Joint Optimization of Resources and Routes: From Communication Networks to Power Grids

Ali Tizghadam and Alberto Leon-Garcia

Abstract In this chapter we are concerned with robustness design in complex communication networks. We define robustness as the ability of a network to adapt to environmental variations such as traffic fluctuations, topology modifications, and changes in the source of external traffic. We present a network theory approach to the joint optimization of resources and routes in a communication network to provide robust network operation. Our main metrics are the well-known point-to-point resistance distance and network criticality (total resistance distance) of a graph. We show that some of the key performance metrics in a communication network, in particular average network utilization, can be expressed as a weighted combination of point-to-point resistance distances. A case of particular interest is when the external demand is specified by a traffic matrix. We extend the notion of network criticality to be a traffic-aware metric. Traffic-aware network criticality (TANC) is then a weighted linear combination of point-to-point resistance distances of the graph. For this reason in this chapter we focus on a weighted linear sum of resistance distances (which is a convex function of link weights) as the main metric and we discuss a variety of optimization problems to jointly assign routes and flows in a network. We provide a complete mathematical analysis of the optimization problem for the case where the routing in the network is already known (network planning), and we then extend the analysis to the more general case involving the simultaneous optimization of resources and flows (routes) in the network (traffic engineering). Furthermore, we briefly discuss the problems of finding the best set of demands that can be matched to a given network topology. We discuss applications of the proposed optimization methods to the design of virtual networks.

Ali Tizghadam
University of Toronto, Toronto, Canada, e-mail: ali.tizghadam@utoronto.ca

Alberto Leon-Garcia
University of Toronto, Toronto, Canada, e-mail: alberto.leongarcia@utoronto.ca

Moreover, we show how our techniques can be used in the design of robust power grids.

1 Centrality Measures

Centrality measures in graph theory quantify the structural importance or prominence of nodes and links. Numerous centrality indices have been introduced and studied in literature, but one can categorize these indices into three major classes: reachability measures, vitality measures, and flow measures. In reachability indices a node is central if it reaches many other nodes, or can be reached by many other nodes. All the centrality measures in this category use some form of distance between two nodes. For instance, degree centrality which is a well-known reachability index counts the number of nodes that can be reached within distance 1.

Vitality measures are the second class of commonly used centrality index. Given a real-valued function on a graph G, a vitality measure quantifies the difference between the value of the function on G with the presence or absence of a node or a link. For example, in a wireless mobile network, the main goal is to keep connectivity among all the nodes via peripheral (or relay) nodes. Algebraic connectivity (smallest non-zero eigenvalue of the graph Laplacian matrix) is an appropriate choice for real-valued functions on the graph of the mobile network. In this case the vitality of a node or link denotes the change of this function (algebraic connectivity) if that node or link was removed from the network.

Finally, we have flow centrality measures. Let γ_{sd} denote the amount of flow entering at node s destined for node d. Flow indices quantify how much of this flow traverses a specific node or link. In this chapter our focus is on flow centralities, and in particular we are interested in different variations of betweenness centrality as the most useful flow measure. We refer the reader to [1] for a complete review of centrality measures.

2 Betweenness Centrality Measures

Freeman [2] introduced a very useful metric in graph theory referred to as shortest-path betweenness centrality. For node k the shortest-path betweenness centrality with respect to flows from source node s to destination node d is defined as the proportion of instances of the shortest paths from node s to d that traverse node k. This can be interpreted as the probability that node k is involved in any communication between s and d with the assumption that all communication is along equiprobable shortest paths. The overall shortest-path betweenness centrality of node k is the sum of the centralities over all source-destination pairs. Link betweenness is defined similarly. The concept of betweenness centrality is closely related to the principle of conservation of flow in a communication network. According to this principle for any intermediate node (or link) in a communication path, the incoming flow is equal to the outgoing flow. When we count the proportion of instances that a node is involved in a communication along shortest paths (or more generally a path), we implicitly assume the conservation of flow.

A major drawback of the shortest-path betweenness is that in communications networks it is frequently desirable to take a path other than the shortest path. To overcome this issue other betweenness centrality metrics have been proposed. In [3] the authors introduce flow betweenness centrality. Suppose that each link of the network is capable of transferring a unit flow of information. The flow betweenness of a node k is defined as the proportion of flow through node k when maximum flow is transmitted from source s to destination d averaged over all $s-d$ pairs. The maximum flow that can be sent between two nodes is in general more than a unit flow since one can use multiple paths to transmit information. The flow betweenness is in fact a vitality measure because the amount of flow passing along node k can be found in this way. Let f_{sd} denote the maximum flow that can be transmitted from source node s to destination d. Further, suppose we remove node k (and its incident links) from the graph, and let f_{sd}^k denote the maximum flow that can be sent from s to d in the new graph. Then the flow traversing node k is: $f_{sd} - f_{sd}^k$. In fact, the flow betweenness measures the betweenness centrality of network nodes when maximum possible flows are fed into the network (between every pair of nodes). While the flow betweenness considers paths other than shortest paths, it suffers from some of the limitations in the definition of shortest path betweenness. In maximum flow problems we still have one (or more) ideal path(s) mandating the communication (just like shortest path), however in many practical situations flow does not take ideal paths either shortest path, max-flow path, or any other type of ideal path.

Deterministic betweenness is a straightforward extension of shortest path betweenness and is proposed in [4]. Deterministic betweenness of a node (or link) k is the fraction of total paths between a source s and destination node d traversing node (or link) k, averaged over all active traffic sources and sinks, which we refer to as the community of interest. One can easily recognize two main differences between deterministic betweenness and the original shortest-path betweenness. First, in the former all the paths are involved, whereas in the latter only shortest paths are considered. Second, in deterministic betweenness only the active path set (paths within a community of interest) is involved in definition of betweenness, whereas in original shortest path betweenness, all possible node pairs are considered. Unfortunately the enumeration of paths does not lend itself to tractable analytic results.

In order to overcome these tractability problems Newman [5] proposed random-walk betweenness which is a probabilistic approach to define and analyze the betweenness of a node/link. We begin with a graph model.

An undirected graph $G(V,E,W)$ consists of a finite node set V which contains n nodes, together with a link set $E \in V \times V$, and each link has an associated non-negative weight w_{ij}. The weight of a node i is defined as $W_i = \sum_j w_{ij}$. We can define a transition probability matrix $P = [p_{ij}]$ of an irrecucible Markov random walk through the undirected graph which satisfies $\sum_j p_{ij} = 1 \ \forall i \in V$. In this work we consider a connected graph and use the weighted random walk whose transition probability is defined as $p_{ij} = \frac{w_{ij}}{W_i}$. Moreover, we define weighted graph Laplacian as $L = D - W$, where D is a diagonal matrix whose main diagonal entries are: $D(i,i) = W_i$.

We are now ready to define random-walk betweenness. Consider the set of trajectories that begin at node s and terminate when the walk first arrives at node d, that is, destination node d is an absorbing node. The random-walk betweenness $b_{sk}(d)$ of node k for the $s-d$ trajectories is defined as the number of times node k is visited in trajectories from s to d averaged over all possible $s-d$ trajectories and the total betweenness of node k is $b_k = \sum_{s,d} b_{sk}(d)$.

Let $B_d = [b_{sk}(d)]$ be the $n \times n$ matrix of betweenness metrics of node k for walks that begin at node s and end at node d. Note that the d^{th} row of the matrix is zero. It is shown in [5] that matrix B_d can be written as:

$$B_d = (I - P_d)^{-1} \Theta_d \tag{1}$$
$$\Theta_d = [\theta_{sk}(d)] = \begin{cases} 1 \ if \ s = k \neq d \\ 0 \ otherwise \end{cases}$$

Matrix P_d is the same as P except that its d^{th} row and d^{th} column are zero vectors.In this paper we are interested in a special type of random-walks referred to as weight-based random-walk. The weight-based random-walk is defined on a Markov chain with transition probability matrix P according to the following rule:

$$p_{sk}(d) = \frac{w_{sk}}{\sum_{q \in A(s)} w_{sq}} (1 - \delta(s - d)) \tag{2}$$

where $A(s)$ is the set of adjacent nodes of s and w_{sk} is the weight of link (s, k) (if there is no link between node s and k, then $w_{sk} = 0$), and $\delta(.)$ is the Kronecker delta function (i.e. $\delta(x) = 1$ if $x = 0$ and $\delta(x) = 0$ otherwise). The delta function in equation (2) is due to the fact that the destination node d is an absorbing node, and any random-walk coming to this state, will be absorbed or equivalently $p_{dk}(d) = 0$. Clearly, equation (2) defines a Markovian random-walk.

Finally, traffic-aware betweenness [6] is a natural extension to explicitly account for the effect of traffic demands in the definition of betweenness centrality index. Let $\Gamma = [\gamma_s(d)]$ and γ denote the traffic matrix and the total external traffic (i.e. $\gamma = \sum_{s,d} \gamma_s(d)$) respectively. We define traffic-aware betweenness (TAB) of node k as:

$$b'_{sk}(d) = b_{sk}(d) + \frac{\gamma_s(d)}{\gamma} b_{sk}(d)$$

$$b'_{sk}(d) = (1 + \frac{\gamma_s(d)}{\gamma}) b_{sk}(d) \tag{3}$$

$$b'_k = \sum_{s,d} (1 + \frac{\gamma_s(d)}{\gamma}) b_{sk}(d) \tag{4}$$

If traffic matrix is zero, then we have original topological betweenness as a special case. This definition is quite general and applicable for all types of betweenness. If we consider $b_{sk}(d)$ as the shortest-path betweenness of node k for source-destination pair sd, then we will have traffic-aware shortest-path betweenness, and so on.

2.1 Network Criticality and Resistance Distance

We now introduce *network criticality*, which is introduced in [7] to quantify the robustness of a network. We start by defining node/link criticality.

Definition 1. Node criticality is defined as the random-walk betweenness of a node normalized by its weight value. in other works node criticality is the node random-walk betweenness divided by the node weight. Likewise, link criticality is defined as the betweenness of the link over its weight.

Let η_k be the criticality of node k and η_{ij} be the criticality of link $l = (i, j)$. It is shown in [7] that η_i and η_{ij} can be obtained by the following expressions:

$$\frac{b_{sk}(d)}{W_k} = l^+_{dd} - l^+_{sd} - l^+_{dk} + l^+_{sk} \tag{5}$$

$$\tau_{sd} = l^+_{ss} + l^+_{dd} - 2l^+_{sd} \;\; or \;\; \tau_{sd} = u^t_{sd} L^+ u_{sd} \tag{6}$$

$$\tau_{sd} = \frac{b_{sk}(d) + b_{dk}(s)}{W_k} \tag{7}$$

$$\eta_k = \frac{b_k}{W_k} = \frac{1}{2}\tau \;\; , \quad \tau = \sum_s \sum_d \tau_{sd} \tag{8}$$

$$\eta_{ij} = \frac{b_{ij}}{w_{ij}} = \tau \tag{9}$$

where L^+ is the Moore-Penrose inverse of graph Laplacian matrix L [8], n is the number of nodes, and $u_{ij} = [0 \ldots 1 \ldots -1 \ldots 0]^t$ (1 and −1 are in i^{th} and j^{th} positions respectively). We define the average network criticality as $\bar{\tau} = \frac{1}{n(n-1)}\tau$.

Equations (5) to (9) show that node criticality (η_k) and link criticality (η_{ij}) are independent of the node/link position and only depend on τ (or $\bar{\tau}$) which is a global quantity of the network.

Definition 2. We refer to τ_{sd} as *point-to-point network criticality* and τ as *network criticality*.

One can see that τ is a global quantity on the network graph. Equations (8) and (9) show that node (link) betweenness consists of a local parameter (weight) and a global metric (network criticality). τ can capture the effect of topology through the betweenness values. The higher the betweenness of a node/link, the higher the risk (criticality) in using the node/link. Furthermore, one can define node/link capacity as the weight of a node/link, then the higher the weight of a node/link, the lower the risk of using the node/link. Therefore network criticality can quantify the risk of using a node/link in a network which in turn indicates the degree of robustness of the network.

