

Reliable Routings in Networks with Generalized Link Failure Events*

Stamatis Stefanakos

Dept. of Computer Science,
University of Rome “La Sapienza,” 00198 Rome, Italy
stefanak@di.uniroma1.it

Abstract. We study routing problems in networks that require guaranteed reliability against multiple correlated link failures. We consider two different routing objectives: The first ensures “local reliability,” i.e., the goal is to route so that each connection in the network is as reliable as possible. The second ensures “global reliability,” i.e., the goal is to route so that as few as possible connections are affected by any possible failure. We exhibit a trade-off between the two objectives and resolve their complexity and approximability for several classes of networks. Furthermore, we propose approximation algorithms and heuristics. We perform experiments to evaluate the heuristics against optimal solutions that are obtained using an integer linear programming solver. We also investigate up to what degree the routing trade-offs occur in randomly generated instances.

1 Introduction

As high-speed networks become widely deployed and more commercial services depend on them, it is extremely important to provide reliable connections to the end users. In circuit-switched networks, connections are established by reserving resources along end-to-end paths that are kept up for the duration of the communication. Such networks are, for example, the so-called all-optical networks [12]. In all-optical networks, connections are established through light-paths and several light-paths are multiplexed in the same optical fiber that can carry data at rates of the magnitude of Terabits/sec. In these networks, reliable communications are even more crucial since a short network outage can lead to massive data losses and can affect many connections.

Traditional methods for ensuring reliable transmissions in circuit-switched networks rely on the precomputation of a backup path for every working path or for every network link (see [9] and [15] and the references therein for studies of single link protection in all-optical networks). These methods work fine as long as the network experiences only single link failures. They do not guarantee undisturbed communication, however, in the case of multiple link failures. Such failures are not seldom and often are correlated: a single failure in the physical

* Work partially done while the author was with the Computer Engineering and Networks Laboratory of the Swiss Federal Institute of Technology in Zurich.

network (a cut in the conduit carrying wiring or fibers used for several links) results in several failures in the abstract network layer (see [6] for a discussion on multiple link failures). This type of link failures can be modeled using the notion of *generalized failure events*. A single generalized failure leads to the failure of several links in the network. Links that belong to the same failure event are also said to be in the same shared risk link group [5].

In this paper, we consider the problem of computing reliable routings in networks with generalized failure events under two different notions of reliability. In what we call “local reliability,” we seek routings in which each connection spans as few as possible different failure events. In a sense, we minimize the failure probability of each connection, assuming that all failure events occur equally likely. In what we call “global reliability,” we seek routings in which any single failure event affects as few as possible connections. Under this objective, we are interested in minimizing the distortion to the network operation in case of a failure event.

1.1 Problem Definitions and Preliminaries

We model the network via an undirected multi-graph $G = (V, E)$. Connection requests are pairs of vertices in the graph. A connection (s, t) is established via an undirected simple path from s to t (an s - t path). We represent failure events by assigning colors (labels) to the edges of the graph. That is, each failure event corresponds to a single color and all edges bearing a particular color fail when the corresponding event occurs.

We consider “locally” reliable routings, in the sense that each single established connection should be as reliable as possible. Formally, we speak about the MINIMUM COLOR PATH problem: We are given an undirected multi-graph $G = (V, E)$, an assignment of colors to the edges of G , and a set of connection requests $R = \{(s_1, t_1), \dots, (s_k, t_k)\}$. We seek a *routing* of R , i.e., an $s_i - t_i$ -path for $1 \leq i \leq k$. The goal is to minimize the average number of colors contained in a path. Observe that the problem can be solved for each path separately: we simply seek a path with the minimum number of colors for each request.

“Globally” reliable routings are routings in which any failure event only affects a few of the connections. Formally, we speak about the MINIMUM COLOR CONGESTION ROUTING problem: We are given an undirected multi-graph $G = (V, E)$, an assignment of colors to the edges of G , and a set of connection requests $R = \{(s_1, t_1), \dots, (s_k, t_k)\}$. We seek a routing of R , such that the maximum number of paths that contain the same color is minimized.

