

Thesis Proposal

Efficient Algorithms for Alternate Paths and Bridges in Networks and Geometric Graphs

Amit M. Bhosle

Department of Computer Science

University of California

Santa Barbara, CA 93106

bhosle@cs.ucsb.edu

1 Introduction

Transient single element (node/link) failures dominate the failures in today's large communication networks like the Internet [MIB⁺04]. Thus, having the routing paths globally recomputed on a failure does not pay off since the failed element recovers fairly quickly, and the recomputed routing paths need to be discarded. A popular technique for dealing with such issues is that of using a *proactive recovery scheme*. Such a scheme works typically by precomputing a set of alternate paths that avoid a node or a link, and then using these alternate paths to re-route the traffic when the failure actually occurs. The information of the failed network element is not advertised across the network in the hope that it is transient, and the failed element would recover shortly.

In this thesis, we consider the two basic versions of this problem: (a) single link failure and (b) single node failure, and present efficient algorithms for computing alternate paths for such failure scenarios. We also study some variations of these problems, that include variations in the graph properties (primarily, the edge weights).

We achieve near-optimal upper bounds on some problems, and establish non-trivial lower bounds on some others. We use the alternate paths computed by our algorithms in the design of a local re-routing based proactive recovery scheme that can be used for handling transient single node failures in communication networks. The protocol also works for handling transient single link failures.

Apart from investigating graph theoretical problems dealing with finding good paths joining components *disconnected* by the failures, we also study a related problem in computational geometry. This problem deals with finding optimal *bridges* under a practically motivated metric connecting two disjoint simple polygons. The metric minimizes the maximum (geodesic) distance from any point in one polygon to any point in the other polygon. Problems similar to this have been studied in the past, but there were no previous results known for this problem. We present an exact algorithm, and a few approximation algorithms, including a fully polynomial time approximation scheme for this problem.

1.1 Node Failure Recovery

The single node failure recovery problem is formally defined as follows:

SNFR: Given a biconnected undirected edge weighted graph $G = (V, E)$, and the shortest paths tree $\mathcal{T}_s(G)$ of a node s in G where $\mathcal{C}_x = \{x_1, x_2, \dots, x_{k_x}\}$ denotes the set of *children* of x in \mathcal{T}_s , for each node $x \in V$ and $x \neq s$, find a path from $x_i \in \mathcal{C}_x$ to s in the graph $G = (V \setminus \{x\}, E \setminus E_x)$, where E_x is the set of edges adjacent to x .

In other words, for each node x in the graph, we are interested in finding alternate paths from each of its children to the source node s when the node x *fails*. Note that the problem is not well defined when node s fails.

Wang and Gao's Backup Route Aware Protocol (BRAP) [WG08] uses some precomputed backup routes in order to handle transient single *link* failures. The BRAP protocol depends on the availability of *reverse paths* at each node. However, the authors of [WG08] do not discuss the computation of these reverse paths. The alternate paths as defined in our SNFR problem qualify as these reverse paths.

For the single node failure case, we present an $O(m \log n)$ time algorithm [BG08]. Although the alternate paths computed by this algorithm are not optimal, we demonstrate via simulation results that they are very close

to optimal. For randomly generated graphs, our algorithm computes paths that are within 15% of the optimal. We compare our algorithm’s performance with that of [ZNY⁺05], wherein the authors have presented a proactive recovery scheme for dealing with transient single node failures. Our algorithm is faster by at least an order of magnitude than the algorithm of [ZNY⁺05], while our paths lengths are usually comparable, and sometimes better.

Using the alternate paths computed by the SNFR algorithm, we design a protocol to be used in communication for recovering from transient single node failures. Each node is assigned as escape edge by the SNFR algorithm. This edge identifies an alternate path to the final destination that does not use the node’s failed parent. The failure recovery protocol embeds the information of the escape edge in the recovery message, and re-routes the messages along the alternate paths. Note that in the failure free case, the protocol does not consume any additional resources.

We further evaluate two heuristics for using shorter alternate paths. One of these is based on storing multiple escape edges for each node, while the other tries to optimize the alternate paths by using short-cuts (if possible) while the message is still within the original subtree of the failed node.

