Frugal Topologies for Saving Energy in IP Networks

Mohammed Hussein, Gentian Jakllari, Beatrice Paillassa
IRIT-ENSEEIHT, University of Toulouse, France
{mohammed.hussein, gentian.jakllari, beatrice.paillassa}@enseeiht.fr

***Abstract*—Recent years have seen the advent of energy conservation as a key engineering and scientific challenge of our time. At the same time, most IP networks are typically provisioned to 30%-40% average utilization, leading to a significant waste of energy. Current approaches for creating more frugal networks rely on instantaneous and global knowledge of the traffic matrix and network congestion levels – a requirement that can be impractical for many network operators.**

We introduce a new traffic-agnostic metric for quantifying the quality of a frugal topology, the Adequacy Index (ADI). We show that the problem of minimizing the power consumption of a network subject to a given ADI threshold is NP-hard and present two polynomial time heuristics – ABStAIn and CuTBAck. We perform extensive simulations using topologies and traffic matrices from 3 real networks. Our results show that ABStAIn and CuTBAck are as effective as an exponential time traffic based solution at creating frugal topologies and outperform a state of the art polynomial time traffic based solution by about 80%. Furthermore, the median link utilization observed with ABStAIn and CuTBAck is similar to that with traffic based solutions, with the maximum link utilization never exceeding 80%.

I. Introduction

The Internet has become a major source of power consumption due to its exponential growth. European telecommunication operators currently consume 21.4 TWh per year, a quantity projected to soon reach 35.8 TWh, if no green networking technologies are adapted [1]. At the same time, network operators overprovision their networks. Most ISPs upgrade their infrastructures when the link utilization reaches above 40% [2], [3], leading to a significant waste of energy. Reducing this waste, however, is challenging. First, a network operator has to build its infrastructure for the worst case and the rule of thumb value of 40% is based on average utilization. Network traffic is inherently variable, with high and low peaks often reflecting human activity. Second, the coarse granularity of the current transmission technology forces network providers to install high bandwidth links during any incremental upgrade [4]. Therefore, designing large networks with average link utilization approaching 100% is infeasible. Given this, several works [5], [4], [6], [7], [8], [9] have proposed adapting the availability of the network infrastructure to the traffic demand. During off-peak hours, networking components – routers, switches, line cards – can be selectively switched off to save energy, only to be turned back on during peak hours. The problem is finding the optimal set of networking components to be switched off such that the resulting frugal network does not suffer from heavy congestion and packet losses. The general method for addressing this problem has been to adopt the capacitated multi-commodity minimum cost flow (CMCF) problem formulation so that the objective function depends on the power consumption [6], [4]. With the problem being NP-Hard, various heuristics have been proposed for creating frugal topologies by switching off the maximum number of links[1] possible subject to the maximum link utilization – the main quality metric used – being below a threshold. However, for these solutions to perform as advertised they require accurate traffic matrices and instantaneous congestion information about the entire network. Given how quickly traffic and congestion levels can change, fulfilling these requirements can be impractical for many network operators.

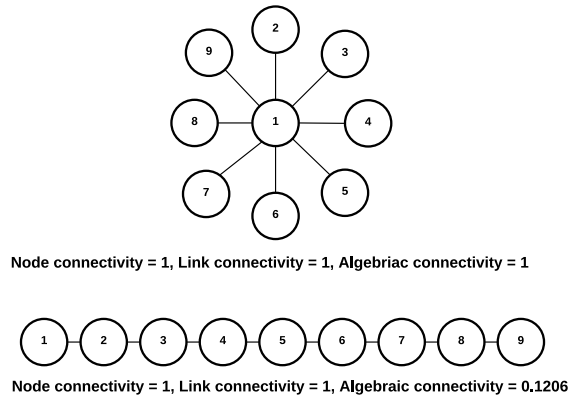


Figure 1: Algebraic connectivity is a better indicator of network robustness.

In this work, we argue for using a quality metric for frugal topologies that does not depend on instantaneous traffic and congestion levels and yet is highly correlated with the quality of the network as a whole. We introduce the Adequacy Index (ADI), a metric for quantifying the quality of the frugal version relative to the full-on network. ADI is based on the concept of algebraic connectivity from spectral graph theory [10]. Compared with more intuitive graph concepts, such as the vertex/edge[2] connectivity – the minimum number of vertices/edges whose deletion from a graph disconnects it – the algebraic connectivity better captures the robustness of a graph [11], [12], [13]. Consider the two simple graphs

[1]Line cards represent a majority of the routers energy consumption [1].

[2]For the rest of the paper, we will use the terms vertex/node and edge/link interchangeably.

in Fig 1. Removing any single edge from the star topology would only result in a single isolated node. Removing a link from the middle of the line topology would split the network in two. While the vertex and edge connectivity fail to capture the difference in robustness between the star and line topology, the algebraic connectivity does.

We formalize the problem of minimizing the power consumption of a network subject to a given ADI and prove that the problem is NP-Complete. We introduce a generic approach for solving the problem and present two polynomial time instantiations: ABStAIn (for **A**lgebraic **Ba**n**S**ed **A**lgorithm for Frugal**I**ty) and CuTBAck (for **C**en**T**rality **B**ased **A**lgorithm for Frugality). ABStAIn and CuTBAck follow the generic approach of creating frugal topologies by removing links in some order until a given ADI threshold is reached. They differ on how the network links are ordered from the most expendable, the first to be switched off for frugality, to the most important. ABStAIn relies on the algebraic connectivity for deciding how important a particular link is, while CuTBAck on the notion of the link betweenness centrality [14].

We evaluate Cutback and Abstain using real topologies and traffic matrices from 3 networks representing a mixture of academic, commercial and educational usage, and compare them to two popular traffic based solutions: Benchmark-MLU [6] and Least-Flow [4]. The data shows that Cutback and Abstain switch off as many links as the exponential time Benchmark-MLU and 80% more than the Least-Flow. The median link utilization observed in the frugal topologies generated by Cutback and Abstain was similar to what was observed in the topologies generated by Benchmark-MLU and Least-Flow, with the maximum link utilization never exceeding 80%.