Network criticality can be interpreted as the total resistance of a corresponding electrical network. Consider an electrical circuit with the same graph as our original network graph, and with link resistances equal to the reciprocal of link weights. It can be shown that the network criticality is equal to the total resistance distance (effective resistance) [12] seen between different pairs of nodes in the electrical circuit. A high network criticality is an indication of high resistance in the equivalent electrical circuit, therefore, in two networks with the same number of nodes, the one with lower network criticality is better connected, hence better positioned to accommodate network flows. Furthermore, network criticality quantifies the sensitivity of a network to the environmental changes. It has been shown that network criticality equals the average of link betweenness sensitivities, where link betweenness sensitivity is defined as the partial derivative of link betweenness with respect to the corresponding link weight [9]:

$$\tau = \frac{1}{m-1} \sum_{(i,j)\in E} \frac{\partial b_{ij}}{\partial w_{ij}} \tag{10}$$

Equation (10) states that minimization network criticality results in minimization of the average sensitivity of link betweennesses with respect to the changes in link weights (which in turn captures sensitivity to environmental changes). Therefore, a control algorithm for minimum network criticality balances the betweenness of the links in such a way to keep the average sensitivity below a desired level. From another point of view, the lower the network criticality, the better distributed is the traffic between all the links of a network, and the better balanced the load of the traffic among all active links. This implies better fairness in routing the traffic in the nodes of the network. More detailed information on properties and interpretations of network criticality can be found in [10, 11].

Another advantage of having low network criticality is the robustness enhancement of the network. Suppose that a node is failing or becoming inaccessible so that it is not able to route the traffic passing through it. If we adapt the routing to minimize the criticality, the result is to adjust the betweenness in such a way that traffic is re-routed to other nodes instead of the impaired one and that the resulting flows provide higher robustness against additional unpredictable deleterious situations.

It has been shown that τ_{sd} is a convex function of link weights and τ is a strictly convex function of link weights [13].

Definition 3. In analogy to equation (8) we define traffic-aware node criticality (TANOC) as the traffic-aware node betweenness normalized by node weight (sum of the link weights incident to the node):

$$\tau'_{sk}(d) = 2\frac{b'_{sk}(d)}{W_k}, \text{ where } W_k = \sum_j w_{kj}$$

$$\tau'_k = 2\frac{b'_k}{W_k} \tag{11}$$

Furthermore, we define the traffic-aware network criticality (TANC) as the average traffic-aware node criticality:

$$\tau' = \frac{1}{n}\sum_k \tau'_k \tag{12}$$

3 Random-Walk Betweenness in Data Networks

Random-walk betweenness is closely related to packet network models. We consider a packet switching network in which external packets arrive to packet switches according to independent arrival processes. Each packet arrival has a specific destination and the packet is forwarded along the network until it reaches said destination. We assume that packet switches are interconnected by transmission lines that can be modeled as single-server queues. Furthermore, we suppose that packet switches use a form of routing where the proportion of packets at queue i forwarded to the next-hop queue j is p_{ij}.

We calculate the total arrival/departure rate of the traffic to/from each node. The total input rate of node k (internal plus external) is denoted by x_k. After receiving service at the i^{th} node, the proportion of customers that proceed to node k is p_{ik}. To find x_k we need to solve the following set of linear equations (see [14]):

$$x_k = \gamma_k + \sum_{i=1}^{n} x_i p_{ik} \tag{13}$$

where γ_k is the external arrival rate to node k. Note that equation (13) is essentially the KCL (Kirchhoff's Current Law). If we denote $\overrightarrow{x} = [x_1, x_2, ..., x_n]$ and $\overrightarrow{\gamma} = [\gamma_1, \gamma_2, ..., \gamma_n]$, then equation (13) becomes:

$$\overrightarrow{x} = \overrightarrow{\gamma} + \overrightarrow{x} P \tag{14}$$

Suppose we focus on traffic destined to node d, then node d is an absorbing node, and we suppose that the arrival rate at node d is zero (since said arrivals do not affect other nodes) and equation (14) can be written as:

$$\overrightarrow{x_d} = (\overrightarrow{\gamma_d} + \overrightarrow{x_d} P_d) \times \Theta_d \tag{15}$$

where $\overrightarrow{x_d}$ and $\overrightarrow{\gamma_d}$ are the same as $\overrightarrow{x}$ and $\overrightarrow{\gamma}$ except for the d^{th} element which is 0. Matrix P_d is also the same as P except that its d^{th} row and d^{th} column are zero vectors. Equation (15) can be solved for $\overrightarrow{x_d}$.

$$\overrightarrow{x_d} = \overrightarrow{\gamma_d} \times \Theta_d \times (I - P_d \times \Theta_d)^{-1} \tag{16}$$

To find the relationship of betweenness B_d and the input arrival rate x_k we notice that $p_{dk}(d) = 0$ which means that $P_d = \Theta_d \times P_d$. Thus:

$$\begin{aligned}
P_d \times \Theta_d &= \Theta_d \times P_d \times \Theta_d \\
\Theta_d - P_d \times \Theta_d &= \Theta_d - \Theta_d \times P_d \times \Theta_d \\
(I - P_d) \times \Theta_d &= \Theta_d \times (I - P_d \times \Theta_d) \\
\Theta_d \times (I - P_d \times \Theta_d)^{-1} &= (I - P_d)^{-1} \times \Theta_d
\end{aligned}$$

Using equation (1) we will have:

$$\Theta_d \times (I - P_d \times \Theta_d)^{-1} = B_d \tag{17}$$

We substitute equation (17) in (16) and obtain the relationship between the node traffic and node betweenness.

$$\overrightarrow{x_d} = \overrightarrow{\gamma_d} \times B_d \tag{18}$$

If we denote the k^{th} element of $\overrightarrow{x_d}$ and $\overrightarrow{\gamma_d}$ by $x_k(d)$ and $\gamma_k(d)$ respectively, we have:

$$x_k(d) = \sum_s \gamma_s(d) b_{sk}(d) \tag{19}$$

Now we can find the total load at node k by adding the effect of all destinations in equation (19).

$$x_k = \sum_d x_k(d) = \sum_{s,d} \gamma_s(d) b_{sk}(d) \tag{20}$$

It is constructive to establish the relationship of node betweenness and node traffic in a more intuitive way. Consider the traffic generated by packets that arrive at s and are destined for d. Each packet in this flow generates $b_{sk}(d)$ arrivals on average at node k. Let $\gamma_s(d)$ be the number of external packets per second that arrive at node s with destination d. Over a large number of such trials, say N, the average number of times node k is visited will be approximately $N \times b_{sk}(d)$. Suppose that it takes T seconds to have N arrivals at node s, then the average number of visits per second to node k is $\frac{N \times b_{sk}(d)}{T} = \gamma_s(d) \times b_{sk}(d)$, since the average arrival rate at s for d is approximately $\frac{N}{T}$.

We only consider external arrivals with destinations other that the originating node so $\gamma_{dd} = 0$. The total traffic $x_{sk}(d)$ generated by the $s-d$ flow at node k is then $\gamma_s(d) \times b_{sk}(d)$, where s is not equal to d. Recalling that $b_{sd}(d) = 1$, we obtain:

$$x_{sk}(d) = \begin{cases} \gamma_s(d) b_{sk}(d) & if\ s \neq d\ \&\ d \neq k \\ \gamma_s(d) & if\ s \neq d\ \&\ k = d \\ 0 & if\ s = d \end{cases}$$

The total traffic into node k is obtained by summing over all s and d, with s not equal to d

$$\begin{aligned} y_k &= \sum_{s,d} x_{sk}(d) \\ &= \sum_{s \neq k} \gamma_s(k) + \sum_{s \neq d} \sum_{d \neq k} \gamma_s(d) b_{sk}(d) \end{aligned} \tag{21}$$

The first sum on the right hand side of equation (21) is the total network packet arrival rate destined for k, that is, the total flow absorbed at node k.

The second term is the total traffic that flows across queue k, that is, the flow through k that originates at nodes other than d and that are not destined for k. This second term, the transit flow through queue k, accounts for the effect of the network topology, so we let x_k denote this flow:

$$x_k = \sum_{d \neq k} \gamma_k(d) b_{kk}(d) + \sum_{s \neq k} \sum_{d \neq k} \gamma_s(d) b_{sk}(d) \tag{22}$$

The first sum in equation (22) is the arrivals at k destined for d, including revisits. The second sum is the total transit traffic through k that did not originate locally. x_k can be viewed as a measure of betweenness of queue k that takes the different arrival rates into account.

Suppose that different queues have different total external arrival rates but the fraction of external traffic destined for d does not depend on s, that is,

$$\gamma_s(d) = \gamma_s a_d$$

where

$$a_d \geq 0, \quad \sum_d a_d = 1$$

The total traffic through queue k is then

$$\begin{aligned} x_k &= \sum_{d \neq k} \gamma_k a_d b_{kk}(d) + \sum_{s \neq k} \gamma_s [\sum_{d \neq k} a_d b_{sk}(d)] \\ &= \gamma_k [\sum_{d \neq k} a_d b_{kk}(d)] + \sum_{s \neq k} \gamma_s [\sum_{d \neq k} a_d b_{sk}(d)] \end{aligned} \tag{23}$$

The terms inside the square brackets in equation (24) can be viewed as betweenness measures that have been weighted by the differential preferences for destinations according to a_d. These weighted betweenness measures are in turn scaled according to the arrival rates at different queues.

In the case where arriving packets are equally likely to be destined to any destination (other than the arriving node), we have $a_d = \frac{1}{n-1}$, so

$$\begin{aligned} x_k &= \frac{\gamma_k}{n-1} [\sum_{d \neq k} b_{kk}(d)] + \sum_{s \neq k} \frac{\gamma_s}{n-1} [\sum_{d \neq k} b_{sk}(d)] \\ &= \gamma_k \bar{b}_{kk} + \sum_{s \neq k} \gamma_s \bar{b}_{sk} \end{aligned} \tag{24}$$

Finally suppose that the arrival rate at every node is equal, that is, $\gamma_s = \frac{\gamma}{n}$, where γ is the total external packet arrival rate to the network then

$$x_k = \frac{\gamma}{n(n-1)}(\sum_{d\neq k} b_{kk}(d)) + \frac{\gamma}{n(n-1)}\sum_{s\neq k}\sum_{d\neq k} b_{sk}(d)$$
$$= \frac{\gamma}{n(n-1)} b_k \quad (25)$$

where we define b_k as the random walk betweenness for node k:

$$b_k = \sum_s \sum_{d\neq k} b_{sk}(d)$$

We have derived the following theorem.