1.2 Link Failure Recovery

The single link failure recovery problem is formally defined as follows:

SLFR: Given a 2-connected graph G , a specified node s , and a shortest paths tree $\mathcal{T}_s = \{e_1, e_2, \dots, e_{n-1}\}$ of s , where $e_i = (x_i, y_i)$ and $x_i = \text{parent}_{\mathcal{T}_s}(y_i)$, find a shortest path from y_i to s in the graph $G \setminus e_i$ for $1 \leq i \leq n - 1$.

We present an algorithm [BG03] for the SLFR problem that computes optimal alternate paths in near-optimal running time of $O(m + n \log n)$, where m and n denote the number of edges and nodes in the graph. This solution improves upon an $O(n^3)$ solution presented by [SL00]. Also, this algorithm can be trivially modified to find the anti-block vital edge of a shortest path in $O(m + n \log n)$ time. The anti-block vital edge problem was studied in [SXX07] wherein the authors presented an $O(n^3)$ algorithm for the problem.

We derive faster algorithms [Bho05, BG04a] for some variations of the basic problem including the cases

where (a) graph is unweighted, (b) edge weights are integers and (c) graph is planar. For these restricted graphs, we achieve the same improvements for the *most vital edge of a shortest path* problem [MMG89], also known as the replacement-paths problem [HS01, HSB07], defined as follows:

MVA-SP : Given a biconnected, undirected, edge weighted graph $G(V, E)$, two nodes $s, t \in V$, and a shortest path $\mathcal{P}_G(s, t) = \{e_1, e_2, \dots, e_p\}$ from s to t in G , compute the shortest path from s to t in each of the p graphs $G \setminus e_i$ for $1 \leq i \leq p$, where $G \setminus e_i$ represents the graph G with the edge e_i removed.

We present linear time algorithms for the cases where the graph is *unweighted*, or *planar*. For integer edge weighted graphs, our algorithm runs in $O(m + T_{\text{sort}}(n))$, where $T_{\text{sort}}(n)$ is the time required to sort n integers, and is currently $O(n \log \log n)$ [Han02].

We further adapt our single link failure recovery algorithm to find the *k-replacement-edges* for the tree edges of a minimum spanning tree in $O(m + n \log n)$ time, improving the previous bound of $O(n^2)$ [Lia01]. This directly improves the complexity of Shen's randomized algorithm [She99] for the *k most vital arcs of a minimum spanning tree* problem from $O(n^2)$ to near-optimal $O(m + n \log n)$.

For the directed version of the single link failure recovery problem, we establish a lower bound of $\Omega(\min(m\sqrt{n}, n^2))$ in the path comparison model for shortest path algorithms introduced in [KKP91] and further explored in [HSB07].

Apart from the single link failure scenario, we study the problem of computing replacement paths between two given nodes, s and t , if any two edges of the original $s - t$ shortest path fail. We present an $O(n^3)$ algorithm for this problem, based on a new algorithm for the directed version of the replacement paths problem [BG04b]. We also show that the problem admits a lower bound of $\Omega(mn)$ in the path comparison model for shortest path algorithms.

Our single node failure recovery protocol [BG08] can be modified to handle single link failures as well. The alternate paths used by the failure insensitive routing protocol proposed in [LYN⁺04] for handling single link failures were shown (experimentally) to be within 15% of the optimal. Also, their path computation algorithm is based on recomputation, and has a time complexity of $\Omega(mn \log n)$. On the other hand, our algorithm runs in

near-optimal $O(m + n \log n)$ time, and uses optimal alternate paths.

1.3 Connecting Polygons

Apart from the above graph problems, we have also studied the problem of finding optimal bridges for connecting two simple polygons.

Given two simple polygons P and Q , let the weight of a bridge (p, q) , with $p \in \rho(P)$ and $q \in \rho(Q)$, where $\rho()$ defines the compact region enclosed by the boundary of the polygon, between the two polygons be defined as $gd(p, P) + d(p, q) + gd(q, Q)$, where $d(p, q)$ is the Euclidean distance between the points p and q , and $gd(x, X)$ is the geodesic distance between x and its geodesic furthest neighbor on X .

