}$$

We can also incorporate the following valid inequalities to reinforce our formulation:

$$\sum_{(k,l) \in A} P^r_{kl} \leq \max_{\{k,l\} \in E} p_{kl}, \qquad \forall r \in R. \tag{1.14}$$

Therefore, we have the following MILP formulation for (HLPIHLF-1BP):

$$\begin{aligned} \min & \sum_{k \in V} f_k z_k + \sum_{\{k,l\} \in E} h_{kl} y_{kl} + \sum_{r \in R} \sum_{(k,l) \in A} C^r_{kl} \Big( (1 - p_{kl}) x^r_{kl} + P^r_{kl} \Big) \\ \text{s.t. } & (1.1) - (1.14). \end{aligned} \tag{HLPIHLF-1BP}$$

Below we state some simple optimality conditions that can be used to reduce the set of decision variables.

**Proposition 4.1** *There is an optimal solution to* (HLPIHLF-1BP) *such that* $x^r_{lk} = \bar{x}^r_{lk} = P^r_{lk} = 0$ *for all* $r \in R$, $(l,k) \in A$, *with* $C^r_{kl} \leq C^r_{lk}$.

*Proof.* The proof is straightforward. Indeed, the value of any solution with $x^r_{lk} = 1$ where $C^r_{kl} < C^r_{lk}$ will improve by changing the direction in which edge $\{k,l\}$ is traversed, i.e., by doing $x^r_{lk} = 0$, $x^r_{kl} = 1$. When $C^r_{kl} = C^r_{lk}$ the value of the new solution will not change.

The same argument can be applied for setting $\bar{x}^r_{lk} = 0$ and thus, $P^r_{lk} = 0$. □

Note that the above result allows to reduce to half the number of decision variables.

In practice, it is likely that there are few possible values for failure probabilities, and the edges of the network are clustered in groups such that, within each group, all edges have the same failure probability. We next analyze such a situation.

**Remark 4.1 (Clustered sets of edges)** *Let us assume that the edges of* $E$ *are clustered in* $K$ *groups* $E_1, \dots, E_K$ *such that all edges in* $E_s$ *have the same failure probability* $\rho_s \in [0,1]$, *for* $s = 1, \dots, K$. *Then, in the term of the objective function of* (HLPIHLF-1BP) *corresponding to the expected routing cost of the commodities, variables* $P^r_{kl}$ *can be substituted by a new set of variables as follows. For* $r \in R$, $(k,l) \in A_s$, *being* $A_s$ *the arc set induced by* $E_s$, $s = 1, \dots, K$, *let* $\xi^r_{kls}$ *be a binary variable that takes value one if and only if the original route of commodity* $r$ *uses some hub arc in the* $s$*-th cluster (with failure probability* $\rho_s$*) and in the backup route it uses hub arc* $(k,l)$.

*Then, the expected routing cost of commodity* $r \in R$ *can be rewritten as:*

$$\sum_{s=1}^{K} \Big[ (1-\rho_s) \sum_{(k,l) \in A_s} C^r_{kl} x^r_{kl} + \rho_s \sum_{(k,l) \in A} C^r_{kl} \xi^r_{kls} \Big]. \tag{8}$$

*Using similar arguments as for the linearization of variables* $P^r_{kl}$, *the values of the* $\xi^r_{kls}$ *variables can be determined by the following sets of constraints:*

$$\xi^r_{kls} \geq \sum_{(k',l') \in A_s} x^r_{k'l'} + (\bar{x}^r_{kl} - 1) \qquad \forall r \in R, (k,l) \in A, s = 1, \dots, K \tag{9}$$

$$\sum_{s=1}^{K} \xi^r_{kls} = \bar{x}^r_{kl} \qquad \forall r \in R, (k,l) \in A \tag{10}$$

$$\xi^r_{kls} \geq 0 \qquad \forall r \in R, (k,l) \in A, s = 1, \dots, K. \tag{11}$$

*The particular case of one single cluster (*$K = 1$*) where all edges have the same failure probability, i.e.,* $p_{kl} = \rho$ *for all* $\{k,l\} \in E$, *allows to further simplify the above formulation. Now the index* $s$ *can be dropped from variables* $\xi$ *and Constraints* (9)-(11) *are no longer needed, as Constraints* (10) *reduce to* $\xi^r_{kl} = \bar{x}^r_{kl}$, $(k,l) \in A$. *Then, the expected routing cost of commodity* $r \in R$ *simplifies to:*

$$\sum_{(k,l) \in A} C^r_{kl} \Big( (1-\rho) x^r_{kl} + \rho \overline{x}^r_{kl} \Big). \tag{12}$$

## 5. HLPIHLF with $\lambda$-connected backbone networks

In this section we introduce a different model for the HLPIHLF, that will be referred to as $\lambda$-connected HLPIHLF (HLPIHLF-$\lambda$). Again we make the assumption that the original routing paths contain exactly one inter-hub arc, although we follow a different modeling approach as for how to protect the backbone network $(H, E_H)$ against potential failures. On the one hand, we extend the set of alternative paths that can be used when the hub edge of some original path fails, and allow for any arbitrarily long chain of arcs connecting the OD pair of each commodity, provided that all its intermediate arcs are activated inter-hub arcs. On the other hand, we no longer make explicit the alternative routing paths for the commodities. Instead, we impose that the backbone network is $\lambda$-connected, in the sense that it must contain at least $\lambda$ routing paths connecting any pair of activated hubs $k$, $l \in H$ with $k \neq l$, where $\lambda \geq 2$ is a given integer parameter. This implies that if some hub arc of the original path fails, then the backbone network contains at least $\lambda - 1$ alternative paths connecting the activated hubs. Note that, this forces the backbone network to have at least $\lambda$ activated hub nodes. The particular case of HLPIHLF-$\lambda$ with $\lambda = 2$, extends the HLPIHLF-1BP studied in the previous section, as it enforces at least one backup path in the backbone network in addition to the original one, which can be arbitrarily long.

We recall that for any non-empty subset of nodes $S \subset V$, the *cutset* associated with $S$ is precisely the set of edges connecting $S$ and $V \backslash S$ namely:

$$\delta(S) = \{\{k, l\} \in E \mid k \in S, l \in V \setminus S\}.$$

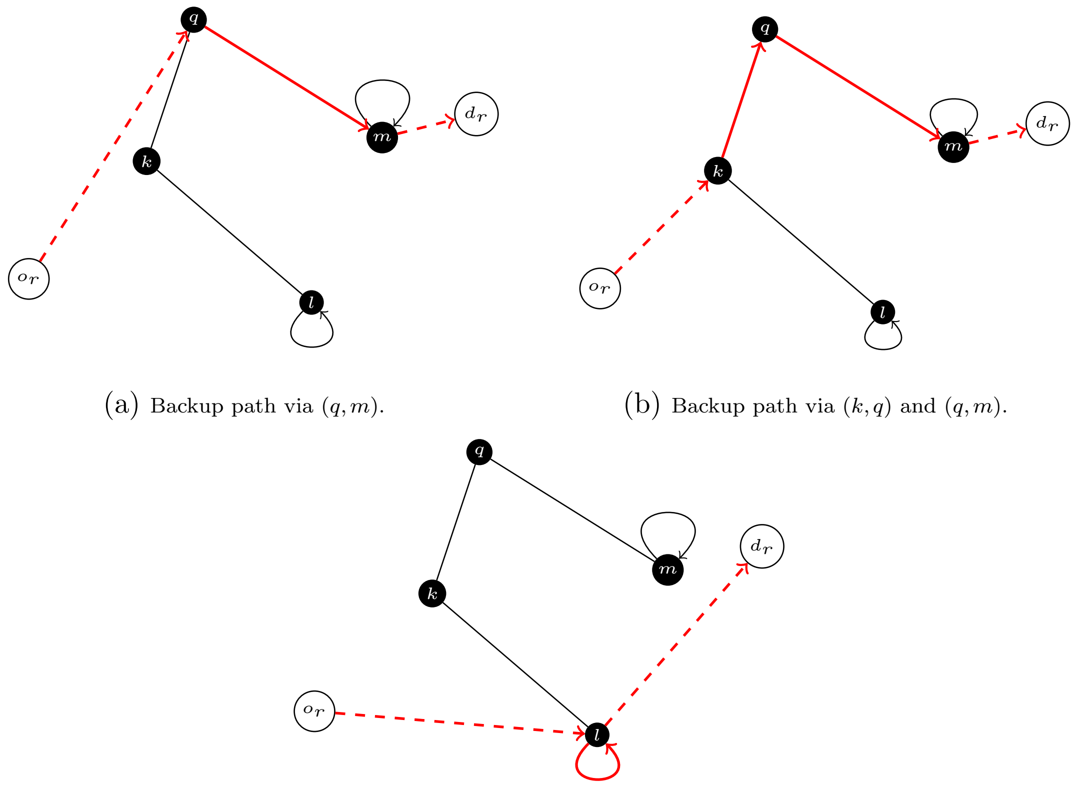
Observe that the backbone network $(H, E_H)$ depicted in Figure 1a is 2-connected since any cutset has at least two edges (one of them being possibly a loop).

