This paper was presented at RNDM 2013 - 5^{th} International Workshop on Reliable Networks Design and Modeling, co-located with ICUMT 2013 Congress (Sept. 10-12, 2013, Almaty, KZ)

Network Design and Optimisation Based on Cost and Algebraic Connectivity

Mohammed J.F. Alenazi*, Egemen K. Çetinkaya*, and James P.G. Sterbenz*†
*Information and Telecommunication Technology Center
Department of Electrical Engineering and Computer Science
The University of Kansas, Lawrence, KS, 66045, USA
{malenazi, ekc, jpgs}@ittc.ku.edu
†School of Computing and Communications (SCC) and InfoLab21
Lancaster LA1 4WA, UK
jpgs@comp.lancs.ac.uk
www.ittc.ku.edu/resilinets

***Abstract*—Network design and optimisation has been one of the major focuses of the research community over the past decades. Connectivity of topologies can be improved by simply adding links; however, this incurs cost for addition of links for increased resilience. Therefore, topological design and optimisation requires developing algorithms so that a designer can select optimum parameters to achieve resilience in the least costly manner. In this paper, we develop a heuristic algorithm that optimises a topology based on algebraic connectivity metric that is defined as the second smallest eigenvalue of the Laplacian matrix. Furthermore, the connectivity of a topology is improved based on the available budget, for which we capture network cost in terms of euclidian distance between two connected nodes. We apply our algorithm on three realistic sets of backbone service provider graphs and compare the utility of our algorithm. The heuristic algorithm we introduce in this paper optimises topologies and is computationally less costly than an exhaustive optimisation.**

***Keywords*-Network design, optimisation, heuristic algorithm; Network cost model; Network resilience, connectivity, robustness, dependability, reliability; Algebraic connectivity; Backbone network**

I. Introduction and Motivation

Networks in general, and communication networks in particular, are prone to a variety of challenges and attacks that can have costly consequences. However, network connectivity can be improved with careful planning and optimisation, and the impact of such challenges can be reduced. The design and optimisation of cost-efficient networks that are resilient against challenges and attacks has been studied by many researchers over the past few decades, but the resilient network design problem is NP-hard.

In this paper, we approach resilient network design from a graph theoretic perspective. We develop a *heuristic algorithm* that improves the connectivity of a graph in terms of the *algebraic connectivity* metric by adding links. Algebraic connectivity $a(G)$ is defined as the second smallest eigenvalue of the Laplacian matrix [1] and it is widely used for topological optimisations [2]–[4]. A secondary objective of our algorithm is to select the links that improve the algebraic connectivity of the graph in the least costly fashion in which we capture the cost of network as the total link length. Furthermore, we parameterise our optimisation algorithm such that connectivity and cost are weighted depending on a cost-effect parameter named γ.

The heuristic to increase algebraic connectivity in a graph is based on adding links to the nodes that have least incident links (i.e. minimal degree nodes) [2], [4]. Our parameterised heuristic algorithm identifies and selects the links that increases the algebraic connectivity of a graph depending on the available budget. Moreover, the search of the optimal links is computationally less expensive in our algorithm compared to an exhaustive search. We use three commercial service provider networks (AT&T, Level 3, and Sprint) with their corresponding physical- and logical-level graphs to analyse our algorithm. Our algorithm provides the cost-efficient new links to improve a network's resilience measured by the algebraic connectivity metric.

The rest of the paper is organised as follows: We present brief background on network design and optimisation in Section II. The assumptions, objective functions, and our heuristic algorithm is presented in Section III. The dataset for the communication networks as well as evaluation of these topologies using our algorithm is presented in Section IV. Finally, we summarise our findings as well as propose future work in Section V.

II. Background and Related Work

Network design is a NP-hard problem [5], [6] that has been studied in the past decades by many network researchers [7]–[16]. The design process includes constructing the network from the ground up including placement of nodes [10], [11] and providing connectivity among nodes to enable services. The optimisation process includes improvement of the network for one or multiple objectives. Network optimisation can be accomplished by means of rewiring while keeping the number of edges constant [4] or by means of adding new links to improve the connectivity of graphs [2]. Moreover, the design process is different for backbone and access networks, since the topological structure of these networks fundamentally differ [10]–[12].

978-1-4799-1177-6/13/$31.00

Network design and optimisation objectives are cost, capacity, reliability, and performance [9]–[11]. Network cost is incurred by the number of nodes required, capacity of nodes required, and number of links. Previously, we provided a network cost model as:

$$C_{i,j} = f + v \times d_{i,j} \tag{1}$$

where f is the fixed cost associated with the link (including termination), v is the variable cost per unit distance for the link, and $d_{i,j}$ is the length of the link [17], [18]. Moreover, in a modest attempt to capture the total cost of fibre topologies, if we assume that the fibre length dominates wide-area network cost and ignore the fixed cost associated with each link, the network cost can be written as:

$$C = \sum_i l_i \tag{2}$$

where l_i is the length of the i-th link [19], [20].

Topological connectivity is another objective that can be measured by means of many graph metrics such as average degree, betweenness, closeness, and graph diversity [4], [7], [8], [21]–[23]. In this paper, we measure the connectivity of a graph in terms of algebraic connectivity metric. Algebraic connectivity $a(G)$ is defined as the second smallest eigenvalue of the Laplacian matrix [1]. The Laplacian matrix of G is: $L(G) = D(G) - A(G)$ where $D(G)$ is the diagonal matrix of node degrees, $d_{ii} = \deg(v_i)$, and $A(G)$ is the symmetric adjacency matrix with no self-loops. The algebraic connectivity of a complete graph (i.e. full mesh) is n where n is the number of nodes, and it is 0 for a disconnected graph with more than one component.

