Building Multiobjective Resilient Networks

Crina Grosan[1], Ajith Abraham[2] and Bjarne E. Helvik[2]
[1]*Department of Computer Science*
Babes-Bolyai University, Cluj-Napoca, Romania
[2]*Centre for Quantifiable Quality of Service, Centre of Excellence*
Norwegian University of Science and Technology, Trondheim, Norway
cgrosan@nessie.cs.ubbcluj.ro, ajith.abraham@q2s.ntnu.no, bjarne@q2s.ntnu.no

Abstract

The paper deals with the design of resilient networks that are fault tolerant against link failures. Usually, fault tolerance is achieved by providing backup paths, which are used in case of an edge failure on a primary path. We consider this task as a multiobjective optimization problem: to provide resilience in networks while minimizing the cost subject to capacity constraint. We propose a stochastic approach, which can generate multiple Pareto solutions in a single run. The feasibility of the proposed method is illustrated by considering two network design problems.

1. Introduction

There has been some recent research interest in developing algorithms for problems of guaranteeing resilience against failures. Due to the fact that the assurance of continuity in traffic is a vital demand in today's networks, we have to be able to provide backup paths at the moment a failure on a edge (or in multiple edges) on the primary path occurs. For this, the backup path has to be built a priory.

There are several attempts to solve the resilience and path allocation problem in different ways. Chekuri et al. [1] deal with edge failure by building simultaneous primary and backup paths but they are only considering the case of uncapacitated networks. A hybrid genetic algorithm which deals with alternative backup paths is proposed in [1]. But the problem is not considered as multiobjective. A multiobjective network design approach is proposed by Banarjee and Kumar [3] and by Yeh [11], but not directly treating resilience.

There is a huge amount of work related to network resilience that can be found in the literature: [4][5][6][7][8][9][10] are some of them. Some surveys on resilient network technology, planning and optimization can be found in [12][13][14]. We focus on this paper on the simultaneous allocation of primary and backup paths. We also formulate the problem as dealing with multiple criteria in the same time. Several objectives can be formulated. In this research, we deal with three criteria: (*1*) minimize the network cost, (*2*) minimize the number of common links between primary and its corresponding backup path (so that in case an edge (link) failure occurs on the primary path the chances that the same edge belongs to the backup path also and makes the traffic impossible to continue is minimal) and (*3*) maximize the number of common links between all the backup paths (in the situation that there are several source-destination flows to be fulfilled at the same time). We also take into consideration that an edge failure can affect only one of the primary paths at one time and there no need of using both backup paths as reserve in the same time. One approach plan and develop such networks is by simulation, where a stochastic optimization technique is implemented.

Rest of the paper is organized as follows. We provide a detailed description of the proposed algorithm in the following Section. The ways in which a solution is initialized and improved are widely described and an illustrative example is presented in the Section 2.1. In the Section 3, the proposed approach is tested for a well known benchmark: the Europe network. Conclusions are provided towards the end.

2. Problem formulation

Since in today's networks, due to the demand's importance we cannot simply rely on a single path. We have to a priori build a path, which can be used to reroute the traffic in case a failure occurs on the primary path. It is necessary to predefine a backup path (or backup sub-paths) if:

- the recovery of full capacity after failure shall be guaranteed;
- the recovery shall be almost immediate (~ 50 ms), i.e. no noticeable service reduction.

Recovery by predefining backup paths is referred to as "protection". Recovery by finding back-up paths when (after) a failure has occurred is referred to as "restoration" [12][14].

This article focuses on the cases where it is guaranteed that no more than a fixed number of edge-failures can occur. We then consider the problem of simultaneous primary and backup path allocation. We are given specifications of the traffic to be handled, and we want to have a provision for both the primary network as well as the backup network. On the arrival of a pair (source, destination) we must find both a primary path and a backup path between them.

Network parameters

We consider the following network parameters:

- A bi-directed graph G= (V, E);
  - the adjacency matrix;
- the capacity of each link: cap : $E \rightarrow R^+$;
- the cost associated to each link: co : $E \rightarrow R^+$;
- a pair source – destination (s, d) $\in$ V (or a set of pairs (s_i, d_i), i=1..,N, $N \geq 1$, in the general case) and the requirements.