The MINIMUM COLOR CONGESTION ROUTING problem is closely related to the MINIMUM LOAD ROUTING problem. The latter asks for a routing so that the maximum number of paths per edge (i.e., the maximum load) is minimized. MINIMUM COLOR CONGESTION ROUTING is a generalization of MINIMUM LOAD ROUTING and it reduces to the special case if each edge is assigned a different color.

We give some formal definitions regarding algorithms and graph classes that we will need later on: An algorithm A for a minimization problem Π is a ρ -*approximation algorithm* if for every instance I of Π , A runs in time polynomial

in $|I|$ and delivers a solution with objective value at most $\rho \cdot OPT(I)$, where $OPT(I)$ denotes the objective value of an optimal solution for I . We also say that ρ is the *approximation ratio* of algorithm A . A graph is a *chain* if it is a path (a multi-chain can have multiple edges between adjacent vertices; in this paper when we refer to a chain it is implied to be a multi-chain). A graph is a *ring* if it is a cycle. A (multi-)graph is a *series-parallel graph* if it can be obtained from a collection of trees by repeatedly replacing edges by parallel edges or by chains. We refer the reader to [11] for formal definitions of complexity classes.

1.2 Related Work

The topics of path protection and network survivability have received significant attention in the last years, especially for all-optical networks. We do not provide a comprehensive survey of the literature here but rather focus on work that closely relates to the optimization problems that we study. Note that as, to the best of our knowledge, the MINIMUM COLOR CONGESTION ROUTING problem has not been studied before, we will review only results related to the MINIMUM COLOR PATH problem.

Carr et al. [3], motivated by a data mining application, introduce the RED-BLUE SET COVER problem: given a set $R = \{r_1, \dots, r_{|R|}\}$ of red elements, a set $B = \{b_1, \dots, b_{|B|}\}$ of blue elements, and a family \mathcal{S} of subsets of $R \cup B$, one seeks a subfamily of \mathcal{S} that covers all blue elements but covers only the minimum possible number of red elements. Carr et al. show that this problem reduces to MINIMUM COLOR PATH. They also show that the RED-BLUE SET COVER can not be approximated within a ratio of $O(2^{\log^{1-\varepsilon} n})$ for any positive ε , unless $NP \subseteq DTIME(n^{\text{polylog}(n)})$, where $n = |\mathcal{S}|^4$. This translates to a similar inapproximability result for MINIMUM COLOR PATH: unless $NP \subseteq DTIME(n^{\text{polylog}(n)})$, no $O(2^{(\log k)^{1-\varepsilon}})$ -approximation algorithm for any positive ε can exist for MINIMUM COLOR PATH, where $k \in \Omega(n^{1/4})$ [14]. Note that the graph in the constructed instance of MINIMUM COLOR PATH in the reduction of RED-BLUE SET COVER to MINIMUM COLOR PATH given in [3] is series-parallel; thus, this hardness result applies already to instances restricted to series-parallel graphs.

Yuan, Varma, and Jue [16] show that the MINIMUM COLOR PATH problem is NP -hard in chains. They propose heuristics for general graphs which they evaluate using experiments on randomly generated instances. They also study variations of MINIMUM COLOR PATH in which for each request two disjoint paths are sought that have the minimum number of total colors, or the minimum color overlap.

Other authors have studied the problem of establishing a spanning tree using as few colors as possible. Chang and Leu [4] prove this problem to be NP -hard and propose two heuristics and an exact exponential-time algorithm. Krumke and Wirth [8] analyze the performance of the two heuristics of [4]. They show that one of the two can be arbitrarily bad while the other achieves a logarithmic approximation ratio. Wan, Chen, and Hu [13] slightly improve the approximation ratio shown by Krumke and Wirth [8].

1.3 Our Results

We begin, in Section 2, by exhibiting trade-offs between the different routing objectives. We show that routing with respect to local reliability can be arbitrarily bad for global reliability and vice versa. Also, we consider failure-oblivious routings and show that they can be arbitrarily bad with respect to the reliability objectives.

