Proceedings of the 10th International Conference "Reliability and Statistics in Transportation and Communication" (RelStat'10), 20–23 October 2010, Riga, Latvia, p. 10-24. ISBN 978-9984-818-34-4
Transport and Telecommunication Institute, Lomonosova 1, LV-1019, Riga, Latvia

OPTIMAL RELIABILITY DESIGN OF TRANSPORTATION NETWORK

Ilya Gertsbakh[1], Yoseph Shpungin[2]

[1]*Department of Mathematics, Ben Gurion University*
P.O.Box 653, Beer-Sheva, 84105, Israel
E-mail: elyager@bezeqint.net

[2]*Software Engineering* Department, *Shamoon College of Engineering*
Beer-Sheva, 84105, Israel
E-mail: yosefs@sce.ac.il

In this paper we consider transportation network optimal reinforcement problem. We mean by that network reliability improvement achieved by reinforcement of a certain number of its most important links (road segments), which are subject to failure (destruction). This reinforcement is made given a budgetary constraint, and its goal is to maximize the global network performance measure, which depends on the source-terminal reliability of a fixed set of origin-destination routes.

The central role in the proposed optimization is played by a combinatorial Monte-Carlo algorithm for estimating the network $s-t$ reliability and its gradient vector. The proposed optimization method has been tested on an example described in [1]of a road network of Istanbul. The link failures are caused by natural disasters like earthquakes, and link e reinforcement means its replacement for a cost $c(e)$ by a more reliable one.

Keywords: *Network Monte Carlo, reliability gradient vector, network design, link failure, earthquakes, $s-t$ network reliability*

1. Introduction

Optimal transportation network (TN) reliability design is not a well-defined notion. Its formulation demands first of all answering the following principal questions:

1) What are the network components subject to failure;

2) What is the definition of component failure;

3) How network reliability is connected or related to its component failures;

4) What is the whole network performance criterion (NPC) and how is it expressed via its reliability parameters.

After these issues are clarified, it is necessary to define the relevant resources which are at the disposal of the network designers. Then it will become possible to formulate the problem of finding the best policy of using these resources to achieve the maximal possible NPC under given constraints.

Following recently published papers [1,2], we make the following assumptions:

a) TN components subject to failure are the links (or arcs),i.e. the road segments connecting network nodes;

b) Link failure formally means that the link $e=(a,b)$ connecting nodes (a,b) becomes not operational, i.e. the connection between a and b via this link is completely disrupted. Physically it might be due a natural disaster (earthquake, flood), road accident or, speaking in military terms, an "enemy attack."

c) Each link e has its failure probability $q(e)$, and link failures are viewed as independent events.

d) The TN performance criterion is a function of the network ability to provide a connection between two sets of its nodes - the origins and destinations. The simplest case of NPC will be a functional depending on the probability to provide connection between a "source" and "terminal", or so-called probability of $s-t$ connectivity. The model considered in [1], singled out several most important $s-t$ pairs in the TN and assumed that the NPC depends on the set of $s-t$ connection probabilities of these pairs.

e) A natural way to formalize network improvement or reinforcement (termed in [1] as predisaster investment) is to carry out link *e* "repair" to reduce its failure probability from *q*(*e*) to $q_{\min}(e) < q(e)$ by investing *c*(*e*). In physical terms, this means reconstruction of the road segment and/or reinforcement of its weak elements, like bridges, tunnels, etc. Since there are always budget constraints, the total amount of money invested in the network component reinforcement must be limited by a certain budget *D*.

The above assumptions a)-e) allow to formulate our problem in the following way: find the "best" set of reinforced components (links) in order to maximize the NPC subject to the budget constraint *D*.

The paper is organized as follows. In Section 2 we formulate the problem in formal terms and present our principal optimization algorithm. We suggest using a greedy-type semi-heuristic procedure. Its most important and difficult part is the estimation of network reliability and the reliability gradient vector for each of the selected $s-t$ pairs. We shortly describe this procedure in Appendix. Section 3 presents an account of our optimization results for an Istanbul network analyzed in [2].

2. Problem Formulation and Algorithm

By transportation network (TN) $N = (V, E, T)$ we denote an undirected graph with a node-set $V, |V| = n$, an edge-set $E, |E| = m$, and a set $T \subseteq V$ of special nodes called *terminals*. Instead of the term *edge* we will often use the word *link* or *arc*. Each network link *e* is associated with a probability p_e of being *up* and probability $q_e = 1 - p_e$ of being *down*. We postulate that link failures are mutually independent events. Link failure means its elimination (erasing) from *E*.

Let *s* and *t* be two distinct terminals, $s, t \in T$. The TN will be called $s-t$ (or source-terminal) connected if there is a path connecting *s* and *t*, with all its edges being in the *up* state. Denote by $R(s-t; \vec{p})$ the probability that the TN is $s-t$ connected. Here $\vec{p}$ denotes the vector of link *up* probabilities: $\vec{p} = (p_1, p_2, \ldots, p_m)$. In TN, typically there are several $s-t$ pairs of nodes whose connection is of special importance. We assume that there are *k* such pairs and denote them as $Z = \{s_1 - t_1, \ldots, s_k - t_k\}$.