Objective functions

The goal is to find a minimal (cost wise) path between these nodes as well as to assure that the capacity of the edges is not overloaded, which is formulated in the objectives considered below.

Objective 1: minimize the total cost of the paths (primary and backup) between source and destination:

minimize

$$\sum_{i=1}^{N} \text{cost(primary path } (s_i, d_i) + \sum_{i=1}^{N} \text{cost(backup path(} s_i, d_i)$$

Objective 2: Suppose an edge fails; in order to ensure the network survivability we have to use the backup path. But for this we have to assure that the failed edge is not part of the backup path also. This is formulated as a simple objective (criterion) in our model: minimize the number of common edges between primary and backup paths:

minimize

$$\sum_{i=1}^{N} \begin{array}{l}\text{no of common edges (primary path } (s_i, d_i), \\ \text{backup path(} s_i, d_i))\end{array}$$

Objective 3: This objective is used while dealing with shared backup path protection (the objective requires that the backup paths (in the situation in which we have multiple source-destination pairs) should share as many edges as possible):

maximize

$$\sum_{i=1, j=1, i \neq j}^{N} \begin{array}{l}\text{no of common edges (backup path(} s_i, d_i), \\ \text{backup path(} s_j, d_j))\end{array} .$$

Constraints:

Capacity is treated as a constraint. Once a solution is generated or obtained by modifying an existing one (as explained in the following sections) its validity is verified. This validity refers to the capacity restriction. If capacity is overloaded in at least one link, then the solution is not taken into consideration.

2.1 Proposed Pareto Resilience Model (PRM)

A PRM solution represents all the required paths to be designed together with their corresponding backup paths. For example, if we have 2 source nodes and 2 destination nodes and we have to find 4 primary paths between sources and destinations as well as 4 backup paths, a solution will represent all the 8 paths. If we have just one source node and one destination node then the solution will only consists of two paths (the primary and the backup one).

In order to build a solution, a set of valid paths between each source and each destination node required are to be generated. From each set, the primary and backup paths, which will compose of the solution, are then selected. This selection is done on a random basis. A set of such feasible solutions is generated and maintained during the search process. Improvement techniques are further used in order to assure a convergence to a better solution.

Solution design

Following are the steps to build a path:

Step 1. Start with the source node as current node.

Step 2. If there is a direct path (link) from the current node to the destination node then move to destination node. Otherwise randomly chose another node from the network, which is connected to the current node and set it as new current node.

Step 3. If the current node is the desired destination node then stop. Otherwise go to Step 2.

The following constraints are to be taken into account:

- each node can be used at most one time in a path (for avoiding cycles);
- if a node is reached from where further movement to another node is impossible (because all the connected nodes were previously used), the solution is aborted.

Solution improvement

There are two ways to improve a solution:

1. rebuild an existing path (primary or backup) from a given node;
2. replace an existing path with another one from the set of paths generated in the beginning of the search process.

Both these techniques for improving a solution are considered.

First improvement technique

In order to improve a solution designed using the above procedure, some modifications are performed on the existing paths as follows:

- one primary path is randomly selected from the ones initially designed. One node one this path is also randomly generated. From the selected node, the path is re-built by considering another path to reach the destination different from the current one.
- one backup path is randomly selected from the ones initially generated and a procedure similar to the one used for the primary path is applied.

Second improvement technique

Another way in which a solution can be improved consists of exchanging a whole path with another one from the existing set (initially generated) of paths. One path (either primary or backup) between each source and destination is picked at one time and exchanged with another one. The newly obtained solution is again compared with the initial solution and the rules described above are followed.

The improvement procedures are applied for each of the solutions. These improvement steps are repeated for a successive number of iterations. At the end of these iterations, the non-dominated solutions among all the obtained solutions are considered. The newly obtained solution is compared to the previous solution (which was modified). In order to choose between two solutions and compare them, Pareto dominance relationship is used:

- If it dominates the previous solution, then the new solution is kept;
- If it is dominated by the previous solution then the previous solution is kept;
- If the initial solution and the new obtained solution are nondominated then one of them (randomly chosen between the two) is kept.