Fundamentally, Cutback and Abstain provide a novel control option to network operators – it can be programmed to be enabled during off-peak hours for creating energy frugal topologies and disabled during peak hours.

II. Primer on Algebraic Connectivity

"How well connected is a graph?" This is a fundamental question to any problem modeled using graphs and that unfortunately defies a simple answer. Even producing a simple definition as to what well connected exactly means is challenging. The algebraic connectivity – the second smallest eigenvalue of the graph's Laplacian matrix – was established by Fiedler in his seminal work [10] as an elegant answer to this fundamental question.

Let $G = (V, E)$ be a simple graph with $|V|$ vertices and $|E|$ edges; its algebraic connectivity is a function of its adjacency and degree matrices. In the following we introduce formal definitions for all these quantities along with some key results on algebraic connectivity.

Definition 1 (Adjacency Matrix)**.** *Given a simple graph $G = (V, E)$ with $|V| = n$, its adjacency matrix $A(G)$ is a $n \times n$ binary matrix where the entry a_{ij} is equal to 1 if $\{i, j\} \in E$ and 0 otherwise.*

Definition 2 (Degree Matrix)**.** *Given a simple graph $G = (V, E)$ with $|V| = n$, its degree matrix $D(G)$ is a $n \times n$ diagonal matrix where the entry d_{ii} is equal to the degree of vertex i.*

Definition 3 (Laplacian Matrix). *Given a simple graph $G = (V, E)$ with $|V| = n$, its Laplacian matrix $L(G)$ is a $n \times n$ matrix defined as:*

$$L(G) = D(G) - A(G)$$

From Definition 3, it follows that the entry $l_{i,j}$ of the Laplacian matrix for graph G is

$$l_{i,j} = \begin{cases} deg(i) & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } \{i, j\} \in E \\ 0 & \text{otherwise} \end{cases}$$

where $deg(i)$ is the degree of vertex i.

The eigenvalues of the Laplacian matrix are usually referred to as the graph spectra. The number of zero-valued eigenvalues of the Laplacian matrix is equal to the number of connected components in the graph G. Consequently, the *second smallest* eigenvalue being 0 is equivalent to the graph having at least two connected component and thus being disconnected. Therefore, this eigenvalue is referred to as the algebraic connectivity of the graph [10]. More formally:

Definition 4 (Algebraic Connectivity $a(G)$). *Let $N \geq 2$ and $0 = \lambda_1 \leq \lambda_2 \cdots \leq \lambda_N$ be the eigenvalues of the Laplacian matrix $L(G)$. The algebraic connectivity $a(G)$ of the graph G is equal to the second smallest eigenvalue, λ_2.*

The algebraic connectivity has become essential to the study of the network robustness not only because a non-zero value proves end-to-end connectivity but more importantly because of Lemma 1 proved by Fiedler [10]. It connects the algebraic connectivity to two important graph properties. One, the *vertex connectivity*, the minimum number of vertices whose deletion from a graph disconnects it. Two, the *edge connectivity*, the minimum number of edges whose deletion from a graph G disconnects it.

Lemma 1 (Bound on Connectivity)**.** *Let $k(G)$ and $\eta(G)$ be the vertex and edge connectivity of the graph G, respectively. Then*

$$a(G) \leq k(G) \leq \eta(G).$$

Finally, we present a property that will be useful in Sections III and IV.

Lemma 2. *The function $a(G)$ is non-decreasing for graphs with the same set of vertices, i.e. $a(G_1) \leq a(G_2)$, if $V_1 = V_2$, and $E_1 \subseteq E_2$.*

III. New Metric and Problem Definition

In this section we introduce a new metric, *Adequacy Index (ADI)*, for quantifying the quality of a network topology. We then use this metric to formally define the problem of computing frugal topologies.

A. Adequacy Index

Our goal is to compute a frugal yet adequate version of an IP network topology. For this, we first need to quantify the notions of *frugal* and *adequate*. The notion of frugal is easy to quantify – it is the non-trivial version of the full network topology that minimizes energy consumption. Adequate has been traditionally defined as a topology whose maximum link utilization is bounded by a given threshold [6], [4]. The advantage of this definition is that it guarantees a given level of congestion and quality of service in the network. Unfortunately, guaranteeing a given level of link utilization requires accurate and instantaneous information as to the level of congestion and traffic matrix in the network. To circumvent this impractical requirement, we introduce a new definition for adequate:

Definition 5 (**Adequacy Index, ADI).** *Let $G = (V, E)$ be a simple graph. Let $G^f = (V, E^f)$ such that $E^f \subseteq E$ be a frugal version of graph G. The adequacy index, ADI, of the frugal graph G^f is defined as follows:*

$$ADI(G^f) = \frac{a(G^f)}{a(G)} \tag{1}$$

where $a()$ denotes the algebraic connectivity.

The following lemma describes a basic property of the Adequacy Index.

Lemma 3. *Let $G = (V, E)$ be a simple graph and $G^f = (V, E^f)$ such that $E^f \subseteq E$ a frugal version of graph G. Then*

$$0 \leq ADI(G^f) \leq 1.$$

Proof: Follows from Lemma 2. ■

The Adequacy Index has the advantage of depending on the topological properties of the network and not the instantaneous traffic level. At the same time, as it depends on the algebraic connectivity it is related to the level of connectivity and redundancy. In Section V, using real topologies and traffic matrices, we show that the Adequacy Index provides a nob that enables changing the level of frugality as well as congestion in the network.

B. Problem Definition

We model a network of IP routers, such as an Autonomous System or backbone network, as a simple graph $G = (V, E)$, with V the set of vertices representing the routers and E the set of edges representing the physical links between routers. Let $p_{(u,v)}$ be the power consumption of the link from router u to v. Let $x_{(u,v)}$ be a binary variable denoting whether link (u, v) is switched off for saving energy. The objective is to turn off as many links as possible so as to create a frugal yet adequate topology of routers. The problem can be formulated as follows:

$$\text{minimize} \quad P_{tot} = \sum_{(u,v)\in E} p_{(u,v)} x_{(u,v)} \tag{2}$$

$$\text{s.t. } ADI(G^f) \geq Threshold \tag{3}$$

Equation 2 minimizes the total power consumption of the network. Equation 3 forces the Adequacy Index of the frugal graph to be above a threshold value. Finally, Theorem 1 shows the difficulty of solving this problem.