Let us introduce the following additional notation. For a given indicator vector $\bar{y} \in \{0,1\}^{|E|}$:

$$\bar{y}(\delta(S)) = \sum_{\{k,l\} \in \delta(S)} \bar{y}_{kl}. \tag{13}$$

That is $\bar{y}(\delta(S))$ gives the number of edges in the cutset $\delta(S)$, that are activated relative to vector $\bar{y}$. When $\bar{y} = y$, with $y$ being the vector of hub-edge decision variables as defined in HLPIHLF-1BP, then $y(\delta(S))$ gives precisely the number of inter-hub edges in the cutset $\delta(S)$.

Continuing with the example drawn in Figure 1, Figure 2 shows different choices for backup paths in case edge $\{k, l\}$, used in the original path for commodity $r = (o_r, d_r, w_r)$, fails. Note that, whereas the backup paths drawn in Figures 2a and 2c are also valid for HLPIHLF-1BP, the backup path shown in Figure 2b uses two inter-hub arcs, thus not being valid for HLPIHLF-1BP. Similarly to HLPIHLF-1BP, loops also account for the $\lambda$-connectivity, as they can be used both in original and backup paths. Note also that in case the original path for commodity $r = (o_r, d_r, w_r)$ uses the loop $(m, m)$, the backup paths in Figure 2 are also feasible.




(a) Backup path via $(q, m)$.

(b) Backup path via $(k, q)$ and $(q, m)$.

(c) Backup path via $(l, l)$.

**Figure 2 Alternative possibilities for backup paths in the $\lambda$-connected model.**

With the above notation, and taking into account that, by definition, $\delta(S)$ contains no loops, the $\lambda$-connectivity of the backbone network can be stated by means of the following constraints, associated with each subset $S \subset V$, and each pair of potential hubs $k, l \in V$ with $k \in S$, $l \notin S$:

$$y(\delta(S)) + y_{kk} \geq \lambda \left(z_k + z_l - 1\right). \tag{14}$$

The right hand side of the above constraint can take a strictly positive value only when $k, l$ are activated hub nodes, that is, when $\delta(S)$ is a cutset of the backbone network. In this case, the inequality imposes that $\delta(S)$ contains at least $\lambda - y_{kk}$ activated hub edges. As indicated, the loop $\{k, k\}$ must also be accounted for a potential hub edge since it can be used in routing paths. Hence, if the loop $\{k, k\}$ is activated as a hub edge, it should be subtracted from the number of hub edges in $\delta(S)$ that must be activated. Summarizing, the above constraint imposes that, if nodes $k, l$, $k \in S$, $l \notin S$, are activated hubs, then the number of hub edges in the cutset $\delta(S)$ must be at least $\lambda - 1$ if the loop $\{k, k\}$ is activated as a hub edge or $\lambda$ otherwise. The $\lambda$-connectivity of singletons can be imposed by means of constraints $y(\delta(k)) + y_{kk} \geq \lambda z_k$, $k \in V$, which have an analogous interpretation.

Below we develop a MILP formulation for the HLPIHLF-$\lambda$, which incorporates $\lambda$-connectivity by means of the family of Constraints (14) introduced above. The formulation uses the same $z$, $y$, and $x$ variables as before. Still, since backup paths are no longer made explicit, variables $\bar{x}$ used in formulation HLPIHLF-1BP of the previous section are no longer needed. As explained, the $y$ variables will be used to impose the $\lambda$-connectivity condition, which will be stated by means of an exponential set of constraints.

Given that backup routes are no longer made explicit, we no longer have closed expressions for their expected routing costs and we must estimate their values. Let us denote by $\bar{C}^r_{kl}$ an estimation of the backup routing cost of commodity $r$ when the hub arc $(k,l)$ of its original routing path fails. The resulting formulation for the HLPIHLF-$\lambda$ is:

$$\min \sum_{k\in V} f_k z_k + \sum_{\{k,l\}\in E} h_{kl} y_{kl} + \sum_{r\in R}\sum_{(k,l)\in A} \left[(1-p_{kl})C^r_{kl} + p_{kl}\bar{C}^r_{kl}\right] x^r_{kl}$$

$$\text{s.t.} \sum_{(k,l)\in A} x^r_{kl} = 1 \qquad \forall r\in R \tag{2.1}$$

$$x^r_{kl} + x^r_{lk} \le y_{kl} \qquad \forall r\in R, \{k,l\}\in E, k\neq l \tag{2.2}$$

$$x^r_{kk} \le y_{kk} \qquad \forall r\in R, k\in V \tag{2.3}$$

$$y_{kl} \le z_k \qquad \forall \{k,l\}\in E \tag{2.4}$$

$$y_{kl} \le z_l \qquad \forall \{k,l\}\in E, k\neq l \tag{2.5}$$

$$y(\delta(S)) + y_{kk} \ge \lambda(z_k + z_l - 1) \qquad \forall S\subset V,\ |S|\ge 2,\ k\in S,\ l\notin S \tag{2.6}$$

$$y(\delta(k)) + y_{kk} \ge \lambda z_k \qquad \forall k\in V \tag{2.7}$$

$$x^r_{kl}\in\{0,1\} \qquad \forall r\in R, (k,l)\in A \tag{2.8}$$

$$z_k\in\{0,1\} \qquad \forall k\in V \tag{2.9}$$

$$y_{kl}\in\{0,1\} \qquad \forall \{k,l\}\in E, \tag{2.10}$$

where Constraints (2.1)-(2.5) are similar to (1.1)-(1.6) but referring to the original path only, and (2.6) and (2.7) are the $\lambda$-connectivity constraints described above.

Note that once the hub backbone network $(H, E_H)$ is obtained by solving the above problem, one can explicitly compute a backup path for a commodity $r\in R$, whose original path is $(o_r, k_r, l_r, d_r)$, by solving (in polynomial time) a shortest path problem from source $o_r$ to destination $d_r$ on the graph $G_r = (V_r, E_r)$ with nodes $V_r = \{o_r, d_r\}\cup H$ and edges $E_r = \Big\{\{o_r, h\} : h\in H\Big\}\cup E_h \cup \Big\{(h, d_r) : h\in H\Big\}\setminus\{k_r, l_r\}$.

We next show that, in some cases, Constraints (2.6) can be reinforced.

**Proposition 5.1** *Let $S \subset V$ be a nonempty subset of nodes with $|S| \leq \lambda - 1$. Then, the associated Constraints* (2.6) *are dominated by the following set of inequalities, which are also valid for* (HLPIHLF$-\lambda$)*:*

$$y(\delta(S)) + y_{kk} \geq \lambda z_k, \qquad \forall k \in S. \tag{2.11}$$

*Proof.* We first see that (2.11) are valid for (HLPIHLF$-\lambda$). Taking into account that $\lambda$-connectivity with $\lambda \geq 2$ implies that any feasible solution has at least $\lambda$ open hubs and that $|S| \leq \lambda - 1$, when $k \in S$ is activated as a hub node (i.e. $z_k = 1$), there will be at least one more open hub $\bar{l} \notin S$ (i.e. $z_{\bar{l}} = 1$). That is, when $z_k = 1$ there will be at least one active constraint in the set (2.6) with right-hand-side value $\lambda(z_k + z_{\bar{l}} - 1) = \lambda z_k$. When $k$ is not activated as a hub node (i.e. $z_k = 0$) none of Constraints (2.6), nor (2.11) will be active.
Furthermore, (2.11) dominate (2.6), since $z_l \leq 1$ implies that $\lambda z_k \geq \lambda(z_k + z_l - 1)$.
Therefore, the result follows. □

Note that when $S$ is a singleton, i.e., $S = \{k\}$, the set of Constraints (2.11) reduces precisely to (2.7).

### 5.1. Incorporation of $\lambda$-cutset constraints: a branch-and-cut approach

As already mentioned, in (HLPIHLF$-\lambda$), the size of the family of Constraints (2.6) is exponential in the number of potential hub nodes, $n$. It is thus not possible to solve the formulation directly with some off-the-shelf solver, even for medium size instances. In this section we present an exact branch-and-cut algorithm for this formulation in which, as usual, the family of constraints of exponential size (2.6) is initially relaxed. The strategy that we describe below is embedded within an enumeration tree and it is applied not only at the root node but also at all explored nodes. Our separation procedure is an adaptation of the separation procedure for classical connectivity constraints (Padberg and Grötschel 1985), and follows the same vein of those applied to more general connectivity inequalities in node and arc routing problems (see, e.g., Belenguer and Benavent 1998, Aráoz et al. 2009, Rodríguez-Pereira et al. 2019, for further details).

The initial formulation includes all Constraints (2.1)-(2.5), and (2.7). Furthermore, all integrality conditions are relaxed.

Let $(\overline{x}, \overline{y}, \overline{z})$ be the solution to the current linear programming relaxation and let $G(\overline{y}) = (V(\overline{y}), E(\overline{y}))$ denote its associated support graph where $E(\overline{y})$ consists of all the edges of $E$ such that $\overline{y}_{kl} > 0$ and $V(\overline{y})$ the set of end nodes of the edges of $E(\overline{y})$. Each edge $(k,l) \in E(\overline{y})$ is associated with a capacity $\overline{y}_{kl}$. The separation for inequalities (2.6) is to find $S \subset V$, and $k \in S$, $l \notin S$, with $\overline{y}(\delta(S)) < \lambda(\overline{z}_k + \overline{z}_l - 1) - \overline{y}_{kk}$ or to prove that no such inequality exists. Note that, when they exist, violated $\lambda$-connectivity Constraints (2.6) can be identified from a tree of min-cuts associated with $G(\overline{y})$ relative to the capacities vector $\overline{y}$, $T(\overline{y})$.