The PRM description

The PRM pseudo code is summarized as follows:

```
Begin
For each pair source-destination
initialize a set of N possible paths.
Initialize a set of No_sol solutions.
Set t = 1.
Repeat
        For i = 1 to No_sol do
        Begin
                Improvement 1 (i, j).
                If dominates (j, i)
                Then replace i by j
                Else if dominates (i, j)
                Then keep i
                Else keep any of i or j.

                Improvement 2 (i, j).
                If dominates (j, i)
                Then replace i by j
                Else if dominates (i, j)
                Then keep i
                Else keep any of i or j.
        end
        t = t+1
Until t = No_of_Iterations.
Print all the nondominated solutions
obtained at the final iteration.
End.
```

Remarks:

(i) The *Improvement 1* and *Improvement 2* procedures refer to the solutions obtained by applying that specific improvement.

(ii) *No of iterations* is a priori known and represents the stopping criterion.

3. Experiments and Discussions

In order to emphasise the performance of the proposed approach, we considered two examples. The first example is described in detail so that the reader can easily follow all the explanations about the PRM provided in the previous Section. The second example is the well known Europe network benchmark.

Experimental example

We consider the network given in Figure 1. The cost associated to each pair of connected nodes is considered to be one. We have two source nodes (node 1 and node 5) and two destination nodes (node 8 and node 13). The goal is to efficiently design two primary paths and two corresponding backup paths.

Solution initialization

Suppose we have to determine 4 paths (two primary paths and two backup paths). A set of paths between node 1 and node 5 and a similar set of paths between node 8 and node 13 are generated. From these sets, paths are picked randomly and the initial set of solutions is constructed. Then, an initialization of these sets (having 10 elements) might look as follows:

```
1 3 11 8 9 10 12 13 6 5
1 3 4 5
1 7 3 2 9 8 10 12 13 11 6 5
1 3 11 6 5
1 3 11 12 13 6 5
1 3 2 9 8 10 12 11 6 5
```

```
1 7 3 4 5
1 2 3 11 6 5
1 7 8 11 6 5
1 7 3 2 9 12 13 6 5
8 10 12 13
8 7 1 3 4 5 6 13
8 7 3 4 6 13
8 7 11 13
8 9 12 13
8 9 11 13
8 11 13
8 10 9 11 13
8 7 3 11 13
8 7 1 3 2
```

6 of the solutions are given below:

```
P⟹ 1 3 11 8 9 10 12 13 6 5
B⟹ 1 3 4 5
P⟹ 8 7 1 3 4 5 6 13
B⟹ 8 7 3 4 6 13

P⟹ 1 3 2 9 8 10 12 11 6 5
B⟹ 1 3 4 5
P⟹ 8 9 12 13
B⟹ 8 11 13

P⟹ 1 3 11 12 13 6 5
B⟹ 1 7 3 2 9 8 10 12 13 11 6 5
P⟹ 8 11 13
B⟹ 8 7 3 11 13

P⟹ 1 3 2 9 8 10 12 11 6 5
B⟹ 1 2 3 11 6 5
P⟹ 8 7 1 3 4 5 6 13
B⟹ 8 7 1 3 2 9 10 12 13

P⟹ 1 7 8 11 6 5
B⟹ 1 7 8 11 6 5
P⟹ 8 11 13
B⟹ 8 9 12 13

P⟹ 1 7 3 2 9 12 13 6 5
B⟹ 1 3 2 9 8 10 12 11 6 5
P⟹ 8 10 12 13
B⟹ 8 11 13
```

Solution Improvement

First improvement technique

Let us consider the first solution given above.

```
P⟹ 1 3 11 8 9 10 12 13 6 5
B⟹ 1 3 4 5
P⟹ 8 7 1 3 4 5 6 13
B⟹ 8 7 3 4 6 13
```

Suppose the first and forth paths are selected (this means, primary path is between nodes 1 and 5 and the backup path between nodes 8 and 13). A number is generated between 1 and the length of the primary path selected (which is 10) minus 1. Suppose we generated number 2. This means, starting with the second position (which is node 3) this path is rebuild.