Topology design using algebraic connectivity has been studied by several researchers [2]–[4]. It has been shown that algebraic connectivity is more informative and accurate than average node degree when characterising network resilience [3]. Moreover, we have shown algebraic connectivity [20], [24] and diversity [23] are predictive of flow robustness of graphs. Three synthetically generated topologies (i.e. Watts-Strogatz, Gilbert, Barabási-Albert) have been optimised using edge rewiring in which the objective is to increase the algebraic connectivity [4]. It was shown that algebraic connectivity increases the most if edges are rewired between weakly connected nodes. Another study optimised synthetically generated Erdős-Rényi and Barabási-Albert graphs in terms of adding links to the existing topology [2]. It was concluded that adding links between a low degree node and a random node is computationally less expensive than an exhaustive search. In this paper, we present an algorithm for topological optimisation in terms of adding links, which maximises algebraic connectivity and aims to choose links so that the cost is minimal among given choices.

III. Topology Optimisation Algorithm

In this section, we describe our algorithm that optimises connectivity and cost of a topology. Our heuristic algorithm is implemented using Python. Furthermore, we assume that node locations are given for a graph to apply optimisation algorithm, as would be the case for a deployed service provider.

A. Objectives

The objective of this algorithm is to identify the best links to be added to improve the connectivity of the graph. In this paper, we use algebraic connectivity as a measure of connectivity, but we note that any graph connectivity property, such as average node degree, clustering coefficient, or diversity can be used with or instead. For example, the clustering coefficient can be used to replace the algebraic connectivity or both the clustering coefficient and the algebraic connectivity can be used with a tuning parameter to control their effect in selecting the links.

B. Algorithm

The topology optimisation algorithm has three inputs: an input graph G_i, a number of required links L_r, and a cost-effect parameter γ. The input graph G_i has a number of nodes n_i with a number of links l_i. The number of required links L_r is the number of links that should be added to the graph. The cost-effect parameter γ is a tuning parameter between cost and algebraic connectivity. When $\gamma = 0$, the cost term of the rank function is neglected since it is zeroed. As a result, the algorithm selects the link that maximises the algebraic connectivity. On the other hand, when $\gamma = 1$, the algebraic connectivity is neglected and the least link cost is selected in each iteration. The algorithm adds links to the graph with L_r iterations. To keep track of the selected links in each iteration, the algorithm adds these links to a list. In each iteration, the algorithm starts by adding the selected links from previous iterations to the graph. Then, the rank value is computed for each candidate link and the link with the maximum rank value is selected to be added. A ranking function is used to select the best candidate in each iteration. The rank value r is computed using:

$$r = (1 - \gamma)a(G) + \gamma(1 - C) \tag{3}$$

where C represents the length of the ranked link. This algorithm uses four functions: cost function $\mathrm{cost}(L)$, algebraic connectivity function $\mathrm{algConn}(G)$, $\mathrm{maxLink}(D)$, and $\mathrm{candidate}(G)$. The cost function $\mathrm{cost}(L)$ returns the cost of adding a link L. In this paper, the cost is defined as the euclidean distance between the two ends of the link. The algebraic connectivity function $\mathrm{algConn}(G)$ takes a graph G and returns the second smallest eigenvalue of its Laplacian matrix. The $\mathrm{maxLink}(D)$ function returns the maximum ranked link. The $\mathrm{candidate}(G)$ takes a graph G as input and returns a set of candidate links to be added to the graph. The candidate links are a set of links that are examined every time a link is added to a graph. One option to use for the candidate links is the set of complement links of a graph is denoted as $\bar{G}$, which can be determined as the set of links in full mesh subtracted from the current links in a graph G. The number of complement links (cf. shown in column 4 Table II) is computed as:

$$\frac{n_i(n_i-1)}{2} - l_i \tag{4}$$

However, this number is computationally expensive as the number of nodes n_i gets larger, which results in a complexity of $O(L_r n_i^2)$. In an attempt to decrease the number of candidate links, we only examine the links connected to the lowest degree node in the graph. As a result, the algorithm complexity decreases to $O(L_r n_i)$.

Both the $\mathrm{algConn}(G)$ and $\mathrm{cost}(L)$ functions are normalised to have a maximum value of one. Since the theoretical maximum value for the algebraic connectivity of a given graph is the number of its nodes, it is normalised by dividing it by the number of nodes. To normalise the cost function, it is divided by the maximum possible distance between any nodes in the graph. The pseudocode of our algorithm is shown in Algorithm 1.

Functions:
$\mathrm{cost}(L)$:= cost function
$\mathrm{algConn}(G)$:= algebraic connectivity function
$\mathrm{candidate}(G)$:= candidate links function
$\mathrm{maxLink}(D)$:= max value of a dictionary
Input:
G_i:= input graph
L_r:= number of required links
γ:= cost effect parameter
Output:
an ordered list of the added links
begin
    selectedLinks = [] ; empty ordered list
    rank = {} ; empty dictionary
    while $L_r > 0$ **do**
        G = $\mathrm{G_i}$
        G.addlinks(AddedLinks)
        for link in $\mathrm{candidate}(G)$ **do**
            rank[link]=$(1-\gamma)$algConn(G) + $\gamma(1-$cost(link))
        end
        selectedLinks.add(maxLink(rank))
        $L_r = L_r - 1$
    end
    return selectedLinks
end

Algorithm 1: Topology Optimisation Algorithm

C. Options

In this paper, we target two graph types: physical and logical level graphs [24]. For the logical level graph, the algorithm is applied with no additional conditions. However, for the physical level graph, we have added a filter that removes very long links from the candidate links set. This is because it is not practical to add very long links between cities such as a physical fibre link between Los Angeles and New York City. Therefore this raises the question of what the maximum length should be chosen. In our implementation, we have it as a variable that can be set by the user. We choose the maximum length link in the input graph to be the threshold for long links in the dataset, which gives a good indicator for the maximum link length a provider can afford.