Theorem 1. *The problem of finding the most frugal network topology subject to a given adequacy index threshold is NP-hard.*

Proof: The proof is simple so we provide a sketch. We show that our problem is NP-Hard by reducing the maximum algebraic connectivity augmentation problem [15], hereto $P2$, to our problem, hereto $P1$. To this end, we consider the instance of $P1$ in which once the most frugal topology is found we are asked whether the number of edges in this topology is higher than a non-negative integer k. Solving this instance of $P1$ consists of solving an instance of $P2$. Since $P2$ has been shown to be NP-Hard [15] that concludes the proof. ■

In Section IV, we focus on designing heuristic approaches to solve our problem.

IV. Creating Frugal Topologies

In this section, we present heuristics for computing the most frugal network topology subject to a given adequacy index threshold. At first, we introduce a generic approach that uses the adequacy index metric. Then, we present two specific instantiations of the generic approach. The first leverages the algebraic connectivity while the second leverages concepts from the network centrality [14].

A. Generic Approach

The generic approach, Algorithm 1, uses a greedy strategy for solving the problem. It starts with the complete topology and renders it frugal by removing edges iteratively (lines 5-12). In every iteration it selects the most expendable edge (line 6) and checks whether removing it would lower the adequacy index of the frugal graph below a given threshold, ADI_T, given as input (line 8). If not, the edge is removed and the remaining edges are re-sorted (line 12) – removing an edge changes the graph structure and the relative importance of the remaining edges (see Fig. 2 in Section IV-B). Otherwise the edge is kept. Obviously, the key part of this approach is sorting the edges from the most to the least expendable (lines 4,12). Depending on how the sorting procedure is implemented, we can have a rich set of solutions for the most frugal adequate topology problem. In the following we show two such examples.

B. ABStAIn: Algebraic BaSed Algorithm for FrugalIty

To implement the generic approach, ABStAIn proposes a sorting algorithm that establishes a direct link between every edge and the Adequacy Index. The straightforward approach would be to remove every edge, compute the impact this would have on the Adequacy Index and sort edges based on this impact. This, however, would entail computing the algebraic connectivity of the graph as many times as there are edges – a costly proposition given that a single computation of the eigenvalues using the popular QR algorithm [16] takes $O(|V|^3)$ operations. Instead, we

Algorithm 1: Generic Approach.

input : Complete Network Graph: $G(V, E)$; Adequacy Index Threshold: ADI_T; Sorting Function: $SortEdges()$

output : Frugal Graph: $G^f(V, E^f)$

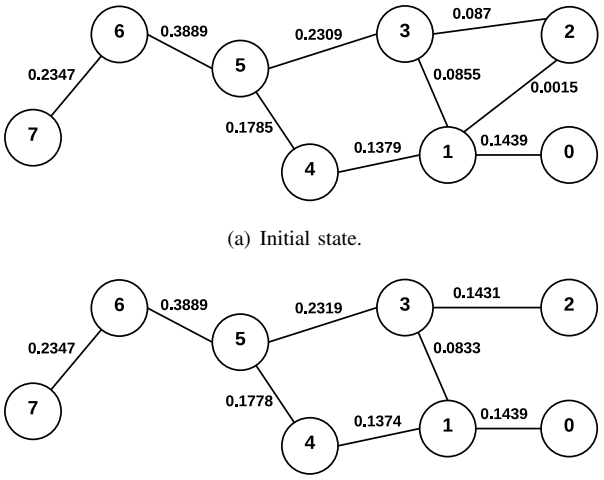
1: **begin**
2: $G^f(V, E^f) \leftarrow G(V, E)$;
3: $a(G) \leftarrow$ AlgebraicConnectivity $(G(V, E))$
`//Sort the edges from least to most important using the specific ordering algorithm.`
4: $SE\_List \leftarrow$ SortEdges $(G(V, E^f))$;
`//Remove edges starting from the most expandable if doing so does not lower the adequacy index below the threshold, $ADI_T$.`
5: **for** $i = 1 \rightarrow sizeof(SE\_List)$ **do**
6: $e \leftarrow MostExpendable(SE\_List)$;
7: $E^f \leftarrow E^f - \{e\}$;
8: $a(G^f) \leftarrow$AlgebraicConnectivity $(G^f(V, E^f))$
9: **if** $\frac{a(G^f)}{a(G)} \leq ADI_T$ **then**
`//Do not remove this edge.`
10: $E^f \leftarrow E^f + \{e\}$;
11: **else**
`//Removing an edge changes the structure of the graph and the relative importance of the remaining edges. Thus, the edges queue is re-sorted.`
12: $SE\_List \leftarrow$ SortEdges $(G(V, E^f))$;
13: **return** $G^f(V, E^f)$;

Algorithm 2: ABStAIn's Method for Sorting Edges.

input : Complete Network Graph: $G(V, E)$

output : A sorted list of the graph edges.

1: **begin**
2: $F[V] \leftarrow$ ComputeFiedlerVector $(G(V, E))$
`//Compute the Fiedler Factor, $FF$, for all edges.`
3: **for** $e(u, v) \in E$ **do**
4: $FF(e(u, v)) = |F[u] - F[v]|$;
`//Sort edges in nonincreasing order of Fiedler Factor.`
5: $SE\_List \leftarrow$ Sort $(E, FF[E], nonincreasing)$;
6: **return** SE_List;



(a) Initial state.

(b) After ABStAIn's first iteration.