In Section 3, we consider the MINIMUM COLOR PATH problem. We give approximation algorithms achieving logarithmic approximation ratio for the special cases of chains, rings, and trees. We also show that this is best possible by providing a matching inapproximability result.

In Section 4, we consider the MINIMUM COLOR CONGESTION ROUTING problem. We obtain inapproximability results for general network topologies using the existing results for the MINIMUM LOAD ROUTING problem. Furthermore, we show that MINIMUM COLOR CONGESTION ROUTING is NP -hard even in the special case of chain networks and can not be approximated within a factor better than 2 unless $P = NP$. Finally, we present a heuristic for MINIMUM COLOR CONGESTION ROUTING that routes the requests sequentially along shortest paths that are computed with respect to edge-weights that are exponential in the current maximum color congestion.

We present experimental results in Section 5. We use integer linear programming formulations of MINIMUM COLOR PATH and MINIMUM COLOR CONGESTION ROUTING to obtain optimal solutions for the two problems.

We note that several technical details and proofs have been omitted from this version due to space limitations.

2 Routing Trade-Offs

2.1 Local vs. Global Reliability

Observe that while the MINIMUM COLOR CONGESTION ROUTING problem tries to minimize the *maximum* number of paths per color, the MINIMUM COLOR PATH problem seeks to minimize the *average* number of paths per color. This renders the two problems considerably different: it is easy to construct instances where the optimal solution for one is suboptimal for the other. We omit the details from this version because of space limitations.

2.2 Oblivious vs. Failure-Aware Routings

Oblivious routing is done without taking into consideration the link failure events of the network, i.e., routing is performed assuming that each edge fails independently of any other. This can be a choice of the network designer in order to simplify the network protocols or it can be because of lack of information. We are interested to see how “bad” such oblivious routings can be with respect to the best possible routings if we take into consideration the failure events of the network. It can be shown that routing under limited knowledge can be arbitrarily bad with respect to both routing objectives. Nevertheless, failure-aware

routings can be arbitrarily bad with respect to traditional routing objectives. It is fairly easy to see that routing with the objective of MINIMUM COLOR PATH can produce routings with very large link-load and very large delays (hop-count). Minimizing the color congestion can also result in similar routings. In Section 5, we will investigate whether such routings occur in practice.

3 Locally Reliable Routings

As we have discussed in Section 1.2, already in series-parallel graphs there can not exist any reasonable approximation algorithm for MINIMUM COLOR PATH, unless $NP \subseteq DTIME(n^{\text{polylog}(n)})$ (i.e., all problems in NP can be solved by deterministic algorithms that run in quasi-polynomial time) [14]. In the simplest topologies, while we can still not hope for constant approximation algorithms, we can achieve logarithmic approximation algorithms, as the following two theorems show.

Theorem 1. *There exists an $O(\log |V|)$ -approximation algorithm for MINIMUM COLOR PATH in chains, rings, and trees.*

Theorem 2. *MINIMUM COLOR PATH can not be approximated within a ratio of $c \log |V|$ for some positive constant c , unless $P = NP$, even in chains.*

4 Globally Reliable Routings

It is easy to see that MINIMUM COLOR CONGESTION ROUTING is a generalization of the MINIMUM LOAD ROUTING problem. Given an instance of MINIMUM LOAD ROUTING we can reduce it to MINIMUM COLOR CONGESTION ROUTING simply by assigning a different color to every edge of the graph. In [1], Andrews and Zhang show that MINIMUM LOAD ROUTING can not be approximated within a ratio of $(\log \log m)^{1-\varepsilon}$ for any positive ε , unless $NP \subseteq ZTIME(n^{\text{polylog}(n)})$, where m is the number of edges of the input graph. We thus obtain the following hardness result for MINIMUM COLOR CONGESTION ROUTING.

Theorem 3. *The problem MINIMUM COLOR CONGESTION ROUTING can not be approximated within a ratio of $(\log \log |E|)^{1-\varepsilon}$ for any positive ε , unless $NP \subseteq ZTIME(n^{\text{polylog}(n)})$.*

Even in the very simple topology of chains, the problem remains NP -hard.