Theorem 1. *Consider a network with n nodes, and assume that the average traffic rate on all of the nodes is $\gamma_n = \frac{\gamma}{n}$ where γ is the total external input traffic rate of the network. Let x_k be the total arrival rate of a node k and b_k be the total betweenness of this node, then:*

$$x_k = \frac{\gamma_n}{n-1} b_k = \frac{\gamma}{n(n-1)} \times b_k$$

4 Network Utilization and Network Criticality

In this section we derive a general expression for the average network utilization (and individual node utilization). Theorem 1 establishes a connection between the load of a node and its betweenness when the average input rate to all the nodes is uniform. In general, for a traffic matrix $\Gamma = [\gamma_i(j)]$ the utilization of a node can be expressed as a linear combination of point-to-point network criticalities subject to considering weight-based random-walks as defined in equation (2). To see this consider equation (19):

$$x_k = \sum_{s,d} \gamma_s(d) b_{sk}(d)$$
$$= \frac{1}{2}\sum_{s,d}(\gamma_s(d) b_{sk}(d) + \gamma_d(s) b_{dk}(s))$$
$$= \frac{1}{2}\sum_{s,d}(\gamma_s(d) b_{sk}(d) + \gamma_d(s)(W_k \tau_{sd} - b_{sk}(d)))$$
$$= \frac{W_k}{2}\sum_{s,d}\gamma_d(s)\tau_{sd} + \frac{1}{2}\sum_{s,d}(\gamma_s(d) - \gamma_d(s)) b_{sk}(d) \quad (26)$$

where we have used equation (7) to obtain equation (26). Now we write $b_{sk}(d)$ in terms of different elements of matrix $\Omega = [\tau_{sd}]$. We have:

$$\tau_{sd} = l^+_{ss} + l^+_{dd} - 2l^+_{sd}$$
$$\tau_{dk} = l^+_{dd} + l^+_{kk} - 2l^+_{dk}$$
$$\tau_{sk} = l^+_{ss} + l^+_{kk} - 2l^+_{sk}$$

Therefore

$$\begin{aligned}\tau_{sd} + \tau_{dk} - \tau_{sk} &= 2(l^+_{dd} - l^+_{sd} - l^+_{dk} + l^+_{sk}) \\ &= 2\frac{b_{sk}(d)}{W_k} \\ b_{sk}(d) &= \frac{W_k}{2}(\tau_{sd} + \tau_{dk} - \tau_{sk}) \qquad (27)\end{aligned}$$

Using equation (27) in (26), we have

$$\begin{aligned}x_k &= \frac{W_k}{2}\sum_{s,d}\gamma_d(s)\tau_{sd} + \frac{W_k}{4}\sum_{s,d}(\gamma_s(d) - \gamma_d(s))(\tau_{sd} + \tau_{dk} - \tau_{sk}) \\ \frac{x_k}{W_k} &= \frac{1}{4}\sum_{s,d}(\gamma_s(d) + \gamma_d(s))\tau_{sd} + \frac{1}{4}\sum_{s,d}(\gamma_s(d) - \gamma_d(s))(\tau_{dk} - \tau_{sk}) \qquad (28)\end{aligned}$$

Node utilization is defined as the load of a node normalized by its capacity (or in a more general sense by its weight). We denote the utilization of node k by $V_k = \frac{x_k}{W_k}$ and the average network utility by $\bar{V} = \frac{\sum_k V_k}{n}$. For the average network utilization $\bar{V}$ we have:

$$\bar{V} = \frac{1}{4}\sum_{s,d}(\gamma_s(d) + \gamma_d(s))\tau_{sd} + \frac{1}{4n}\sum_{s,d}(\gamma_s(d) - \gamma_d(s))(\tau_{d*} - \tau_{s*}) \qquad (29)$$

where $\tau_{i*} = \sum_k \tau_{ik}$. Equation (29) can be simplified as:

$$\bar{V} = \sum_{s,d}\beta_{sd}\tau_{sd} \qquad (30)$$

where $\beta_{sd} = \frac{\gamma_{sd}+\gamma_{ds}}{4} + \frac{\gamma_{*s}-\gamma_{s*}}{2n}$.

We can easily express the average network utilization $\bar{V}$ in terms of network criticality and traffic-aware network criticality. Since $x_k = \sum_{sd}\gamma_{sd}b_{sk}(d)$, from equation (4) we conclude that:

$$\begin{aligned}b'_k &= b_k + \frac{1}{\gamma}x_k \\ V_k &= \frac{x_k}{W_k} = \gamma\frac{b'_k - b_k}{W_k} \\ V_k &= \frac{\gamma}{2}(\tau'_k - \tau)\end{aligned}$$

Finally:

$$\bar{V} = \frac{1}{n}\sum_{k=1}^{n} V_k = \frac{\gamma}{2}(\tau' - \tau) \tag{31}$$

Proceeding as we did for $\bar{V}$, one can see that:

$$\tau' = \sum_{s,d}(1 + \frac{\gamma_{sd} + \gamma_{ds}}{2\gamma} + \frac{\gamma_{*s} - \gamma_{s*}}{n\gamma})\tau_{sd} \tag{32}$$

It will be easily verified that the coefficients of τ_{sd} in equation (32) (i.e. $1 + \frac{\gamma_{sd}+\gamma_{ds}}{2\gamma} + \frac{\gamma_{*s}-\gamma_{s*}}{n\gamma}$) are always non-negative; therefore, TANC is a convex function of link weights since τ_{sd} is always convex. Consequently, form equation (31) one can see that the average network utilization is in most general form the difference of two convex functions (or equivalently the sum of a convex and a concave function). Minimizing the difference of two convex functions can be converted to a convex maximization problem, which can be numerically solved with methods like branch-and-bound [15].

We can find a subset of traffic matrices, for which the average network utilization $\bar{V}$ is pure convex. In fact the average network utilization is a convex function of link weights if and only if in equation (30) we have $\forall s,d \;\; \beta_{sd} + \beta_{ds} \geq 0$ (note that $\tau_{sd} = \tau_{ds}$), or:

$$\gamma_{sd} + \gamma_{ds} \geq \frac{1}{n}(\gamma_{s*} - \gamma_{*s} + \gamma_{d*} - \gamma_{*d}) \tag{33}$$

Equation (33) defines a subset of all possible traffic matrices for which the average network utilization is convex. In the following, we assume that the traffic matrices are within this subset.

This motivates the rest of this chapter. In order to minimize the average network utilization (or to minimize the maximum of node utilization) we have to effectively solve a convex optimization problem, which is investigated in next section.

5 Minimizing Weighted Network Criticality

We first consider a general weighted version of network criticality (WNC) defined as follows.

$$\tau_\alpha = \sum_{i,j} \alpha_{ij}\tau_{ij}, \;\; \forall i,j \in N \;\; \alpha_{ij} + \alpha_{ji} \geq 0 \tag{34}$$

Obviously, the average network utilization and individual node utilization are special cases of WNC by appropriate selection of coefficients. To study the minimization of WNC, we rewrite WNC in a matrix form as follows:

$$\begin{aligned}\tau_\alpha &= \sum_{i,j} \alpha_{ij}\tau_{ij} \\ &= \sum_{ij} \alpha_{ij} u_{ij}^t L^+ u_{ij} \\ &= \sum_{ij} \alpha_{ij} Tr(u_{ij}u_{ij}^t L^+) \\ &= Tr(U_\alpha L^+) \end{aligned} \tag{35}$$

where $U_\alpha = \sum_{ij} \alpha_{ij} U_{ij}$ and $U_{ij} = u_{ij}u_{ij}^t$.

It is easy to see that the sum of the rows in U_α is zero, and for $\alpha_{ij} \geq 0$ it is a symmetric and positive semidefinite matrix. One example of U_α for $n = 3$ (number of nodes) is given in the following:

$$U_\alpha = \begin{pmatrix} \alpha'_{12} + \alpha'_{13} & -\alpha'_{12} & -\alpha'_{13} \\ -\alpha'_{12} & \alpha'_{12} + \alpha'_{23} & -\alpha'_{23} \\ -\alpha'_{13} & -\alpha'_{23} & \alpha'_{13} + \alpha'_{23} \end{pmatrix}$$

where $\alpha'_{ij} = \alpha_{ij} + \alpha_{ji}$.

We now consider minimization of WNC. First we show that the minimization is viable. To this end we need the following lemma.

Lemma 1. *The partial derivative of τ_α with respect to link weight w_{ij} is always non-positive and can be obtained from the following equation.*

$$\frac{\partial \tau_\alpha}{\partial w_{ij}} = -\left\| F_\alpha L^+ u_{ij} \right\|^2$$

where F_α is a matrix such that $U_\alpha = F_\alpha^t F_\alpha$. This decomposition is always possible because U_α is a positive semidefinite matrix.

Proof. See Appendix 9.

Since WNC is a convex function and its derivative with respect to the weights is always non-positive (according to lemma 1), the minimization of τ_α subject to some convex constraint set is possible.

In formulating the optimization problem, we add a maximum budget constraint to the problem. We assume that there is a cost z_{ij} to deploy each unit of weight on link (i, j). We also assume that there is a maximum budget of C to spend across all network links. This constraint means that $\sum_{(i,j)\in E} w_{ij} z_{ij} \leq C$. Now we can write our optimization problem as follows:

$$\begin{array}{c} Minimize \quad \tau_\alpha \\ Subject\ to \quad \sum_{(i,j)\in E} w_{ij} z_{ij} \le C \quad ,C\ is\ fixed \\ w_{ij} \ge 0 \ \ \forall (i,j) \in E \end{array} \tag{36}$$

Assuming $\Gamma = L + \frac{J}{n}$, and considering the fact that $L = \sum_{i,j} w_{ij} u_{ij} u_{ij}^t$ (definition of Laplacian) and $L^+ = \Gamma^{-1} - \frac{J}{n}$ [16], where J is a square $n \times n$ matrix with all entries equal to 1, we can write the optimization problem (36) as:

$$\begin{array}{c} Minimize \quad Tr(U_\alpha L^+) \\ Subject\ to \quad \Gamma = \sum_{(i,j)\in E} w_{ij} u_{ij} u_{ij}^t + \frac{J}{n} \\ L^+ = \Gamma^{-1} - \frac{J}{n} \\ \sum_{(i,j)\in E} w_{ij} z_{ij} = C \quad ,C\ is\ fixed \\ w_{ij} \ge 0 \ \ \forall (i,j) \in E \end{array} \tag{37}$$

Note that $U_\alpha J = 0$, consequently $Tr(U_\alpha L^+) = Tr(U_\alpha \Gamma^{-1})$.

Lemma 2. *The condition of optimality for optimization problem (36) can be written as:*

$$\min_{(i,j)\in E} \frac{C}{z_{ij}} \frac{\partial \tau_\alpha}{\partial w_{ij}} + \tau_\alpha \ge 0$$

Moreover:

$$w_{ij}^* \left(C \frac{\partial \tau_\alpha}{\partial w_{ij}} + z_{ij} \tau_\alpha\right) = 0 \ \ \forall (i,j) \in E \tag{38}$$

where w_{ij}^* denotes the optimal weight for link (i,j).

Proof. See Appendix 10.

Lemma 3. *The dual of the optimization problem (37) is as follows:*

$$\begin{array}{ll} maximize & \frac{1}{C} Tr^2(U_\alpha X) \\ subject\ \ to & \frac{1}{\sqrt{z_{ij}}} \left\| F_\alpha X u_{ij} \right\| \le 1 \ \ \forall (i,j) \in E \\ & X \overrightarrow{1} = 0 \\ & X \ge 0 \end{array}$$

where $X \ge 0$ means that X is a positive semi-definite matrix. More precisely $X = \frac{1}{\sqrt{\lambda}} L^+$ where L^+ is the Moore-Penrose inverse of Laplacian matrix, and $\lambda = \max_{(i,j)\in E} \frac{1}{z_{ij}} \left\| F_\alpha L^+ u_{ij} \right\|^2$.

Proof. See Appendix 11.

We are ready to to give an upper bound for the optimality gap in the optimization problem. The following theorem summarizes the result.

Theorem 2. *Consider the following optimization problem:*

$$\begin{array}{c} Minimize \quad \tau_\alpha \\ Subject\ to \quad \sum_{(i,j)\in E} z_{ij} w_{ij} = C \quad ,C\ is\ fixed \\ w_{ij} \geq 0\ \forall (i,j) \in E \end{array}$$

For any sub-optimal solution of the convex optimization problem, the deviation from optimal solution (optimality gap) has the upper bound of $\frac{\tau_\alpha}{C \min_{(i,j)\in E} \frac{1}{z_{ij}} \frac{\partial \tau_\alpha}{\partial w_{ij}}} (C \min_{(i,j)\in E} \frac{1}{z_{ij}} \frac{\partial \tau_\alpha}{\partial w_{ij}} + \tau_\alpha)$.

Proof. We denote the duality gap (the difference between the objective function of the dual and primal optimization problem) by d_{gap}. Using Lemma (3), we have:

$$\begin{aligned} d_{gap} &= Tr(U_\alpha L^+) - \frac{1}{C} Tr^2(U_\alpha X) \\ &= Tr(U_\alpha L^+) - \frac{1}{C} \frac{1}{\max_{(i,j)\in E} \frac{1}{z_{ij}} \left\| F_\alpha L^+ u_{ij} \right\|^2} Tr^2(U_\alpha L^+) \\ &= Tr(U_\alpha L^+)(1 + \frac{1}{C} \frac{Tr(U_\alpha L^+)}{\min_{(i,j)\in E} - \frac{1}{z_{ij}} \left\| F_\alpha L^+ u_{ij} \right\|^2}) \end{aligned} \tag{39}$$

Now it is enough to simplify equation (39) using lemma 1.

$$\begin{aligned} d_{gap} &= \tau_\alpha (1 + \frac{1}{C} \frac{\tau_\alpha}{\min_{(i,j)\in E} \frac{1}{z_{ij}} \frac{\partial \tau_\alpha}{\partial w_{ij}}}) \\ &= \frac{\tau_\alpha}{C \min_{(i,j)\in E} \frac{1}{z_{ij}} \frac{\partial \tau_\alpha}{\partial w_{ij}}} (C \min_{(i,j)\in E} \frac{1}{z_{ij}} \frac{\partial \tau_\alpha}{\partial w_{ij}} + \tau_\alpha) \end{aligned} \tag{40}$$

According to the duality theorem, this completes the proof of theorem 2.