In this paper we suggest as the NPC the *minimal* reliability of the $s-t$ connectivity over the set of all source-terminal pairs *Z*:

$$G(\vec{p}) = \min_{i \in Z}[R(s_i - t_i; \vec{p})]. \tag{1}$$

Let us introduce two additional vectors: $p^* = (p_1^*, \ldots, p_k^*)$ and $c^* = (c_1, \ldots, c_k)$, where c_j is the cost of *increasing* link *j up* probability from p_j to p_j^*.

Let $\vec{B} = (b_1, \ldots, b_k)$ be a *k*-dimensional vector with binary 0/1 components. The set of all 2^k binary vectors we denote by Ω. Suppose we choose a particular vector $\vec{B} \in \Omega$ and decide to reinforce those links whose coordinates in $\vec{B}$ are not zeroes. Then the vector of link *up* probabilities will become

$$\vec{p}(\vec{B}) = (p_1 + (p_1^* - p_1) \cdot b_1, \ldots, p_k + (p_k^* - p_k) \cdot b_k).$$

The cost of this action will be

$$C(\vec{B}) = c_1 \cdot b_1 + \ldots + c_k \cdot b_k.$$

Suppose that the total reinforcement budget must not exceed the given value *B*.

Now we are ready to formulate our optimization problem in the following compact form: find such $\vec{B} \in \Omega$ which *maximizes*

$$G(\vec{p}(\vec{B})) \quad \text{subject to} \quad C(\vec{B}) \le D. \tag{2}$$

Now let us approach the maximization procedure. It will be based on an efficient Monte Carlo procedure of estimating network reliability gradient function. It does not demand the knowledge of the analytic form of $R(s_i - t_i; \vec{p})$. Let us postpone to the Appendix the details of the gradient estimation procedure and suppose that we have an accurate estimate of

$$\Delta_i R(s-t; \vec{p}) \approx \frac{\partial R(s-t; \vec{p})}{\partial p_i} \cdot (p_i^* - p_i).$$

Now we are ready describe a greedy type semi-heuristic solution algorithm for solving problem (2). Denote by *S* the set of all indices {1,2,...k} and all *pairs* of indices $\{(i, j), 1 \le i < j \le k\}$.

Algorithm MinGradKnapsack

1. Set $X := 0$.
2. $Y := G(\vec{p})$. (This is the initial value of the NPC)
3. For each element $\alpha \in S$, estimate $\Delta_i G(\vec{p})$. [Calculate the increase of the minimal probability of $s-t$ connectivity as a result of the reinforcement of link (or links) α].
4. Calculate $\eta_\alpha = \dfrac{\Delta_i G(\vec{p})}{c_\alpha}$, for each $\alpha \in S$ and arrange them in *decreasing* order. Denote by *w* the index of the *maximal* η_i. [If α is a pair of indices, e.g $\alpha = (r, s)$, then put $c_\alpha = c_r + c_s$].
5. Put $X := X + c_w; S := S \setminus \alpha$. [Delete α from the set S; if $\alpha = j$, delete *j* and all pairs containing *j*. Act similarly if $\alpha = (r, s)$].
6. Recalculate p by replacing p_α by p_α^*. [If α is a pair (*r*,*s*), replace the *r*-th and the *s*-th component of *p* by p_r^*, p_s^*].
7. If $X > D$, then Stop. Otherwise GOTO 2. #

In simple words, the algorithm finds on each step the link or a pair of links whose reinforcement provides the maximal performance measure *increase per unit cost*, recalculates the reliabilities of all $s-t$ connections involved and the gradient vector after the links have been reinforced, repeats this procedure in a loop and stops when the budget becomes exhausted.

3. Example: 30-link network of Istanbul [1]

We describe in this section a road network of Istanbul, borrowed from the paper [1] by Srinivas Peeta, F.Sibel Salman, Dillek Gunnec, and Kannan Viswanath. The data for this example are summarized in Table 1 and are cited with the kind permission of the authors.

The network has 30 links and 20 nodes. The first and the fourth columns contain data on the links and the nodes they connect. Columns 2 and 5 are the link reinforcement costs; columns 3 and 6 give the link *up* probability. As suggested in [1], five $s-t$ routes are of primary importance: *a* = (14,20), *b* = (14,7), *f* = (12,18), *g* = (9,7), and *h* = (4,8).

Our goal is to find the links which are to be reinforced in order to *maximize* the minimal reliability of the $s-t$ connectivity over the routes *a*, *b*, *f*, *g*, *h*. As assumed in [1], reinforcement of a link *e* raises its *up* probability from p_e to $p_e^* = 1$ for all *e*. We assumed the budget constraint *D* = 1700.