Theorem 4. *MINIMUM COLOR CONGESTION ROUTING is NP -hard in chains.*

The proof uses a reduction from the DOMATIC NUMBER problem. (The *domatic number* of a graph $G = (V, E)$ is the maximum number k such that V can be partitioned into k dominating sets. A dominating set $V' \subseteq V$ for a graph $G = (V, E)$ has the property that every vertex $v \in V$ is in V' or has a neighbor in V' .) One can also show the following:

Theorem 5. MINIMUM COLOR CONGESTION ROUTING *can not be approximated within a ratio of $2 - \varepsilon$ for any positive ε in chains, unless $P = NP$.*

Observe that if each failure event affects only a constant number of links in the network, then we can employ an algorithm for MINIMUM LOAD ROUTING and lose only a constant factor in the approximation ratio:

Theorem 6. *A ρ -approximation algorithm for MINIMUM LOAD ROUTING implies a $c \cdot \rho$ -approximation algorithm for MINIMUM COLOR CONGESTION ROUTING for the case where every color is used in at most c edges.*

Given the inherent intractability of the MINIMUM COLOR CONGESTION ROUTING problem, we propose a heuristic to tackle it in the general case. The heuristic is shown below (Algorithm 1). It computes a shortest path for each of the requests. At every iteration one request is routed. The path assigned to each request is the shortest path with respect to edge-weights which are exponential in the current congestion of each color. When a path is assigned to a request, it does not change until the end of the execution.

Algorithm 1. Heuristic for MINIMUM COLOR CONGESTION ROUTING

Input: Graph $G = (V, E)$, color assignment $\mu : E \rightarrow \{1, \dots, c\}$, set of requests $R = \{(s_1, t_1), \dots, (s_k, t_k)\}$

Output: Set of paths $P = \{p_1, \dots, p_k\}$

 Guess optimal color congestion L^*

$\Lambda := 2L^* \log(2m)$

for $i = 1$ to k **do**

$L_j(i) :=$ Load of color j before routing request i

$w_e(i) := (2 \cdot |E|)^{L_{\mu(e)}(i)/\Lambda}$

$p_i :=$ shortest s_i - t_i path w.r.t. the edge-weights $w_e(i)$

end for

The heuristic is an adaptation of the $O(\log |V|)$ -approximation algorithm for MINIMUM LOAD ROUTING by Aspnes et al. [2]. The difference with the original algorithm is that for MINIMUM COLOR CONGESTION ROUTING we use weights that are exponential in the congestion of each color and not in the link load. Also, note the use of the parameter L^* , which should be equal to the optimal load. This can be guessed by trying all k possible values (since the optimal congestion lies between 1 and $|R| = k$) and then picking the best solution.

5 Experiments

5.1 Goal of the Experiments

The experiments have two goals: First, we want to investigate up to what degree the worst-case trade-off behavior of the instances described in Section 2 occurs in randomly generated instances. Second, we want to see how far from the optimal are the routings returned by the heuristic for MINIMUM COLOR CONGESTION ROUTING.

The concrete questions we want to answer are the following:

- How does the algorithm for MINIMUM LOAD ROUTING of Aspnes et al. [2] perform with respect to the MINIMUM COLOR PATH and MINIMUM COLOR CONGESTION ROUTING objectives?
- How does a simple routing algorithm that routes every request along a shortest path perform in practice with respect to the MINIMUM COLOR PATH, the MINIMUM COLOR CONGESTION ROUTING, and the MINIMUM LOAD ROUTING objectives?
- How does the heuristic algorithm for MINIMUM COLOR CONGESTION ROUTING compare to an optimal algorithm for MINIMUM COLOR CONGESTION ROUTING? Also, how does it perform with respect to the MINIMUM LOAD ROUTING and MINIMUM COLOR PATH objectives? What is the average length of the routings returned by the heuristic and how do they compare to the shortest average length?
- How does an optimal algorithm for MINIMUM COLOR PATH perform with respect to the MINIMUM LOAD ROUTING and MINIMUM COLOR CONGESTION ROUTING objectives? What is the average length of MINIMUM COLOR PATH routings and how do they compare to the shortest average length?