IV. Analysis

In this section, first we describe our algorithm on a small size graph. Next, we present the topological dataset we use to apply our algorithm, followed by cost and connectivity analysis of commercial backbone providers.

A. Algorithm Evaluation

In this section, we explain how our heuristic algorithm optimises a topology on a small-size graph. Figure 1 shows a sample graph with 8 nodes and 9 links as solid lines. The initial algebraic connectivity of this sample graph is 0.3432 and the initial cost (i.e., total link length in km) of the graph is 8,203. Our heuristic algorithm adds links to the least connected nodes, which in the example are nodes 0 and 7. The six candidate links for node 0 are shown as square dots, whereas five candidate links for node 7 are shown as long dashes and dots. Throughout this example, we describe how our algorithm operates if we are going to add *one* link $L_r = 1$ to the sample graph shown in Figure 1.

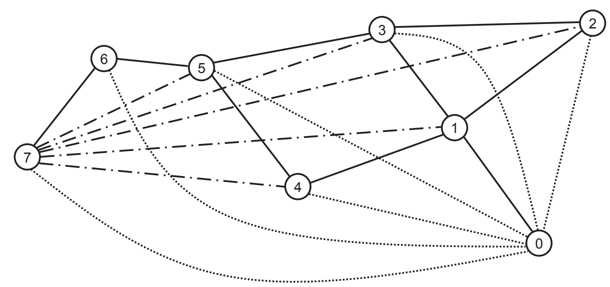


Fig. 1. Sample graph

There can be a maximum of 28 links in this 8-node graph (maximum links can be calculated by $\frac{n(n-1)}{2}$). Since there are 9 links in the graph, if we were to do an exhaustive search, there would be $28 - 9 = 19$ candidate links (i.e. the complement links). In the sample graph shown in Figure 1, there are six candidate links that can be added to node 0 and there are five links for node 7 using our heuristic algorithm. Therefore, the candidate link set is reduced to 11, because our algorithm only considers candidate links from the least connected nodes. The algebraic connectivity and cost value of adding each link individually for $\gamma = 0$ and $\gamma = 1$ is shown in Table I.

When $\gamma = 0$, our algorithm ignores the cost associated with adding a link and selects the additional link that increases the algebraic connectivity of the graph the most. For $\gamma = 0$, the algorithm adds the link between node 1 and 7 in the example graph since it provides the highest algebraic connectivity among the 11 candidate links. When $\gamma = 1$, the cost is the dominant factor determining the addition of a link. Therefore, our heuristic algorithm selects the link between node 0 and

TABLE I
$a(G)$ AND COST VALUES FOR THE SAMPLE GRAPH

Link	$\gamma = 0$		$\gamma = 1$	
	$a(G)$	$\Delta a(G)$	cost	Δ cost
$0 \leftrightarrow 2$	0.3485	0.0053	**9,275**	**1,072**
$0 \leftrightarrow 3$	0.3588	0.0156	9,405	1,202
$0 \leftrightarrow 4$	0.3659	0.0227	9,848	1,645
$0 \leftrightarrow 5$	0.4079	0.0647	10,624	2,421
$0 \leftrightarrow 6$	0.5908	0.2476	11,228	3,025
$0 \leftrightarrow 7$	0.7713	0.4281	11,843	3,640
$7 \leftrightarrow 1$	**0.8345**	**0.4913**	11,302	3,099
$7 \leftrightarrow 2$	0.7071	0.3639	12,061	3,858
$7 \leftrightarrow 3$	0.6651	0.3219	10,915	2,712
$7 \leftrightarrow 4$	0.5918	0.2486	10,207	2,004
$7 \leftrightarrow 5$	0.5075	0.1643	9,463	1,260

2, since it incurs the lowest cost among the candidate set of links. The selection of links via our heuristic algorithm is highlighted bold in Table I. Moreover, we performed an exhaustive search on the sample graph shown in Figure 1, and find that the link between node 1 and 7 has the highest algebraic connectivity among 19 possible links. The result of the exhaustive search for the least incurred cost link indicated that the link between node 3 and 4 is the best option, however, as mentioned above, our algorithm adds links to the minimal degree nodes. Therefore our algorithm selects the link between node 0 and 2 when $\gamma = 1$. We note that for physical-level networks $\gamma \rightarrow 1$ makes sense due to the significant cost of deploying fibre. On the other hand, $\gamma \rightarrow 0$ is more appropriate since the cost of virtual link deployment is negligible, whereas delay is a dominant factor in logical overlays. To conclude, our heuristic algorithm optimises graphs cost-efficiently while selecting the links that improves the algebraic connectivity the most based on the γ parameter value.

B. Topological Dataset

We study physical- and logical-level topologies of three tier-1 service provider networks. We use Rocketfuel-inferred [25] logical-level topologies of AT&T, Level 3, and Sprint. Physical-level topologies of the three service providers were constructed using a third party map [26]. The details of generating physical-level topologies are presented in our other work [19], [24].