Our algorithm found that the following links have to be reinforced: 10, 17, 20, 21, 22, 23, the total cost of which is 1680. Our algorithm carried out 5 cycles. On each of the first four cycles, one link has been reinforced, and two links were selected on the last, fifth cycle.

The minimal reliability has route *h*: *R*(*h*) = 0.686; the reliability of other $s-t$ pairs are as follows:

$$R(a) = 0.773, R(b) = 0.723, R(f) = 0.999, R(g) = 0.834.$$

In spite of the fact that [1] considers quite different NPC, a convex combination of functions of paths lengths for 5 selected pairs, our optimization results greatly coincide: [1] recommends the reinforcement of the links 10, 20, 21, 22, 23, 25. The cost of this replacement is smaller - 1640, the minimal $s-t$ probability 0.682 have routes $b, h.$ The average reliability of the above 5 pairs differ only by 0.003 and is 0.800.

Table 1. Links, their repair costs and reliability

Link *e*=(*a*,*b*)	c_e	p_e	Link *e*=(*a*,*b*)	c_e	p_e
1=(1,3)	80	0.80	16=(11,13)	940	0.55
2=(2,4)	80	0.80	17=(12,13)	300	0.70
3=(3,4)	320	0.80	18=(12,16)	520	0.60
4=(3,5)	260	0.70	19=(12,25)	40	0.80
5=(4,6)	160	0.80	20=(13,14)	800	0.55
6=(5,8)	420	0.60	21=(14,15)	40	0.80
7=(5,10)	160	0.80	22=(15,18)	160	0.70
8=(6,7)	620	0.60	23=(16,17)	40	0.80
9=(7,10)	120	0.80	24=(17,21)	620	0.60
10=(7,11)	340	0.70	25=(18,20)	260	0.70
11=(8,9)	940	0.55	26=(18,21)	780	0.60
12=(8,10)	160	0.80	27=(19,20)	800	0.55
13=(9,11)	620	0.60	28=(20,22)	120	0.80
14=(9,12)	180	0.50	29=(21,23)	220	0.70
15=(9,24)	40	0.80	30=(22,23)	500	0.60

4. Appendix : $s-t$ Reliability and Reliability Gradient

To estimate the $s-t$ connectivity we suggest one of the Monte Carlo procedures described in [3], e.g. the turnip algorithm, see Chapter 9. For small networks, it might be even the crude Monte Carlo.

Let us describe an efficient approach to estimating the network reliability gradient vector. It is based on identifying so-called network *border states* [3], Chapter 5.

Definition A1.

Reliability gradient vector ∇R is defined as $\nabla R = (\frac{\partial R}{\partial p_1}, \ldots, \frac{\partial R}{\partial p_k})$.#

Denote by a binary variable e_i the state of the *i*-th network component: 0 means *down*, 1 means *up*.

Definition A2.

Network state $V = (e_1, \ldots, e_k)$ is called *border state* if:

1. This state is a *DOWN* state;
2. There exists such state $W, W \in UP,$ that the states *V* and *W* differ in exactly one position (i.e. the Manhattan distance between vectors *W* and *V* equals 1). #

The following formula [3] binds together the partial derivative $\frac{\partial R}{\partial p_i}$ and the set of such border states that each of them may be transferred into the *UP* state by turning a single *down* element $e_i = 0$ into *up* state $e_i = 1$:

$$\frac{\partial R}{\partial p_i} = q_i^{-1}\{\sum_{\{v \in BD, v+(0,\ldots,0,1_i,0,\ldots 0) \in UP\}} P(v)\}, \qquad i = 1, \ldots, k. \qquad (3)$$

Here *BD* is the set of all border states, and *P*(*v*) stands for the static probability of the state *v*. In words: to calculate the *i*-th gradient's vector component, it is necessary to find the probability of all border states which become *UP* by "activating" a single component e_i.

The last formula is the key for Monte Carlo algorithm for computing reliability gradient. Note that this algorithm is rather efficient and avoids the so called *rare event phenomenon*. The algorithm works as follows.

Simulate permutation of the network elements. Suppose that initially all elements are *down*, and they are arranged in the order determined by this permutation. Start turning the elements from *down* to *up* from left to right. Suppose that after adding a new *up* element we arrive at a border state. Clearly, from this moment the network remains in border states until the moment it becomes *UP*. Calculate and sum up $P(v)$ for all elements which may transfer the border state v into an *UP* state.

References

1. Peeta Srinivas, Salman Sibel F., Gunnec Dillek, Viswanath Kannan. 2010. Predisaster Investment Decisions for Strengthening a Highway Network. *Computers & Operations Research* 37, 2010, 1708-1719.
2. Sanchez-Silva M., Daniels M., Lleras G., Patino D. A Transport Network Reliability Model for Efficient Assignment of Resources. *Transportation Research, Part B*, 39, 2005, pp. 47-63.
3. Gertsbakh Ilya and Shpungin Yoseph. Models of Network Reliability: Analysis, Combinatorics and Monte Carlo. *CRC Press*, 2010.