The problem of finding the optimal bridge connecting two *convex* polygons has been solved to optimality. The problem was first studied by Cai, Xu and Zhu [CXZ99] for the case when the polygons are convex and they presented an $O(n^2 \log n)$ time algorithm. Because this problem is related to other geometric problems (e.g. diameter problems, minimum separations problems, and minimum spanning tree problems), much progress was made immediately after the problem was posed. For the convex polygon case, different optimal (linear time) algorithms have been developed independently by Bhattacharya and Benkoczi [BB01], Tan [Tan00], and Kim and Shin [KS01]. The high-dimensional version of the problem has been studied in [Tan00, Tok01].

For the problem of finding an optimal bridge between two simple polygons, we show that an optimal bridge always exists such that the endpoints of the bridge lie on the boundaries of the two polygons. Using this property, we present an algorithm [BG05] to find an optimal bridge (of minimum weight) in $O(n^2 \log n)$ time. We present an approximation algorithm that, given any positive integer k , constructs a bridge with objective function value within $(1 + \frac{2}{k+1})$ of optimal in $O(kn \log kn)$ time. We also present a polynomial time approximation scheme that for any $\epsilon > 0$ generates a bridge with objective function within a factor of $1 + \epsilon$ of optimal in $O(kn \log kn)$ time, where $k = 2 * \lceil \frac{1}{\log(1+\epsilon)} \rceil$.

2 Future Work

We now discuss the problems that we are currently investigating, or plan on studying over the course of the next few months. Our primary focus would be on improving some of our current results, and/or study variations of the problems we have solved in the past.

2.1 SNFR Problem

As mentioned earlier, the paths computed by our SNFR algorithm are not optimal alternate paths. We plan on investigating the existence of an efficient algorithm that computes the optimal alternate paths required by the SNFR problem. No non-trivial algorithm for the problem is known. The trivial algorithm is based on recomputation, and has the time complexity of $O(mn + n^2 \log n)$. Does a theoretically or experimentally faster algorithm exist for this problem?

Another interesting problem is to design approximation algorithms for the SNFR problem. None of the known algorithms guarantee the computed paths to be within a constant factor of the optimal paths. Consequently, an algorithm running faster than the naive one, and which computes paths that are guaranteed to be close to optimal would be a welcome solution. We plan on investigating these problems, starting with graphs that are *complete* and *metric*.

2.2 Distributed Algorithms for SLFR/SNFR Problems

All known algorithms for computing alternate paths that avoid a node or a link are centralized algorithms that work using the information of the entire network graph. These include the algorithms presented in [BG08, LYN⁺04, SL00, WG08, ZNY⁺05]. For small networks, centralized algorithms are usually acceptable, but not for large ones. The drawbacks are more pronounced if other steps executed at the network setup time are distributed algorithms. Also, for large networks, centralized algorithms require proportionally higher memory and other computing resources available at a single computer.

We are developing a distributed algorithm for the SNFR/SLFR problems, and hope to come up with an efficient

solution for these problems. The key properties and structure of the alternate paths would be the same as in the case of the centralized solutions, but each node would have to work with only its local view of the network.

2.3 Beyond Single Network Element Failures

An obvious generalization of the SNFR and SLFR problems is the two-element failure cases. In the [BG04b] we considered the failure of two edges of a particular $s - t$ shortest path. However, for the case where *any* two network elements (nodes/edges) can fail has not been considered. Using the SNFR or SLFR algorithm, the two-element failure versions of the problems can be solved in $O(mn + n^2 \log n)$ time. A trivial lower bound for these problems is $\Omega(n^2)$, which is significantly smaller than the upper bound. We propose to investigate whether this gap can be narrowed in either direction.

How about the case where the failed elements are somehow related to each other? For instance, does the case where the two failed node are connected by a direct edge admit a more efficient solution? What if a group of nodes failed where the group is comprised of a node and all of its k -hop neighbors, for a fixed k . Failure of a single node and a single edge would also be an interesting problem to look at.

Also, can the algorithm for the edge-pairs replacement paths problem [BG04b] be generalized to the node version where replacement paths need to be computed when any two *nodes* of a particular $s - t$ path fail?

References

- [BB01] B. Bhattacharya and R. Benkoczi. On computing the optimal bridge between two convex polygons. *Inform. Proc. Lett.*, 79(5):215–221, 2001.
- [BG03] A. M. Bhosle and T. F. Gonzalez. Efficient algorithms for single link failure recovery and its application to atm networks. In *Proc. 15th IASTED Intl. Conf. on Parallel and Distributed Computing and Systems*, 2003.
- [BG04a] A. M. Bhosle and T. F. Gonzalez. Algorithms for single link failure recovery and related problems. *Journal of Graph Algorithms and Applications*, 8(2):275–294, 2004.