Therefore, to solve the above separation problem we proceed as follows. For each min-cut $\delta(S)$ of $T(\overline{y})$ of value $\overline{y}(\delta(S))$, we identify $(\overline{k},\overline{l}) \in \arg\max\{\lambda\,(\overline{z}_k + \overline{z}_l - 1) - \overline{y}_{kk} : k \in S, l \notin S\}$. Then, if $\overline{y}(\delta(S)) < \lambda\,(\overline{z}_{\overline{k}} + \overline{z}_{\overline{l}} - 1) - \overline{y}_{\overline{k}\overline{k}}$, the inequality (2.6) associated with $S$ and $(\overline{k},\overline{l})$ is violated by $\overline{y}$.

In case $|S| \leq \lambda - 1$, according to Proposition 5.1, we incorporate cuts in the shape of (2.11) instead of (2.6).

In our computational experiments, we use the procedure proposed by Gusfield (1993) to identify $T(\overline{y})$. Such an algorithm computes $V(\overline{y})$ max-flows in $G(\overline{y})$, so its overall complexity is $\mathcal{O}(|V(\overline{y})| \times |V(\overline{y})|^3)$.

## 6. Computational Experience

In this section we report the results of an extensive battery of computational tests, which have been carried out to analyze the performance of the two modeling approaches for obtaining *robust* hub networks protected under inter-hub failures, discussed in the previous sections. For the experiments, we have used a large set of benchmark instances based on the well-known CAB (O'Kelly (1987)), AP (Ernst and Krishnamoorthy (1996)) and TR (Tan and Kara (2007)) datasets (taken from ORLIB `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/` and `https://ie.bilkent.edu.tr/~bkara/dataset.php`), for varying settings of the failure probabilities and other parameters as described below. All instances were solved with the Gurobi 9.1.1 optimizer, under a Windows 10 environment on an Intel(R) Core(TM) i7-6700K CPU @ 4.00 GHz 4.01 GHz processor and 32 GB of RAM. Default values were used for all parameters of Gurobi solver and a computing time limit of 7200 seconds was set.

### 6.1. Instance generation

We have generated several instances based on the entire CAB, AP and TR datasets with a number of nodes ($n$) initially ranging in $\{10, 15, 20, 25\}$ for the instances based on the CAB and TR datasets and in $\{10, 20, 25\}$ for the instances based on the AP dataset. Let $c'_{kl}$ be the standard unit transportation costs provided in ORLIB for CAB and AP instances or the travel distances provided for the TR instances and let $\mathrm{w}$ be the normalized vector of service demands, taken from the provided datasets.

Table 1 summarizes the main characteristics of the testing instances and the selected parameters. A detailed description of how the different parameters have been generated can be found in Section *Instance Generation* of the Online Supplement.

The files of the randomly generated probabilities, as well as the Python codes used for such a generation, are available in the Github repository `https://github.com/vblancoOR/HLPLF`.

For each combination of parameters $n$, $\alpha$, $\rho$, and each dataset (CAB, AP with type T and L hub nodes activation costs, and TR) five different instances have been generated for scenario of failure

| Instances | $n$ | $\lambda$ | $\alpha$ | $\beta$ |
|---|---|---|---|---|
| CAB | $\{10,15,20,25\}$ | | | |
| AP | $\{10,20,25,40,50\}$ | $\{2,3,4\}$ | $\{0.2,\ 0.5,\ 0.8\}$ | $\{0.5,1\}$ |
| TR | $\{10,15,20,25,40,50\}$ | | | |

Costs

| Instances | $c_{kl}$ | $f_k$ | $h_{kl}$ | $\bar{C}^r_{kl}$ |
|---|---|---|---|---|
| CAB | | 100 | $\begin{cases} 100\frac{c_{kl}/\mathrm{w}_{kl}}{\mathrm{MAXW}} & \text{if } k \neq l, \\ 100\frac{c_{kl}/\bar{\mathrm{w}}}{\mathrm{MAXW}} & \text{if } k = l. \end{cases}$ | |
| AP | $\alpha(a_k + c'_{kl} + d_l)$ | $T$ and $L$ data files | | $(1+\beta)C^r_{kl}$ |
| TR | | Data file | Data file | |

$a_k = d_k = \min\{\min_{j\neq k} c'_{kj}, \min_{j\neq k} c'_{jk}\}$; $\bar{\mathrm{w}} :=$ Mean of $\mathrm{w}$; $\mathrm{MAXW} = \max\{\frac{c_{ij}}{\mathrm{w}_{ij}} : i,j \in V, \mathrm{w}_{ij} > 0\}$

Failure probabilities

| | | |
|---|---|---|
| Random probabilities (RP) | $p_{kl} \sim U[0,\rho]$, | $\rho \in \{0.1, 0.3\}$ |
| Clustered probabilities (CP) | $p_{kl} \in \{0.1, 0.2, 0.3\}$ | |
| Same probability (SP) | $p_{kl} = \rho$, | $\rho \in \{0.1, 0.3\}$ |

**Table 1 Summary of instances and parameters.**

probabilities RP and one instance has been considered for scenario SP. Five instances have been also generated for scenario CP and each combination of $n$, $\alpha$, and each dataset. Thus, (HLPIHLF-1BP), hereafter called M1, has been solved on a total of 714 instances.

Concerning formulation (HLPIHLF-$\lambda$), we considered three different values for the parameter $\lambda$, namely $\lambda = 2$, $\lambda = 3$ and $\lambda = 4$ (we call the corresponding models M2_2, M2_3 and M2_4, respectively) and two values for $\beta$. Thus, (HLPIHLF-$\lambda$) has been solved on a total of 4284 instances. Additionally, for comparative purposes, we have solved 42 instances of the uncapacitated HL problem, in which no protection under failures is considered. This model will be referred to as M0.

Finally, to test the scalability of our formulations, a second experiment was carried out on a set of larger instances ($n \in \{40, 50\}$) based on the AP and TR datasets considering only (HLPIHLF-$\lambda$),