Figure 2: Edge $(5, 6)$ has the highest weight, consistent with the fact that it is the only edge whose removal would lead to two non-trivial connected components. Fig. a – ABStAIn removes edge $(1, 2)$ first and then recalculates the edge weights. Fig. b – $(2, 3)$ is what now connects vertex 2 to the rest of the graph so its weight has increased significantly, leaving $(1, 3)$ as the next most expendable edge.

present an approach that requires a single calculation of the algebraic connectivity. To accomplish this, ABStAIn relies on what we refer to as the *Fiedler Factor*. Before defining the Fiedler Factor we introduce the definition of the Fiedler Vector.

Definition 6 (Fiedler Vector)**.** *Let $G = (V, E)$ be a simple graph and $L(G)$ its Laplacian matrix. The eigenvector of $L(G)$ corresponding to its second smallest eigenvalue is known as the Fiedler Vector.*

Definition 7 (Fiedler Factor). *Let F be the Fiedler Vector resulting from the Laplacian matrix of the simple graph $G = (V, E)$. For every edge $e(u, v) \in E$ its Fiedler Factor is*

$$FF(e(u, v)) = |F[u] - F[v]| .$$

The following Lemma shows how the Fiedler Factor is related to the algebraic connectivity.

Lemma 4. *Let $G = (V, E)$ be a simple graph and a(G) its algebraic connectivity. Then for all edges $e \in E$ the following holds [10]:*

$$a(G - e) \leq a(G) - FF(e)^2 \tag{4}$$

Eq. 4 indicates that removing the edge whose Fiedler Factor is smallest will have the least negative impact on the algebraic connectivity and, by extension, the adequacy index of the frugal graph. Therefore, ABStAIn's sorting function, Algorithm 2, starts by assigning every edge a weight equal to its Fiedler Factor (lines 3-4). It then simply uses a standard sorting algorithm for sorting edges in non-increasing order of Fiedler Factor (line 5).

Fig. 2 shows an illustration of how ABStAIn works on a simple 8-node graph, shown in its initial state (Fig. 2(a)) and after one iteration of ABStAIn (Fig. 2(b). Every edge is given a weight equal to its Fiedler Factor. ABStAIn considers edge $(5, 6)$ the least expendable, consistent with the fact that it is the only edge whose removal would create two non-trivial connected component. On the other hand, ABStAIn considers edge $(1, 2)$ the most expendable – removing it would still leave the graph connected and vertices 1,3,4,5 would still have two edge-disjoint paths – and gives it the smallest weight. ABStAIn's implementation of the Generic

Algorithm 3: CuTBAck's Method for Sorting Edges.

```
   input  : Complete Network Graph: G(V, E)
   output : A sorted list of the graph edges.
1: begin
       //Compute edge betweenness using Eq.5.
2:     for v ∈ V do
3:         Dijstra (G(V, E), v);
4:         for e ∈ E do
5:             if e ∈ ShortestPathTree then
                   //As G is connected there are
                     |V| − 1 paths on the
                     shortest-path tree.
6:                 btwn(e) ← btwn(e) + 1/((|V|−1));
       //Sort edges in nonincreasing order of
         betweenness.
7:     SE_List ← Sort (E, btwn[E], nonincreasing);
8:     return SE_List;
```

Approach selects edge $(1, 2)$ first (line 6, Algorithm 1), removes it, recalculates the weights and re-sorts the edges.

C. CuTBAck: CenTrality Based Algorithm for Frugality

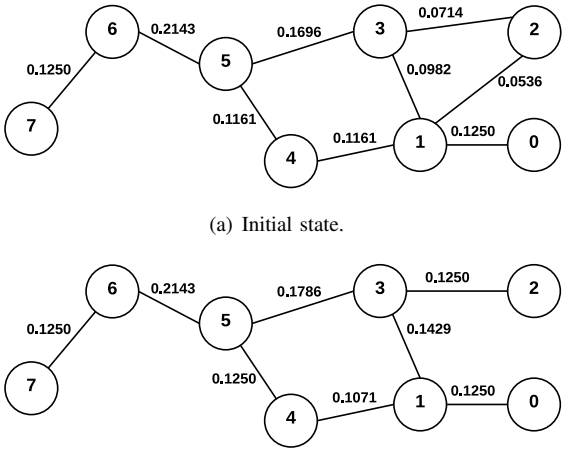
ABStAIn provides an elegant sorting method working directly with the algebraic connectivity on which the adequacy index is founded. However, for matrices of order 5 or more, the eigenvalues and eigenvectors cannot be obtained by an explicit algebraic formula, and must therefore be computed by approximate numerical methods. The popular QR Algorithm takes $O(|V|^3)$ operations per iteration even though in most cases it converges in two iterations [16]. CuTBAck relaxes the requirement for eigenvalue calculations by relying on the edge betweenness centrality – the fraction of all-pairs shortest paths passing through a particular edge [14]. More formally:

Definition 8 (**Link Betweenness Centrality**). *Let $G = (V, E)$ be a simple graph. For every edge $e \in E$ its betweenness centrality is given by the expression:*

$$btwn(e) = \sum_{s \in V} \frac{\sigma(s|e)}{\sigma(s)} \tag{5}$$

where $\sigma(s)$ is the total number of shortest paths on G rooted at vertex s and $\sigma(s|e)$ the number of shortest paths in $\sigma(s)$ passing through edge e.

The betweenness centrality captures the importance of an edge to the shortest paths. The higher the betweenness centrality the higher the disruption will be to the transfer of the data in the network should the link be removed. On the other hand, a link with small betweenness centrality could be removed for frugality without significantly affecting the graph's adequacy. Therefore, CuTBAck builds its sorting algorithm, Algorithm 3, around the betweenness centrality. The algorithm starts by computing the betweenness centrality of all the edges (lines 2-6). This computation is simple – it suffices to run Dijkstra's algorithm from every vertex in the graph. Finally, the edges are sorted in non-increasing order of centrality betweenness by using a standard sorting algorithm (line 7).



(a) Initial state.

(b) After CuTBAck's first iteration.

