

# A GRASP/VND Heuristic for the Generalized Steiner Problem with Node-Connectivity Constraints and Hostile Reliability

Sebastián Laborde, Franco Robledo,  
Pablo Romero, and Omar Viera

Instituto de Computación, INCO  
Facultad de Ingeniería - Universidad de la República  
Montevideo Uruguay  
(nlaborde, frobledo, promero, viera)@fing.edu.uy

**Abstract.** The object under study is a combinatorial optimization problem motivated by the topological network design of communication systems, meeting reliability constraints. Specifically, we introduce the Generalized Steiner Problem with Node-Connectivity Constraints and Hostile Reliability, or GSPNCHR for short. Since the GSPNCHR belongs to the class of  $\mathcal{NP}$ -Hard problems, approximative algorithms are adequate for medium and large-sized networks. As a consequence, we develop a GRASP/VND methodology. The VND includes three local searches, that replace special elementary paths or trees, preserving feasibility. Our goal is to find a minimum-cost solution, meeting a reliability threshold, where both nodes and links may fail with given probabilities. We adapted TSPLIB benchmark in order to highlight the effectiveness of our proposal. The results suggest that our heuristic is cost-effective, providing highly-reliable networks.

**Keywords:** Combinatorial Optimization Problem, Computational Complexity, Network Reliability, GSPNCHR, GRASP, VND.

## 1 Motivation

Currently, the backbone of the Internet infrastructure is supported by fiber-optics communication. Fiber-To-The-Home (FTTH) services have a large penetration throughout the world, and provides high data rates to the final customers. However, there are several shortcomings that should be addressed urgently. The physical design of FTTH is not suitable for large-scale natural disasters and/or malicious attacks [16]. The monitoring and detection of failures are sometimes slow, and a service disruption of hours is extremely harmful for business models. Furthermore, FTTH services are suffering the capacity crunch problem, and elastic optical networks (EON) combined with smart traffic engineering and additional redundancy is currently in order.

Consequently, a smart augmentation of the physical network is mandatory. Since the deployment of fiber-optics is an important economical investment,

the topological network design of FTTH networks should be revisited. The goal is to interconnect distinguished nodes, called terminals, using large level of redundancy, and simultaneously, meeting large reliable constraints.

Reliability analysis deals with probabilistic failures on the components of a system. The reliability is precisely the probability of correct operation of the whole system, subject to random failures. Here, we consider a realistic *hostile model*, where both nodes and links could fail. Our goal is to understand the cost-reliability trade-off, and how the reliability is naturally increased adding levels of redundancy between distinguished terminals. The contributions of this paper can be summarized in the following items:

1. The Generalized Steiner Problem with Node-Connectivity Constraints and Hostile Reliability (GSPNCHR), is introduced.
2. We formally prove that the GSPNCHR belongs to the  $\mathcal{NP}$ -Hard class.
3. As a consequence, a GRASP/VND methodology is proposed.
4. Our results highlight that the model is robust under non-terminal node-failures, rather than link-failures.

The document is organized in the following manner. The related work is presented in Section 2. A formal description for the GSPNCHR is presented in Section 3; its  $\mathcal{NP}$ -Hardness is also established. A GRASP/VND solution is introduced in Section 4. Numerical results are presented in Section 5. Section 6 contains concluding remarks and trends for future work.

## 2 Related Work

Here, we extend the Generalized Steiner Problem (GSP), adding node-connectivity requirements and a hostile network reliability model with probabilistic failures on its components. We cover the main body of related works in the fields of network reliability analysis, topological network design and joint problems from the scientific literature.

Scarce works jointly deal with a topological network optimization under reliability constraints. Javiera Barrera et. al. proposed a topological network optimization, trying to minimize costs subject to  $K$ -terminal reliability constraints [3]. The authors consider Sample Average Approximation (SAA) method, which is a powerful tool for stochastic optimization and for the resolution of  $\mathcal{NP}$ -Hard combinatorial problems with a target probability maximization [13]. They conclude that suboptimal solutions could be found if dependent failures are ignored in the model. The scientific literature also offers topological optimization problems meeting reliability constraints, or reliability maximization under budget constraints, which is known as network synthesis. The reader can find a survey on the synthesis in network reliability in [4]. More recent works propose a reliability optimization in general stochastic binary systems [7], even under the introduction of Sample Average Approximation [20]. Building uniformly most-reliable graphs is an active and challenging research field, where the goal is to find graphs with fixed nodes and

links with maximum reliability evaluation in a uniform sense. The interested reader can consult [2] for conjectures in this field. A close problem to ours is to consider topological transformations (i.e., moving links or path/tree replacements) in order to increase the reliability measure. This problem is not mature, and a recent work propose a novel reliability-increasing network transformation [5]. There, E. Canale et. al. show that any graph with a cut-point can be transformed into a biconnected graph with greater reliability. Using this remarkable property, our design does not include cut-points.