| | | | CPUTime | | | | | MIPGAP | | | | | | | | | | | | %Solved | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | RP | | CP | SP | | RP | | | | CP | | SP | | | | RP | | CP | SP | |
| | | | | | | | | 0,1 | | 0,3 | | | | 0,1 | | 0,3 | | | | | | |
| n | $\alpha$ | Data | 0,1 | 0,3 | | 0,1 | 0,3 | Av | Max | Av | Max | Av | Max | Av | Max | Av | Max | 0,1 | 0,3 | | 0,1 | 0,3 |
| 10 | 0,2 | $AP_T$ | 4 | 7 | 15 | 1 | 1 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| | | $AP_L$ | 7 | 93 | 13 | 2 | 3 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| | | CAB | 143 | 5198 | TL | 1 | 1 | 0,0 | 0,0 | 0,5 | 1,1 | 1,1 | 1,9 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 40 | 0 | 100 | 100 |
| | | TR | 4 | 9 | 44 | 0 | 1 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| | 0,5 | $AP_T$ | 6 | 21 | 14 | 1 | 1 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| | | $AP_L$ | 10 | 112 | 11 | 2 | 3 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| | | CAB | 62 | 5391 | 1864 | 1 | 1 | 0,0 | 0,0 | 1,0 | 3,5 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 40 | 100 | 100 | 100 |
| | | TR | 5 | 11 | 63 | 0 | 1 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| | 0,8 | $AP_T$ | 7 | 13 | 15 | 1 | 1 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| | | $AP_L$ | 9 | 99 | 12 | 1 | 3 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| | | CAB | 25 | TL | 727 | 1 | 1 | 0,0 | 0,0 | 0,6 | 1,2 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 0 | 100 | 100 | 100 |
| | | TR | 5 | 19 | 53 | 0 | 1 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| 15 | 0,2 | CAB | TL | TL | TL | 2 | 7 | 0,6 | 1,0 | 8,3 | 9,6 | 9,4 | 10,8 | 0,0 | 0,0 | 0,0 | 0,0 | 0 | 0 | 0 | 100 | 100 |
| | | TR | 51 | 71 | 287 | 4 | 6 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| | 0,5 | CAB | 4577 | TL | TL | 5 | 8 | 0,4 | 0,7 | 9,8 | 12,5 | 7,6 | 8,4 | 0,0 | 0,0 | 0,0 | 0,0 | 40 | 0 | 0 | 100 | 100 |
| | | TR | 52 | 89 | 619 | 3 | 5 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| | 0,8 | CAB | 511 | TL | TL | 4 | 6 | 0,0 | 0,0 | 8,8 | 13,3 | 3,7 | 5,5 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 0 | 0 | 100 | 100 |
| | | TR | 55 | 138 | 412 | 3 | 4 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| 20 | 0,2 | $AP_T$ | 1812 | TL | 6336 | 43 | 30 | 0,0 | 0,0 | 5,9 | 15,8 | 1,4 | 2,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 0 | 20 | 100 | 100 |
| | | $AP_L$ | TL | TL | TL | 3443 | 2080 | 14,8 | 15,4 | 20,5 | 21,8 | 17,8 | 19,1 | 0,0 | 0,0 | 0,0 | 0,0 | 0 | 0 | 0 | 100 | 100 |
| | | CAB | TL | TL | TL | 13 | 121 | 3,1 | 3,7 | 14,4 | 16,5 | 15,4 | 17,9 | 0,0 | 0,0 | 0,0 | 0,0 | 0 | 0 | 0 | 100 | 100 |
| | | TR | 360 | 1402 | 4131 | 32 | 32 | 0,0 | 0,0 | 0,0 | 0,0 | 1,1 | 5,3 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 80 | 100 | 100 |
| | 0,5 | $AP_T$ | 1520 | TL | 6611 | 53 | 34 | 0,0 | 0,0 | 2,9 | 7,4 | 1,3 | 3,4 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 0 | 20 | 100 | 100 |
| | | $AP_L$ | TL | TL | TL | 4308 | 1257 | 13,1 | 14,5 | 20,3 | 21,9 | 17,2 | 19,2 | 0,0 | 0,0 | 0,0 | 0,0 | 0 | 0 | 0 | 100 | 100 |
| | | CAB | TL | TL | TL | 20 | 51 | 3,0 | 3,8 | 15,7 | 19,5 | 15,5 | 19,8 | 0,0 | 0,0 | 0,0 | 0,0 | 0 | 0 | 0 | 100 | 100 |
| | | TR | 363 | 1850 | 5497 | 21 | 36 | 0,0 | 0,0 | 0,0 | 0,0 | 3,6 | 12,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 60 | 100 | 100 |
| | 0,8 | $AP_T$ | 1693 | 7137 | 5780 | 40 | 33 | 0,0 | 0,0 | 3,8 | 12,1 | 1,6 | 4,3 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 20 | 40 | 100 | 100 |
| | | $AP_L$ | TL | TL | TL | 2343 | 1809 | 12,6 | 13,6 | 19,0 | 20,6 | 15,6 | 16,5 | 0,0 | 0,0 | 0,0 | 0,0 | 0 | 0 | 0 | 100 | 100 |
| | | CAB | TL | TL | TL | 12 | 27 | 2,8 | 3,9 | 16,4 | 19,3 | 16,2 | 18,3 | 0,0 | 0,0 | 0,0 | 0,0 | 0 | 0 | 0 | 100 | 100 |
| | | TR | 422 | 1864 | 5625 | 19 | 33 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 100 | 100 | 100 | 100 |
| 25 | 0,2 | $AP_T$ | 4774 | 6835 | 4731 | 128 | 129 | 2,4 | 11,8 | 12,5 | 21,6 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 80 | 20 | 100 | 100 | 100 |
| | | $AP_L$ | TL | TL | TL | TL | TL | 16,6 | 17,9 | 22,8 | 24,1 | 19,8 | 20,4 | 11,5 | 11,5 | 14,2 | 14,2 | 0 | 0 | 0 | 0 | 0 |
| | | CAB | OoM | TL | TL | 62 | 3208 | 4,3 | 5,4 | 19,5 | 21,7 | 19,2 | 23,2 | 0,0 | 0,0 | 0,0 | 0,0 | 0 | 0 | 0 | 100 | 100 |
| | | TR | 2332 | TL | TL | 119 | 169 | 0,0 | 0,0 | 12,0 | 13,2 | 17,3 | 22,4 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 0 | 0 | 100 | 100 |
| | 0,5 | $AP_T$ | 6389 | TL | 4789 | 129 | 179 | 5,0 | 9,5 | 12,6 | 19,3 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 40 | 0 | 80 | 100 | 100 |
| | | $AP_L$ | TL | TL | TL | TL | TL | 14,7 | 16,3 | 20,8 | 24,4 | 18,1 | 20,2 | 8,6 | 8,6 | 13,4 | 13,4 | 0 | 0 | 0 | 0 | 0 |
| | | CAB | OoM | OoM | TL | 98 | 2403 | 4,7 | 6,7 | 23,2 | 24,2 | 20,4 | 23,3 | 0,0 | 0,0 | 0,0 | 0,0 | 0 | 0 | 0 | 100 | 100 |
| | | TR | 1935 | TL | TL | 82 | 128 | 0,0 | 0,0 | 11,5 | 13,3 | 13,3 | 14,3 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 0 | 0 | 100 | 100 |
| | 0,8 | $AP_T$ | 6465 | 6789 | 3490 | 125 | 316 | 5,1 | 8,7 | 10,3 | 20,6 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 0,0 | 40 | 20 | 100 | 100 | 100 |
| | | $AP_L$ | TL | TL | TL | 2927 | TL | 13,8 | 14,8 | 20,5 | 21,8 | 17,5 | 22,5 | 0,0 | 0,0 | 11,9 | 11,9 | 0 | 0 | 0 | 100 | 0 |
| | | CAB | OoM | OoM | TL | 71 | 2740 | 7,1 | 9,2 | 24,0 | 25,1 | 18,9 | 20,6 | 0,0 | 0,0 | 0,0 | 0,0 | 0 | 0 | 0 | 100 | 100 |
| | | TR | 2483 | TL | TL | 53 | 80 | 0,0 | 0,0 | 12,7 | 14,6 | 14,2 | 15,1 | 0,0 | 0,0 | 0,0 | 0,0 | 100 | 0 | 0 | 100 | 100 |

**Table 2 Results for (HLPIHLF-1BP).**

which, as we will see, is the most promising formulation, for $\lambda \in \{2,4\}$ and $\beta = 1$. We have solved a total of 612 instances in this second study. Overall, 5652 instances have been solved.

## 6.2. Numerical results with (HLPIHLF-1BP) and (HLPIHLF-$\lambda$)

In the following sections we present and compare the results obtained in our computational study with (HLPIHLF-1BP) and (HLPIHLF-$\lambda$).

**6.2.1. Results with (HLPIHLF-1BP)** The results produced by (HLPIHLF-1BP) are summarized in Table 2. In that table, “RP”, “CP” and “SP” stand for the scenarios with random failure probabilities (with $\rho = 0.1$ and $\rho = 0.3$), clustered failure probabilities and same failure probability (with $\rho = 0.1$ and $\rho = 0.3$), respectively. The values of $n$, $\alpha$ and “Data” indicate the number of nodes in the network, the value for the discount factor applied to the routing cost through inter-hub arcs, and the dataset that has been used to obtain the costs and the flows, respectively. "$\text{AP}_T$" and "$\text{AP}_L$" refer to AP dataset using type T and type L fixed costs for the hub nodes, respectively. For scenarios RP and CP, the information contained in each row refers to average values over the five instances with the corresponding combination of parameters, whereas for scenario SP the values of the entries correspond to the unique instance with this combination of parameters. The maximum MIPGap values over the five instances are also shown under "Max" columns.

The numerical results are grouped in three blocks of columns. Block "CPUTime" gives the computing times, in seconds, required to solve the instances, block "MIPGap" the percentage MIP gaps returned by Gurobi at termination, and block "%Solved" the percentage of instances solved to proven optimality within the time limit. An entry "`TL`" in the CPUTime block means that the time limit of 7200 seconds was reached in all five instances of the group. The "`OoM`" entry indicates that the flag "Out of memory" was the output of the solver in at least one of the instances in the row, and then, the remaining information of the row refers to average values (or maximum values when corresponding) over the solved instances only (even if none of these instances could be solved to proven optimality). We observe that there is no significant difference between the maximum and the average MIP gap values. Thus, in what follows we only report average values in our tables and figures.

Table 2 shows a different performance of (HLPIHLF-1BP) among the instances corresponding to the different scenarios. Based on the computing times, MIPGaps, and percentage of solved instances, scenario SP produces the easiest instances, for all configurations of parameters, as expected. One can observe that, for instances with the same probability, all instances generated from the CAB, $\text{AP}_T$, and TR datasets with up to $n = 25$, as well as the instances generated from $\text{AP}_L$ with up to $n = 20$ have been optimally solved within the time limit.

On the other hand, note that instances based on CAB dataset are more difficult to solve than instances based on TR and AP datasets. TR based instances are the easiest to solve: all instances with up to $n = 15$, 95% for $n = 20$ and 35% for $n = 25$ have been optimally solved within the time limit. Regarding AP based instances, all instances with $n = 10$ have been optimally solved within the time limit, although $\text{AP}_T$ instances consumed, in general, less computing time. The difference between $\text{AP}_T$ and $\text{AP}_L$ instances becomes more evident for $n > 10$, since approximately 50% of the $\text{AP}_T$ instances were optimally solved whereas no $\text{AP}_L$ instance with random or clustered probability (scenarios RP and CP) was solved to proven optimality within the time limit. As mentioned before, CAB instances are the most difficult ones. For $n = 10$, 30% of these instances could not be optimally solved solved within the time limit. This percentage increases up to 90% for $n = 20$ and up to 100% for $n = 25$. Additionally, for $n = 25$, the execution was stopped due to an Out of Memory flag for 30% of the CAB instances under a random probability (RP) scenario.

**6.2.2. Results with (HLPIHLF-$\lambda$)** Using formulation (HLPIHLF-$\lambda$), all instances could be solved to proven optimality within the maximum computing time for all three considered values of $\lambda \in \{2,3,4\}$.