TABLE II
PHYSICAL AND LOGICAL TOPOLOGICAL DATASET

Network	Nodes	Links	Complement links
AT&T phy.	383	488	72,665
Level 3 phy.	99	132	4,719
Sprint phy.	264	313	34,403
AT&T log.	107	140	5,531
Level 3 log.	38	376	3,276
Sprint log.	28	76	302

In general, logical-level topologies are richly connected compared to their physical-level graphs and they are smaller in size and order. Comparison of characteristics of these different topologies is beyond the scope of this paper and presented in our earlier work [24]. The number of nodes, links, and complement links of these graphs are shown in Table II.

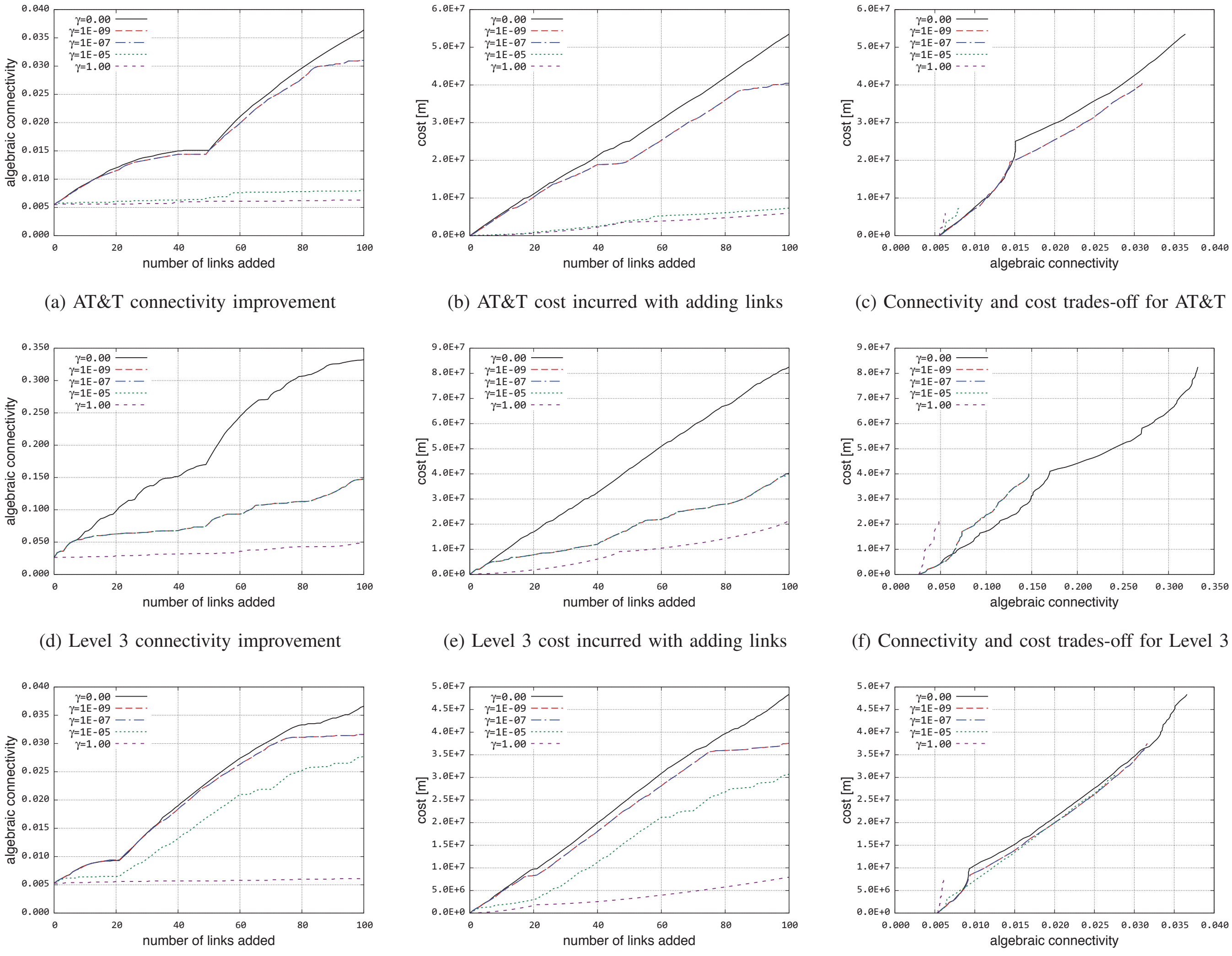
C. Backbone Provider Network Analysis

Our algorithm is applied to three ISPs by adding 100 links. We show the graph algebraic connectivity and the cost incurred in terms of meters after adding each link. Moreover, we show the relation of cost and algebraic connectivity and the slope in these figures shows how the cost increases as the graph connectivity improves.

1) Selection of γ values: γ parameter that ranges 0 to 1 controls the outcome of the algorithm as described in Section III-B. In Equation 3, we have two terms: $(1-\gamma)a(G)$ and $\gamma(1-C)$. The $a(G)$ is the normalised algebraic connectivity value, which is low for sparse graphs and one for a full mesh graph. The value of C denotes the normalised cost of adding a link and it is low when the maximum possible link length in the input graph is larger than the average link length in the candidate set. Therefore, choosing the value of γ depends on the initial properties of the input graph. For the physical graphs, we choose for $\gamma = \{0, 10^{-9}, 10^{-7}, 10^{-5}, 1\}$ because the cost term is larger than the γ term about six order of magnitude for physical level graphs. For the logical level graphs, we choose different values of $\gamma = \{0, 0.0001, 0.001, 0.01, 1\}$ because the cost term is larger than the algebraic connectivity term about two order of magnitude.

