PEDECIBA Informática
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay

Reporte Técnico RT 15-12

A survivable & reliable network topological design model

Franco Robledo, Omar Viera

2015

A survivable & reliable network topological design model
Franco Robledo, Omar Viera
ISSN 0797-6410
Reporte Técnico RT 15-12
PEDECIBA
Instituto de Computación – Facultad de Ingeniería
Universidad de la República
Montevideo, Uruguay, 2015

A Survivable & Reliable Network Topological Design Model

Franco Robledo Amoza [1] Omar Viera [2]

Operations Research Department
Engineering School-UDELAR
Montevideo, Uruguay

Abstract

This paper is focused on the resolution of a mixed model for the design of large size networks which will be topologically robust regarding its connectivity and reliability. More precisely, we combined the Network Survivability & Network Reliability approaches. The problem of the topological design has been modeled based on the Generalized Steiner Problem with Node-Connectivity Constraints (GSP-NC), which is NP-Hard. Our aim is to heuristically solve the GSP-NC model by designing low cost highly connected topologies and to measure the reliability of such solutions with respect to a certain prefixed lower threshold. We introduce a Greedy Randomized algorithm for the construction of feasible solutions for the GSP-NC and a local search algorithm based on the Variable Neighbourhood Search (VNS) method customized for the GSP-NC. To compute the built networks reliabilities we adapted the Recursive Variance Reduction (RVR) technic as simulation method since the exact evaluation of this measurement is also NP-Hard. The experimental tests were performed over a wide set of testing cases which contained heterogeneous topologies, including instances of more than 200 nodes. The computational results showed highly competitive execution times, achieving minimal local optimal solutions of good quality fulfilling the imposed survivability and reliability conditions.

Keywords: Survivability, Metaheuristic, VNS, VND, Reliability, Simulation, RVR.

---

1 Introduction

The arrival of the optical fiber and its great capacity implied network topologies more dispersed. As a consequence, the design of the network takes more relevance and has become a much more relevant task. The networks must keep operative when occurring a failure at some component (link or central office). In this context, survivability means that there must exist a certain number of disjoint paths pre-established among any pair of relevant offices. In our case we will require node-disjoint paths which is a stronger constraint than the edge-disjunction one. Assuming that both the links and the nodes have associated certain operation probabilities (elementary reliability), our main objective is to build a sub-network of minimum cost that satisfies the node-connectivity requirements, and moreover, its reliability surpasses a certain lower bound prefixed by the network designer. In this way the model takes into account the robustness of the topology to be designed from the point of view of its structure and also in probabilistic terms. Formally, the proposed model that combines the Network Survivability and Network Reliability approaches is the following.

Let us consider $G = (V, E)$ a non-directed simple graph, $T \subseteq V$ a sub-set of distinguished nodes (denominated terminals), $C = \{c_{ij}\}_{(i,j)\in E}$ a costs matrix associated to the links of G and $R = \{r_{ij}\}_{i,j\in T}$ a connection requirements matrix among pairs of terminal nodes (r_{ij} node-disjoint paths between $i, j \in T$). Let us suppose in addition to this that the edges of E and the nodes of $V \setminus T$ have associated operation probabilities given by two vectors: $P_E = \{p_e\}_{e\in E}$ and $P_{V\setminus T} = \{p_v\}_{v\in V\setminus T}$, where the failures are assumed to be statistically independent. Given a certain probability p_{min} set as reliability lower threshold, the objective is to find a subgraph $G_S \subseteq G$ of minimum cost that satisfies the connection requirements matrix R and furthermore its T-terminal reliability has to satisfy $\mathcal{R}_T(G_S) \geq p_{min}$ (i.e. the probability that all nodes in T are connected by working edges exceed p_{min}). We are going to denote this model by GSP-SRC (Generalized Steiner Problem with Survivable&Reliable Constraints).

The current paper is organized as it follows. In Section 2 we propose an approximated algorithm to solve the GSP-SRC. Section 3 introduces experimental results obtained over a set of heterogeneous test instances as well as the most important contributions and conclusions of this work.