To get optimal solutions for MINIMUM COLOR CONGESTION ROUTING and MINIMUM COLOR PATH we use a general purpose integer linear program solver. The formulations are straight-forward and are omitted from this version.

5.2 Implementation Details

The implementation was done in C++ using LEDA [10] (version 4.3.1) for the basic data types and graph algorithms and CPLEX [7] (version 8.1) for solving the integer linear programs. The code was compiled with the GNU C++ compiler (version 2.95.3) on SunOS 5.8. All experiments were run on a SunFire 480R with 4GB RAM and two 900 MHz processors (our code uses only one processor) each with 8 MB (4ns) L2 Cache. The workstation was available to other users as well during the experiments.

5.3 Experimental Results

For the remainder of this section, we will use the following notation to refer to the tested algorithms: MLR stands for the $O(\log |V|)$ -approximation algorithm for MINIMUM LOAD ROUTING by Aspnes et al. [2]; SPR stands for the simple algorithm that routes each request through some shortest path; MCCR stands for our heuristic (Algorithm 1) for MINIMUM COLOR CONGESTION ROUTING; MCCRIIP stands for the ILP-based optimization algorithm for MINIMUM COLOR CONGESTION ROUTING; finally, MCPIIP stands for the ILP-based optimization algorithm for MINIMUM LOAD ROUTING.

We begin with a brief discussion on how the experiments to be performed were chosen. An important limitation in the design of the experiments has been the running time of MCCRIIP and MCPIIP, i.e., the CPLEX optimization time.

We chose to set an upper bound of one hour to the optimization time in order to be able to perform a large number of experiments in a reasonable amount of time. If this amount of time elapses, CPLEX either returns a possibly sub-optimal integer solution if it has found one or exits without returning a solution. In a preliminary experiment on instances with 50 nodes, 500 edges, and 50 requests, MCCRIP exceeded the time limit in all 20 instances that we tested with more than ten colors without producing an integer solution. MCPIP was faster: in the same experiment it found an optimal solution in seven out of 20 runs and returned a sub-optimal integer solution when it exceeded the time limit in the remaining 13 runs. In order to be able to get optimal or good sub-optimal solutions from the ILP optimizations we restricted the experiments to relatively small-sized instances. We performed experiments on networks with 15–65 nodes, 25–95 edges, 10–90 colors, using 50 requests for each instance. Each experiment was performed ten times and the plots that follow show the average of these executions. With these instance sizes, and with a time limit of one hour, we were able to solve all instances, although some not optimally. We give plots of the running times of MCCRIP and MCPIP in Figures 1(a) and 1(b). Both algorithms require more computation time as the number of colors decreases. MCPIP runs generally faster than MCCRIP.

In Figures 1(c) and 1(d) we have plotted the maximum color congestion and the average number of colors per path in the solutions of MLR along with the corresponding solutions from MCCRIP and MCPIP. The network consists of 15 nodes with edges ranging from 15 to 95, 50 requests, and 10 (Figure 1(c)) and 90 colors (Figure 1(d)). One notices immediately that the number of colors per path in the routings of MLR is roughly the same with that of MCPIP. The actual number is pretty low, around and below two colors per path (and decreases as the graph becomes denser). This is due to the small size of the instances (15 nodes) and their small diameter. (The number of paths per color increases as we increase the network size.) With respect to the color congestion, the routings from MLR are by around a factor of $3/2$ away from MCCRIP in sparse graphs, but improve as the graphs get denser. We can conclude that routing to minimize the load is a reasonable choice for producing routings with small number of colors per path. Minimum load routings, however, can have large color congestion, especially in sparse graphs.

In Figures 1(e) and 1(f), we have plotted the maximum color congestion, the maximum load, and the average number of colors per path in the solutions of SPR along with the corresponding values from MCCRIP, MLR, and MCPIP. The network consists of 15 nodes with edges ranging from 15 to 95, 50 requests, and 10 (Figure 1(e)) and 90 colors (Figure 1(f)). Algorithm SPR performs similarly to MLR regarding the color congestion and the number of colors per path. The maximum load of SPR is by around a factor of $3/2$ away from MLR in sparse graphs, but gets closer to MLR as the graphs get denser.