2) Physical level topology analysis: As explained in Section III, an option is added in our heuristic algorithm to discard the links that are longer than the actual maximum link of the graph. Furthermore, physical level graphs have more nodes than the logical level graphs, which increases the number of shorter links for the candidate set. For these reasons, optimisation on physical level graphs results in selection of shorter links. The connectivity and cost optimisation of physical level provider graphs are shown in Figure 2. Algebraic connectivity improvement of the three physical level topologies after adding 100 links iteratively is depicted in Figures 2a, 2d, and 2g. The algebraic connectivity is higher for $\gamma = 0$ than the other values of γ, and for $\gamma = 1$ our algorithm considers minimising the cost, but not improving the algebraic connectivity. Moreover, we observe the occurrence of possible phase transition when $\gamma = 1$ for the physical-level graphs. For example, algebraic connectivity improvement of the AT&T physical topology starts with a moderate increase, and after about 50th link addition, the improvement (i.e., the slope of the curve) gets steeper. The reasons for the occurrence of this phenomenon will be the subject of future work.

The cost incurred when adding 100 links iteratively to the physical level topologies are shown in Figures 2b, 2e, and 2h. The cost in physical topology is the length of links to be laid between nodes, thus, short links are favorable in physical level topology optimisation for $\gamma = 1$. The relationship between connectivity and cost for physical level topologies are shown in Figure 2c, 2f, and 2i. For the Level 3 example shown in Figure 2f, if the cost is the constraint (i.e. $\gamma = 1$), the designer



(a) AT&T connectivity improvement (b) AT&T cost incurred with adding links (c) Connectivity and cost trades-off for AT&T

(d) Level 3 connectivity improvement (e) Level 3 cost incurred with adding links (f) Connectivity and cost trades-off for Level 3

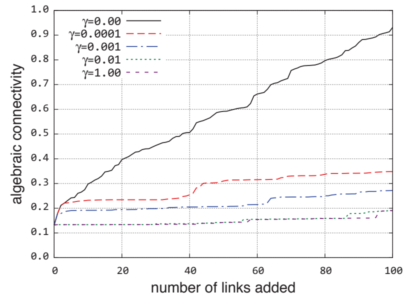
(g) Sprint connectivity improvement (h) Sprint cost incurred with adding links (i) Connectivity and cost trades-off for Sprint

Fig. 2. Analysis of physical topologies

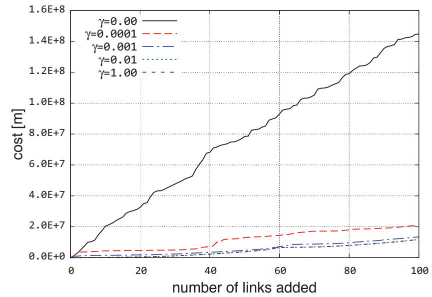
can improve the algebraic connectivity to 0.05 by adding 100 links. On the other hand, if there is available budget (i.e. $\gamma = 0$) the algebraic connectivity of the Level 3 topology can be improved more than 0.3.

3) Logical level topology analysis: The optimisation of logical level topologies are shown in Figure 3. The algorithm has more candidate link options since it is not constrained by the maximum length of links in the input graph for logical level topologies. Therefore, the improvement of algebraic connectivity as links are added is higher than for physical level topologies. The algebraic connectivity improvement of up to two orders of magnitude, can be seen clearly for the logical level topologies in Figures 3a, 3d, and 3g. The cost incurred after adding 100 links for the logical level topologies is shown in Figures 3b, 3e, and 3h. Similar to the physical level topologies, as the value of γ increases, the cost of building more connected graphs decreases. The trade-offs between cost and connectivity for logical level topologies is shown in Figures 3c, 3f, and 3i. For example, to improve the algebraic connectivity of the Level 3 logical topology to a value of 10.0 in Figure 3f, we should select the links returned from the algorithm when γ is 0.01 since it incurs the lowest cost. However, to improve the algebraic connectivity of AT&T to 0.7 as shown in Figure 3c, we have no choice but to select the links when $\gamma = 0$ since the other γ values do not improve the algebraic connectivity to 0.7.

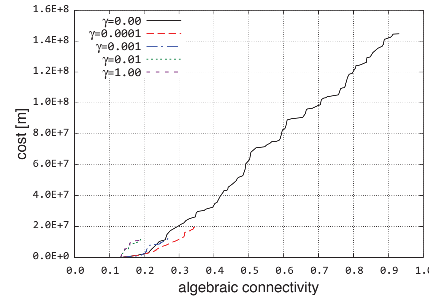
4) Optimisation comparison of backbone networks: Finally, we compare the optimisation output of the three backbone provider topologies using $\gamma = 0$ and $\gamma = 1$ as shown in Figure 4 and Figure 5, respectively. Even though Sprint and AT&T physical level topologies have a different number of nodes and links, the optimisation for $\gamma = 0$ results in about the same algebraic connectivity as when we add 100 links as shown in Figure 4a. On the other hand, the Level 3 physical level topology starts from an even higher initial algebraic connectivity and significantly improves to a higher algebraic