The newly obtained path can be as follows:

```
P⟹ 1 3 11 6 5
```

Same procedure is applied for the backup selected for improvement. Let the new obtained path be as follows:

```
B⟹ 8 7 3 11 12 13
```

Then, the newly obtained solution is:

```
P⟹ 1 3 11 6 5
B⟹ 1 3 4 5
P⟹ 8 7 1 3 4 5 6 13
B⟹ 8 7 3 11 12 13
```

A set of 50 paths between each source and destination are initially considered. Also, the number of solutions initialized and maintained during the search process is equal to 50. The number of iterations considered is 50. 13 nondominated solutions were obtained. 4 of them are depicted in Figure 1.

Europe network example

We consider the Europe network benchmark, which consists of 37 nodes. The problem is to find primary and backup paths between the pairs (11, 31) and (25, 26). The cost of each existing link is set to be one.

The parameters used by PRM are the same as used for the first example. A set of 8 feasible solutions is obtained at the end of the search process. 4 of the solutions are depicted in Figure 2. The value of the second objective is 0 for the obtained solutions, which means there are no common edges between a primary path and its corresponding backup.

It is worth to mention that an advantage of this technique is the computational time, which is very less for both considered experiments (less than one second).

4. Conclusions

The paper proposd an algorithm dealing with multiobjectivity in resilience network. The Pareto Resilience Model (PRM) proposed herein focuses on simultaneous allocation of both primary and backup paths. Several criteria to be optimized at one time can be considered but in this article we used only 3 objectives. Capacitated networks are considered while capacity is treated as constrained. The performance of the proposed approach is tested for a network randomly generated with 13 nodes and for the Europe network benchmark with 37 nodes. The proposed approach is able to detect multiple feasible solutions within a very short time (less than a second).