This means that (HLPIHLF-$\lambda$) is notably easier to solve than (HLPIHLF-1BP). This can be explained by its smaller number of decision variables.

The difficulty of (HLPIHLF-$\lambda$) increases with the value of $\lambda$, as reflected by a decrease in its performance for higher values of this parameter. This could be expected, as (HLPIHLF-$\lambda$) becomes more restrictive as the value of $\lambda$ increases. When $\lambda = 2$, the average computing time over all the instances is approximately 64 seconds, being two seconds for the CAB instances, 70 seconds for the AP$_T$ instances, 89 seconds for the AP$_L$ instances, and 102 for the TR instances. Unlike (HLPIHLF-1BP), CAB instances are computationally less demanding than AP and TR instances. This behavior was also observed for $\lambda = 3$ and $\lambda = 4$. For $\lambda = 4$ the average computing time over all the instances is approximately 218 seconds, being 30 seconds for the CAB instances, 168 seconds for the AP$_T$ instances, 225 seconds for the AP$_L$ instances, and 438 for the TR instances. We have observed that the value of $\alpha$ also affects the performance of (HLPIHLF-$\lambda$), instances being more difficult for smaller $\alpha$ values, specially for the AP instances. We also noted that, unlike (HLPIHLF-1BP), with (HLPIHLF-$\lambda$) there seem to be no noticeably differences among scenarios.

Detailed results on the computing times required in the implementation of (HLPIHLF-$\lambda$) with the tested instances can be found in the Online Supplement.

Given the good performance of (HLPIHLF-$\lambda$) with instances up to 25 nodes, we carried out a second set of computational experiments, on larger instances ($n \in \{40,50\}$) based on the AP and TR datasets. For this second set of experiments we only considered (HLPIHLF-$\lambda$) for $\lambda \in \{2,4\}$ and $\beta = 1$, but did not consider (HLPIHLF-1BP), since it most instances could not be optimally solved with it within the time limit already for $n = 25$. Table 3 summarizes the obtained results.

We can observe that, for $\lambda = 2$, all the TR instances, 99% of the AP$_T$ instances, and 80% of the $AP_L$ instances have been solved to proven optimality within the time limit, whereas for $\lambda = 4$ the percentage of solved instances decreased to 70% for AP$_T$, 40% for $AP_L$, and 6% for TR. This shows that (HLPIHLF-$\lambda$) is able to solve larger instances with up to $n = 50$, even if instances become more challenging as the value of the parameter $\lambda$ increases.

## 6.3. Managerial Insight

In this section we derive some managerial insight from the results obtained in our first set of experiments, i.e., $n \leq 25$ when the instances were solved with both formulations, as well from the solutions of these instances for M0 (the uncapacitated HLP with no protection under failures).

| | | | | CPUTime | | | | | MIPGap | | | | | %Solved | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | RP | | | SP | | RP | | | SP | | RP | | | SP | |
| | n | $\alpha$ | Data | 0.1 | 0.3 | CP | 0.1 | 0.3 | 0.1 | 0.3 | CP | 0.1 | 0.3 | 0.1 | 0.3 | CP | 0.1 | 0.3 |
| $\lambda=2$ | 40 | 0.2 | $AP_T$ | 371 | 363 | 382 | 361 | 310 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | | $AP_L$ | 1608 | 1689 | 5418 | 1181 | 1059 | 0.00 | 0.00 | 4.43 | 0.00 | 0.00 | 100 | 100 | 60 | 100 | 100 |
| | | | TR | 870 | 775 | 773 | 666 | 537 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | 0.5 | $AP_T$ | 344 | 479 | 313 | 322 | 432 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | | $AP_L$ | 3579 | 3155 | 2487 | 2602 | 1351 | 0.00 | 3.16 | 0.00 | 0.00 | 0.00 | 100 | 80 | 100 | 100 | 100 |
| | | | TR | 434 | 600 | 601 | 337 | 414 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | 0.8 | $AP_T$ | 331 | 376 | 325 | 303 | 337 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | | $AP_L$ | 2067 | 1663 | 2126 | 1604 | 692 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | | TR | 499 | 489 | 430 | 512 | 390 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | 50 | 0.2 | $AP_T$ | 3986 | 3470 | 5128 | 4147 | 3651 | 0.00 | 0.00 | 2.46 | 0.00 | 0.00 | 100 | 100 | 80 | 100 | 100 |
| | | | $AP_L$ | 6427 | 5362 | 6472 | 3273 | TL | 13.79 | 6.56 | 12.40 | 0.00 | 15.64 | 20 | 60 | 40 | 100 | 0 |
| | | | TR | 2877 | 2520 | 3385 | 2180 | 3817 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | 0.5 | $AP_T$ | 2635 | 3789 | 4319 | 4823 | 1590 | 0.00 | 0.00 | 2.02 | 0.00 | 0.00 | 100 | 100 | 80 | 100 | 100 |
| | | | $AP_L$ | 3819 | 3406 | TL | 2761 | 2131 | 0.00 | 3.45 | 17.09 | 0.00 | 0.00 | 100 | 80 | 0 | 100 | 100 |
| | | | TR | 1296 | 3037 | 3264 | 1201 | 1349 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | 0.8 | $AP_T$ | 2862 | 3732 | 2836 | 3736 | 2474 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | | $AP_L$ | 5872 | 5151 | 7183 | 6279 | 1172 | 9.93 | 2.99 | 14.02 | 0.00 | 0.00 | 40 | 80 | 20 | 100 | 100 |
| | | | TR | 1010 | 1495 | 1356 | 852 | 1050 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| $\lambda=4$ | 40 | 0.2 | $AP_T$ | 2919 | 2907 | 2300 | 2582 | 6060 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | | $AP_L$ | TL | TL | 5643 | 5427 | 5674 | 49.57 | 50.23 | 19.51 | 0.00 | 0.00 | 0 | 0 | 60 | 100 | 100 |
| | | | TR | 6958 | 7044 | 6002 | TL | TL | 13.64 | 9.16 | 5.89 | 7.14 | 10.61 | 20 | 20 | 60 | 0 | 0 |
| | | 0.5 | $AP_T$ | 2807 | 2335 | 2293 | 2186 | 2939 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | | $AP_L$ | 5589 | 4865 | 3875 | 5535 | 2960 | 10.03 | 10.03 | 0.00 | 0.00 | 0.00 | 80 | 80 | 100 | 100 | 100 |
| | | | TR | TL | 7030 | 7024 | TL | TL | 20.15 | 14.00 | 12.62 | 22.24 | 13.99 | 0 | 20 | 20 | 0 | 0 |
| | | 0.8 | $AP_T$ | 2315 | 2400 | 2272 | 1640 | 1750 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 100 | 100 | 100 | 100 | 100 |
| | | | $AP_L$ | 4959 | 5691 | 3512 | 6253 | 2767 | 19.83 | 9.12 | 0.00 | 0.00 | 0.00 | 60 | 80 | 100 | 100 | 100 |
| | | | TR | TL | 6991 | 7139 | TL | TL | 17.73 | 13.56 | 12.25 | 17.51 | 14.20 | 0 | 20 | 20 | 0 | 0 |
| | 50 | 0.2 | $AP_T$ | 7089 | 7198 | 6966 | 7134 | 6004 | 30.03 | 40.96 | 27.71 | 0.00 | 0.00 | 40 | 20 | 40 | 100 | 100 |
| | | | $AP_L$ | TL | TL | TL | TL | TL | 49.99 | 49.23 | 50.31 | 48.24 | 46.73 | 0 | 0 | 0 | 0 | 0 |
| | | | TR | TL | TL | TL | TL | TL | 17.12 | 13.76 | 16.23 | 17.77 | 16.54 | 0 | 0 | 0 | 0 | 0 |
| | | 0.5 | $AP_T$ | 6482 | 6995 | 6711 | 5310 | TL | 20.67 | 39.79 | 25.85 | 0.00 | 49.76 | 60 | 20 | 40 | 100 | 0 |
| | | | $AP_L$ | TL | TL | TL | TL | TL | 49.68 | 48.38 | 50.03 | 48.53 | 48.19 | 0 | 0 | 0 | 0 | 0 |
| | | | TR | TL | TL | TL | TL | TL | 24.59 | 16.95 | 17.05 | 22.29 | 20.88 | 0 | 0 | 0 | 0 | 0 |
| | | 0.8 | $AP_T$ | 6933 | 7081 | 6259 | 7202 | 7201 | 28.66 | 39.67 | 0.00 | 29.26 | 46.94 | 40 | 20 | 100 | 0 | 0 |
| | | | $AP_L$ | TL | TL | TL | TL | TL | 48.91 | 47.81 | 47.93 | 48.24 | 46.94 | 0 | 0 | 0 | 0 | 0 |
| | | | TR | TL | TL | TL | TL | TL | 25.94 | 21.03 | 22.78 | 27.43 | 23.51 | 0 | 0 | 0 | 0 | 0 |

**Table 3 Average results for (HLPIHLF-$\lambda$) for $n \geq 40$.**

We first analyze the contribution to the objective function value of the different types of costs: routing costs, fixed costs for activating hubs (Hubs_Costs) and fixed costs for activating inter-hub edges (Links_Costs). Since results are similar for $AP_L$ and $AP_T$ data sets, for this analysis we differentiate, for each formulation, between datasets CAB, AP and TR, as well as among the three values of the $\alpha$ parameter.