Theorem 2 is in fact an extension for the results of [13] in which the total resistance distance is considered as the main metric of interest. Those results can be derived as special cases of lemma 3 and theorem 2.

5.1 *Network Planning Using An Interior Point Algorithm*

Optimization problem (36) can be solved with various methods developed for convex optimization problems. In this paper we use a modified version of interior-point method which is developed in [13] based on the duality gap obtained in theorem 2. In this method we use logarithmic barrier for the non-negativity constraints $w_{ij} \geq 0 \ \forall (i,j) \in E$:

$$\Phi = -\sum_{(i,j)\in E} \log w_{ij}$$

In the interior-point method we minimize $t\tau_\alpha + \Phi$, using Newton's method (t is a parameter), subject to $\sum_{(i,j)\in E} z_{ij} w_{ij} = C$. The sub-optimality in this case would be at most $\frac{m}{t}$, where m is the number of links. On the other hand, theorem 2 provides an upper bound for sub-optimality:

$$\hat{t} = \frac{\tau_\alpha}{C \min_{(i,j)\in E} \frac{1}{z_{ij}} \frac{\partial \tau_\alpha}{\partial w_{ij}}} (C \min_{(i,j)\in E} \frac{1}{z_{ij}} \frac{\partial \tau_\alpha}{\partial w_{ij}} + \tau_\alpha)$$

We can use this bound to update parameter t in each step of our interior-point method, by taking $t = \frac{m}{\hat{t}}$. In each step, for a relative precision ϵ, the Newton's method finds change of $\Delta\overrightarrow{w}$ for the vector of all weights (denoted by $\overrightarrow{w}$) by solving:

$$(t\nabla^2\tau_\alpha + \nabla^2\Phi)\Delta\overrightarrow{w} = -t\nabla\tau_\alpha + \nabla\Phi$$

The next task is to find the Newton step length s by backtracking line search [19], and then updating the weight vector by $\overrightarrow{w} = \overrightarrow{w} + s\delta\overrightarrow{w}$. The algorithm exits when we have $\tau_\alpha - \tau_\alpha^{opt} \leq \hat{t} \leq \epsilon\tau_\alpha$. We can choose ϵ small enough to have a desired precision.

We note that, in order to use this recursive method, we need to have the gradient vector $\nabla\tau_\alpha$ and Hessian matrix $\nabla^2\tau_\alpha$. Lemma 1 provides the entries of the gradient vector and using the lemma, it is easy to see that the entries of Hessian matrix can be found from the following equation:

$$\frac{\partial^2\tau_\alpha}{\partial w_{pq}\partial w_{ij}} = 2u_{ij}^t L^+ u_{pq} u_{pq}^t L^+ U_\alpha L^+ u_{ij} \ \forall (i,j),(p,q) \in E$$

In matrix form this can be shown as:

$$\nabla^2\tau_\alpha = (B^t L^+ B) \ o \ (B^t L^+ U_\alpha L^+ B)$$

where B is the incidence matrix of the graph, and o denotes Hadamard (componentwise) product.

6 Applications

This section considers applications of weighted network criticality. First, we develop a semi-definite programming formulation for general weight planning to minimize τ_α. We then develop optimization problems to jointly optimize the resources (weights) and routes (link flows) in a communication network. We will also discuss a joint optimization of demands, flows, and resources in order to maximize a concave utility function of demands, while keeping network criticality below a certain threshold. Finally, we discuss robust optimization of network weights in order to protect the network against k link failures.

6.1 Network Planning Using Semi-Definite Programming

Optimization problem (37) (and problem (36)) provides an approach for robust network design via optimal allocation of network link weights to minimize (traffic-aware) network criticality. Optimization problem (37) can be converted to a semi-definite program (SDP) as stated in the following.

$$\begin{aligned} & \text{Minimize} \quad Tr(Y) \\ \text{Subject to} \quad & \textstyle\sum_{(i,j)\in E} w_{ij} z_{ij} \leq C \quad ,C \text{ is fixed} \\ & w_{ij} \geq 0 \ \ \forall (i,j) \in E \\ & \begin{pmatrix} L + \frac{J}{n} & U_\alpha^{\frac{1}{2}} \\ U_\alpha^{\frac{1}{2}} & Y \end{pmatrix} \geq 0 \end{aligned} \tag{41}$$

where $\geq$ means positive semi-definite.

Capacity allocation is an important case in which we define the weight of a link to be its capacity. In this case optimization of network criticality results in capacity planning. A quite general case of capacity planning problem is when a routing mechanism is already designed for the network and it is known that each link of the network is supposed to carry a known amount of traffic demands. Suppose we know that our routing scheme is shortest path and traffic matrix $[\gamma_{ij}]$ is given for the network. Assume that we have found the values of flows for each link to meet the given traffic matrix via shortest-path routing and the result is flow λ_{ij} for link (i,j). Then by applying the change of variable $w_{ij} = c_{ij} + \lambda_{ij}$ to the optimization problem (41), we will have the following convex optimization problem for optimal capacity allocation.

$$\begin{array}{c} \textit{Minimize} \quad Tr(Y) \\ \textit{Subject to} \quad \sum_{(i,j)\in E} c_{ij} z_{ij} = C' \quad ,C' \textit{ is fixed} \\ c_{ij} \geq 0 \;\; \forall (i,j) \in E \\ \begin{pmatrix} L + \frac{J}{n} & U_{\alpha}^{\frac{1}{2}} \\ U_{\alpha}^{\frac{1}{2}} & Y \end{pmatrix} \geq 0 \end{array} \tag{42}$$

where $C' = C - \sum_{(i,j)\in E} z_{ij}\lambda_{ij}$. This has the same form of optimization problem (41), and both are SDP representation of optimization problem (36) (with $w_{ij} \to c_{ij}$ and $C \to C'$); therefore, all the results developed for optimization problem (36) are applicable for the capacity assignment problem.

Solving this SDP problem is much faster and can be done with a variety of existing packages (for example see [17, 18]) to solve SDP problems.

6.2 Joint Optimization of Routes and Resources

Optimization problem (36) can be extended to a more general case where the weights of the network links (resources) and the link flows (routes) are unknown. In this section we focus on capacity as the resource and assume that the link weight equals the capacity (or available capacity depending on the context) of the link. In order to account for flows, it is enough to add the equations for the conservation of flow at each node (and for each entry of the traffic matrix) to the constraints of the problem. For a specific node k and entry γ_{sd} of the traffic matrix, the conservation of flow can be stated as:

$$\sum_{i\in A(k)} f_{ik}^{(sd)} - \sum_{j\in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d)$$

where $A(k)$ denotes the set of neighbor nodes of node k, $f_{ik}^{(sd)}$ denotes the flow of link (i,k) for traffic entry between source s and destination d, and $\delta(x)$ is Kronecker delta function. Furthermore, the flow of each link should not exceed the capacity of the link, therefore we will need the following constraints for each link of the network:

$$\begin{array}{c} f_{ij} = \sum_{sd} f_{ij}^{(sd)} \;\; \forall (i,j) \in E \\ f_{ij} \leq w_{ij} \;\; \forall (i,j) \in E \\ f_{ij} \geq 0 \;\; \forall (i,j) \in E \end{array}$$

Now we can write the optimization for robust joint flow assignment and resource allocation as follows.

$$\begin{array}{ll} & Minimize \quad \tau_\alpha \\ Subject\ to & \sum_{(i,j)\in E} w_{ij} z_{ij} \le C \quad ,C\ is\ fixed \\ & w_{ij} \ge 0 \ \ \forall (i,j)\in E \\ & \sum_{i\in A(k)} f_{ik}^{(sd)} - \sum_{j\in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d) \ \ \forall k\in N,\ \ \forall \gamma_{sd} \\ & f_{ij} = \sum_{sd} f_{ij}^{(sd)} \ \ \forall (i,j)\in E \\ & f_{ij} \le w_{ij} \ \ \forall (i,j)\in E \\ & f_{ij} \ge 0 \ \ \forall (i,j)\in E \end{array} \tag{43}$$

One efficient method to solve optimization problem (43) is *dual decomposition* [19, 20] which can separate the network flow problem from resource allocation. We form a dual problem for (43) by introducing Lagrangian multiplier matrix Λ for constraint set $f_{ij} \le w_{ij} \ \ \forall (i,j) \in E$. The resulting partial Lagrangian is $L = \tau_\alpha - Tr(\Lambda(F-W))$ where $F = [f_{ij}]$ and $W = [w_{ij}]$ are the matrices of link flows and link weights respectively. The dual objective function is then:

$$\begin{aligned} d(\Lambda) = & \inf_W \left(\tau_\alpha + Tr(\Lambda W)\right)\Big|_{\sum_{(i,j)\in E} w_{ij} z_{ij} = C,\ \ w_{ij}\ge 0\ \ \forall (i,j)\in E} \\ & + \inf_F \left(-Tr(\Lambda F)\right)\Big|_{\sum_{i\in A(k)} f_{ik}^{(sd)} - \sum_{j\in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d)\ \ \forall k\in N,\ \ f_{ij} = \sum_{sd} f_{ij}^{(sd)}\ \ \forall (i,j)\in E} \end{aligned}$$

Note that two infimum functions in the dual objective are working on separate variables (first infimum on weights, and second one on flows). The dual function can be seen as the sum of the following two functions:

$$d_W(\Lambda) = \inf_W \left(\tau_\alpha + Tr(\Lambda W)\right) \left\| \begin{array}{l} \sum_{(i,j)\in E} w_{ij} z_{ij} = C,\ C\ is\ fixed \\ \quad w_{ij} \ge 0 \qquad \forall (i,j)\in E \end{array} \right. \tag{44}$$

$$d_F(\Lambda) = \sup_F \ Tr(\Lambda F) \left\| \begin{array}{c} \sum_{i\in A(k)} f_{ik}^{(sd)} - \sum_{j\in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d) \\ \forall k\in N, \\ f_{ij} = \sum_{sd} f_{ij}^{(sd)} \ \ \forall (i,j)\in E \end{array} \right. \tag{45}$$

The dual problem associated with the primal optimization problem (43) is:

$$\begin{array}{c} Maximize \quad d(\Lambda) = d_W(\Lambda) + d_F(\Lambda) \\ Subject\ to \quad \Lambda \ge 0 \end{array} \tag{46}$$

where $\ge$ is a component-wise operator (i.e. $\Lambda = [\lambda_{ij}] \ge 0$ means $\lambda_{ij} \ge 0 \ \ \forall i,j$). Optimization problem (46) is convex because the dual function is always convex [19]. We assume that the Slater's condition [19] is satisfied in optimization problem (43) in order to guarantee that strong duality holds, i.e. the solution of optimization problem (43) and its dual (46) are equal. Note that in general this is not true when the primal objective function is not strictly convex. By assuming Slater's condition we can solve optimization problem (46) instead of the primal one.

To find the solution of dual optimization problem (46) we study dual functions $d_W(\Lambda)$ and $d_F(\Lambda)$ separately, and then we add them together. Considering equation (44), $d_W(\Lambda)$ is the solution of the following optimization problem:

$$\begin{array}{c} Minimize \quad \tau_\alpha + Tr(\Lambda W) \\ Subject\ to \quad \sum_{(i,j)\in E} w_{ij} z_{ij} \leq C \quad ,C\ is\ fixed \\ w_{ij} \geq 0 \ \ \forall (i,j) \in E \end{array} \tag{47}$$

Optimization problem (47) can be viewed as the network planning subproblem which will assign the optimal values of weights. We refer to this as the **resource (weight) allocation subproblem** associated with problems (43) and (46). Similarly, equation (45) implies that $d_F(\Lambda)$ can be found by solving optimization problem (48) as follows:

$$\begin{array}{c} Maximize \quad Tr(\Lambda F) \\ Subject\ to\ f_{ij} = \sum_{sd} f_{ij}^{(sd)} \ \ \forall (i,j) \in E \\ f_{ij} \geq 0 \ \ \forall (i,j) \in E \\ \sum_{i\in A(k)} f_{ik}^{(sd)} - \sum_{j\in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d) \ \ \forall k \in N, \ \ \forall \gamma_{sd} \end{array} \tag{48}$$

Problem (48) determines the optimum routing given the optimal values of weights and we refer to it as **flow assignment subproblem** associated with problems (43) and (46).