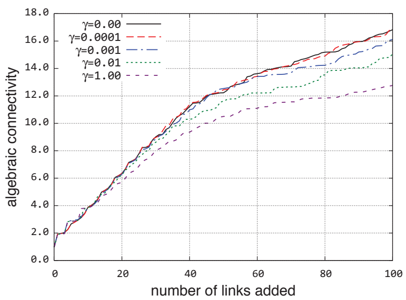
(a) AT&T connectivity improvement



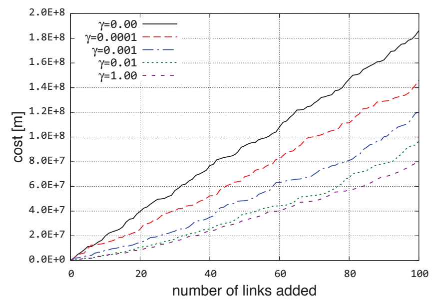
(b) AT&T cost incurred with adding links



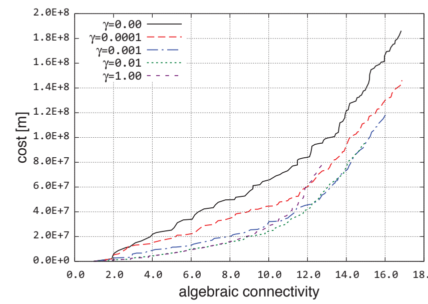
(c) Connectivity and cost trades-off for AT&T



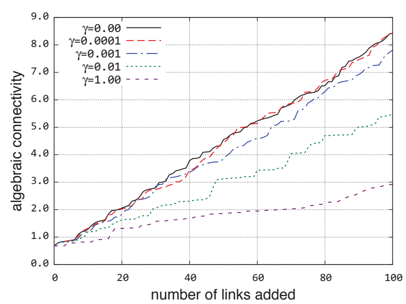
(d) Level 3 connectivity improvement



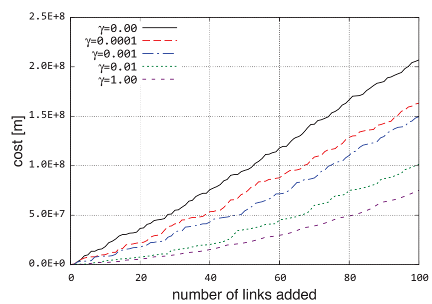
(e) Level 3 cost incurred with adding links



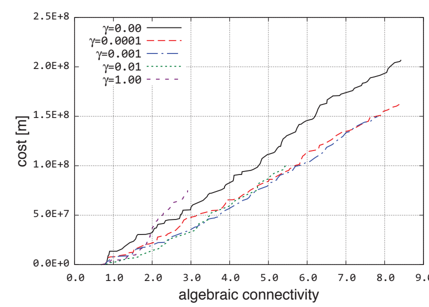
(f) Connectivity and cost trades-off for Level 3



(g) Sprint connectivity improvement



(h) Sprint cost incurred with adding links



(i) Connectivity and cost trades-off for Sprint

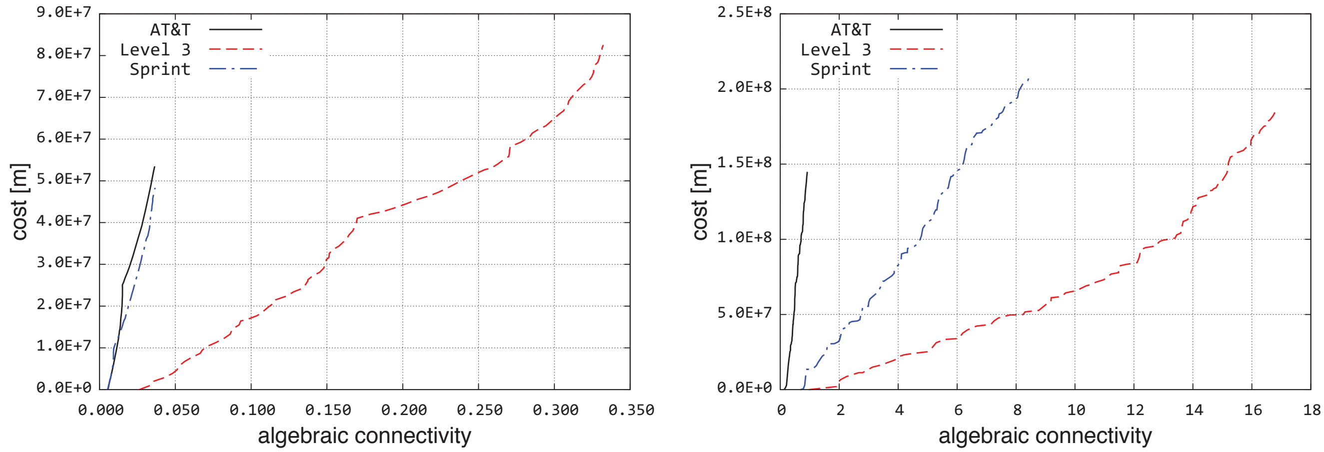
Fig. 3. Analysis of logical topologies

connectivity with larger cost than the others as shown in Figure 4a. Similar conclusions can also be drawn when we compare our optimisation algorithm output for $\gamma = 1$ as shown in Figure 5a. The main difference is that when there is no budget constraint (i.e. $\gamma = 0$), the algebraic connectivity and cost is an order of magnitude higher. Moreover, the slope in these figures shows how the cost increases as we improve the connectivity of the graph. Small slope value of a curve implies high gain in the graph connectivity for a given budget, which is favorable. On the other hand, large slope value means that the cost is high while the improvement is low.