Figure 3: Edge $(5, 6)$ has the highest weight as all the shortest paths to vertices 6 and 7 have to pass through it. Removing link $(1, 2)$ will have the minimal effect of increasing the shortest path between vertices 1 and 2 by a hop. Fig. a – CuTBAck removes edge $(1, 2)$ first and then recalculates the edge weights. Fig. b – $(2, 3)$ is what now connects vertex 2 to the rest of the graph so its weight has increased significantly, leaving $(1, 4)$ as the next most expendable edge.

Fig. 3 shows an illustration of how CuTBAck works on a simple 8-node graph, shown in its initial state (Fig. 3(a)) and after one iteration of CuTBAck (Fig. 3(b). Every edge is given a weight equal to its betweenness centrality. CuTBAck considers edge $(5, 6)$ the least expendable in the graph. This is consistent with the fact that removing edge $(5, 6)$ would cut the flow of data to two nodes, 5 and 6, more than any other edge. On the other hand, removing edge $(1, 2)$ would cause a minimal disruption to the data flow in the network and thus this link is given the smallest weight by CuTBAck. CuTBAck's implementation of the Generic Approach selects edge $(1, 2)$ first (line 6, Algorithm 1), removes it, recalculates the weights and re-sorts the edges.

V. Performance Evaluation

In this section, we evaluate the performance of ABStAIn and CuTBAck in terms of network links switched off, path stretch and link utilization distribution using real network topologies and traffic matrices. In summary, we make the following main observations:

1) In Section V-B, we show that ABStAIn and CuTBAck, when applied on real network topologies, can switch off between 20%-30% of the network links. For Adequacy Index values between 0.4 and 1, the percentage of switched off links is very close to the optimal solution.
2) In Section V-C, we show that the Adequacy Index correlates very well with the maximum link utilization in the network.
3) In Section V-D, we show that despite the high number of links switched off by ABStAIn and CuTBAck, the shortest paths are only stretched by around 20% on

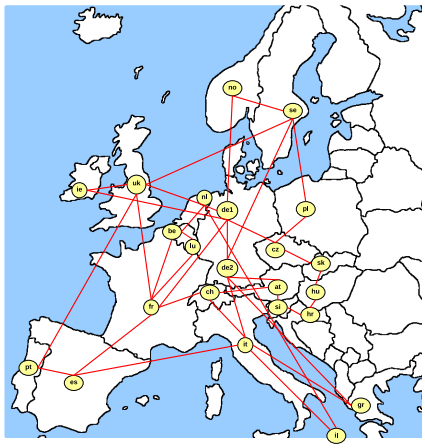


Figure 4: The GÉANT Network.

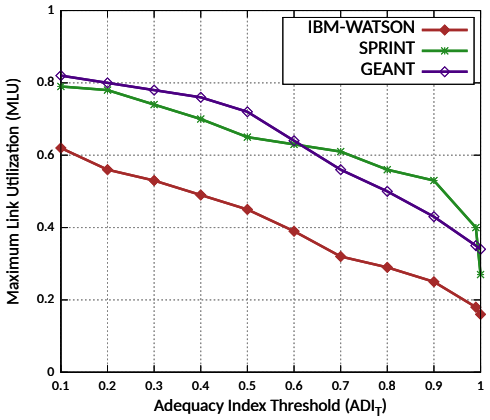


Figure 6: ADI is shown to correlated well with the maximum link utilization. The data is collected on frugal topologies generated by the Optimal solution so as to isolate the {ADI, link utilization} relation from any bias the particularities of a heuristic may introduce.

average.

4) In Section V-E, comparing to two well known traffic based solution, the exponential time Benchmark-MLU [6] and polynomial time Least-Flow [4], ABStAIn and CuTBAck are shown to perform as well as Benchmark-MLU and clearly outperform Least-Flow.
5) ABStAIn, as expected, outperforms CuTBAck but the gap is small enough to justify selecting the latter in applications where running time is a primary concern.

A. Experimental Setup

Network	Usage	Location	Nodes	Links
IBM-Watson	Research	USA	16	49
SPRINT	Commercial	USA	44	212
GÉANT	Academia	Europe	23	74

Table I: Network topologies used in the performance evaluation.

Implementation: A custom simulator is written in Matlab to implement ABStAIn and CuTBAck while the MATLAB code for the solutions proposed in [6] was provided by the authors. The optimization toolbox "OPTI" in Matlab is used for generating optimal frugal topologies.

Topologies and Traffic Matrices: We use 3 real network topologies, representing a mixture of research, education and commercial applications (Table I): IBM-Watson selected from SNDLib [17], SPRINT selected from Rocketfuel [18] and GÉANT, a pan-European (see Fig. 4) research and education network [19].

SND-lib provides real traffic matrices for the IBM-Watson topology but does not provide the link capacities. Thus, we select the link capacities using the method described in [2] – high degree nodes have high capacity links (10Gb/s) with the rest having smaller capacity links (2.5Gb/s). Since Rocketfuel does not provide link capacities and traffic matrices, we assign capacities to links using the same method as for IBM-Watson and generate traffic matrices using the gravity model [20]. In particular, we assume that each router generates traffic for all the other routers: 40% of it is high bit-rate traffic, between 1 Mbit/s and 80 Mbit/s, and the remaining 60%, low bit-rate traffic, up to 1 Mbit/s. For GÉANT, the topology and traffic matrices, measured every 15 minutes, are provided by the authors of [19].

Basis for Comparison: We compare ABStAIn and CuTBAck with the optimal solution computed using the MATLAB "OPTI" tool and two well known traffic based solutions [6], [4]. To measure the value of a carefully designed solution we also compare to a solution that simply removes links at random.