In Figures 2(a) and 2(b), we have plotted the maximum color congestion, the maximum load, the average number of colors per path, and the average hop-length in the solutions of MCCR along with the corresponding objective values

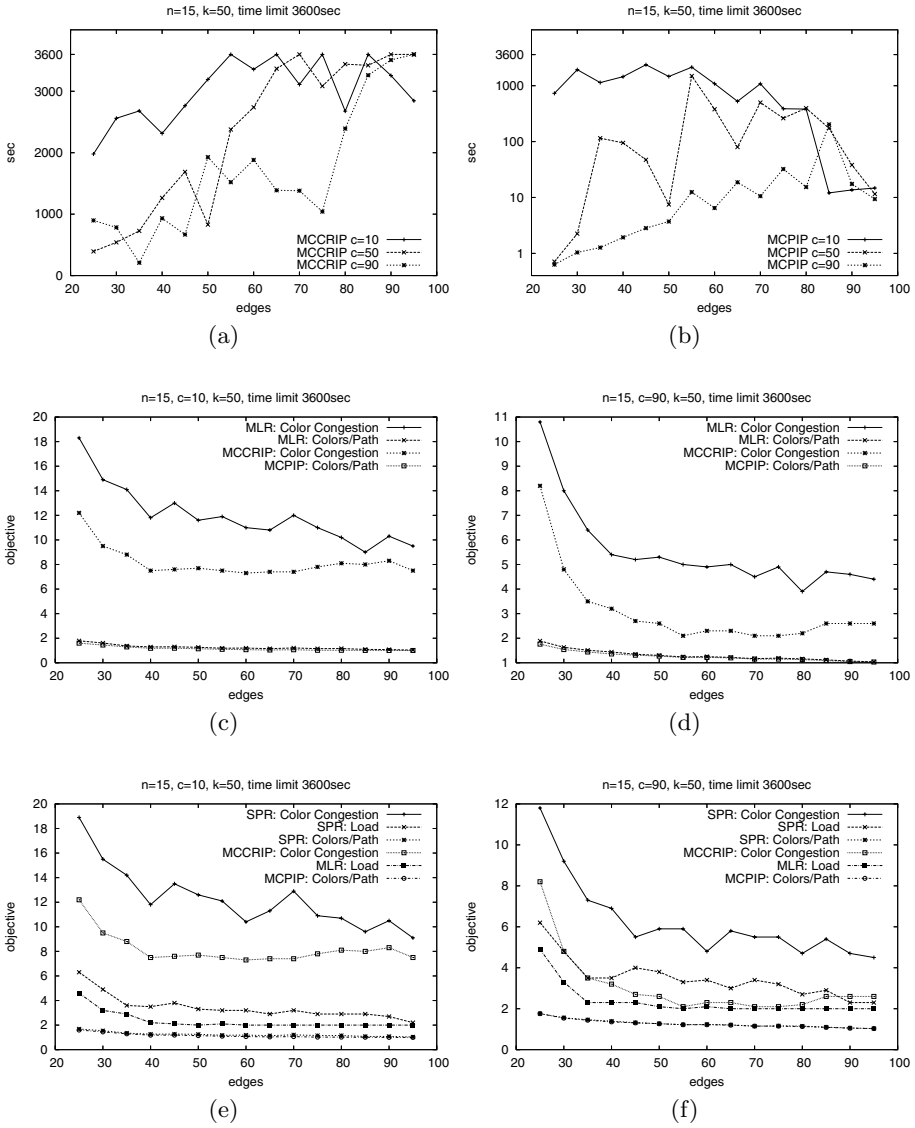


Fig. 1. (a), (b): Running times of MCCRIP and MCPIP. (c), (d): Performance of MLR with respect to the MINIMUM COLOR CONGESTION ROUTING and MINIMUM COLOR PATH objectives. (e), (f): Performance of SPR with respect to the MINIMUM COLOR CONGESTION ROUTING, MINIMUM COLOR PATH, and MINIMUM LOAD ROUTING objectives.