Most works in the field of network reliability analysis deal with its evaluation rather than its maximization. The literature on network reliability evaluation is abundant, and here we can mention distinguished works on this field. A trade-off between accuracy and computational feasibility is met by simulations, given the hardness of the classical network reliability models [11]. Macroscopically, Monte Carlo methods consider independent replications of a complex system, and by means of statistical laws find pointwise estimations, in order to make decisions on the system. The reader is invited to consult an excellent book on Monte Carlo methods authored by Fishman [9], which was inspirational for network reliability, numerical integration, statistics and other fields of knowledge. In our particular case we deal with the hostile network reliability model, where both links and non-terminal nodes fail independently. Its reliability evaluation belongs to the class of  $\mathcal{NP}$ -Hard problems as well [11]. Recursive Variance Reduction (RVR) is an outstanding technique for the reliability estimation [8]. This formulation allows a meaningful variance reduction, and the product between time and variance is also reduced when compared to Crude Monte Carlo. Furthermore, the variance is mathematically proved to be always better in RVR than in CMC. More recently, the applicability of RVR is extended to Stochastic Monotone Binary Systems [6]. Since the hostile model is an SMBS, in this work we consider RVR for the network reliability estimation.

In the Generalized Steiner Problem (GSP), the goal is to communicate a given subset of terminal-nodes at the minimum cost, meeting connectivity requirements either by link-disjoint (GSP-EC) or node-disjoint (GSP-NC) paths. Since the problem is  $\mathcal{NP}$ -Hard, the literature offers approximation algorithms as well as metaheuristics. Agrawal, Klein and Ravi [1] developed an approximation algorithm with logarithmic factor for the GSP-EC. Jain [12] presented a factor-2 approximation algorithm for the GSP-EC, using the primal-dual schema for linear programming. A deep inapproximability result for the GSP-NC without Steiner nodes was introduced by Kortsarz in [14]. In [24] an enumeration of optimal solutions for the GSP is carried out with a compact data structure, called Zero-Suppressed Binary Decision Diagrams (ZDD). The authors show that this method works for several real-world instances. Heuristics are also available for the GSP. Sartor and Robledo developed a GRASP methodology to address the GSP-EC [23]. In [19], S. Nasmachnow presents an empirical evaluation of several simple metaheuristics (VNS is included) to address the GSP, with promising results. Several implementations of VNS have been developed for the particular Traveling

Salesman Problem as well, showing that VNS is competitive. Exact solutions for the TSP and extensions can also be found in [22]. The CPU-times provided by the exact solutions are longer than the heuristics, and in particular VNS proposal for the TSP.

### 3 Problem and Complexity

In this section, we first present a general description of the GSPNCHR. Then, a formal combinatorial optimization problem is introduced, and the hardness is established.

**Definition 1 (GSPNCHR).** *Consider a simple undirected graph  $G = (V, E)$ , terminal-set  $T \subseteq V$ , link-costs  $\{c_{i,j}\}_{(i,j) \in E}$  and connectivity requirements  $R = \{r_{i,j}\}_{i,j \in T}$ . Further, we assume that both links and non-terminal (Steiner) nodes fail with respective probabilities  $P_E = \{p_e\}_{e \in E}$  and  $P_{V-T} = \{p_v\}_{v \in V-T}$ . Given a reliability threshold  $p_{min}$ , the goal is to build a minimum-cost topology  $G_S \subseteq G$  meeting both the connectivity requirements  $R$  and the reliability threshold:  $R_K(G_S) \geq p_{min}$ , being  $K = T$  the terminal-set.*

Recall that the  $K$ -Terminal reliability is the probability that all the terminals belong to the same component, after node and link failures. The exact computation of the reliability  $R_K(G)$  is  $\mathcal{NP}$ -Hard [11]. Consider an instance  $(G, C, R, T, P_E, P_{V-T}, p_{min})$  of the GSPNCHR, and the following decision variables:

$$y_{(i,j)}^{u,v} = \begin{cases} 1 & \text{if } (i,j) \in E \text{ is used in a path } u - i - j - v \\ 0 & \text{otherwise} \end{cases}$$

$$x_{(i,j)} = \begin{cases} 1 & \text{if } (i,j) \in E \text{ is used in the solution} \\ 0 & \text{otherwise} \end{cases}$$

$$\hat{x}_i = \begin{cases} 1 & \text{if the Steiner node } i \in V - T \text{ is used in the solution} \\ 0 & \text{otherwise} \end{cases}$$