Entry λ_{ij} of matrix Λ can be interpreted as the price of allocating unit weight to link (i, j) in the weight matrix. Therefore, the resource allocation subproblem (47) tries to minimize weighted network criticality incremented by total price of deploying a complete weight matrix, and the flow assignment subproblem (48) will try to maximally utilize the allocated weights to run the network flows.

A detailed discussion of numerical methods to solve optimization problem (46) is beyond the scope of our work in this chapter; however, we indicate how we can iteratively solve optimization problem (46) in the following manner. We start with an initial value for Λ, and then we find the optimal weight and flow matrices using resource allocation and flow assignment subproblems. Then we find an update for price matrix Λ using subgradient method. Using the updated value of Λ we reoptimize the weight and flow matrices by solving optimization problems (47) and (48). This process continues until we arrive at a stable price matrix. This procedure allows us to view the resource allocation subproblem as an optimization working on physical layer, while the flow assignment subproblem operates independently on a network layer and that provides optimal routing scheme for the problem. The connection between these two layers is through price matrix Λ as illustrated in Fig. 1.

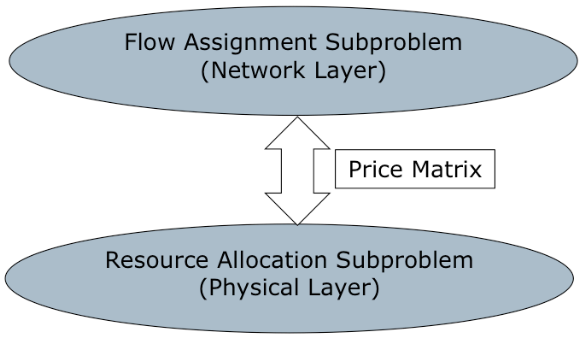

Fig. 1 Layered View of Resource Allocation and Flow Assignment Subproblems

Solution of optimization problem (43) is a symmetric set of link weights representing total capacity of links and associated link flows. We can modify the optimization problem to provide an asymmetric capacity assignment (i.e. we change the undirected graph model to a directed one). We interpret link weights as the **available capacity** and reformulate the optimization problem accordingly. Let c_{ij} and w_{ij} denote the capacity and weight (available capacity) of link (i, j) respectively. Optimization problem ((43) can be converted to the following problem:

$$
\begin{aligned}
& \textit{Minimize} \quad \tau_\alpha \\
\textit{Subject to} \quad & \sum_{(i,j)\in E} c_{ij} z_{ij} \leq C \quad , C \textit{ is fixed} \\
& \sum_{i\in A(k)} f_{ik}^{(sd)} - \sum_{j\in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d) \quad \forall k \in N, \ \forall \gamma_{sd} \\
& f_{ij} = \sum_{sd} f_{ij}^{(sd)} \quad \forall (i, j) \in E \\
& f_{ij} = c_{ij} - w_{ij} \quad \forall (i, j) \in E \\
& f_{ij} \geq 0 \quad \forall (i, j) \in E \\
& w_{ij} \geq 0 \quad \forall (i, j) \in E
\end{aligned}
\tag{49}
$$

Note that optimization problem (49) provides optimal solutions of weights (available capacities), capacities, and flows for all the links. The weights will be symmetric; however, the link capacities (c_{ij}'s) need not be symmetric. This means that the optimization problem allocates capacities and flows in such a way that the final residual bandwidth or available capacities of link (i, j) and link (j, i) are equal (i.e. $w_{ij} = w_{ji}$), while the total capacity of link (i, j) is not necessarily equal to that of link (j, i) (i.e. $c_{ij} \neq c_{ji}$). The main difference between the solution of optimization problem (43) and (49) is in the way they minimize τ_α. In the former, the link capacity is designed such that τ_α is minimized before applying any flow to the network, while the latter determines flows and capacities in such a way that τ_α for the residual network is minimized.

6.3 Joint Optimization of Demands, Flows, and Resources

We consider one more extension for the optimization problem which tries to find the joint optimal assignment of external demands (traffic matrix) and link flows when the weight assignment (capacity) is known. As in the previous case, we assume that the link weights represent the available capacity. In this problem we optimize a concave utility function of traffic matrix (denoted by $\Psi(\Gamma)$) subject to the condition that the average network utilization $\bar{V}$ (see equation (31)) is less than a given maximum value. A well-known example of concave utility functions is the logarithmic function. In this work we are interested in a subset of all possible traffic matrices for which the constraint for network utilization $\bar{V}$ is convex; therefore, we add the set of inequalities (33) as constraints to the optimization problem. Clearly, the flow conservation and capacity constrains are also necessary. Summarizing all above, we can write the following optimization problem to find the optimal joint assignment of traffic demands and link flows.

$$\begin{gathered}
Maximize \quad \Psi(\Gamma) \\
Subject\ to\ f_{ij} = \sum_{sd} f_{ij}^{(sd)} \ \ \forall (i,j) \in E \\
f_{ij} = c_{ij} - w_{ij} \ \ \forall (i,j) \in E \\
f_{ij} \geq 0 \ \ \forall (i,j) \in E \\
w_{ij} \geq 0 \ \ \forall (i,j) \in E \\
\sum_{i \in A(k)} f_{ik}^{(sd)} - \sum_{j \in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d) \ \ \forall k \in N, \ \ \forall \gamma_{sd} \\
\sum_{sd} (\tfrac{\gamma_{sd}+\gamma_{ds}}{4} + \tfrac{\gamma_{*s}-\gamma_{s*}}{2n})\tau_{sd} \leq a, \quad a\ is\ fixed \\
\gamma_{sd} + \gamma_{ds} \geq \tfrac{1}{n}(\gamma_{s*} - \gamma_{*s} + \gamma_{d*} - \gamma_{*d}) \\
\gamma_{ij} \geq 0 \ \ \forall i,j \in N
\end{gathered} \tag{50}$$

where a is a known maximum acceptable value for average network utilization.

Finally, we can jointly optimize resources (weights), traffic demands, and routes. We assume that the goal is to maximize a concave function of traffic demands subject to a maximum value for network criticality. Moreover, we assume a maximum budget for the total weights (given a cost for deploying a unit of weight). Thus, considering the flow conservation and capacity constraints we will have the following convex optimization problem for joint optimization of weights, acceptable demands, and routes (flows).

$$\begin{aligned}
&Maximize \quad \Psi(\Gamma)\\
&Subject\ to \sum_{ij} z_{ij} c_{ij} \leq C\\
&\tau \leq b, \quad b\ is\ fixed\\
&\sum_{i\in A(k)} f_{ik}^{(sd)} - \sum_{j\in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d) \quad \forall k \in N, \quad \forall \gamma_{sd}\\
&f_{ij} = \sum_{sd} f_{ij}^{(sd)} \quad \forall (i,j) \in E\\
&f_{ij} = c_{ij} - w_{ij} \quad \forall (i,j) \in E\\
&f_{ij} \geq 0 \quad \forall (i,j) \in E\\
&w_{ij} \geq 0 \quad \forall (i,j) \in E\\
&\gamma_{ij} \geq 0 \quad \forall i,j \in N
\end{aligned} \tag{51}$$

The solution of optimization problem (51) plans network weights, gives the optimal set of demands $\Gamma = [\gamma_{ij}]$ (maximizing concave utility function Ψ) which can be accommodated by the network, and provides link flows subject to the condition that the network criticality does not exceed a known value b.

It is possible to find a layered approach for optimization problem (51). The steps are similar to what we did for joint optimization of flows and resources (see equations (46), (47), (48)). Here we construct a dual problem for (51) by introducing Lagrangian multiplier matrix Λ for constraint set $f_{ij} = c_{ij} - w_{ij} \ \ \forall (i,j) \in E$. The resulting partial Lagrangian is $L = \Psi(\Gamma) - Tr(\Lambda(F - C + W))$ where $F = [f_{ij}]$, $C = [c_{ij}]$ and $W = [w_{ij}]$ denote the matrices of link flows, link capacities, and link weights respectively. The dual objective function is then $d(\Lambda) = d_{F,\Gamma}(\Lambda) + d_W(\Lambda) + d_C(\Lambda)$, where:

$$d_W(\Lambda) = \sup_W \ -Tr(\Lambda W) \left\| \begin{array}{l} w_{ij} \geq 0 \ \forall (i,j) \in E \\ \tau \leq b \end{array} \right.$$

$$d_C(\Lambda) = \sup_C \ Tr(\Lambda C) \left\| \begin{array}{l} \sum_{(i,j)\in E} c_{ij} z_{ij} = C,\ C\ is\ fixed \\ c_{ij} \geq 0 \qquad \forall (i,j) \in E \end{array} \right.$$

$$d_{F,\Gamma}(\Lambda) = \sup_{F,\Gamma} \ (\Psi(\Gamma) - Tr(\Lambda F)) \left\| \begin{array}{c} \sum_{i\in A(k)} f_{ik}^{(sd)} - \sum_{j\in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d) \\ \forall k \in N, \\ f_{ij} = \sum_{sd} f_{ij}^{(sd)} \quad \forall (i,j) \in E \\ f_{ij} \geq 0 \quad \forall (i,j) \in E \\ \gamma_{ij} \geq 0 \quad \forall i,j \in N \end{array} \right.$$

The dual problem associated with the primal optimization problem (51) is:

$$\begin{aligned} Minimize \quad & d(\Lambda) = d_W(\Lambda) + d_C(\Lambda) + d_{F,\Gamma}(\Lambda) \\ Subject\ to \quad & \Lambda \geq 0 \end{aligned} \tag{52}$$

Since in general the primal objective function is not strictly concave, we assume that Slater's condition is satisfied in the optimization problem (51). This guarantees that the strong duality holds for this problem and that the solution of dual optimization problem (52) is equal to the solution of primal problem (51).

As we discussed before, in order to solve dual problem (52) we can evaluate $d_W(\Lambda)$, $d_C(\Lambda)$, and $d_{F,\Gamma}(\Lambda)$. We note that $d_{F,\Gamma}(\Lambda)$ is the solution of the following optimization problem:

$$\begin{aligned} & Maximize \quad \Psi(\Gamma) - Tr(\Lambda F) \\ & Subject\ to\ f_{ij} \geq 0 \ \ \forall (i,j) \in E \\ & \gamma_{ij} \geq 0 \ \ \forall i,j \in N \\ & \textstyle\sum_{i \in A(k)} f_{ik}^{(sd)} - \sum_{j \in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d) \ \ \forall k \in N, \ \ \forall \gamma_{sd} \end{aligned} \tag{53}$$

Optimization problem (53) can be further divided into two optimization problems separating the effect of demands and flows by introducing another set of Lagrange multipliers. Similarly, $d_W(\Lambda)$ is the solution of optimization problem (54).

$$\begin{aligned} & Minimize \quad Tr(\Lambda W) \\ & Subject\ to \quad \tau \leq b, \quad b\ is\ fixed \\ & w_{ij} \geq 0 \ \ \forall (i,j) \in E \end{aligned} \tag{54}$$

The solution of optimization problem (55) provides us with $d_c(\Lambda)$.

$$\begin{aligned} & Maximize \quad Tr(\Lambda C) \\ & Subject\ to\ \textstyle\sum_{(i,j)\in E} c_{ij} z_{ij} = C,\ C\ is\ fixed \\ & c_{ij} \geq 0 \ \ \forall (i,j) \in E \end{aligned} \tag{55}$$

If we interpret Λ as the price matrix, then optimization problem (53) tries to maximize a utility function discounted by total price of assigning link flows. Optimization problem (54) finds the minimum required weights (available capacities) in order to guarantee that the network criticality of the residual network is not more than a pre-specified value b. Optimization problem (55) finds the best capacity assignment under price model mandated by matrix Λ.