B. Experiment 1: Frugality

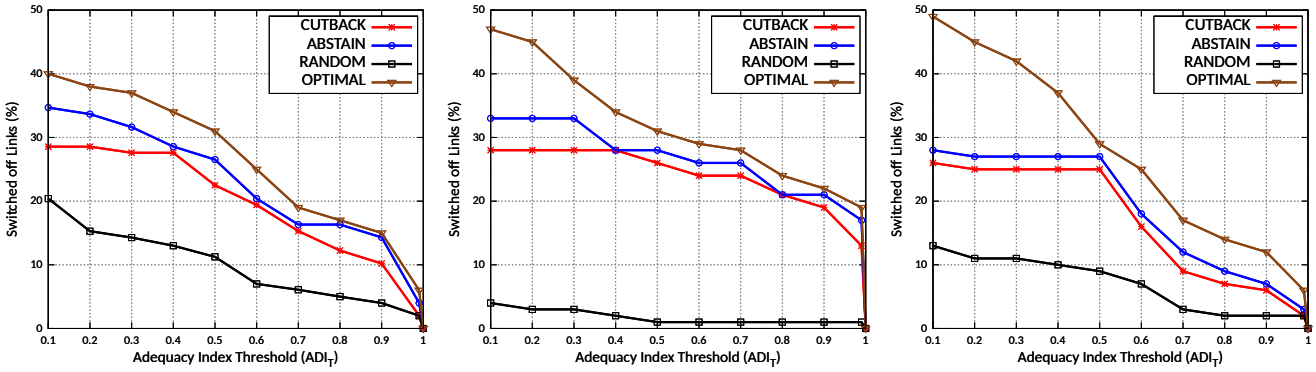
Fig. 5 shows the number of switched off links on all 3 topologies for different values of the Adequacy Index threshold. The data points to three interesting conclusions. First, ABStAIn and CuTBAck are able to switch off a large number of links, between 20%-30%, several times more than the strategy of switching links off at random. Second, ABStAIn and CuTBAck are very competitive when compare to the Optimal solution – for Adequacy Index threshold between 0.4 and 1 ABStAIn and CuTBAck are almost as good as the Optimal and only for values under 0.3 does the Optimal start performing significantly better. Third, while ABStAIn, built on the algebraic connectivity, is expected to outperform CuTBAck, built on the betweenness centrality and simpler to compute, the difference is small.

C. Experiment 2: Adequacy Index vs. Maximum Link Utilization

In this experiment, we evaluate the relation between the Adequacy Index and maximum link utilization in the network.

Method: Optimal frugal topologies for IBM-Watson, Sprint and GÉANT are generated using MATLAB's "OPTI" tool using Adequacy Index threshold values varying from 0.1 to 1. The Optimal solution is favored over using our heuristics so as to focus on the value of Adequacy Index as a metric without the results being biased by the way a particular heuristic works. Once the optimal frugal topology is generated, we introduce off-peak hour[3] traffic using the

[3]We recommend disabling the mechanism for creating frugal topologies during peak hours.



(a) IBM-Watson (b) SPRINT (c) GÉANT

Figure 5: The percentage of switched off links as function of the Adequacy Index threshold.

traffic matrices described in Section V-A and measure the maximum link utilization in the network.

Results: Fig. 6 shows that the Adequacy Index correlates very well with the maximum link utilization measured in the network. What is more, the data for $ADI_T = 1$, equivalent to no frugality (see Eq. 1), demonstrates that networks are severely underutilized, leaving substantial room for saving energy. Indeed, reducing the Adequacy Index threshold leads to more frugal topologies (as shown in Fig. 5) without any link getting saturated. Even for very aggressive values of the Adequacy Index threshold the maximum link utilization is around 80% for GÉANT and Sprint, and only 60% for IBM-Watson.

D. Experiment 3: Effect of Frugality on Shortest Paths

In this experiment, we evaluate the effect of being frugal on the shortest paths. Most works in the subject quantify the effect of frugality by measuring link utilization. However, removing links inevitably leads to longer routes and the end-to-end delay depends not only on buffer delay but also the number of hops a packet has to cross on the way to the destination.

Method: For all 3 topologies under consideration, we compute all-pairs shortest paths before and after applying ABStAIn and CuTBAck. The *path stretch* is defined as the ratio between a shortest path on the frugal topology divided by the shortest path between the same source destination pair on the original graph.

Results: Fig. 7 shows that while there is a price to be paid in terms of expected end-to-end delay for being energy frugal, it is quite low. Even for very aggressive Adequacy Index threshold values the path stretch is between 20% and 30%.

E. Experiment 4: Comparison with Traffic Based Solutions

In this experiment we compare ABStAIn and CuTBAck to two well known traffic based solutions: Least-Flow [4] and what we refer to as Benchmark-MLU [6]. The authors of [6] have proposed a solution to the problem of computing the most frugal topology subject to a maximum link utilization threshold relying on multiple calls to the CPLEX optimizer.

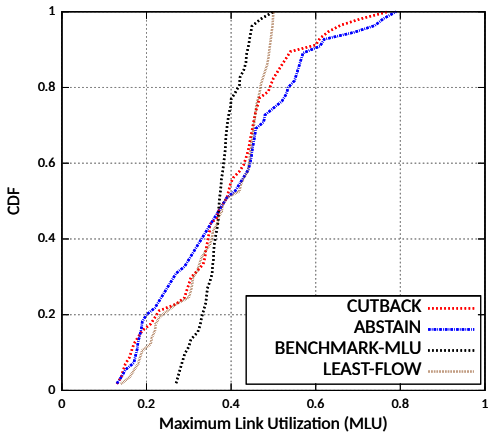


Figure 8: CDF of the link utilization for all heuristics.

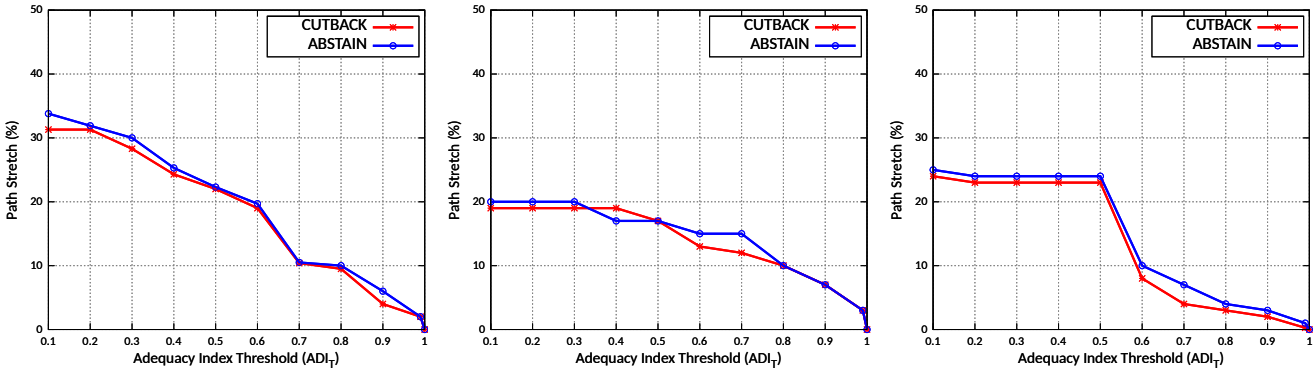
Therefore, while shown to be one of the best traffic based solutions, it can only be used as a benchmark[4].