Here, we introduce the GSPNCHR as the following combinatorial optimization problem:

$$\begin{aligned}
& \min \sum_{(i,j) \in E} c_{i,j} x_{i,j} \\
& \text{s.t. } x_{ij} \geq y_{(i,j)}^{u,v} + y_{(j,i)}^{u,v} \quad \forall (i,j) \in E, \forall u,v \in T, u \neq v \quad (1) \\
& \sum_{(u,i) \in E} y_{(u,i)}^{u,v} \geq r_{u,v} \quad \forall u,v \in T, u \neq v \quad (2) \\
& \sum_{(j,v) \in E} y_{(j,v)}^{u,v} \geq r_{u,v} \quad \forall u,v \in T, u \neq v \quad (3) \\
& \sum_{(i,p) \in I(p)} y_{(i,p)}^{u,v} - \sum_{(p,j) \in I(p)} y_{(p,j)}^{u,v} \geq 0, \quad \forall p \in V - \{u,v\}, \forall u,v \in T, u \neq v \quad (4) \\
& \sum_{(s,i) \in E} x_{s,i} \leq M \hat{x}_s, \quad \forall s \in V - T \quad (5) \\
& R_K(G_S) \geq p_{min} \quad (6) \\
& x_{(i,j)} \in \{0,1\} \quad \forall (i,j) \in E \quad (7) \\
& \hat{x}_i \in \{0,1\} \quad \forall i \in V - T \quad (8) \\
& y_{(i,j)}^{u,v} \in \{0,1\} \quad \forall (i,j) \in E, \forall u,v \in T, u \neq v \quad (9)
\end{aligned}$$

The goal is to minimize the global cost of the solution. The set of Constraints 1 state that links are one-way. The connectivity requirements are expressed by means of Constraints 2 and 3. Constraints 4 represent Kirchhoff law, or flow conservation. Constraints 5 state that an incident link to a Steiner node can be used only if the Steiner node is considered in the solution. Observe that  $M$  is a large real number;  $M = |E|$  can be used in the model without loss of generality. The minimum reliability threshold is established with Constraint 6, being  $G_S \subseteq G$  the subgraph with all the constructed links  $x_{i,j}$ . Finally, the set of constraints 7-9 state that all the decision variables belong to the binary set  $\{0,1\}$ . Now, we establish the hardness for the GSPNCHR.

**Theorem 1.** *The GSPNCHR belongs to the class of  $\mathcal{NP}$ -Hard problems.*

*Proof.* By inclusion. Recall that Hamilton Tour belongs to Karp list of  $\mathcal{NP}$ -Complete problems [11]. Consider a simple graph  $G = (V, E)$ . Consider the *trivial instance*  $(G, C, R, T, P_E, P_{V-T}, p_{min})$  with unit costs, perfect nodes/links, no Steiner nodes and requirements  $r_{i,j} = 2$  for all  $i, j \in V$ . The cost is not greater than  $n = |V|$  if and only if  $G$  has a Hamilton tour.

The GSPECHR is also  $\mathcal{NP}$ -Hard; the proof is analogous. Theorem 1 can be strengthened considering strong inapproximability results [10].

In order to tackle the GSPNCHR, first we provide full solution for the relaxed version without the reliability threshold, this is, without Constraint 6. Then, we count the number of feasible solutions that meet that constraint. We use this approach, since we want to determine if the topological robustness has an impact in the resulting network reliability.

## 4 Solution

GRASP and VND are well known metaheuristics that have been successfully used to solve many hard combinatorial optimization problems. GRASP is a powerful multi-start process which operates in two phases [21]. A feasible solution is built in a first phase, whose neighborhood is then explored in the Local Search Phase [21]. The second phase is usually enriched by means of different variable neighborhood structures. For instance, VND explores several neighborhood structures in a deterministic order. Its success is based on the simple fact that different neighborhood structures do not usually have the same local minimum. Thus, the resulting solution is simultaneously a locally optimum solution under all the neighborhood structures. The reader is invited to consult the comprehensive Handbook of Heuristics for further information [17]. Here, we develop a GRASP/VND methodology.

### 4.1 NetworkDesign

A pseudocode of our full proposal is presented in Figure 1. It receives the ground graph  $G_B$ , a number of iterations  $iter$  and a positive integer  $k$  to find the  $k$  shortest paths during the Construction Phase, a reliability threshold  $p_{min}$ , the elementary reliabilities  $P_E$ ,  $P_{V-T}$  and number of iterations  $simiter$  during the simulations carried out in the Reliability Phase. If the resulting solution respects Constraint 6, it is included in the set  $sol$ , that is returned in Line 10. Observe that RVR method is considered in order to test this reliability constraint [8]. Our goal is to determine how many solutions for the relaxed problem meet Constraint 6, as a function of the robustness (connectivity matrix  $R$ , elementary reliabilities and threshold  $p_{min}$ ).

---

**Algorithm 1**  $sol = NetworkDesign(G_B, iter, k, p_{min}, P_E, P_{V-T}, simiter)$

---

```
1:  $i \leftarrow 0$ ;  $P \leftarrow \emptyset$ ;  $sol \leftarrow \emptyset$ 
2: while  $i < iter$  do
3:    $\bar{g} \leftarrow Construction(G_B, P, k)$ 
4:    $\bar{g} \leftarrow VND(\bar{g}, P)$ 
5:    $reliability \leftarrow RVR(g_{sol}, P_E, P_{V-T}, simiter)$ 
6:   if  $reliability > p_{min}$  then
7:      $sol \leftarrow sol \cup \{g_{sol}\}$ 
8:   end if
9: end while
10: return  $sol$ 
```