In order to find the optimum solution of dual optimization problem (52), we start from an initial guess for matrix Λ. Solution of optimization problems (54) and (55) provide optimum values of weights and capacities at this stage, then the difference between capacity and weight of each link is equal to the flow of the link. Using optimization problem (53) we can then find best possible demand set. Then a new approximation for Lagrangian matrix Λ can be obtained using subgradient method and the iteration continues until we arrive at the stable matrix Λ.

6.4 Robust Network Design: Protecting Against Multiple Link Failures

The solution of optimization problem (36) provides a robust network design method via optimal weight assignment; however, it does not necessarily protect the network against multiple link failures. Link (node) failures can be the result of unplanned random failures or due to targeted attacks. In this section we extend optimization problem (36) to account for multiple link failures.

Let $D \in \{0,1\}^m$ be a binary matrix representing the location of link failures, i.e. $d_{ij} = 0$ for failed link (i,j) and $d_{ij} = 1$ for operational ones. We replace the weight matrix W with DoW (o is the Hadamard operator) and redo the optimization of WNC. Now if we want that the network to be robust to up to k link failures (we refer to such a network as **k-robust** network), we need to minimize the following objective function.

$$\max_{\sum_{i,j} d_{ij}=n-k} Tr(U_\alpha L^+(DoW))$$

. where n is the total number of nodes in the network. Note that the above function is convex because it is a point-wise maximum of a set of convex functions. By minimizing this function we find a k-robust topology (along with its optimal link weights). Therefore, a general optimization problem to provide a k-robust network can be written as:

$$\begin{array}{ll} & Minimize \quad \max_{\sum_{i,j} d_{ij}=n-k} Tr(U_\alpha L^+(DoW)) \\ Subject\ to & \sum_{(i,j)\in E} w_{ij} z_{ij} \le C \quad ,C\ is\ fixed \\ & w_{ij} \ge 0 \ \ \forall (i,j) \in E \end{array} \tag{56}$$

Interpreting weight as capacity, equation (56) can be extended to provide simultaneous solution for k-robust flow assignment and weight allocation, just by adding the flow conservation equations and link capacity constraints:

$$\begin{aligned}
&\textit{Minimize} \quad \max_{\sum_{i,j} d_{ij}=n-k} Tr(U_\alpha L^+(DoW)) \\
&\textit{Subject to} \quad \sum_{(i,j)\in E} w_{ij}z_{ij} \leq C \quad , C \textit{ is fixed} \\
&w_{ij} \geq 0 \;\; \forall (i,j) \in E \\
&\sum_{i\in A(k)} f_{ik}^{(sd)} - \sum_{j\in A(k)} f_{kj}^{(sd)} = \gamma_{sd}\delta(k-s) - \gamma_{sd}\delta(k-d) \;\; \forall k \in N, \;\; \forall \gamma_{sd} \\
&f_{ij} = \sum_{sd} f_{ij}^{(sd)} \;\; \forall (i,j) \in E \\
&f_{ij} \geq 0 \;\; \forall (i,j) \in E \\
&f_{ij} \leq w_{ij} \;\; \forall (i,j) \in E
\end{aligned} \tag{57}$$

6.5 *Case of Directed Networks*

Most of the discussion in this section was based on the assumption that our network is modeled with an undirected graph. We considered the case of having asymmetric capacities and link flows by interpreting the link weight as available capacity; however, even in this case the residual network has symmetric link weights. The main reason for this assumption is that the concept of resistance distance is only available on reversible Markov chains. However, there is another nice interpretation for network criticality which provides guidlines to extend the notion of network criticality to directed graphs.

Suppose that there are costs associated with traversing links along a path and consider the effect of network criticality on average cost incurred by a message during its walk from source s to destination d. It is shown in [10] that the average incurred cost is the product of network criticality and the total cost of all link weights. Therefore, if we set a fixed maximum budget for the cost of assigning weights to links, then the average travel cost is minimized when network criticality is minimized.

While the analogy between resistance distance and random-walks does not hold in directed graphs, we can still find the hitting times and commute times for a directed graph, and the interpretation of average travel cost (or equivalently average commute time) still holds. In fact we have shown that the average travel time in a directed graph can be found using the exact same analytical machinery [21], i.e. the trace of generalized inverse of the combinatorial Laplacian matrix of a directed graph (L) which is defined as $L = \Phi(I-P)$, where Φ is a diagonal matrix with main diagonal entry i equal to the i^{th} entry of stationary probability vector corresponding to transition probability matrix P. We propose to use the average travel time as the objective of our optimization problem in the case of directed graphs. In this case the optimization problem is not necessarily convex (with regards to the link weights).

7 Case Study

In this section we consider some simple applications of the optimization problems discussed in section 6. In particular we are interested in virtual network design to assign robust resources and flows to different customers of the network. Furthermore, we will apply the proposed optimization problems in designing robust power grids.

7.1 Virtual Network Assignment

In our first case study, we investigate the problem of assigning virtual networks to different customers of a communication network in order to meet their contracted service levels. In this study we are interested in end-to-end bandwidth requirements as the main service.

For illustrative purposes we consider a simple topology which is shown in Fig. 2. This network is referred to as trap topology in networking literature. The trap topology is well-known in the context of survivable routing. Suppose there is a demand from node 1 for node 6. The min-hop path from node 1 to 6 is the straight line $1 \to 3 \to 4 \to 6$. It appears that this path is the best choice to run the demand, but in survivable routing we need to assign backup paths to each primary route. In trap network there is no link-disjoint backup path for $1 \to 3 \to 4 \to 6$. Therefore it would be desirable to choose path $1 \to 2 \to 4 \to 6$ ($1 \to 3 \to 5 \to 6$) as the primary route for demands from 1 to 6. Then the link-disjoint backup path will be $1 \to 3 \to 5 \to 6$ ($1 \to 2 \to 4 \to 6$).

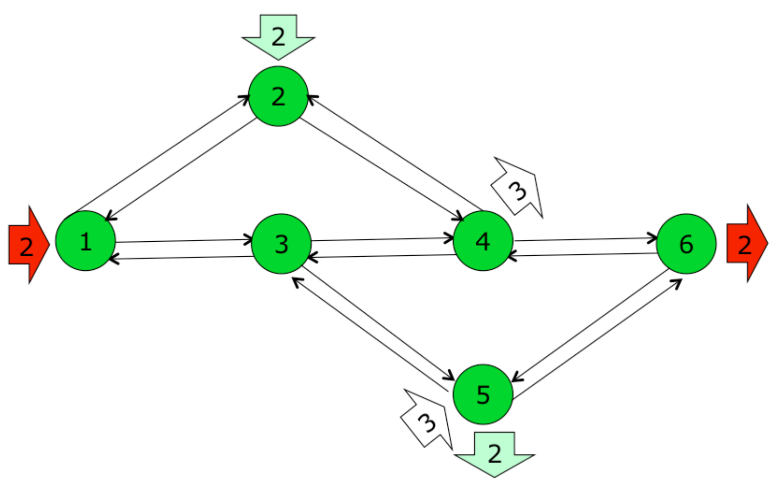

Fig. 2 Trap Network with Given Traffic Matrix

We assume each customer in the network is defined by a set of demands identified by their source, destination and required bandwidth, i.e. each demand σ_i is defined by a triple (s_i, d_i, b_i), where s_i, d_i, b_i denote source, destination and required bandwidth of demand σ_i. In this study, we assume two customers (CS1 and CS2) exist on trap network with the following demands:

$$\begin{aligned} \sigma_1 &= (1,6,2) \\ \sigma_2 &= (2,5,2) \\ \sigma_3 &= (5,4,3) \\ CS1 &= \{\sigma_1, \sigma_2\} \\ CS2 &= \{\sigma_3\} \end{aligned}$$

From the above description, the requirements of customers can be summarized in separate virtual networks assigned to each customer as shown in Fig. 3.

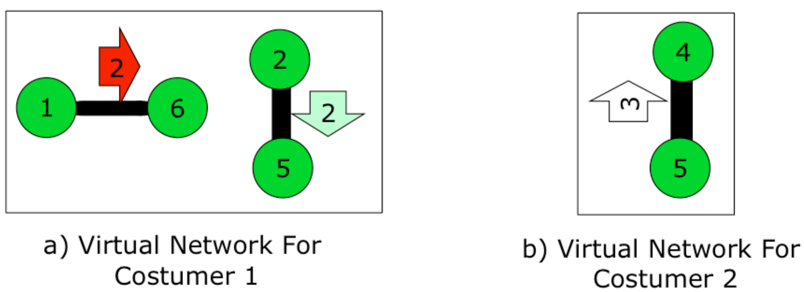

Fig. 3 Desired Virtual Networks For CS1 and CS2

Our goal is to determine the optimal robust allocation of capacities and flows for each customer so as to meet the requirements of all the customers. We will use optimization problem (49) to find optimal capacity allocation and flow assignment simultaneously. Optimization problem (49) permits us to find asymmetric capacity assignment for the links; however, due to the nature of network criticality, the residual capacity of links after flow assignment is symmetric.

We suppose that the cost of deploying all the links (z_{ij}'s) are equal (and assumed to be 1) and the total budget for capacity is 26 (it means that $\sum_{ij} z_{ij} c_{ij} = \sum_{ij} c_{ij} = 26$). Solution of optimization problem (49) will result in the capacity allocation of Fig. 4.

Now we need to find the exact graph embedding for virtual networks of CS1 and CS2. For CS1, based on the solution of optimization problem (49), the optimal flow assignment for the physical substrate (trap network)

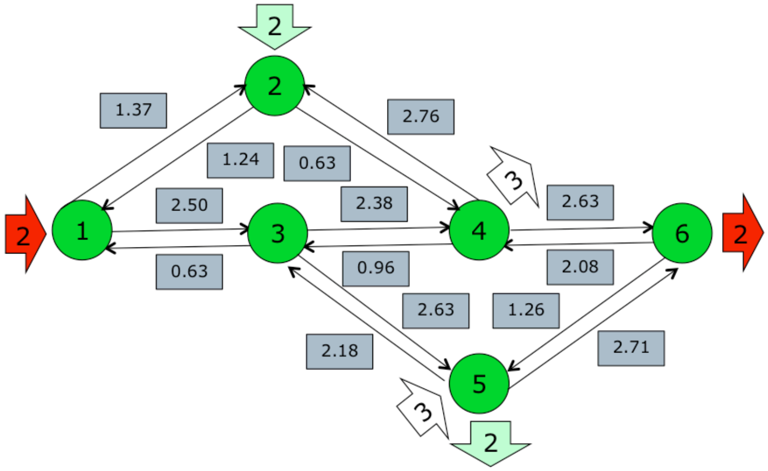

Fig. 4 Optimal Capacity Assignment For Trap Network

is shown in Fig. 5. From Fig. 5 we see that all the nodes of the substrate network involve in providing connection (service) for CS1, however, 4 links ((3,1), (4,2), (5,3), (6,4)) do not contribute in building the virtual network. As a matter of fact the virtual network can be viewed as a subset of link/node resources dedicated to a customer.

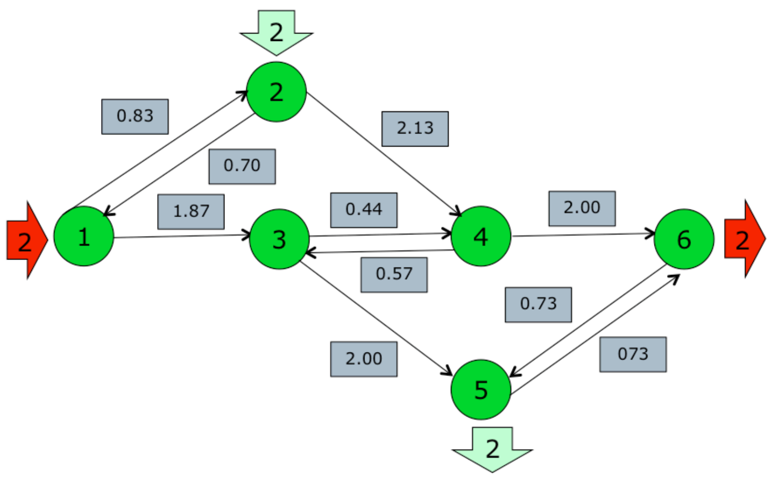

Fig. 5 Optimal Resource Assignment for Virtual Network 1 (Customer 1) On Trap Physical Topology

The optimal flow assignment for CS2 is shown in Fig. 6. It can be seen that for CS2 nodes 1 and 2 do not contribute in providing service and among all the links only 4 links involve in guaranteeing service for CS2.