Algorithm	Switched-off (%)	Path Stretch (%)
Benchmark-MLU (MLU_T = 0.5)	24.4	28.8
Least-Flow (MLU_T = 0.5)	15	20
ABStAIn (ADI_T = 0.5)	27.7	24.6
CuTBAck (ADI_T = 0.5)	25	23

Table II: Comparisons with traffic based approaches.

Method: For this set of experiments we use only the GÉANT network as it is the only network for which we have complete information as to the link capacities as well as real traffic matrixes at 15 mins granularity. For Least-Flow and Benchmark-MLU we use the best settings proposed by their authors. In particular, the maximum link utilization threshold, MLU_T, is set to 0.5 for both. For ABStAIn and CuTBAck the Adequacy Index threshold, ADI_T, is also set

[4]On a Linux PC with eight 3.4 GHz CPUs and 16 GB of RAM a single run of Benchmark-MLU takes from a few hours on GÉANT to around 20 hours on the SPRINT topology.



(a) IBM-Watson (b) SPRINT (c) GÉANT

Figure 7: The impact of frugality on the path stretch.

to 0.5. We run all solutions on the GÉANT topology using off-peak traffic matrices and measure the link utilization in the network, number of links switched off and the path stretch.

Results: Table II shows that ABStAIn switches off the most links, followed closely by CuTBAck and Benchmark-MLU. Least-Flow is a distant forth. Similar results are observed for the path stretch. The fact that ABStAIn and CuTBAck outperform Benchmark-MLU, albeit by a little, is surprising considering the latter uses the CPLEX optimizer for computing topologies very close to the optimal. On the other hand, Benchmark-MLU clearly outperforms all other solutions in terms of traffic distribution in the network, as shown in Fig. 8. As expected, the link utilization never reaches beyond 50% with Least-Flow and Benchmark-MLU. Nevertheless, ABStAIn and CuTBAck, without requiring instantaneous traffic information, still achieve the same median link utilization and no link ever reaches saturation – the maximum link utilization is 80%.

F. Experiment 5: Illustration Using a Power Model

As there is no universally accepted power model and to avoid having results biased by a particular model so far we have used the number of links shut-down as the metric for quantifying the power savings. To demonstrate the potential for power savings in real life, in this experiment we repeat the same experiment as in V-E while using the power model from [9].

Algorithm	Power Saving(%)	Path Stretch(%)
Benchmark-MLU (MLU_T = 0.5)	33	28.8
Least-Flow (MLU_T = 0.5)	19	20
ABStAIn (ADI_T = 0.5)	38	24.6
CuTBAck (ADI_T = 0.5)	34.5	23

Table III: Comparisons using a specific power model.

Table III shows that the results are consistent with those in Table II. ABStAIn and CuTBAck are shown to reduce the power consumption by an impressive 38 and 34.5%, respectively, further underscoring the potential for reducing the energy footprint of IP networks.

VI. Related Work

Traditionally, energy efficiency has not been a major consideration in networking design. However, as reducing energy consumption has emerged as a major scientific and engineering challenge of our time, the subject has attracted more attention from the ICT industry. In a pioneering work [21], Gupta and Singh were among the first to show the benefits of powering down network components and the impact this could have on the network protocols. This stimulated a lot of discussion and many works followed. They can be largely grouped into two major categories.

In the first category fall works that propose to modify the protocols at the heart of IP networks so as to reduce their energy footprints. These include modifications to some of the most popular technologies and protocols, such the OSPF protocol [22], [23], [24] and Ethernet [25], including a new standard for energy efficient Ethernet [26]. However, an energy frugal networking stack does not suffice. ISPs and large organization deploy complex traffic engineering schemes often driven by business policies beyond the reach of the networking stack [27].

In the second category, fall schemes [5], [4], [6], [7], [8], [9], including ABStAIn and CuTBAck, that can be deployed by ISPs and large organizations at the same level as their traffic engineering schemes. They can take the global view of the infrastructure of a large network and render it more energy frugal while complying with the congestion and packet losses levels the particular organization considers acceptable. In [6] the authors define the problem of maximizing the number of links in the network that can be switched off subject to fulfilling specific link utilization and packet delay constraints. The problem is modeled as a mixed integer program and several heuristics capable of reducing the reducing power consumption by 27% to 42% are proposed. However, these schemes have a worst-case exponential complexity and need accurate traffic matrices and network congestion levels. ABStAIn

and CuTBAck run in polynomial time and do not require accurate traffic matrices. Least-Flow [4] is a greedy heuristic that iteratively selects the least loaded link in the network as a candidate for being switched off. However, just because a particular link is lightly loaded does not necessarily mean it is expendable from the perspective of the whole network. This explains why ABStAIn and CuTBAck showed superior performance in our experiments. In [28] a hybrid routing/network design scheme is proposed. However, it depends on solving a mixed integer program and thus is applicable only to very small topologies. A heuristic for switching off links when bundles of multiple physical cables are present is introduced in [5]. This solution can be used to perform an initial pruning of the topology before ABStAIn and CuTBAck are applied.

In summary, ABStAIn and CuTBAck are the first polynomial time schemes that do not require accurate traffic information, can easily be implemented on a central controller, such as an SDN controller [3], and enable large network operators to reduce their energy footprints.