The tradeoffs between cost and algebraic connectivity for the logical topology graphs when $\gamma = 0$ is shown in Figure 4b. For AT&T, we see that algebraic connectivity does not improve for adding the first links, with total cost around 1.5×10^8. On the other hand, for Level 3 and Sprint, the algebraic connectivity increases significantly as links are added as shown in Figure 4b. When $\gamma = 1$, which means the least cost links are selected, the algebraic connectivity does not improve much as links are added to the graph compared to $\gamma = 0$ as shown in Figure 4 and Figure 5. Another interesting result is that the Sprint and Level 3 topologies incur about the same cost after adding 100 links, however the algebraic improvement for Level 3 is twice or more for $\gamma = 0$ and $\gamma = 1$ values. Finally, physical level topologies have lower gain in terms of the algebraic connectivity since long links are removed from the candidate set and these links can be the highest contributors to the algebraic connectivity.

V. Conclusions and Future Work

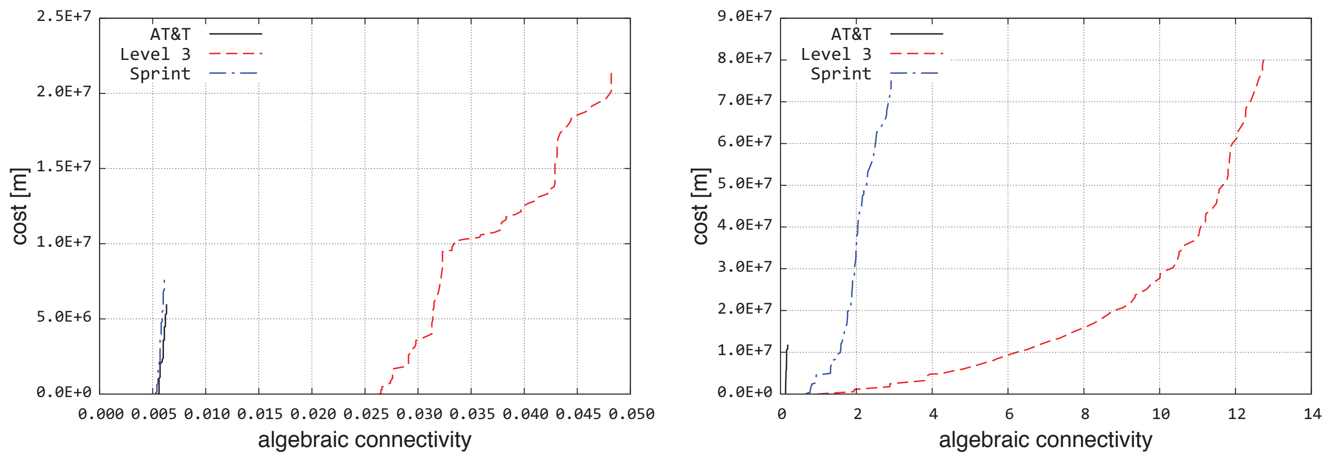
Network design and optimisation is a major area of research. In this paper, we present a new heuristic algorithm that optimises the connectivity of a given graph with node locations. We use algebraic connectivity as a measure to improve the connectivity of the graph. This algorithm minimises the cost of adding new links by selecting shorter links with high



(a) Physical level topologies

(b) Logical level topologies

Fig. 4. Algebraic connectivity and cost effect for $\gamma = 0$



(a) Physical level topologies

(b) Logical level topologies

Fig. 5. Algebraic connectivity and cost effect for $\gamma = 1$

algebraic connectivity. We introduce a tuning parameter γ to control the effect of the cost function while selecting new links. Furthermore, the candidate links that are being added to improve the connectivity of the graph can be constrained by a length limit in our algorithm. We apply this algorithm to physical- and logical-level topologies of three backbone providers. The results show trade-offs between improving algebraic connectivity and minimising cost, from which a cost-efficient set of link addition can be chosen based on the value of γ. We show that the algebraic connectivity improvement for the physical level graphs are less than the logical level graphs because of the differences in their structural characteristics, as well as the link length limitation imposed on the candidate links that are added to the physical level topologies.

For our future work, we would like to run our heuristic algorithm using graph properties such as clustering coefficient, betweenness, and graph diversity [23]. For example, we can add clustering coefficient to replace the algebraic connectivity or we can add both with a parameter to weight their effect in ranking the links that need to be added to improve connectivity of the graph. We will also modify our algorithm to achieve a specified graph metric value with a constrained budget. Finally, we will investigate the occurrence of a phase transition phenomenon in physical-level topologies and compare backbone provider graphs with synthetic topologies.

Acknowledgments

This research was supported in part by NSF grant CNS-1219028 (Resilient Network Design for Massive Failures and Attacks) and by NSF grant CNS-1050226 (Multilayer Network Resilience Analysis and Experimentation on GENI). Mohammed J.F. Alenazi is supported by SACM (Saudi Arabian Cultural Mission) and affiliated with King Saud University.

REFERENCES

[1] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[2] H. Wang and P. Van Mieghem, "Algebraic connectivity optimization via link addition," in *Proceedings of the 3rd ICST International Conference on Bio-Inspired Models of Network, Information and Computing Sytems (BIONETICS)*, (Hyogo, Japan), pp. 22:1–22:8, November 2008.