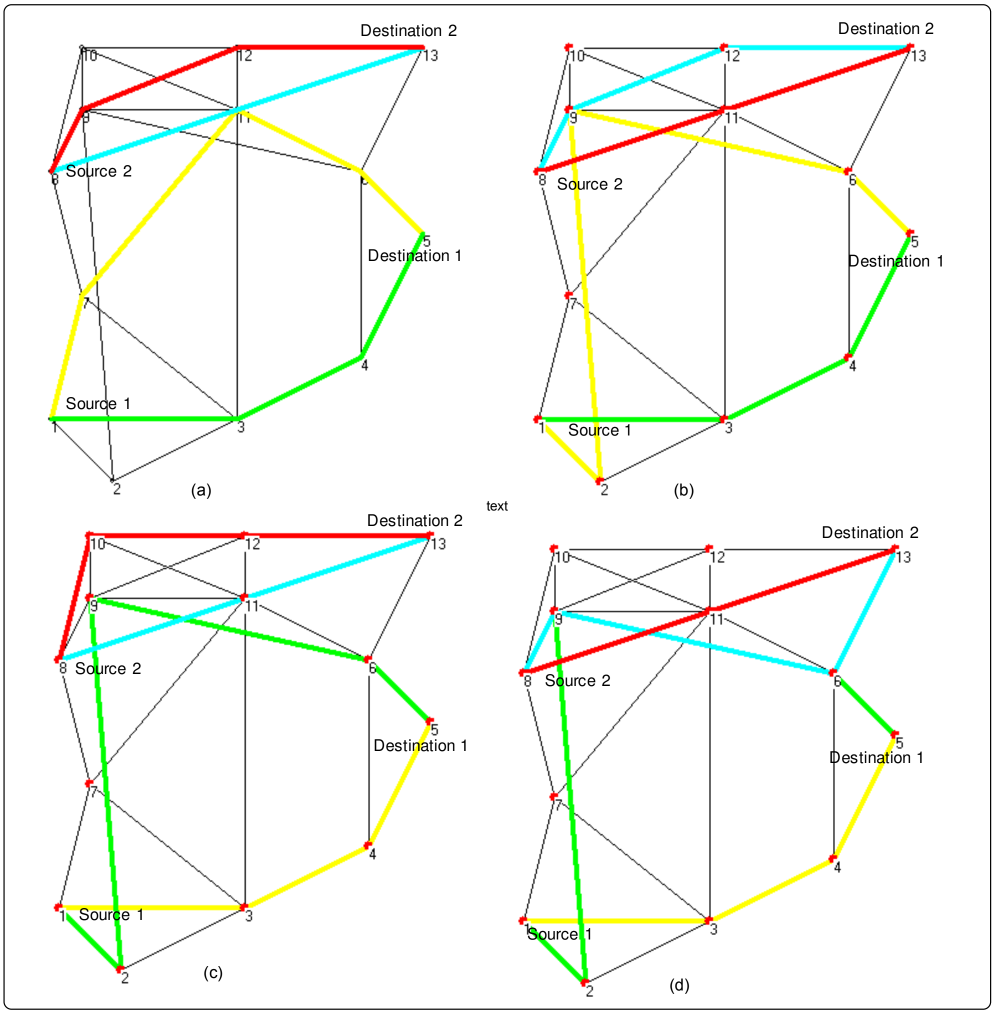

Figure 1. Examples of Pareto solutions obtained by PRM

Since the model proposed here uses only to ways to improve a solution and no modality to combine two existing solutions we propose as future work the introduction of multiple ways to generate and obtain new solutions from the existing ones which can increases the diversity of the results at the end of the search process.

References

[1] C. Chekuri, A. Gupta, A. Kumar, J. Naor, D. Raz, Building Edge-Failure Resilient Networks, In Proceedings of the 9th Integer Programming and Combinatorial Optimization Conference, Cambridge Massachusetts, pp. 439-456, 2002

[2] L. He, N. Mort, Hybrid Genetic Algorithms for Telecommunications Network Back-Up Routing, BT Technology Journal, 18(4), pp. 42-50, 2007

[3] N. Banarjee, R. Kumar, Multiobjective Network Design for Realistic Traffic Models, In Proceedings of Generic and Evolutionary Computation Conference (GECCO'07), London, UK, pp. 1904-1911, 2007

[4] A. Bremler-Barr, Y. Afek, E. Cohen, H. Kaplan, M. Merritt, Restoration by Path Con-catenation: Fast Recovery of MPLS Paths, In Proceedings of the ACM PODC '01, 2001.

[5] A. Balakrishnan, T. Magnanti, J. Sokol, and Y. Wang. Spare-Capacity Assignment For Line Restoration Using a Single-Facility Type. In Operations Research, 50(4), pp. 617-635, 2002.