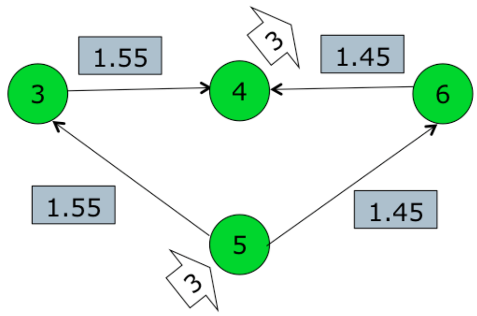

Fig. 6 Optimal Resource Assignment for Virtual Network 2 (Customer 2) On Trap Physical Topology

Adding the flows of two customers, we can have the total flow of each link in trap network as depicted in Fig. 7.

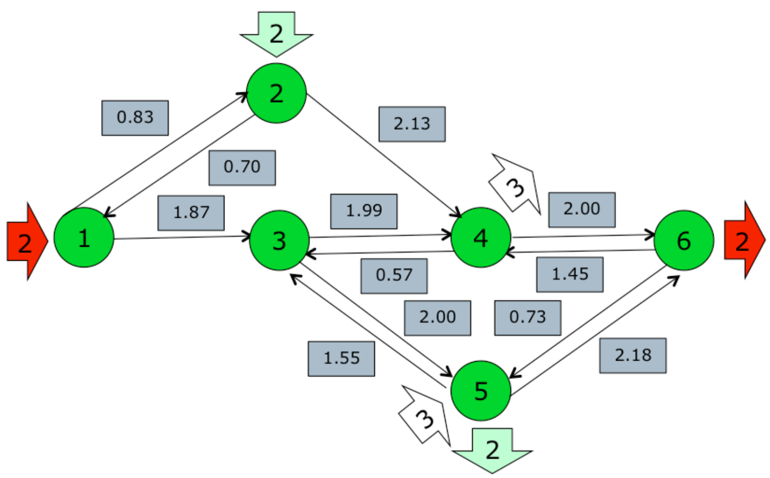

Fig. 7 Total Link Flow Assignment On Trap Network

7.2 *Design of Robust Power Grids*

The concept of traffic-aware network criticality has a nice application in smart power grids. Nowadays the idea of using renewable energy sources has gained considerable attention. Many places with renewable energy (such as places with high wind) are not within the reach of existing power grid network and it is required to extend the existing power grid to the places with renewable energy. Thus, we need to know how to design a robust power network as an extension to the existing one. In addition to the robustness, a power grid should be sparse enough, to avoid unnecessary power lines. This problem is recently investigated and a method of sparsification is also developed [22]. Here we use the method of [22] to generate and optimize our power network and we use an alternative sparsification method to prune the network.

It can be shown that in a DC-model approximation of a power grid, the average power dissipation of the grid is proportional to $Tr(AL^{+})$, where $A = <\overrightarrow{a}\overrightarrow{a}^t>$ and $\overrightarrow{a}$ is the vector of link electrical currents ($< . >$ denotes time average) [22]. Clearly, the power dissipation has the general form of weighted network criticality (or equivalently traffic-aware network criticality if we interpret the power as network flow, and weights as line conductances); therefore, minimization of power dissipation in power grids results in minimization of WNC. We address the optimization of a power grid network with multiple random independent loads supplied by a generator. For consumer nodes, we specify mean load $\bar{a}_i < 0$ and the variance σ_i^2. At transmission (relay) nodes, the average and variance of the load are zero. At the generator we must have $a_0 = -\sum_{i\neq 0} a_i$. Therefore, matrix $A = <\overrightarrow{a}\overrightarrow{a}^t>$ can be written as:

$$\begin{pmatrix} (\sum_{i\neq 0}\bar{a}_i)^2 + \sum_{i\neq 0}\sigma_i^2 & -\overrightarrow{1}^t(\overrightarrow{a}\overrightarrow{a}^t + \Sigma) \\ -(\overrightarrow{a}\overrightarrow{a}^t + \Sigma)\overrightarrow{1} & \overrightarrow{a}\overrightarrow{a}^t + \Sigma \end{pmatrix}$$

We let $\bar{a}_i = -1$ and $\sigma_i^2 = \frac{1}{4}$ for consumer nodes in our tests in this section. We consider an $n - by - n$ grid (n is an odd number) and we let the generator node be the middle node of the grid and consumer nodes on the border nodes (Fig. 8-(a)), or a middle node in one of the border lines of the grid and consumers on the parallel border (Fig. 8-(b)).

First, we optimize the grid for power dissipation (i.e. we minimize weighted network criticality $Tr(AL^{+})$. The optimization problem is essentially the same as problem (41) with appropriate values of α_{ij}, so that $U_\alpha = A$. We solved problem (41) for the given values of $U_\alpha = A$. The optimal networks are shown in Fig. 9-(a), (b), where the thickness of the lines represent the link weights or line conductances (thicker line has higher conductance). We discuss Fig. 9-(b), since it is more vulnerable and needs attention. Fig. 9-(b) shows that by optimizing traffic-aware network criticality, we prune the original grid; however, this network does not provide protection against

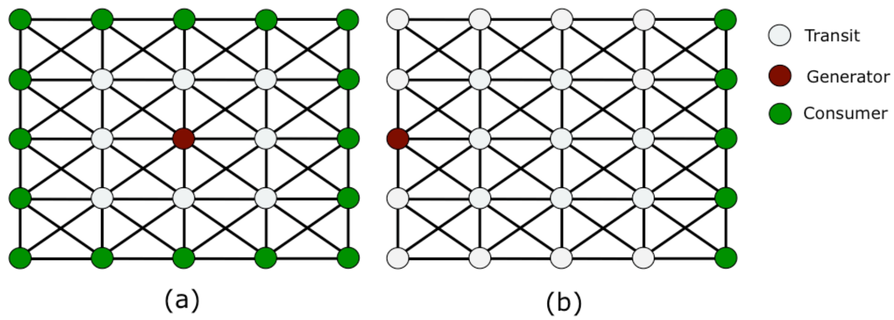

Fig. 8 Power grids with one Generator Node

possible link/node failures. We can use optimization problem (56) to find a k-robust grid power topology. Fig. 10-(a) shows an example of a 1-robust topology.

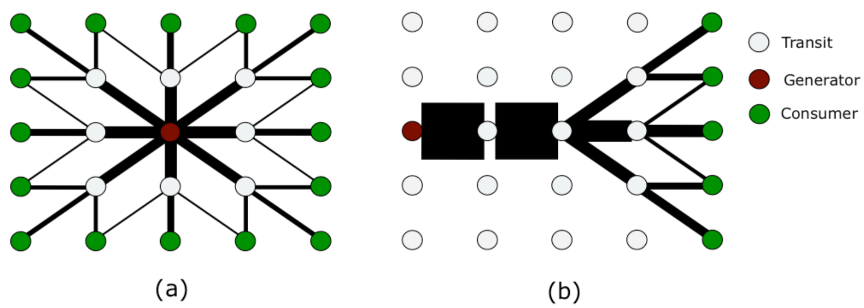

Fig. 9 Optimal Grid Topologies - Thickness of the lines represent the conductances

We provide one more extension that is particularly useful for the case of power grids in which the network should be sparse enough while preserving robustness. We would like to sparsify the robust topology of 10-(a). Fortunately, there is an elegant study on the context of sparsification using resistance distance. In [23] the problem of finding an sparse version of a network so that the total network criticality of the original graph and its sparse version are close enough. The authors have proposed an algorithm to find such sparse networks. The algorithm works as follows. Suppose H is the sparse version of graph G. Choose a random line (i, j) of the network G with probability p_{ij} proportional to $w_{ij}\tau_{ij}$, where τ_{ij} is the point-to-point network criticality or the resistance distance seen between nodes i and j. Add (i, j) to H with weight $\frac{w_{ij}}{qp_{ij}}$, where q is the number of independent samples (we should sum up weights if a line is chosen more than once. We used this algorithm to simplify the optimal robust network of Fig. 10-(a), and the result is shown in Fig. 10-(b).

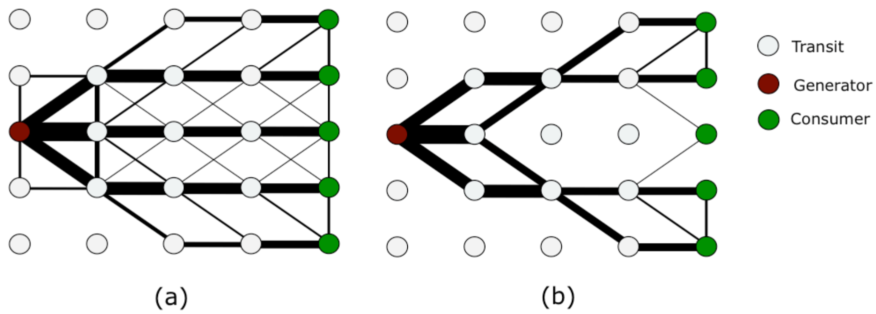

Fig. 10 (a) Optimal Robust Grid Topology (Link Failure), (b) Optimal Robust Sparse Grid topology

The network criticality of the sparse topology in Fig. 10-(b) is close to that of the original topology (Fig. 10-(a)) and it is still 1-robust, but the structure that number of active links (power lines) in the topology of Fig. 10-(b) is much less than the original one.

8 Conclusion

In this chapter we developed a number of optimization problems for simultaneous optimization of resources and flows in a variety of network types including communication networks and power grids. Our goal in the optimization problems was to minimize a weighted linear combination of resistance distances (point-to-point network criticalities) which is a convex function of link weights.

In another development, we discussed the problem off finding the best matched traffic matrix to a given network topology, along with optimal routing strategy (flow assignment) associated with optimal demand. We extended this idea and proposed an optimization problem to jointly optimize demands, routes and resources. Moreover, we discussed the application of network criticality in planning k-robust networks, where the topology is potentially protected against up to k link failures.

We used the proposed optimization problems to design virtual networks for different customers of a communication network. We also applied the k-robust strategy to design robust power networks.

There are different avenues for further development of the proposed ideas in this chapter. We can extend the optimization problems to the case where other QoS constraints, such as delay partitioning constraints, are also requested. Our discussion in this work was mostly on undirected networks (also we explained how we can have asymmetric capacities within the proposed framework). One more extension is to develop similar optimization problems for the general case of a directed graph.

Appendix

9 Proof of Lemma 1

Proof. We use proposition 3 to derive the result:

$$\begin{aligned}
\frac{\partial \tau_\alpha}{\partial w_{ij}} &= \frac{\partial Tr(U_\alpha \Gamma^{-1})}{\partial w_{ij}} \\
&= -Tr(U_\alpha \Gamma^{-1} \frac{\partial \Gamma}{\partial w_{ij}} \Gamma^{-1}) \\
&= -Tr(U_\alpha \Gamma^{-1} u_{ij} u_{ij}^t \Gamma^{-1}) \\
&= -Tr(F_\alpha^t F_\alpha \Gamma^{-1} u_{ij} u_{ij}^t \Gamma^{-1}) \\
&= -Tr(F_\alpha L^+ u_{ij} u_{ij}^t L^+ F_\alpha^t) \\
&= -Tr((F_\alpha L^+ u_{ij})(F_\alpha L^+ u_{ij})^t) \\
&= -Tr((F_\alpha L^+ u_{ij})^t (F_\alpha L^+ u_{ij})) \\
&= -\left\| F_\alpha L^+ u_{ij} \right\|^2
\end{aligned}$$

10 Proof of Lemma 2

Proof. We need the following lemma.