---

**Fig. 1.** Pseudocode for the main algorithm: *NetworkDesign*.

## 4.2 Construction Phase

This algorithm trades simplicity and effectiveness, building paths iteratively. Figure 2 receives the ground graph  $G_B$ , the matrix with link-costs  $C$ , the connectivity matrix  $R$ , and the parameter  $k$ . Denote  $S_D^{(I)}$  the set of terminal nodes, following the terminology of the backbone design from Wide Area Networks. In Line 1, the solution  $g_{sol}$  is initialized only with the terminal nodes  $S_D^I$  without links,  $M = \{m_{i,j}\}_{i,j \in T}$  stores the unsatisfied requirements, so initially  $m_{i,j} = r_i$ , for all  $i, j \in S_D^{(I)}$ , and the matrix  $P = \{P_{i,j}\}_{i,j \in S_D^{(I)}}$  that represents the collection of node-disjoint paths is empty for all  $P_{i,j}$ . Additionally, the matrix  $A = \{A_{i,j}\}_{i,j \in S_D^{(I)}}$  that controls the number of attempts that the algorithm fails to find  $r_{i,j}$  node-disjoint paths between  $i, j$  is initialized correspondingly:  $A_{i,j} = 0 \forall i, j \in S_D^{(I)}$ .

---

**Algorithm 2**  $(sol, P) = Construction(G_B, C, R, k)$

---

```

1:  $g_{sol} \leftarrow (S_D^{(I)}, \emptyset)$ ;  $m_{i,j} \leftarrow r_{i,j}$ ;  $P_{i,j} \leftarrow \emptyset, \forall i, j \in S_D^{(I)}$ ;  $A_{i,j} \leftarrow 0, \forall i, j \in S_D^{(I)}$ 
2: while  $\exists m_{i,j} > 0 : A_{i,j} < MAX\_ATTEMPTS$  do
3:    $(i, j) \leftarrow ChooseRandom(S_D^{(I)} : m_{i,j} > 0)$ 
4:    $\bar{G} \leftarrow G_B \setminus P_{i,j}$ 
5:   for all  $(u, v) \in E(\bar{G})$  do
6:      $\bar{c}_{u,v} \leftarrow c_{u,v} \times 1_{\{(u,v) \notin g_{sol}\}}$ 
7:   end for
8:    $L_p \leftarrow KSP(k, i, j, \bar{G}, \bar{C})$ 
9:   if  $L_p = \emptyset$  then
10:     $A_{i,j} \leftarrow A_{i,j} + 1$ ;  $P_{i,j} \leftarrow \emptyset$ ;  $m_{i,j} \leftarrow r_{i,j}$ 
11:   else
12:     $p \leftarrow Select\_Random(L_p)$ ;  $g_{sol} \leftarrow g_{sol} \cup \{p\}$ 
13:     $P_{i,j} \leftarrow P_{i,j} \cup \{p\}$ ;  $m_{i,j} \leftarrow m_{i,j} - 1$ 
14:     $(P, M) \leftarrow General\_Update\_Matrix(g_{sol}, P, M, p, i, j)$ 
15:   end if
16: end while
17: return  $(g_{sol}, P)$ 

```

---

**Fig. 2.** Pseudocode for the Construction Phase.

The purpose of the *while-loop* (Lines 2-13) is to fulfill all the connectivity requirements in a randomized fashion. Observe that we selected a large Restricted Candidate List (RCL) in our GRASP proposal for diversification purposes. A pair of terminals  $(i, j)$  is uniformly picked at random from the set  $S_D^{(I)}$ , provided that  $m_{i,j} > 0$  (Line 3). The graph  $\bar{G}$  defined in Line 4 discards

the nodes that were already visited in the previous paths. Therefore, if we find some path between  $i$  and  $j$  in  $\overline{G}$ , it will be included. In the *for-loop* of Lines 5-7, an auxiliary matrix with the costs  $\overline{C} = \overline{c}_{i,j}$  allows to use already existent links from  $g_{sol}$  without additional cost, and add them to build a new node-disjoint path. The  $k$ -Shortest Paths from  $i$  to  $j$  are computed in Line 8 using Yen algorithm [18], that finds the  $k$ -Shortest Paths between two fixed nodes in a graph. In Line 9, we test if the list  $L_p$  is empty. In this case we re-initialize  $P_{i,j}$ ,  $m_{i,j}$ , and add a unit to  $A_{i,j}$ , since  $i$  and  $j$  belong to different connected components. If the list  $L_p$  is not empty, a path  $p$  is uniformly picked from the list  $L_p$ , and it is included in the solution (Line 12). The path  $p$  is added to  $P_{i,j}$ , and the requirement  $m_{i,j}$  is decreased a unit (Line 13). The addition of the path  $p$  could build node-disjoint paths from different terminals. Consequently, the function *General\_Update\_Matrix* finds these new paths. *Construction* returns a feasible solution  $g_{sol}$  equipped with all the sets  $P = \{P_{i,j}\}_{i,j \in S_D^{(T)}}$  of node-disjoint pairs between the different terminals (Line 17). The reader can observe that *Construction* returns a feasible solution for the GSPNC, which is the relaxed version of the GSPNCHR.