- [BG04b] A. M. Bhosle and T. F. Gonzalez. Replacement paths for pairs of shortest path edges in directed graphs. In *Proc. 16th IASTED Intl. Conf. on Parallel and Distributed Computing and Systems*, 2004.
- [BG05] A. M. Bhosle and T. F. Gonzalez. Exact and approximation algorithms for finding an optimal bridge connecting two simple polygons. *International Journal of Computational Geometry and Applications*, 15(6):609–630, 2005.
- [BG08] A. M. Bhosle and T. F. Gonzalez. Efficient algorithms and routing protocols for handling transient single node failures. *TR-2008-06, Computer Science, UCSB*, 2008.
- [Bho05] A. M. Bhosle. Improved algorithms for replacement paths problems in restricted graphs. *Operations Research Letters*, 33:459 – 466, 2005.
- [CXZ99] L. Cai, Y. Xu, and B. Zhu. Computing the optimal bridge between two convex polygons. *Inform. Proc. Lett.* 69, pages 127- 130, 1999.
- [Han02] Y. Han. Deterministic sorting in $O(n \log \log n)$ time and linear space. In *34th ACM STOC*, pages 602–608. ACM Press, 2002.
- [HS01] J. Hershberger and S. Suri. Vickrey prices and shortest paths: What is an edge worth? In *42nd IEEE FOCS*, pages 252-259, 2001.
- [HSB07] J. Hershberger, S. Suri, and A. M. Bhosle. On the difficulty of some shortest path problems. *ACM Transactions on Algorithms*, 3(1), 2007.
- [KKP91] D. R. Karger, D. Koller, and S. J. Phillips. Finding the hidden path: Time bounds for all-pairs shortest paths. In *32nd IEEE FOCS*, pages 560-568, 1991.
- [KS01] S. K. Kim and C. S. Shin. Computing the optimal bridge between two polygons. *Theory of Computing Systems*, 34(4):337–352, 2001.
- [Lia01] W. Liang. Finding the k most vital edges with respect to minimum spanning trees for fixed k. *Discrete Applied Mathematics*, 113:319–327, 2001.

- [LYN⁺04] S. Lee, Y. Yu, S. Nelakuditi, Z.-L. Zhang, and C.-N. Chuah. Proactive vs reactive approaches to failure resilient routing. *In Proc. of IEEE INFOCOM*, 2004.
- [MIB⁺04] A. Markopulu, G. Iannaccone, S. Bhattacharya, C. Chuah, and C. Diot. Characterization of failures in an ip backbone. *In Proc. of IEEE INFOCOM*, 2004.
- [MMG89] K. Malik, A. K. Mittal, and S. K. Gupta. The k most vital arcs in the shortest path problem. *In Oper. Res. Letters*, pages 8:223-227, 1989.
- [She99] Hong Shen. Finding the k most vital edges with respect to minimum spanning tree. *Acta Informatica*, 36(5):405–424, 1999.
- [SL00] R. Slosiar and D. Latin. A polynomial-time algorithm for the establishment of primary and alternate paths in atm networks. *In IEEE INFOCOM*, pages 509-518, 2000.
- [SXX07] Bing Su, Qingchuan Xu, and Peng Xiao. Finding the anti-block vital edge of a shortest path between two nodes. *In COCOA*, pages 11–19, 2007.
- [Tan00] X. Tan. On optimal bridges between two convex regions. *Inform. Proc. Lett.*, 76:163–168, 2000.
- [Tok01] T. Tokuyama. Efficient algorithm for the minimum diameter bridge problem. *Lect. Notes Comput. Sci.*, 2098:362–369, 2001.
- [WG08] F. Wang and L. Gao. A backup route aware routing protocol - fast recovery from transient routing failures. *In INFOCOM*, 2008.
- [ZNY⁺05] Z. Zhong, S. Nelakuditi, Y. Yu, S. Lee, J. Wang, and C.-N. Chuah. Failure inferencing based fast rerouting for handling transient link and node failures. *In Proc. of IEEE INFOCOM*, pages 4: 2859-2863, 2005.