The obtained results indicate that the percent contribution of the fixed costs for activating inter-hub edges varies from 0.1% for CAB instances to 13% for TR instances in M0. As expected, the percent contribution of hub fixed costs depends on the value of the parameter $\alpha$, but mainly on the dataset and on the model. For the instances based on the CAB dataset, the percent contribution of hub fixed costs varies from 20% with M0, M1, and M2_2 for $\alpha = 0.8$, to 40% with M2_4 for $\alpha = 0.2$. For the instances based on the AP dataset, the percent contribution of hub fixed costs varies from 45% with M0 and $\alpha \in \{0.5, 0.8\}$ to 75% with M2_4. Regarding the instances based on the TR dataset, the percent contribution of hub fixed costs varies from 21% with M0 and $\alpha \in \{0.5, 0.8\}$ to

69% with M2_4 for $\alpha = 0.2$. The percent contribution of routing costs also depends on the value of the parameter $\alpha$, on the dataset, and on the model. For the instances based on the CAB dataset, this percentage varies from 55% to 80%, with the highest values for M0, M1, and M2_2 and $\alpha = 0.8$. For the instances based on the AP dataset, the percent contribution of routing costs varies from 15% to 55%, corresponding the highest values to M0 with $\alpha = 0.8$. Finally, for the instances based on the TR dataset, the percent contribution of routing costs varies from 22% to 68%, corresponding again the highest values to M0 with $\alpha = 0.8$.

In the Online Supplement we provide the bar diagrams of the percent contribution of each type of cost for the different models.

For analyzing the structure of the backbone networks produced by the different tested models we have studied the number of open hubs and activated inter-hub edges, together with two density indices, denoted by $I_1$ and $I_2$. $I_1$ indicates the density of the backbone network including loops and is computed as the ratio between the number of activated inter-hub edges and the total number of edges in a complete graph with $\#H$ nodes if loops were included:

$$I_1 = \frac{2\#Lk}{\#H(\#H+1)}.$$

$I_2$ indicates the density of the backbone network when loops are not considered, and is computed as the ratio between the number of non-loop inter-hub edges activated and the number of edges in a complete graph with $\#H$ nodes if loops were excluded:

$$I_2 = \frac{2(\#Lk - \#Lp)}{\#H(\#H-1)}.$$

Note that values of the two indices range in $[0,1]$, i.e., $0 \leq I_1, I_2 \leq 1$.

The obtained results show that the number of open hubs mostly depends on the model and also on the dataset. For all the AP instances the number of open hubs is always one for M0, two for M1 and M2_2, three for M2_3, and four for M2_4. For the CAB instances this number ranges, in average, in [2, 5] for M0, in [2.18, 4.29] for M1, in [2, 4.76] for M2_2, in [3, 4.76] for M2_3, and in [4, 4.74] for M2_4. For the TR instances, the number of open hubs is always one for M0, two for M1, three for M2_2, four for M2_3, and five for M2_4.

Concerning the number of activated inter-hub edges, our results indicate that this number is smaller for M0 than for M1 and that, for M1, this number is similar to that for M2_2 but smaller than that for M2_$\lambda$ for $\lambda > 2$, since, as expected, this number increases with the value of $\lambda$. Additionally, with M0 most of the activated inter-hub edges are loops, specially with the AP and the TR instances. In fact, we have observed that M0 gives a density index $I_1$ close to 1 and a density index $I_2$ close to 0, for both the AP instances and TR instances. This indicates that most of the activated inter-hub edges are loops. Instead, for the TR instances the optimal backbone networks produced by M1 have both $I_1 = I_2 = 1$.

We finally point out that, as expected, the density of the backbone network for M2_$\lambda$ increases with the value of the parameter $\lambda$.

In the Online Supplement we provide with several tables and figures that support the above analysis. In particular, we report the average values for the number of hubs (#H), inter-hub edges including loops (#Lk), and loops (#Lp) activated in the optimal backbone networks for the different models; and depict bar diagrams of the values for the density indices $I_1$ and $I_2$, for the different types of instances.

## 7. The price of robustness

For assessing the robustness and reliability of the hub network models proposed in this paper, we evaluate the so-called price of robustness (see e.g. Bertsimas and Sim 2004), defined as the extra cost incurred to design a robust network. In our case, robustness translates into protecting the backbone network under inter-hub edge failures, which, essentially, is attained by incorporating additional inter-hub edges to the backbone network. We thus start our analysis by comparing the overall fixed cost for each of the proposed models M1, M2_2, M2_3, and M2_4, with that of the *unprotected* network obtained with M0. The interested reader is referred to the Online Supplement for detailed information about these costs.

As expected, constructing networks that are robust under inter-hub edge failures has a significant impact in the design costs (fixed costs for activating hub nodes and inter-hub edges) of the network. We have observed that, in this type of instances, the value of $\alpha$ scarcely affects the design costs in any of the models. The percent deviation of the activated hubs and inter-hub edges fixed costs with respect those of M0 ranges, in average, from around 65% for M1 to around 335% for M2_4, increasing for the $\lambda$-connected models with the value of $\lambda$. This can be easily explained as larger backbone networks are required as $\lambda$ increases. Nevertheless, as shown by the results of the experiment that we report next, in case of failure, this increase in the design costs strengthens the possibility of being able to re-route all the commodities of *a posteriori* solutions (after the occurrence of a failure in the inter-hub edges).

For this experiment, we have used all $n = 20$ instances of the TR dataset, whose optimality is guaranteed for all models. For each of the 255 instances generated for the TR dataset with $n = 20$, we have simulated the following scenarios for potential failures of the inter-hub edges of the backbone networks produced by the different models:

- **Failure scenario 1 (FS1):** Only activated inter-hub edges may fail. Each activated hub edge $\{k,l\}$ in a backbone network fails (and removed from the backbone network) according to a Bernoulli distribution with probability $p_{kl}$.