### 4.3 Local Search Phase - VND

The goal is to combine a rich diversity of neighborhoods in order to obtain an output that is locally optimum solution for every feasible neighborhood. Here, we consider three neighborhood structures to build a VND [17]. First, the concept of key-nodes, key-paths and key-trees are in order:

**Definition 2 (key-node).** *A key-node in a feasible solution  $v \in g_{sol}$  is a Steiner (non-terminal) node with degree three or greater.*

**Definition 3 (key-path).** *A key-path in a feasible solution  $p \subseteq g_{sol}$  is an elementary path where all the intermediate nodes are non-terminal with degree 2 in  $g_{sol}$ , and the extremes are either terminals or key-nodes.*

A feasible solution  $g_{sol}$  accepts a decomposition into key-paths:  $K_{g_{sol}} = \{p_1, \dots, p_h\}$ .

**Definition 4 (Key-tree).** *Let  $v \in g_{sol}$  be a key-node belonging to a feasible solution  $g_{sol}$ . The key-tree associated to  $v$ , denoted by  $T_v$ , is the tree composed by all the key-paths that meet in the common end-point (i.e., the key-node  $v$ ).*

Now, we are in conditions to define three neighborhood structures that combine the previous concepts. Consider a feasible solution  $g_{sol}$  for the GSPNC.

**Definition 5 (Neighborhood Structure for key-paths).** *Given a key-path  $p \in g_{sol}$ , a neighbor-solution is  $\hat{g}_{sol} = \{g_{sol} \setminus p\} \cup \{\hat{p}\}$ , where  $\hat{p}$  is other path that connects the extremes from  $p$ . The neighborhood of key-paths from  $g_{sol}$  is composed by the previous operation to the possible members belonging to  $K_{g_{sol}}$ .*

**Definition 6 (Neighborhood Structure for key-tree).** Consider the key-tree  $T_v \in g_{sol}$  rooted at the key-node  $v$ . A neighbor of  $g_{sol}$  is  $\hat{g}_{sol} = \{g_{sol} \setminus T_v\} \cup \{T\}$ , being  $T$  another tree that replaces  $T_v$  with identical leaf-nodes.

**Definition 7 (Neighborhood Structure for key-path replacement).** Given a key-path  $p \subseteq g_{sol}$ , a neighbor solution for  $g_{sol}$  is  $\hat{g}_{sol} = \{g_{sol} \setminus p\} \cup \{m\}$ , being  $m$  the set of nodes and links that will be added to preserve the feasibility of  $\hat{g}_{sol}$ .

Our full algorithm *NetworkDesign* considers a classical VND implementation, calling the three respective local searches in order, after the *Construction* phase:

1. *KeyPathLocalSearch*
2. *KeyTreeLocalSearch*
3. *SwapKeyPathLocalSearch*

This order was selected by computational reasons: the last one is the most demanding in terms of CPU, while the former is the most simple. The respective pseudocodes for the different local searches are presented in Figures 3-5. It is worth to remark that these local searches take effect only if the resulting solution is both feasible and cheaper than the original one. The respective codes from each local search are self-explanatory, and strictly follow the corresponding neighborhood structures, trying to find better replacements. For completeness, two auxiliary functions called during these searches are here explained, in terms of inputs and outputs. We invite the reader to consult [15] for implementation details:

- *General\_RecConnect*: receives the ground graph  $G_B$ , cost-matrix  $C$ , current solution  $g_{sol}$  and a key-node  $v$ . It tries to replace the key-tree  $T_v$  with a better key-tree  $T$  spanning the same leaf-nodes, preserving feasibility. It returns another solution and a boolean *improve* (if *improve* = 0, an identical solution is returned).
- *FindSubstituteKeyPath*: receives the current solution  $g_{sol}$ , the key-path  $p$  and a matrix  $P$  with the collection of disjoint path between the terminals. It replaces the current path  $p$  by  $\hat{p}$ , exploiting the information given by  $P$  in order to reconstruct a new feasible solution. If this solution is cheaper, it returns *improve* = 1 and the resulting solution (otherwise, an identical solution is returned).

In the last step, RVR is introduced in order to determine if Constraint 6 is met. The reader is invited to consult authoritative works on RVR and cites therein [8].