VII. CONCLUSION

We presented ABStAIn and CuTBAck, two polynomial time heuristics for creating frugal IP networks. The new heuristics are based on the Adequacy Index, a novel metric that leverages spectral graph theory for quantifying the quality of a frugal graph. Using topologies and traffic matrices from 3 real networks – IBM-Watson, SPRINT and GÉANT - we showed that ABStAIn and CuTBAck are able to switch off about 25% of the links in the network – 80% as many links as a state of the art polynomial time solution and as many as an exponential time approach. This significant frugality was achieved with the median and maximum link utilization in the network below 40% and 80%, respectively. We believe ABStAIn and CuTBAck give network operators a new option for managing the energy consumption of their infrastructure.

ACKNOWLEDGEMENT

We would like to thank Mingui Zhang for providing us with the source code, topologies and traffic matrices used in [6].

REFERENCES

[1] F. François, N. Wang, K. Moessner, S. Georgoulas, and R. de Oliveira Schmidt, "Leveraging MPLS backup paths for distributed energy-aware traffic engineering," *IEEE Transactions on Network and Service Management*, vol. 11, no. 2, pp. 235–249, 2014.

[2] S. Kandula, D. Katabi, B. S. Davie, and A. Charny, "Walking the tightrope: responsive yet stable traffic engineering," in *ACM SIGCOMM*, 2005, pp. 253–264.

[3] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat, "B4: Experience with a globally-deployed software defined wan," in *ACM SIGCOMM*, 2013.

[4] L. Chiaraviglio, M. Mellia, and F. Neri, "Minimizing isp network energy cost: formulation and solutions," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 2, pp. 463–476, 2012.

[5] W. Fisher, M. Suchara, and J. Rexford, "Greening backbone networks: reducing energy consumption by shutting off cables in bundled links," in *Proceedings of the first ACM SIGCOMM workshop on Green networking*. ACM, 2010, pp. 29–34.

[6] M. Zhang, C. Yi, B. Liu, and B. Zhang, "Greente: Power-aware traffic engineering," in *Network Protocols (ICNP), 2010 18th IEEE International Conference on*. IEEE, 2010, pp. 21–30.

[7] A. Cianfrani, V. Eramo, M. Listanti, M. Marazza, and E. Vittorini, "An energy saving routing algorithm for a green ospf protocol," in *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*. IEEE, 2010, pp. 1–5.

[8] A. P. Bianzino, L. Chiaraviglio, and M. Mellia, "Distributed algorithms for green ip networks," in *Computer Communications Workshops (INFOCOM WKSHPS), 2012 IEEE Conference on*. IEEE, 2012, pp. 121–126.

[9] ——, "Grida: A green distributed algorithm for backbone networks," in *Online Conference on Green Communications (GreenCom), 2011 IEEE*. IEEE, 2011, pp. 113–119.

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

[11] F. R. Chung, *Spectral graph theory*. American Mathematical Soc., 1997, vol. 92.

[12] A. Jamakovic and S. Uhlig, "On the relationship between the algebraic connectivity and graph's robustness to node and link failures," in *Next Generation Internet Networks, 3rd EuroNGI Conference on*. IEEE, 2007, pp. 96–102.

[13] X. F. Wang and G. Chen, "Synchronization in small-world dynamical networks," *International Journal of Bifurcation and Chaos*, vol. 12, no. 01, pp. 187–192, 2002.

[14] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, Mar. 1977.

[15] D. Mosk-Aoyama, "Maximum algebraic connectivity augmentation is np-hard," *Operations Research Letters*, vol. 36, no. 6, pp. 677–679, 2008.

[16] J. G. F. Francis, "The QR transformation — Part 2," *The Computer Journal*, vol. 4, no. 4, pp. 332–345, 1962. [Online]. Available: http://comjnl.oxfordjournals.org/content/4/4/332.abstract

[17] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski, "SNDlib 1.0 — Survivable Network Design Library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.

[18] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," in *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 4. ACM, 2002, pp. 133–145.

[19] S. Uhlig, B. Quoitin, J. Lepropre, and S. Balon, "Providing public intradomain traffic matrices to the research community," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 83–86, 2006.

[20] A. Nucci, A. Sridharan, and N. Taft, "The problem of synthetically generating ip traffic matrices: initial recommendations," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 3, pp. 19–32, 2005.

[21] M. Gupta and S. Singh, "Greening of the internet," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM, 2003, pp. 19–26.

[22] E. Amaldi, A. Capone, L. Gianoli, and L. Mascetti, "Energy management in IP traffic engineering with Shortest Path routing," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, June 2011, pp. 1–6.

[23] F. Cuomo, A. Abbagnale, and S. Papagna, "ESOL: Energy saving in the Internet based on Occurrence of Links in routing paths," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, June 2011, pp. 1–6.

[24] A. Ruiz-Rivera, K.-W. Chin, R. Raad, and S. Soh, "Hotpluz: A bgp-aware green traffic engineering approach," in *Communications (ICC), 2014 IEEE International Conference on*, June 2014, pp. 3721–3726.

[25] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. Maestro, "IEEE 802.3az: the Road to Energy Efficient Ethernet," *Communications Magazine, IEEE*, vol. 48, no. 11, pp. 50–56, November 2010.

[26] "IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 3: CSMA/CD Access Method and Physical Layer Specifications Amendment 5: Media Access Control Parameters, Physical Layers, and Management Parameters for Energy-Efficient Ethernet," *IEEE Std 802.3az-2010 (Amendment to IEEE Std 802.3-2008)*, pp. 1–302, Oct 2010.

[27] H. H. Liu, S. Kandula, R. Mahajan, M. Zhang, and D. Gelernter, "Traffic engineering with forward fault correction," in *ACM SIGCOMM*, 2014.

[28] J. Chabarek, J. Sommers, P. Barford, C. Estan, D. Tsiang, and S. Wright, "Power awareness in network design and routing," in *IEEE INFOCOM 2008*, 2008.