[1] Email: frobledo@fing.edu.uy
[2] Email: viera@fing.edu.uy

2 The Algorithmic Solution for the GSP-SRC

Network Design Algorithm. NetworkDesign is the main algorithm which iteratively executes the different phases that solve the GSP-SRC. The algorithm (shown in Figure 1) receives as entry G the original graph, $MaxIter$ the number of iterations that is going to be executed, k an integer (parameter of the Construction Phase), the $threshold$ of T-terminal reliability required and the number of replications used in Reliability Phase. Each iteration computes: (i) Construction Phase, (ii) Survivability Optimizer Phase, and (iii) Reliability Phase. Construction Phase takes G as input and returns a topology satisfying the node-connectivities given by R. Since the solution built by Construction Phase is not even a local optimum, in order to improve this solution, Survivability Optimizer Phase searches for a local optimum solution by means of a VNS algorithm designed specifically for the GSP-NC. Finally Reliability Phase is computed evaluating the T-terminal reliability of the solution achieved in (ii); if it surpasses the pre-fixed threshold then the local optimal solution is added into the collection L_Sol, otherwise is discarded. The algorithm returns a list L_Sol of feasible solutions that satisfy the pre-established survivability and reliability requirements.

Construction Phase Algorithm. The algorithm (shown in Figure 1) takes as input the graph G of feasible connections, the matrix of connection costs C, the matrix R of connection requirements between terminal nodes, and a parameter k. The current solution $\mathcal{G}_{sol}$ is initialized with the terminal nodes without any connection among them. An auxiliary matrix M is initialized with the values of R. This is used with the purpose of maintaining on each step the connection requirements not yet satisfied between nodes of T. The paths found on each iteration are stored in a data structure $\mathcal{P}$. Iteratively the Construction Phase searches for node-disjoint paths between terminal nodes of T that have not yet satisfied their connection requirements. The algorithm chooses on each iteration a pair of such terminal nodes $i, j \in T$. The current solution is updated by adding a new low cost node-disjoint path between the chosen nodes. For this, we employ an extension of the Takahashi-Matsuyama algorithm [**?**] in order to compute efficiently the k shortest node-disjoint paths from i to j (lines 3-9). These paths are stored in a restricted candidate list $\mathcal{L}_p$. A path is randomly selected from $\mathcal{L}_p$ and incorporated to $\mathcal{G}_{sol}$. This process is repeated until all the connection requirements have been satisfied; then the feasible solution $\mathcal{G}_{sol}$ and the set of node-disjoint paths $\mathcal{P} = \{\mathcal{P}_{ij}\}_{i,j\in T}$ are returned.

Survivability Optimizer Phase: VNS Algorithm for the GSP-SRC. VNS is sustained on the idea of systematically changing the neighbourhood at the moment of performing the local search and therefore requires a finite set of different predefined neighbourhoods. VNS is based on three simple facts: (1) a local mini-

Procedure NetworkDesign;
Input: $G, C, k, MaxIter, umbral, simiter$;

$i \leftarrow 0$; $\mathcal{P} \leftarrow \emptyset$; $L\_Sol \leftarrow \emptyset$;
while $(i < MaxIter)$ do
  $[\bar{\mathcal{G}}_{sol}, \mathcal{P}] \leftarrow \mathrm{ConstructionPhase}(G, C, k)$;
  $\mathcal{G}_{sol} \leftarrow \mathrm{SurvivabilityOptimizerPhase}(\bar{\mathcal{G}}_{sol}, \mathcal{P})$;
  $\tilde{\mathcal{R}}_T(\mathcal{G}_{sol}) \leftarrow \mathrm{RealiabilityPhase}(\mathcal{G}_{sol}, simiter)$;
  if $(\tilde{\mathcal{R}}_T(\mathcal{G}_{sol}) \geq umbral)$ then $L\_Sol \leftarrow \mathrm{Add}(\mathcal{G}_{sol})$;
  else $\mathrm{Discard}(\mathcal{G}_{\mathrm{sol}})$;
  $i \leftarrow i + 1$;
end_while;
return L_Sol;
end NetworkDesign;

Procedure ConstructionPhase;
Input: G, C, R, k;

$\mathcal{G}_{sol} \leftarrow (T, \emptyset)$; $M \leftarrow R$; $\mathcal{P}_{ij} \leftarrow \emptyset \ \forall i, j \in T$;
while $\exists m_{ij} > 0$ do
  Let $i, j \in T$ be two nodes that satisfy $m_{ij} > 0$;
  $\bar{G} \leftarrow (G \setminus \mathcal{P}_{ij})$;
  $\bar{C} \leftarrow C_{|\bar{G} \setminus \mathcal{G}_{sol}}$; /* $\bar{C}_{uv} = 0$ if $(u, v) \in G_{sol}$ */
  $\mathcal{L}_p \leftarrow$ the k shortest paths from i to j on $\bar{G}$, considering $\bar{C}$;
  $p \leftarrow \mathrm{SelectRandom}(\mathcal{L}_p)$;
  $\mathcal{G}_{sol} \leftarrow \mathcal{G}_{sol} \cup \{p\}$; $\mathcal{P}_{ij} \leftarrow \mathcal{P}_{ij} \cup \{p\}$;
  $M \leftarrow \mathrm{UpdateMatrix}(\mathcal{G}_{sol}, M, R, p)$;
end_while;
return $\mathcal{G}_{sol}, \mathcal{P}$;
end ConstructionPhase;