---

**Algorithm 3**  $g_{sol} = KeyPathLocalSearch(G_B, C, g_{sol})$

---

```

1: improve  $\leftarrow TRUE$ 
2: while improve do
3:   improve  $\leftarrow FALSE$ 
4:    $K(g_{sol}) \leftarrow \{p_1, \dots, p_h\}$  {Key-path decomposition of  $g_{sol}$ }
5:   while not improve and  $\exists$  key-paths not analyzed do
6:      $p \leftarrow (K(g_{sol}))$  {Path not analyzed yet, with extremes  $u$  and  $v$ }
7:      $\hat{\mu} \leftarrow \langle NODES(p) \cup S_D \setminus NODES(g_{sol}) \rangle$  {Induced subgraph  $\hat{\mu}$ }
8:      $\hat{p} \leftarrow Dijkstra(u, v, \hat{\mu})$ 
9:     if  $COST(\hat{p}) < COST(p)$  then
10:       $g_{sol} \leftarrow \{g_{sol} \setminus p\} \cup \{\hat{p}\}$ 
11:      improve  $\leftarrow TRUE$ 
12:    end if
13:  end while
14: end while
15: return  $g_{sol}$ 

```

---

**Fig. 3.** Pseudocode for Local Search 1: *KeyPathLocalSearch*.

---

**Algorithm 4**  $g_{sol} = KeyTreeLocalSearch(G_B, C, g_{sol})$

---

```

1: improve  $\leftarrow TRUE$ 
2: while improve do
3:   improve  $\leftarrow FALSE$ 
4:    $X \leftarrow KeyNodes(g_{sol})$  {Key-nodes from  $g_{sol}$ }
5:    $\bar{S} \leftarrow S_D \setminus NODES(g_{sol})$ 
6:   while not improve and  $\exists$  key-nodes not analyzed do
7:      $v \leftarrow X$  {Key-node not analyzed yet}
8:      $[g_{sol}, improve] \leftarrow General\_RecConnect(G_B, C, g_{sol}, v, \bar{S})$ 
9:   end while
10: end while
11: return  $g_{sol}$ 

```

---

**Fig. 4.** Pseudocode for Local Search 2: *KeyTreeLocalSearch*.

---

**Algorithm 5**  $g_{sol} = \text{SwapKeyPathLocalSearch}(G_B, C, g_{sol}, P)$

---

```

1:  $improve \leftarrow TRUE$ 
2: while  $improve$  do
3:    $improve \leftarrow FALSE$ 
4:    $K(g_{sol}) \leftarrow \{p_1, \dots, p_h\}$  {Key-path decomposition of  $g_{sol}$ }
5:   while not  $improve$  and  $\exists$  key-paths not analyzed do
6:      $p \leftarrow (K(g_{sol}))$  {Path not analyzed yet}
7:      $(g_{sol}, improve) \leftarrow \text{FindSubstituteKeyPath}(g_{sol}, p, P)$ 
8:   end while
9: end while
10: return  $g_{sol}$ 

```

---

**Fig. 5.** Pseudocode for Local Search 3: *SwapKeyPathLocalSearch*.

## 5 Numerical Results

In order to understand the effectiveness of this proposal, an extensive computational study was carried out using our main algorithm *NetworkDesign*. The experimental analysis was carried out in a Home-PC (Pentium Core I5, 6GB). Since there are no benchmark for our specific problem we adapted the well-known TSPLIB instances, adding node/link failure probabilities and node connectivity requirements. We selected  $k = 5$  for *Construction*, which showed acceptable results in a training set. In our reliability-centric design, we fixed  $p_{min} = 0.8$ ; lower values make no sense. The elementary reliabilities for both Steiner nodes and links are close to the unit, since we are focused on the design of highly-reliable networks. Specifically, the nine combinations for  $p_v, p_e \in \{0.99, 0.97, 0.95\}$  were considered in different instances, being  $p_v$  and  $p_e$  the elementary reliabilities for Steiner nodes and links  $e = (i, j)$  respectively. The number of iterations for *NetworkDesign* is established in  $iter = 100$ , and the number of iterations for the RVR method is  $10^4$ .

We want to understand the sensibility of the solution to perturbations in the elementary reliabilities. Therefore, different values for the elementary reliabilities for both Steiner nodes and links were used. Table 1 shows the results for each adapted TSPLIB instance. Each column contains, respectively, name of the TSPLIB instance, percentage of terminal nodes (%  $T$ ), relative improvements of *Construction* (% $IC$ ) and *VND* phases (% $IVND$ ), in relation to the cost of the corresponding input graphs, CPU-time per iteration of *NetworkDesign*, reliability estimation  $\bar{R}$  and estimated variance  $\bar{Var}$ . From Table 1, we can appreciate that the cost of the resulting graph after the *Construction* is practically one-half the cost of its input. The improvement of

$VND$  is consistently bounded between 30.55% and 39.45%, according to the instance and its characteristics on the test-set. The minimum threshold  $p_{min} = 0.8$  is widely exceeded in all the instances under study, considering the RVR estimation  $\bar{R}$ . The elementary reliabilities were established in  $p_v = 0.99$  and  $p_e = 0.95$  respectively for nodes and links, and the range of reliabilities is bounded between  $0.8231$  and  $0.967$ , meeting the reliability constraint. The estimated variance  $\overline{Var}$  is reduced in average in all the instances under study. These facts highlight the activity of the  $VND$  phase, the accuracy of RVR and the global effectiveness of our proposal. Furthermore, the CPU times are acceptable, even under large-sized graphs with 400 nodes.