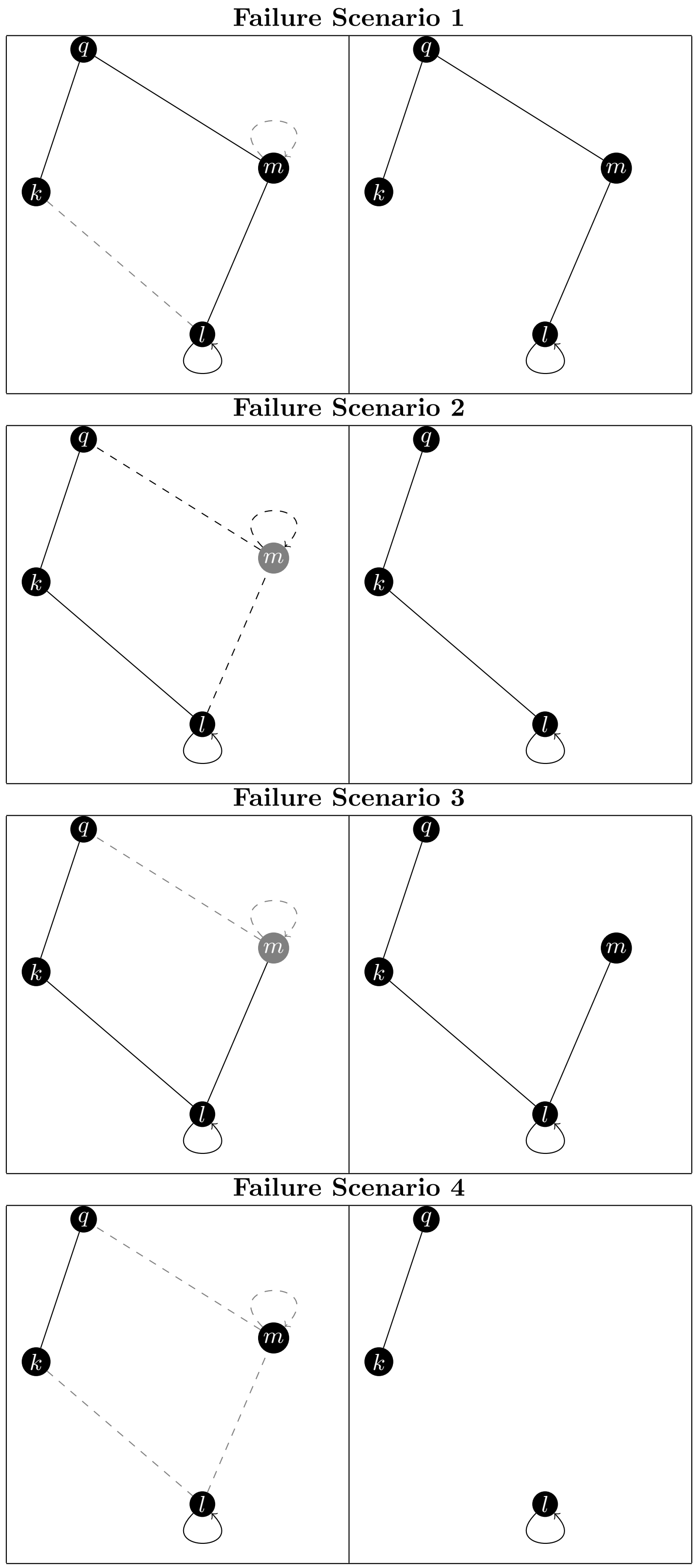



**Figure 3** **Illustration of the four different failure scenarios that we perform.**

- **Failure scenario 2 (FS2):** In this scenario failures are associated with hub nodes. Failure of a hub node implies the failure of all the inter-hub edges incident to the hub. Thus, each activated hub fails with probability $p_{kk}$, and then all its incident inter-hub edges are removed from the backbone network.
- **Failure scenario 3 (FS3):** Failures are simulated for inter-hub edges of the backbone network similarly to FS1. In case an inter-hub edge fails, the failure probability of the loops at the end nodes of the edges is increased by 50%. Then, failures in the loop edges are simulated.
- **Failure scenario 4 (FS4):** First, failures are simulated for inter-hub edges of the backbone network similarly to FS1. The difference is that we now assume that the failure of a considerable number of inter-hub edges incident to any activated hub node, will provoke the failure of the hub node as well, and thus the failure of all its incident inter-hub edges. That is, for each activated hub node, we assume that if at least a given percentage $\gamma\%$ of its incident inter-hub edges have failed, then the whole hub node fails, provoking that its remaining incident inter-hub edges also fail, which are also removed from the network. In our study we fix the value of the parameter $\gamma$ to 75%, i.e., if 75% or more of the inter-hub edges incident to a hub node fails, then, the hub (and the remaining incident inter-hub edges) cannot be used any more for routing the commodities.

Figure 3 illustrates with simple backbone networks $(H, E_H)$ the four failure scenarios that we consider. The backbone network has four hub nodes $H = \{k, l, m, q\}$ and six inter-hub edges, of which two are loops $E_H = \{\{k, l\}, \{k, q\}, \{l, m\}, \{q, m\}, \{l, l\}, \{m, m\}\}$. In FS1, hub edges $\{k, l\}$ and $\{m, m\}$ are chosen to fail, both depicted with dashed lines in the left picture. The network resulting after removing these inter-hub edges is shown in the right picture. In FS2, hub node $m$ (in a gray circle) is chosen to fail, and then, all the hub edges incident with it (depicted with dashed lines), namely $\{q, m\}, \{m, m\}$ and $\{l, m\}$, are removed from the network. In FS3 the inter-hub edge $\{q, m\}$ is chosen to fail, and increases in 50% the failure probability of the loop $\{m, m\}$, which is then randomly chosen to fail. Thus, we chose $\{q, m\}$ and $\{m, m\}$ (depicted with dashed lines), which are then removed from the network. Finally, in FS4, hub edges $\{k, l\}$, $\{l, m\}$, $\{q, m\}$ and $\{m, m\}$ are chosen to fail. Then, since the percentage of inter-hub edges incident to $m$ that fail exceeds $\gamma = 75\%$, all edges incident to $m$ are removed. For the remaining hub nodes, such a percentage is not exceeded so no further inter-hub edges are removed.

We have carried out simulations for each of the above failure scenarios, all of which follow the same general structure, for a given backbone network. $(i)$ We randomly generate the edges that fail according to the corresponding failure scenario and obtain the *after-failure* network by removing from the backbone network these edges. $(ii)$ We try to re-route all the demand through the *after-failure* network. $(iii)$ Since it may happen that it is no longer possible to route the demand in the *after-failure* network, we analyze this circumstance in our study. For each failure scenario, each simulation is repeated 10000 times over each instance. The obtained results indicate that the networks obtained with the proposed models (M1 and M2_$\lambda$) are clearly more robust under inter-hub edge failures than M0. Specifically, in average, with our models all the demand can be re-routed after failure in 90% of the 1000 simulations, whereas with M0 all the demand can be re-routed in only 78% of them. For $M2\_4$ the results are even more impressive since re-routing the demand was possible in 99.5% of the simulations. These results are summarized in the Online Supplement.

On the other hand, the results of failure scenario FS2 show that models $M2\_\lambda$ are not only robust under inter-hub edge failures, but also under failures of the hub nodes. On the contrary, $M0$ and $M1$ have a weaker behaviour under these scenarios, as, in average, the demand could not be routed in nearly 20% of the simulations. This highlights the performance of $M2\_2$, $M2\_3$, and $M2\_4$, for which the percentage of simulations where demand could not be re-routed decreases to 12%, 5%, and 2%, respectively.

Finally, we have defined a function that allows us to estimate the extra routing cost in the *after-failure* network. Specifically, for each simulation, when the demand can be routed in the *after-failure* network, we compute the overall *a posteriori* routing cost $R^F$. We denote by $\tau_F \in [0,1]$ the proportion of simulations for which this cost can be computed. In case a commodity $r \in R$ cannot be routed through the *after-failure* backbone network, we assume that its routing cost is proportional to the *cost of the direct connection* $\bar{c}_{o(r)d(r)}$, i.e. the overall routing cost is $(1+q)\sum_{r\in R} w_r c_{o(r)d(r)}$. The parameter $q > 0$ represents the *extra percent cost* (over the cost of the direct connection) for re-routing a commodity in case the backbone network can no longer be used to route it. Such a cost may represent the outsourcing cost of a direct delivery between the origin and destination of the commodity or the opportunity loss of an unsatisfied user for which the service could not be provided. With this information, we compute the average design and routing cost for the network as:

$$\Phi(q) = \sum_{h\in H} f_h + \sum_{e\in E_H} h_e + \tau_F R^F + (1-\tau_F)(1+q)\sum_{r\in R} w_r c_{o(r)d(r)}.$$

We summarize in Figure 4 the average behavior of the above cost for all the simulations and all the failure scenarios. Each line represents $\Phi(q)$, as a function of the parameter $q$, for each of the *after-failure* networks produced by the simulations constructed with the five different models (M0, M1, M2_2, M2_3, and M2_4). One can observe that for small values of $q$ (the re-routing costs are a small fraction of the direct routing costs) $M0$ is more convenient. This is clear, since when re-routing costs are not very high, one may undertake such costs, even if failures occur very often. As $q$ increases, the most convenient models are M1, M2_2, and M2_3 (in this order). M2_4 is clearly

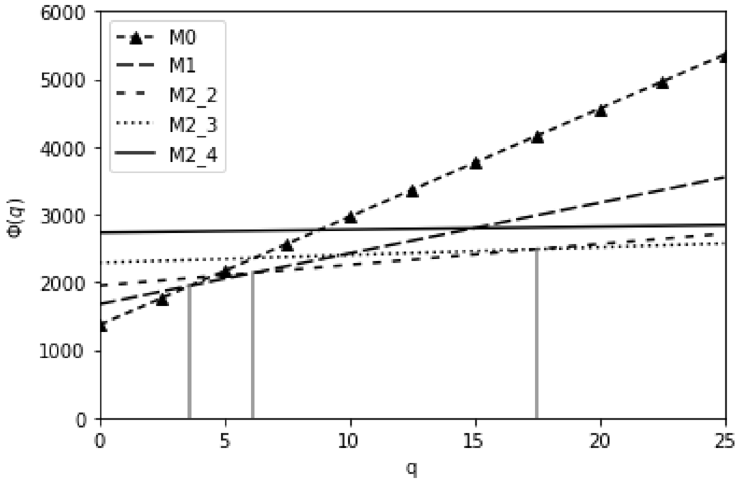



**Figure 4** **Routing costs + fixed costs on the after-failure network as a function of the parameter $q$.**

the most robust one, since the parameter $q$ almost does not affect its routing cost (in this case the percentage of simulations for which the commodities cannot be routed is really small), but its design costs are very high.

Furthermore, it can be observed that, when results are disaggregated by failure scenario (FS1, FS2, FS3, and FS4), the behavior of $\Phi(q)$ is different for the failure scenario FS2, where we simulate failures in hub nodes, from the other failure scenarios. In FS2, models M0, M2_3, and M2_4 outperform M1 and M2_2, in average, whereas in the remaining failure scenarios M0, M1, and M2_2 are more convenient for reasonable values of $q$. This information is shown in the figures provided in the Online Supplement.

We conclude this section by highlighting that the study that we have carried out allows the decision maker to determine the best model to construct the backbone network based on the expected extra cost that should be paid for not providing the service to commodities due to failures in the network.

## 8. Conclusions and further research

In this paper we have developed several models to design hub networks that are robust under link failures. All models ensure that, for each commodity, an additional routing path exists besides its original routing path, which can be used in case of failure of the inter-hub arc of the original path. The first model builds explicitly the backup path for each commodity, whereas the second model ensures that backup paths exist by imposing the $\lambda$-connectivity of the backbone network for a given value of $\lambda \geq 2$. The two models present advantages from the point of view of the robustness of the backbone network. One the one hand, the first model guarantees that backup paths for the commodities are of the same nature than the original ones, although the computational effort to obtain solutions is high. On the other hand, the second model ensures that backup paths exist, with lower computational requirements.