Fig. 1. Global Algorithm and Construction Phase.

mum with respect to a neighbourhood structure it is not necessary a local minimum with respect to another one; (2) a global minimum is a local minimum with respect to all the possible neighbourhood structures; (3) in many problems the local minimum with respect to one or several neighbourhood structures are relatively close. In this work the deterministic variant called VNS descent was used (Variable Neighbourhood Descent, denoted by VND). It consists of iteratively replacing the current solution with the local search result as long as improvements are verified. If a Neighbourhood structure change is performed in a deterministic way every time a local minimum is reached; the descent variable Neighbourhood search is obtained. The final solution given by the VND is a local minimum with respect to all the considered Neighbourhoods. Next, we explain the VND customization for the GSP-NC. Our VND uses three local searches: $\mathrm{SwapKeyPathLocalSearch}$, $\mathrm{KeyPathLocalSearch}$, and $\mathrm{KeyTreeLocalSearch}$ which are explained in detail in *** along with their respective neighbourhood structures.

The algorithm (shown in Figure 2) receives as input $\mathcal{G}_{sol}$ the initial solution graph, $\mathcal{P}$ the matrix of paths (both outputs of Construction Phase), and cls the set of local searches. Initially the cost of $\mathcal{G}_{sol}$ is computed and the local search $\mathrm{SwapKeyPathLocalSearch}$ is applied to it. $\mathrm{SwapKeyPathLocalSearch}$ uses $\mathcal{P}$ the paths matrix as input (lines 1-2). Since only this local search uses the $\mathcal{P}$ information, it is executed only at the beginning of the algorithm a single time. However, its incorporation is fundamental for the purpose of achieving important improvements in the initial solutions generated by the algorithm of construction. Line 3 computes the new cost of $\mathcal{G}_{sol}$ and $notimprove$ is reset to 0. In cycle 4-12 the k-th local search is performed to find a better solution (line 5) until no more improvements can be found by exploring the neighbourhoods set. If an improvement is

Procedure VND_SurvivabiliyOptimizerPhase;
Input: $\mathcal{G}_{sol}$, $\mathcal{P}$, cls;

1: $cost \leftarrow \text{cost}(\mathcal{G}_{sol})$; $k_{max} \leftarrow 2$;
2: $\mathcal{G}_{sol} \leftarrow \text{SwapKeyPathLocalSearch}(\mathcal{G}_{sol}, \mathcal{P})$;
3: $cost \leftarrow \text{cost}(\mathcal{G}_{sol})$; $notimprove \leftarrow 0$;
4: while $(notimprove < k_{max})$ do
5: $\quad \bar{\mathcal{G}}_{sol} \leftarrow cls[k_{max}] \dashrightarrow \text{Search}(\mathcal{G}_{sol}, k)$;
6: $\quad cost \leftarrow \text{cost}(\mathcal{G}_{sol})$; $newcost \leftarrow \text{cost}(\bar{\mathcal{G}}_{sol})$;
7: $\quad$ if $(newcost < cost)$ then
8: $\quad\quad cost \leftarrow newcost$; $notimprove \leftarrow 0$;
9: $\quad\quad \mathcal{G}_{sol} \leftarrow \bar{\mathcal{G}}_{sol}$;
10: $\quad$ else $notimprove \leftarrow notimprove + 1$;
11: $\quad k \leftarrow (k+1) \bmod k_{max}$;
12: end_while;
13: return $\mathcal{G}_{sol}$;
end VND_SurvivabiliyOptimizerPhase;

Procedure RVR_ReliabilityPhase;
Input: $\mathcal{G}_{sol}$, N;

$seed \leftarrow \text{GenerateSeed}()$;
$s \leftarrow 0$; $ss \leftarrow 0$;
for $l = 1$ to N
$\quad \mathcal{G}'_{sol} \leftarrow \mathcal{G}_{sol}$;
$\quad$ /* RVR computes the anti-reliability */
$\quad x \leftarrow 1 - \text{RVR}(\mathcal{G}'_{sol}, P_E, P_{V \setminus T})$;
$\quad s \leftarrow s + x$;
$\quad ss \leftarrow ss + (x * x)$;
end_for;
$\tilde{\mathcal{R}}_{(V \setminus T)}(\mathcal{G}_{sol}) \leftarrow s/N$;
$variance \leftarrow (1/(N \times (N-1))) \times (ss - (s \times s/N))$;
return $\tilde{\mathcal{R}}_{(V \setminus T)}(\mathcal{G}_{sol})$, $variance$;
end RVR_ReliabiliyPhase;