Tables 2 and 3 illustrate the number of feasible solutions obtained when we fix the elementary node-reliability and modify the link-reliabilities, and vice-versa. The suffix  $TXY$  in each instance indicates the percentage  $XY\%$  of terminal nodes. The feasibility is 100% in almost all the instances under study when both  $p_e = p_v = 0.99$ . However, the feasibility is dramatically deteriorated as soon as the link-reliabilities are decreased (see the last column of Table 2). This effect is not pronounced when the node-reliabilities are reduced, as we can appreciate from the last column of Table 3. This fact shows that the system is robust under failures of Steiner nodes.

Table 1: **GRASP/VND Effectiveness**

Instance	% $T$	% $IC$	% $IVND$	$CPU (s)$	$\bar{R}$	$\overline{Var}$
att48	20	99.27	34.61	11.466	0.967	7.608E-07
att48	35	98.6	36.83	29.769	0.943	3.448E-06
att48	50	98.22	37.1	65.904	0.927	5.322E-06
berlin52	20	98.98	30.55	30.605	0.937	3.294E-06
berlin52	35	99.06	33.93	33.433	0.938	3.19E-06
berlin52	50	98.02	33.48	106.945	0.907	6.487E-06
brazil58	20	98.92	31.96	62.377	0.885	6.722E-06
brazil58	35	99.25	39.45	68.891	0.86	8.347E-06
brazil58	50	98.75	35.26	103.553	0.91	7.093E-06
ch150	20	99.76	37.51	222.552	0.8559	1.029E-05
ch150	35	99.72	36.65	546.652	0.88.03	9.033E-05
gr202	20	99.89	32.43	528.162	0.8231	1.224E-05
gr202	35	99.75	34.56	3511.698	0.8414	1.11E-05
gr202	50	99.74	33.36	9505.629	0.8303	1.279E-05
rd400	20	99.94	35.84	88.214	0.8094	14.22E-05
rd400	35	99.94	33.54	504.103	0.8537	11.89E-05
rd400	50	99.93	33.16	980.701	0.8643	11.51E-05

**Table 2.** Feasible solutions with  $R \geq 0.98$ ,  $p_v = 0.99$  fixed and variable link reliability

Instance	$p_e = 0.99$	$p_e = 0.97$	$p_e = 0.95$
att48 T20	100	90	12
att48 T35	100	53	0
att48 T50	100	20	0
berlin52 T20	100	41	0
berlin52 T35	100	50	0
berlin52 T50	100	1	0
brazil58 T20	99	15	0
brazil58 T35	97	0	0
brazil58 T50	100	5	0
ch150 T20	100	0	0
ch150 T35	100	0	0
ch150 T50	100	0	0
gr202 T20	99	0	0
gr202 T35	100	0	0
gr202 T50	100	0	0
rd400 T20	100	0	0
rd400 T35	100	0	0
rd400 T50	100	0	0

**Table 3.** Feasible solutions with  $R \geq 0.98$ ,  $p_e = 0.99$  fixed and variable node-reliability

Instance	$p_v = 0.99$	$p_v = 0.97$	$p_v = 0.95$
tt48 T20	100	100	99
att48 T35	100	98	96
att48 T50	100	100	99
berlin52 T20	100	100	80
berlin52 T35	100	99	93
berlin52 T50	100	100	100
brazil58 T20	99	59	41
brazil58 T35	97	43	9
brazil58 T50	100	99	81
ch150 T20	100	60	20
ch150 T35	100	98	76
ch150 T50	100	100	97
gr202 T20	99	80	30
gr202 T35	100	69	16
gr202 T50	100	100	76
rd400 T20	100	16	2
rd400 T35	100	98	80
rd400 T50	100	100	100

## 6 Conclusions and Trends for Future Work

We studied the topological design of highly reliable networks. Our goal is to combine purely deterministic aspects such as connectivity with probabilistic models coming from network reliability. For that purpose, the Generalized Steiner Problem with Node-Connectivity Constraints and Hostile Reliability (GSPNCHR) is here introduced. The GSPNCHR belongs to the class of  $\mathcal{NP}$ -Hard problems, since it subsumes the Generalized Steiner Problem (GSP). Therefore, exact methods are prohibitive, even for networks with moderate size. A GRASP/VND solution is here developed, which shows to be both flexible and effective. Since the reliability evaluation for the hostile model also belongs to the  $\mathcal{NP}$ -Hard class, we adopted an outstanding pointwise reliability estimation, known as Recursive Variance Reduction (RVR) method. This method is unbiased, accurate and it presents small variance, as the results show. The model is more sensible to link-failures rather than node-failures.

The interplay between topological network design and network reliability is not well understood yet. Some local searches were here proposed, essentially using key-path and key-tree replacements, in order to reduce costs preserving feasibility. A current research line is to introduce reliability-increasing transformations. The development of local searches that increase reliability and reduce costs would enrich the current solution. Another possibility for future work is to enrich the number of local searches and consider probabilistic transitions between them.