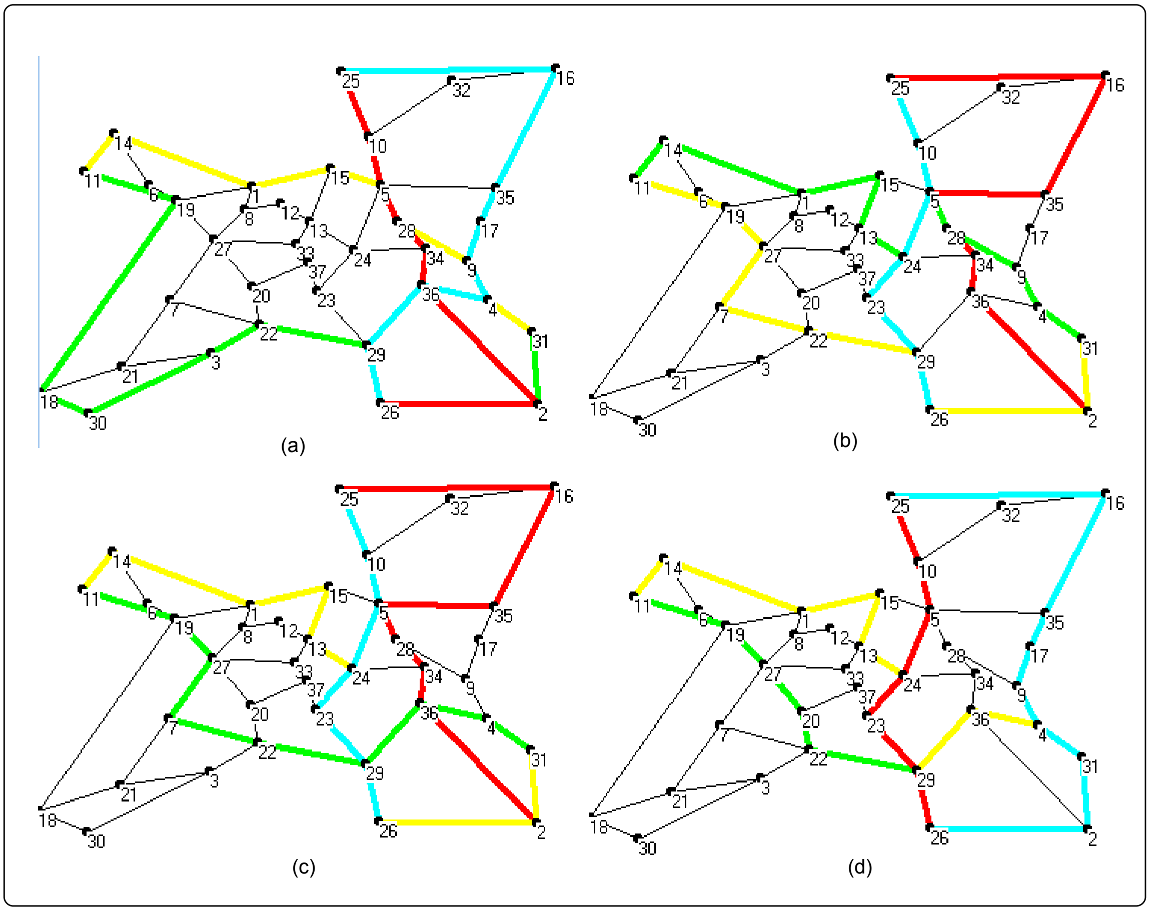

Figure 2. Example of solutions obtained by PRM for the Europe network benchmark. The source- destination pairs are (11, 31) and (25, 26).

[6] A. Balakrishnan, T. Magnanti, J. Sokol, Y. Wang, Telecommunication Link Restoration Planning with Multiple Facility Types. In Annals of Operations Research,Topological Network Design in Telecommunications,P. Kubat and J. M. Smith (Eds.), 106(1-4), pp. 127-154, 2001.

[7] D. Bienstock, G. Muratore, Strong Inequalities for Capacitated Survivable Network Design Problems, Mathematical Programming, 89, pp. 127-147, 2001.

[8] G. Brightwell, G. Oriolo, F. B. Shepherd, Reserving resilient capacity in a network, SIAM Journal of Discrete Mathematics, 14(4), pp. 524-539, 2001.

[9] G. Brightwell, G. Oriolo, F. B. Shepherd, Reserving Resilient Capacity with Upper Bound Constraints, Networks, 41(2), pp. 87-96, 2003.

[10]G. Dahl, M. Stoer, A Cutting Plane Algorithm for Multicommodity Survivable Network Design Problems, INFORMS Journal on Computing, 10, pp. 1-11, 1998.

[11]W.C. Yeh, An interactive augmented max-min MCS-RSM method for the multi-objective network reliability problem, International Journal of Systems Science, 38:2, 87 – 99, 2007

[12]W. D. Grover, Mesh-based Survivable Transport Networks: Options and Strategies for Optical, MPLS, SONET and ATM Networking. Prentice Hall PTR, 2003.

[13]M. Pioro, D. Medhi, Routing, Flow and Capacity Design in Communication and Computer Networks. ISBN 0125571895. Morgan Kaufmann Publishers, March 2004.

[14]J.-P. Vasseur, M. Pickavet, P. Demeester, Network Recovery : Protection and Restoration of Optical, SONET-SDH, IP, and MPLS. The Morgan Kaufmann Series in Networking. Morgan Kaufmann, 2004.