Both models have been computationally tested on an extensive battery of experiments with instances based on three well-known HL benchmarks, namely AP, CAB and TR. Furthermore, we have analyzed the robustness of the models by simulating failures corresponding to different types of failure scenarios on the TR network, assessing their applicability. Some conclusions can be derived from this study. In particular, we observe that our models are very useful for the design of hub networks exposed to link failures, since they produce backbone networks that allow re-routing the demand even when failures occur. As expected, the design costs for obtaining robust hub networks, which guarantee that commodities can be re-routed if inter-hub edges fail, considerably increase with respect to those of models that do not consider potential failures of inter-hub edges. Our simulation study indicates that, even when re-routing costs are small, the robust networks produced by our models allow reducing the overall cost for building and using the hub infrastructure.

Future research on the topic includes the study of valid inequalities for both models in order to alleviate the computational effort for their exact solution. We observed in the results of our computational experiments, that for model (HLPIHLF-1BP), there are already instances with $n = 25$ nodes for which the the Time Limit was reached (even some of them flagged Out of Memory). For model (HLPIHLF-$\lambda$), for $n = 50$ the time limit was also reached, with MIP Gaps close to 50%. In our opinion, with a suitable choice of the heuristic approach, we think it would be possible to find better solutions (smaller MIPGap) in much smaller computing times.

Another topic of research related to this work is the incorporation of the robust strategies that we use here to other network design problems as min-cost spanning trees or min-cost flow networks, where failures in the edges of the network may cause a break in the system.

## Acknowledgments

The authors of this research acknowledge financial support by the Spanish Ministerio de Ciencia y Tecnología, Agencia Estatal de Investigación and Fondos Europeos de Desarrollo Regional (FEDER) via projects PID2020-114594GB-C21 and MTM2019-105824GB-I00. The authors also acknowledge partial support from projects FEDER-US-1256951, Junta de Andalucía P18-FR-422, P18-FR-2369, B-FQM-322-UGR20 (COXMOS), and NetmeetData: Ayudas Fundación BBVA a equipos de investigación científica 2019. The first author was also partially supported by the IMAG-Maria de Maeztu grant CEX2020-001105-M /AEI /10.13039/501100011033 and UE-NextGenerationEU (ayudas de movilidad para la recualificación del profesorado universitario).

## References

Altner DS, Ergun Ö, Uhan NA (2010) The maximum flow network interdiction problem: valid inequalities, integrality gaps, and approximability. *Operations Research Letters* 38(1):33–38.

Alumur S, Nickel S, da Gama FS (2012) Hub location under uncertainty. *Transportation Research Part B: Methodological* 46(4):529–543.

Alumur SA, Kara BY, Karasan OE (2009) The design of single allocation incomplete hub networks. *Transportation Research Part B: Methodological* 43(10):936–951.

An Y, Zhang Y, Zeng B (2015) The reliable hub-and-spoke design problem: Models and algorithms. *Transportation Research Part B: Methodological* 77:103–122.

Andreas AK, Smith JC (2008) Mathematical programming algorithms for two-path routing problems with reliability considerations. *INFORMS Journal on Computing* 20(4):553–564.

Aneja YP, Chandrasekaran R, Nair K (2001) Maximizing residual flow under an arc destruction. *Networks: An International Journal* 38(4):194–198.

Aráoz J, Fernández E, Franquesa C (2009) The clustered prize-collecting arc routing problem. *Transportation Science* 43:287–300.

Belenguer JM, Benavent E (1998) The capacitated arc routing problem: Valid inequalities and facets. *Computational Optimization and Applications* 10(2):165–187.

Bertsimas D, Sim M (2004) The price of robustness. *Operations Research* 52(1):35–53.

Calık H, Alumur SA, Kara BY, Karasan OE (2009) A tabu-search based heuristic for the hub covering problem over incomplete hub networks. *Computers & Operations Research* 36(12):3088–3096.

Campbell J, O'Kelly M (2012) Twenty-five years of hub location research. *Transportation Science* 46(2):153–169.

Campbell JF, Ernst AT, Krishnamoorthy M (2005) Hub arc location problems: part i—introduction and results. *Management Science* 51(10):1540–1555.

Cánovas L, García S, Marín A (2007) Solving the uncapacitated multiple allocation hub location problem by means of a dual-ascent technique. *European Journal of Operational Research* 179(3):990–1007.

Çetiner S, Sepil C, Süral H (2010) Hubbing and routing in postal delivery systems. *Annals of Operations research* 181(1):109–124.

Contreras I, Cordeau JF, , Laporte G (2011) Stochastic uncapacitated hub location. *European Journal of Operational Research* 212(3):518–528.

Contreras I, O'Kelly M (2019) Hub location problems. Laporte G, Nickel S, Saldanha da Gama F, eds., *Location Science*, chapter 8, 177–205 (Springer), 2 edition.

Cormican KJ, Morton DP, Wood RK (1998) Stochastic network interdiction. *Operations Research* 46(2):184–197.

Cui T, Ouyang Y, Shen ZJM (2010) Reliable facility location design under the risk of disruptions. *Operations Research* 58(4):998–1011.

de Camargo RS, Miranda Jr G, Ferreira RPM, Luna H (2009) Multiple allocation hub-and-spoke network design under hub congestion. *Computers & Operations Research* 36(12):3097–3106.

Erdoğan G, Battarra M, Rodríguez-Chía AM (2022) The hub location and pricing problem. *European Journal of Operational Research* 301(3):1035–1047.

Ernst AT, Krishnamoorthy M (1996) Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science* 4(3):139 – 154.

Farahani RZ, Hekmatfar M, Arabani AB, Nikbakhsh E (2013) Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering* 64(4):1096–1109.

Gusfield D (1993) Very simple methods for all pairs network flow analysis. *SIAM Journal on Applied Mathematics* 19(1):143–555.

Huang X, Tian YP (2017) Localization in sensor networks with communication delays and package losses. *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, 3974–3979 (IEEE).

Kim H, O'Kelly ME (2009) Reliable p-hub location problems in telecommunication networks. *Geographical Analysis* 41(3):283–306.

Korani E, Eydi A (2021) Bi-level programming model and KKT penalty function solution approach for reliable hub location problem. *Expert Systems with Applications* 184:115505.

Li Y, Li X, Shu J, Song M, Zhang K (2022) A general model and efficient algorithms for reliable facility location problem under uncertain disruptions. *INFORMS Journal on Computing* 34(1):407–426.

Ma J, Pajouh FM, Balasundaram B, Boginski V (2016) The minimum spanning k-core problem with bounded cvar under probabilistic edge failures. *INFORMS Journal on Computing* 28(2):295–307.

Marín A, Cánovas L, Landete M (2006) New formulations for the uncapacitated multiple allocation hub location problem. *European Journal of Operational Research* 172(1):274–292.

Mohammadi M, Jula P, Tavakkoli-Moghaddam R (2019) Reliable single-allocation hub location problem with disruptions. *Transportation Research Part E: Logistics and Transportation Review* 123:90–120.

O'Kelly ME (1987) A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research* 32(3):393–404.

O'Kelly ME (1992) Hub facility location with fixed costs. *Papers in Regional Science* 71(3):293–306, ISSN 1435-5957, URL `http://dx.doi.org/10.1007/bf01434269`.

Oliveira FA, de Sá EM, de Souza SR (2022) Benders decomposition applied to profit maximizing hub location problem with incomplete hub network. *Computers & Operations Research* 142:105715.

Padberg M, Grötschel M (1985) Polyhedral computations. *The Traveling Saliesman Prolbem: A Guided Tour of Combinatorial Optimization*, 307–360 (Wiley).

Parvaresh F, Hashemi Golpayegany SA, Moattar Husseini SM, Karimi B (2013) Solving the p-hub median problem under intentional disruptions using simulated annealing. *Networks and Spatial Economics* 13(4):445–470, URL `http://dx.doi.org/10.1007/s11067-013-9189-3`.

Rodríguez-Pereira J, Fernández E, Laporte G, Benavent E, Martínez-Sykora A (2019) The Steiner traveling salesman problem and its extensions. *European Journal of Operational Research* 278:615–628.

Rostami B, Kämmerling N, Buchheim C, Clausen U (2019) Reliable single allocation hub location problem under hub breakdowns. *Computers & Operations Research* 96:15–29.

Royset JO, Wood RK (2007) Solving the bi-objective maximum-flow network-interdiction problem. *INFORMS Journal on Computing* 19(2):175–184.

Snyder LV, Daskin MS (2005) Reliability models for facility location: the expected failure cost case. *Transportation Science* 39(3):400–416.

Tan PZ, Kara BY (2007) A hub covering model for cargo delivery systems. *Networks: An International Journal* 49(1):28–39.

Wei N, Walteros JL, Worden MR, Ortiz-Peña HJ (2021) A resiliency analysis of information distribution policies over mobile ad hoc networks. *Optimization Letters* 15(4):1081–1103.

Wood RK (1993) Deterministic network interdiction. *Mathematical and Computer Modelling* 17(2):1–18.

Zeng B, An Y, Zhang Y, Kim H (2010) A reliable hub-spoke model in transportation systems. *Proceedings of the 4th international symposium on transportation network reliability, Minneapolis, Minnesota, USA*, 22–23.

Zetina C, Contreras I, Cordeau JF, Nikbakhsh E (2017) Robust uncapacitated hub location. *Transportation Research Part B: Methodological* 106:393–410.

Zheng Y, Ji Y, Shen Y, Liu B, Du Y (2022) Hub location problem considering spoke links with incentive-dependent capacities. *Computers & Operations Research* 148:105959.