from MCCRIP, MLR, MCPIP, and SPR. The network consists of 15 nodes with edges ranging from 15 to 95, 50 requests, and 10 (Figure 2(a)) and 90 (Figure 2(b)) colors. The routings of MCCR have color congestion which is really close

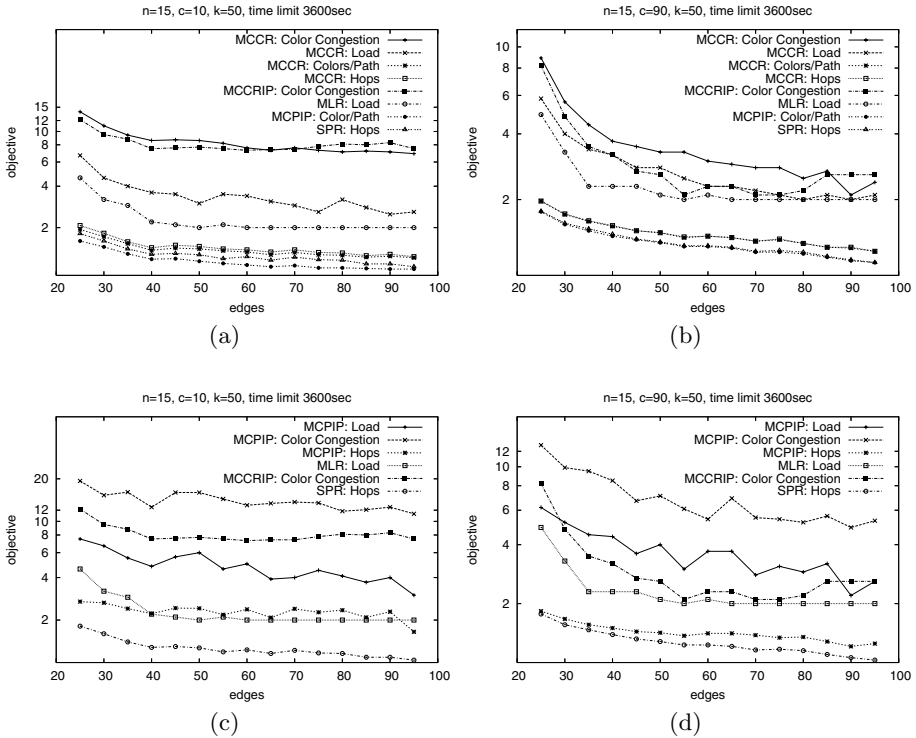


Fig. 2. (a), (b): Performance of MCCR with respect to the maximum color congestion, the maximum load, the average number of colors per path, and the average hop-length. (c), (d): Performance of MCP/IP with respect to the maximum color congestion, the maximum load, and the average hop-length.

to that of MCCRIP. Actually, in some instances MCCR performs better than MCCRIP. This happens mostly in instances with ten colors which are the hardest to solve and MCCRIP more often than not exceeds the time limit of one hour and returns a suboptimal solution. Furthermore, observe that the load of MCCR is close to that of MLR (and gets closer in dense graphs). Also, the number of colors per path in the solutions of MCCR is very close to that of MCP/IP, and the average hop-length is also close to that of SPR. Thus, the heuristic for MINIMUM COLOR CONGESTION ROUTING performs very well with respect to the maximum color congestion objective and produces reasonable routings with respect to the maximum load, the average number of colors per path, and the average hop-length objectives.

In Figures 2(c) and 2(d), we have plotted the maximum color congestion, the maximum load, and the average number of hops in the solutions of MCP/IP along with the corresponding objective values from MCCRIP, MLR, and SPR. The network consists of 15 nodes with edges ranging from 15 to 95, 50 requests, and 10 (Figure 2(c)) or 90 (Figure 2(d)) colors. MCP/IP produces routings with

color congestion which is away from that of MCCRIP by a factor of around $3/2$. The load of MCPIP is also higher than that of MLR and in sparse instances with many colors it even exceeds the color congestion of MCCRIP. It remains however within a factor of 2 from the load of MLR. As expected, MCPIP produces routings with average hop-length close to that of SPR in instances with 90 colors; in instances with 10 colors MCPIP produces routings with greater average hop-length than that of SPR (but within a factor of 2). Hence, MCPIP performs reasonably well with respect to the maximum load, the maximum color congestion, and the average hop-length objectives.