Fig. 2. Survivability and Reliability Phases.

achieved the current cost is updated, $notimprove$ is set in 0 and the new solution $\mathcal{G}_{sol}$ is actualized (lines 8-9). In case of not having improvement $notimprove$ is increased in one (line 10). Ultimately, having or not having had improvements the next neighbourhood is explored in circular form (line 11). In this way and unlike the generic VND, when finding an improvement continues the search for new solutions in the following Neighbourhood instead of returning to Neighbourhoods already explored. Once the loop is finalized (lines 4-12), line 13 returns the best solution found by the VND.

Reliability Phase: RVR Algorithm for the GSP-SRC. The RVR is a powerful variance reduction method which has shown -compared to other methods- to obtain excellent results when applied to different network reliability problems. The construction of the method is done to obtain an estimation of the Q_K measurement (anti-reliability for a K set of terminals). We have adapted the RVR method to estimate the T-terminal reliability of $\mathcal{G}_{sol}$ assuming that the terminal nodes T are perfect (do not fail), and the edges as well as the Steiner nodes (nodes of $V \setminus T$) have operation probabilities $P_E = \{p_e\}_{e \in E}$ and $P_{V \setminus T} = \{p_v\}_{v \in V \setminus T}$ respectively. Details and properties of this adaptation can be found in ***. Figure 2 shows the pseudo-code of the $\mathcal{R}_T(\mathcal{G}_{sol})$ estimated computation using the RVR method.

3 Computational Results and Discussion

All the algorithms were coded using C++. The experimental results were obtained in a Pentium IV 3.0 GHz and 1 GB of RAM running under Windows XP Professional Edition. The parameters $MaxIter$, k, $threshold$ and $simiter$ were set in:

100, 5, 85%, and 10^4 respectively. Since there are no benchmark cases for the GSP-SRC, we have selected 20 instances of the TSP problem from the TSPLIB library and for each of them we have generated three GSP-SRC instances by means of the random selection of a 20%, 35%, y 50% out of the total of nodes, as terminal nodes. The TSP chosen instances were: att48, berlin52, brazil58, ch150, d198, eil51, gr137, gr202, kroA100, kroA150, kroB100, kroB150, kroB200, lin105, pr152, rat195, st70, tsp225, u159, y rd100. The connectivity requirements were randomly set in: $r_{ij} \in \{2, 3, 4\}$, $\forall i, j \in T$. To sum up we introduce the main findings.

We noticed that in all cases the best solutions attained by the VND algorithm were topologically minimal (i.e. by removing an edge the feasibility is lost). The improvement percentage of the VND algorithm with respect to the solution cost delivered by the construction phase was situated (depending on the topological features of the instance) within 25.25% and 39.84%.

For the instances in which the average T-terminal reliability of the L_Sol solutions set returned by NetworkDesign was computed, it widely surpassed the 85% prefixed threshold. Particularly, in those instances in which the operation probabilities of nodes and edges was set in 99% and 90% respectively, the average T-terminal reliability was bounded by 86.0% and 96.7%. On the other hand when setting the values of the operation probabilities of nodes and edges in 99% and 95% respectively, the average T-terminal reliability was bounded by 99.1% and 99.6%. In all these evaluated cases the average variance was small, lower than $1.0E-05$.

Taking into account the NP-hardness of the GSP-SRC model, we can infer that the average times per iteration reached by the NetworkDesign approximated algorithm were highly competitive (less than 173 seconds in the worst case) with respect to each one of the test-set instances that were analyzed.

References

[1] F. Robledo Amoza, "GRASP heuristics for Wide Area Network design", PhD thesis, Université de Rennes I, France, 2005.

[2] S. Laborde, S. Ressi, and A. Rivoir, "Diseño de Topologías de Red Confiables", Proyect, INCO-UDELAR, Montevideo, Uruguay.

[3] N. Mladenović and P. Hansen, "Variable Neighborhood Search", Computers and Operations Research, vol. 24, pp. 1097-1100, 1997.

[4] H. Cancela and M. El Khadiri, "On the RVR Simulation Algorithm for Network Reliability Evaluation", IEEE Tr. on Reliability, vol. 52, no. 2, pp. 207-212, June 2003.

[5] G. Reinelt, TSPLIB: http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/tsplib.html.