[3] W. Liu, H. Sirisena, K. Pawlikowski, and A. McInnes, "Utility of algebraic connectivity metric in topology design of survivable networks," in *Proceedings of the 7th IEEE International Workshop on Design of Reliable Communication Networks (DRCN)*, (Washington, DC), pp. 131–138, October 2009.

[4] A. Sydney, C. Scoglio, and D. Gruenbacher, "Optimizing algebraic connectivity by edge rewiring," *Applied Mathematics and Computation*, vol. 219, no. 10, pp. 5465–5479, 2013.

[5] M. O. Ball, "Complexity of network reliability computations," *Networks*, vol. 10, no. 2, pp. 153–165, 1980.

[6] M. O. Ball, "Computational complexity of network reliability analysis: An overview," *IEEE Transactions on Reliability*, vol. 35, pp. 230–239, August 1986.

[7] R. S. Wilkov, "Analysis and design of reliable computer networks," *IEEE Transactions on Communications*, vol. 20, no. 3, pp. 660–678, 1972.

[8] J. M. McQuillan, "Graph theory applied to optimal connectivity in computer networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 7, pp. 13–41, Apr. 1977.

[9] M. Gerla and L. Kleinrock, "On the topological design of distributed computer networks," *IEEE Transactions on Communications*, vol. 25, pp. 48–60, January 1977.

[10] R. R. Boorstyn and H. Frank, "Large-scale network topological optimization," *IEEE Transactions on Communications*, vol. 25, pp. 29–47, January 1977.

[11] J. G. Klincewicz, "Hub location in backbone/tributary network design: a review," *Location Science*, vol. 6, pp. 307–335, December 1998.

[12] B. Gavish, "Topological design of computer communication networks – the overall design problem," *European Journal of Operational Research*, vol. 58, no. 2, pp. 149–172, 1992.

[13] A. Konak and A. E. Smith, "Network reliability optimization," in *Handbook of Optimization in Telecommunications* (M. G. Resende and P. M. Pardalos, eds.), pp. 735–760, Springer US, 2006.

[14] H. C. Cankaya, A. Lardies, and G. W. Ester, "Network Design Pptimization from an Availability Perspective," in *Proceedings of the 11th International Telecommunications Network Strategy and Planning Symposium (NETWORKS)*, (Vienna), pp. 359–364, June 2004.

[15] S. Chamberland, M. St-Hilaire, and S. Pierre, "On the point-of-presence optimization problem in IP networks," *Canadian Journal of Electrical and Computer Engineering*, vol. 30, no. 3, pp. 137–143, 2005.

[16] D. Fay, H. Haddadi, A. Thomason, A. Moore, R. Mortier, A. Jamakovic, S. Uhlig, and M. Rio, "Weighted Spectral Distribution for Internet Topology Analysis: Theory and Applications," *IEEE/ACM Transactions on Networking*, vol. 18, no. 1, pp. 164–176, 2010.

[17] M. A. Hameed, A. Jabbar, E. K. Çetinkaya, and J. P. Sterbenz, "Deriving Network Topologies from Real World Constraints," in *Proceedings of IEEE GLOBECOM Workshop on Complex and Communication Networks (CCNet)*, (Miami, FL), pp. 400–404, December 2010.

[18] J. P. Sterbenz, E. K. Çetinkaya, M. A. Hameed, A. Jabbar, Q. Shi, and J. P. Rohrer, "Evaluation of Network Resilience, Survivability, and Disruption Tolerance: Analysis, Topology Generation, Simulation, and Experimentation (invited paper)," *Telecommunication Systems*, vol. 52, no. 2, pp. 705–736, 2013.

[19] E. K. Çetinkaya, M. J. F. Alenazi, Y. Cheng, A. M. Peck, and J. P. G. Sterbenz, "On the Fitness of Geographic Graph Generators for Modelling Physical Level Topologies," in *Proceedings of the 5th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, (Almaty), September 2013.

[20] E. K. Çetinkaya, M. J. F. Alenazi, A. M. Peck, J. P. Rohrer, and J. P. G. Sterbenz, "Multilevel Resilience Analysis of Transportation and Communication Networks," *Springer Telecommunication Systems Journal*, 2013. accepted on July 2013.

[21] L. d. F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas, "Characterization of complex networks: A survey of measurements," *Advances in Physics*, vol. 56, no. 1, pp. 167–242, 2007.

[22] I. M. Soi and K. K. Aggarwal, "Reliability indices for topological design of computer communication networks," *IEEE Transactions on Reliability*, vol. R-30, pp. 438–443, December 1981.

[23] J. P. Rohrer, A. Jabbar, and J. P. Sterbenz, "Path Diversification for Future Internet End-to-End Resilience and Survivability," *Springer Telecommunication Systems*, 2012.

[24] E. K. Çetinkaya, M. J. F. Alenazi, J. P. Rohrer, and J. P. G. Sterbenz, "Topology Connectivity Analysis of Internet Infrastructure Using Graph Spectra," in *Proceedings of the 4th IEEE/IFIP International Workshop on Reliable Networks Design and Modeling (RNDM)*, (St. Petersburg), pp. 752–758, October 2012.

[25] N. Spring, R. Mahajan, D. Wetherall, and T. Anderson, "Measuring ISP topologies with Rocketfuel," *IEEE/ACM Transactions on Networking*, vol. 12, no. 1, pp. 2–16, 2004.

[26] KMI Corporation, "North American Fiberoptic Long-haul Routes Planned and in Place," 1999.