6 Conclusion

We have studied routing problems in networks with generalized failure events. We have analyzed the complexity and the approximability of a local (MINIMUM COLOR PATH) and a global (MINIMUM COLOR CONGESTION ROUTING) routing objective for several classes of networks. We have also exhibited trade-offs between the two objectives as well as between the reliability objectives and traditional routing objectives such as minimizing the maximum congestion. Experiments on random instances have shown that the worst-case trade-offs do not occur in practice. Our experiments have also shown that the heuristic algorithm that we proposed to tackle MINIMUM COLOR CONGESTION ROUTING gives routings with maximum color congestion, maximum load, and average number of paths all very close to the optimal ones.

Several interesting directions for future work can be followed. From an algorithmic point of view it is very interesting to further study the MINIMUM COLOR CONGESTION ROUTING problem. Providing a logarithmic approximation algorithm for the general case, or a better inapproximability result would be a first step in that direction. Approximation algorithms for specific graph classes such as chains or rings would also be of great interest. From a practical point of view, it is interesting to study these objectives under additional routing objectives. For example, requiring disjointness among certain requests (e.g., for every request one might want to compute a primary path and a disjoint back-up path) is an important extension to consider.

References

1. M. Andrews and L. Zhang. Hardness of the undirected congestion minimization problem. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC '05)*, 2005.
2. J. Aspnes, Y. Azar, A. Fiat, S. Plotkin, and O. Waarts. On-line routing of virtual circuits with applications to load balancing and machine scheduling. *Journal of the ACM*, 44(3):486 – 504, 1997.
3. R. D. Carr, S. Doddi, G. Konjevod, and M. Marathe. On the red-blue set cover problem. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '00)*, pages 345 – 353, 2000.

4. R.-S. Chang and S.-J. Leu. The minimum labeling spanning trees. *Information Processing Letters*, 63(5):277–282, 1997.
5. S. Chaudhuri, G. Hjálmtýsson, and J. Yates. Control of lightpaths in an optical network. IETF Internet Draft, March 2000.
6. H. Choi, S. Subramaniam, and H.-A. Choi. On double-link failure recovery in WDM optical networks. In *Proceedings IEEE INFOCOM 2002, The 21st Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 2, pages 808–816, 2002.
7. ILOG CPLEX. *CPLEX 8.1*, 2004. <http://www.cplex.com/>.
8. S. Krumke and H.-C. Wirth. On the minimum label spanning tree problem. *Information Processing Letters*, 66(2):81–85, 1998.
9. M. Médard, R. A. Barry, S. G. Finn, W. He, and S. Lumetta. Generalized loop-back recovery in optical mesh networks. *IEEE/ACM Transactions on Networking*, 10(1):153–164, 2002.
10. K. Mehlhorn and S. Näher. *LEDA: A platform for combinatorial and geometric computing*. Cambridge University Press, 1999.
11. C. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.
12. R. Ramaswami and K. N. Sivarajan. *Optical Networks: A Practical Perspective*. Morgan Kaufmann Publishers, 2nd edition, 2002.
13. Y. Wan, G. Chen, and Y. Xu. A note on the minimum label spanning tree. *Information Processing Letters*, 84:99–101, 2002.
14. H.-C. Wirth. *Multicriteria Approximation of Network Design and Network Upgrade Problems*. PhD thesis, University of Würzburg, 2001.
15. Y. Xin and G. N. Rouskas. A study of path protection in large-scale optical networks. *Photonic Network Communications*, 7(3):267–278, 2004.
16. S. Yuan, S. Varma, and J. P. Jue. Minimum-color path problems for reliability in mesh networks. In *Proceedings of IEEE INFOCOM 2005, The 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2005.