## 7 Acknowledgements

This work is partially supported by Project ANII FCE\_1.2019.1\_156693 *Teoría y Construcción de Redes de Máxima Confiabilidad*, MATHAMSUD 19-MATH-03 *Rare events analysis in multi-component systems with dependent components* and STIC-AMSUD ACCON *Algorithms for the capacity crunch problem in optical networks*.

## References

1. Ajit Agrawal, Philip Klein, and R. Ravi. When Trees Collide: An Approximation Algorithm for the Generalized Steiner Problem on Networks. *SIAM Journal on Computing*, 24(3):440–456, 1995.
2. Kassie Archer, Christina Graves, and David Milan. Classes of uniformly most reliable graphs for all-terminal reliability. *Discret. Appl. Math.*, 267:12–29, 2019.
3. J. Barrera, H. Cancela, and E. Moreno. Topological optimization of reliable networks under dependent failures. *Oper. Res. Lett.*, 43(2):132–136, 2015.
4. F.T. Boesch, A. Satyanarayana, and C.L. Suffel. A survey of some network reliability analysis and synthesis results. *Networks*, 54(2):99–107, 2009.
5. E. Canale, F. Robledo, P. Romero, and J. Viera. Building Reliability-Improving Network Transformations. In *Proceedings of the 15th International Conference on the Design of Reliable Communication Networks*, pages 107–113. IEEE, 2019.

6. Eduardo Alberto Canale, Héctor Cancela, Juan Piccini, Franco Robledo, Pablo Romero, Gerardo Rubino, and Pablo Sartor. Recursive Variance Reduction method in Stochastic Monotone Binary Systems. *Proceedings of the 7th International Workshop on Reliable Networks Design and Modeling*, pages 135–141, 2015.
7. H. Cancela, G. Guerberoff, F. Robledo, and P. Romero. Reliability Maximization in Stochastic Binary Systems. In *Proceedings of the 21st Conference on Innovation in Clouds, Internet and Networks and Workshops*, pages 1–7. IEEE, 2018.
8. Héctor Cancela, Mohamed El Khadiri, and Gerardo Rubino. A new simulation method based on the RVR principle for the rare event network reliability problem. *Annals of Operations Research*, 196(1):111–136, 2012.
9. G. Fishman. *Monte Carlo*. Springer Series in Operations Research and Financial Engineering. Springer, 1996.
10. Harold N. Gabow, Michel X. Goemans, and David P. Williamson. An efficient approximation algorithm for the survivable network design problem. *Mathematical Programming*, 82:13–40, 1998.
11. Michael R. Garey and David S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., USA, 1990.
12. Kamal Jain. A Factor 2 Approximation Algorithm for the Generalized Steiner Network Problem. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, FOCS 98, page 448, USA, 1998. IEEE Computer Society.
13. Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de-Mello. The Sample Average Approximation Method for Stochastic Discrete Optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.
14. G. Kortsarz, R. Krauthgamer, and J. Lee. Hardness of Approximation for Vertex-Connectivity Network Design Problems. *SIAM Journal on Computing*, 33(3):704–720, 2004.
15. Sebastián Laborde. Topological Optimization of Fault-Tolerant Networks meeting Reliability Constraints. Master’s thesis, Universidad de la República, 2020.
16. Rafael B.R. Loureno, Gustavo B. Figueiredo, Massimo Tornatore, and Biswanath Mukherjee. Data evacuation from data centers in disaster-affected regions through software-defined satellite networks. *Computer Networks*, 148:88 – 100, 2019.
17. Rafael Mart, Panos M. Pardalos, and Mauricio G. C. Resende. *Handbook of Heuristics*. Springer Publishing Company, Incorporated, 1st edition, 2018.
18. Ernesto Martins and Marta Pascoal. A new implementation of Yen’s ranking loopless paths algorithm. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1(2):121–133, 2003.
19. Sergio Nesmachnow. Evaluating simple metaheuristics for the Generalized Steiner Problem. *Journal of Computer Science and Technology*, 5(4), 2005.
20. J.L. Pulsipher and V.M. Zavala. Measuring and optimizing system reliability: a stochastic programming approach. *TOP*, pages 1 – 20, 2020.
21. Mauricio Resende and Celso Ribeiro. *Optimization by GRASP*. Springer, 2016.
22. Jessica Rodríguez-Pereira, Elena Fernández, Gilbert Laporte, Enrique Benavent, and Antonio Martínez-Sykora. The Steiner Traveling Salesman Problem and its extensions. *European Journal of Operational Research*, 278(2):615 – 628, 2019.
23. Pablo Sartor and Franco Robledo. GRASP Algorithms for the edge-survivable Generalized Steiner Problem. *International Journal of Control and Automation*, 5:27–44, 2012.
24. Hirofumi Suzuki, Masakazu Ishihata, and Shin-ichi Minato. Designing Survivable Networks with Zero-Suppressed Binary Decision Diagrams. In *WALCOM: Algorithms and Computation*, pages 273–285, Cham, 2020. Springer International Publishing.