Lemma 4. *For any weight matrix W of links of a graph: $Vec(W)^t \nabla \tau + \tau = 0$, where $Vec(W)$ is a vector obtained by concatenating all the rows of matrix W to get a vector of w_{ij}'s.*

Proof. In [[13]] it has been shown that if we scale all the link weights with t, the effective resistance τ_{ij} will scale with $\frac{1}{t}$. Since τ_α is a linear combination of point-to-point effective resistances, τ_α will also scale with $\frac{1}{t}$:

$$\tau_\alpha(tVec(W)) = \frac{1}{t}\tau_\alpha(Vec(W)) \tag{58}$$

By taking the derivative of τ with respect to t, we have

$$Vec(W)^t \nabla\tau_\alpha = \frac{-1}{t^2}\tau_\alpha(W) \tag{59}$$

It is enough to consider equation 59 at $t = 1$ to get $Vec(W)^t\nabla\tau_\alpha + \tau_\alpha = 0$.

In general, one can apply the condition of optimality [24, 19] on optimization problem (36) to get necessary condition for a weight vector to be optimal. Let W^* be the optimal weight matrix, and let W_t be another weight matrix satisfying the constraints of optimization problem (36), then according to the condition of optimality:

$$\nabla\tau_\alpha.(Vec(W_t) - Vec(W^*)) \geq 0$$

Now, we choose W_t as follows:

$$W_t = [w_{uv}] = \begin{cases} \frac{C}{2z_{ij}} & if \;\; u = i \;\&\; v = j \\ \frac{C}{2z_{ij}} & if \;\; u = j \;\&\; v = i \\ 0 & otherwise \end{cases}$$

Clearly, W_t satisfies the constraints of optimization problem (36), therefore, using the condition of optimality and considering lemma 4 we have:

$$\begin{aligned} \nabla\tau_\alpha.(Vec(W_t) - Vec(W^*)) &\geq 0 \\ \nabla\tau_\alpha.Vec(W_t) - \nabla\tau_\alpha.Vec(W^*) &\geq 0 \\ \frac{C}{z_{ij}}\frac{\partial\tau_\alpha}{\partial w_{ij}} + \tau_\alpha &\geq 0 \quad \forall(i,j) \in E \\ \min_{(i,j)\in E} \frac{C}{z_{ij}}\frac{\partial\tau_\alpha}{\partial w_{ij}} + \tau_\alpha &\geq 0 \end{aligned} \tag{60}$$

Now, to prove the second part of the theorem we write the constraint of the optimization problem as an inner product of costs and weights.

$$(Vec(Z).Vec(W^*))\tau_\alpha = (\sum_{(i,j)\in E} w^*_{ij} z_{ij})\tau_\alpha = C\tau_\alpha \tag{61}$$

Combining lemma 4 and equation 61 one can see:

$$
\begin{aligned}
C\nabla\tau_\alpha.Vec(W^*) + Vec(Z).Vec(W^*)\tau_\alpha &= 0 \\
Vec(W^*).(C\nabla\tau_\alpha + \tau_\alpha Vec(Z)) &= 0 \\
w^*_{ij}(C\frac{\partial\tau_\alpha}{\partial w_{ij}} + \tau_\alpha z_{ij}) &= 0
\end{aligned}
$$

This completes the proof of lemma 2.

11 Proof of Lemma 3

Proof. The Lagrangian of optimization problem is:

$$
\begin{aligned}
L(\Gamma, W, T, \lambda, \rho) = {} & Tr(U_\alpha \Gamma^{-1}) + Tr(T\Gamma) - C\lambda \\
& + \sum_{(i,j)\in E} w_{ij}(-u^t_{ij}Tu_{ij} + \lambda z_{ij} - \rho_{ij}) - \frac{1}{n}\overrightarrow{1}^t T\overrightarrow{1}
\end{aligned}
$$

To find the dual formulation, it is enough to take the infimum of the Lagrangian over Γ, W.

$$
\begin{aligned}
d(T, \lambda, \rho) &= inf_{\Gamma,W} L(\Gamma, W, T, \lambda, \rho) \\
&= inf_\Gamma Tr(U_\alpha\Gamma^{-1} + T\Gamma) + inf_W(\\
&- \sum_{(i,j)\in E} w_{ij}(-u^t_{ij}Tu_{ij} + \lambda z_{ij} - \rho_{ij}) - \frac{1}{n}\overrightarrow{1}^t T\overrightarrow{1} - C\lambda)
\end{aligned} \tag{62}
$$

The second term in equation (62) is minimized if its derivative with respect to all link weights is zero. The minimum of the first term is $-\infty$ unless matrix T is positive semi-definite. Therefore

$$
d(T, \lambda, \rho) = \begin{cases} inf_\Gamma Tr(U_\alpha\Gamma^{-1} + T\Gamma) - \frac{1}{n}\overrightarrow{1}^t T\overrightarrow{1} - C\lambda \\ if -u^t_{ij}Tu_{ij} + \lambda z_{ij} - \rho_{ij} = 0, \;\; \rho_{ij} \geq 0 \;\; \forall (i,j) \in E \\ and \;\; T \geq 0 \\ -\infty \;\; otherwise \end{cases} \tag{63}
$$

where $T = [t_{ij}]$, and $T \geq 0$ means that matrix T is positive semi-definite.

Term $inf_\Gamma Tr(U_\alpha\Gamma^{-1} + T\Gamma)$ can also be obtained analytically using some known facts from matrix algebra.

Proposition 3 *For any non-singular square ($n\times n$) matrix X and any $n\times n$ matrices A and B, we have :*

$$\frac{d}{dX}Tr(AX) = A$$
$$\frac{d}{dX}Tr(AX^{-1}B) = -X^{-1}BAX^{-1}$$

where in general the derivative $\frac{d}{dX}f(X)$ *of a scalar-valued differentiable function* $f(X)$ *of a matrix argument* $X \in R^{p\times q}$ *is the* $q\times p$ *matrix whose* $(i,j)^{th}$ *entry is* $\frac{\partial f(X)}{\partial X(j,i)}$ *[16].*

Proof. See [16].

Using proposition 3 one can find $inf_{\Gamma}(U_\alpha Tr\Gamma^{-1} + T\Gamma)$ as follows.

$$\frac{d}{d\Gamma}Tr(U_\alpha\Gamma^{-1} + T\Gamma) = \frac{d}{d\Gamma}Tr(U_\alpha\Gamma^{-1}) + \frac{d}{d\Gamma}Tr(T\Gamma) = 0$$
$$T = \Gamma^{-1}U_\alpha\Gamma^{-1} \quad (64)$$

Considering the fact that U_α and Γ^{-1} are symmetric matrices, after some calculations we have:

$$inf_{\Gamma}Tr(U_\alpha\Gamma^{-1} + T\Gamma) = 2Tr(U_\alpha\Gamma^{-1}) \quad (65)$$

We also note that $T\overrightarrow{1} = 0$ (because $\Gamma^{-1}\overrightarrow{1} = (L^+ + \frac{J}{n})\overrightarrow{1} = \overrightarrow{1}$ and $U_\alpha\overrightarrow{1} = 0$). Now we consider a change of variable as $X = \frac{1}{\sqrt{\lambda}}L^+$, clearly $X\overrightarrow{1} = 0$. It is easier to write the optimization problem based on new variable X. Applying this change of variable, equation (65) we have:

$$2Tr(U_\alpha\Gamma^{-1}) = 2\sqrt{\lambda}Tr(U_\alpha X) \quad (66)$$

On the other hand:

$$\begin{aligned} T &= \Gamma^{-1}U_\alpha\Gamma^{-1} \\ &= \lambda XU_\alpha X \end{aligned} \quad (67)$$

Finally, we observe from constraint part of equation (63) that:

$$\frac{1}{z_{ij}}u_{ij}^tTu_{ij} \le \lambda \;\; \forall(i,j)\in E$$
$$\frac{1}{z_{ij}}u_{ij}^t\lambda XU_\alpha Xu_{ij} \le \lambda \;\; \forall(i,j)\in E$$
$$\frac{1}{\sqrt{z_{ij}}}\left\|F_\alpha Xu_{ij}\right\| \le 1 \quad (68)$$

where F_α is an $m \times n$ matrix (m and n are the number of links and nodes of the graph respectively) such that $U_\alpha = F_\alpha^t F_\alpha$. This matrix decomposition always exists, since U_α is a positive semidefinite matrix.

Now it is enough to simplify the dual objective function of equation (63) (i.e. $d(X,\lambda) = 2\sqrt{\lambda}Tr(U_\alpha X) - C\lambda$) using equation (66). In order to maximize the dual function with respect to the dual variable λ, one should have: $\frac{d}{d\lambda}d(X,\lambda) = 0$. By applying this, and after some calculations, the dual objective will be equal to $\frac{1}{C}Tr^2(U_\alpha X)$, which is now only a function of dual variable X.

Therefore, considering equation (68), one can write the dual optimization problem as:

$$\begin{aligned} &\textit{maximize} && \tfrac{1}{C}Tr^2(U_\alpha X) \\ &\textit{subject to} && \tfrac{1}{\sqrt{z_{ij}}}\left\|F_\alpha X u_{ij}\right\| \le 1 \ \ \forall (i,j) \in E \\ &&& X\overrightarrow{1} = 0 \\ &&& X \ge 0 \end{aligned}$$

Finally, we observe that:

$$\begin{aligned} \lambda &= \max_{(i,j)\in E} \frac{1}{z_{ij}} u_{ij}^t T u_{ij} \\ &= \max_{(i,j)\in E} \frac{1}{z_{ij}} u_{ij}^t L^+ U_\alpha L^+ u_{ij} \\ &= \max_{(i,j)\in E} \frac{1}{z_{ij}} \left\|F_\alpha L^+ u_{ij}\right\|^2 \end{aligned} \tag{69}$$

This completes the proof of lemma 3.

References

1. U. Brandes, T . Erlebach (Editors). Network Analysis: Methodological Foundations. *Lecture Notes in Computer Science, Springer-Verlag*, vol. 3418, 2005.
2. L. C. Freeman. Centrality in Networks: I. Conceptual Clarification. *Social Networks*, (1):215–239, 1978/79.
3. L.C. Freeman, S.P. Borgatti, and D.R. White. Centrality in valued graphs: a measure of betweenness based on network flow. *Social Networks*, 13:141–154, 1991.
4. A. Tizghadam and A. Leon-Garcia. A Robust Routing Plan to Optimize Throughput in Core Networks. *Managing Traffic Performance in Converged Networks, Lorne Mason, Tadeusz Drwiega and James Yan (Eds.), Springer*, pages 117–128, 2007.
5. M. Newman. A Measure of Betweenness Centrality Based on Random Walks. *Social Networks*, 27:39-54, 2005.
6. A. Tizghadam and A. Leon-Garcia. Robust Network Planning in Nonuniform Traffic Scenarios. *Submitted to Computer Communications*.

7. A. Tizghadam and A. Leon-Garcia. A Graph Theoretical Approach to Traffic Engineering and Network Control Problem. In 21^{th} *International Teletraffic Congress (ITC21)*, Paris, France, September 2009.
8. C. R. Rao and S. K. Mitra. *Generalized Inverse of Matrices and its Applications*. John Weily and Sons Inc., 1971.
9. A. Tizghadam and A. Leon-Garcia. Autonomic Traffic Engineering for Network Robustness. *IEEE Journal of Selected Areas in Communications (J-SAC)*, 28(1):39 – 50, January 2010.
10. A. Tizghadam and A Leon-Garcia. Survival Value of Communication Networks. In *INFOCOM Workshop on Network Science for Communications (NetSciCom)*, pages 1–6, Rio de Janeiro, Brazil, April 2009.
11. A. Tizghadam and A Leon-Garcia. Betweenness Centrality and Resistance Distance in Communication Networks. In *IEEE Network*, vol. 24, no. 6, pages 10–16, November-December 2010.
12. D. J. Klein and M. Randic. Resistance Distance. *Journal of Mathematical Chemistry*, 12(1):81–95, December 1993.
13. A. Ghosh, S. Boyd, and A. Saberi. Minimizing Effective Resistance of a Graph. *SIAM Review, problems and techniques section*, 50(1):37–66, February 2008.
14. L. Kleinrock. Queueing Systems, volume II. *John Wiley & Sons*, 1975.
15. AH. Land and AG. Doig. An automatic method of solving discrete programming problems. *Econometrica*, vol. 28, no. 3, pages 497-520, July 1960.
16. Dennis S. Bernstein. Matrix Mathematics. *Princeton University Press*, 2th edition, 2009.
17. M. Grant and S. Boyd. CVX: Matlab Software for Disciplined Convex Programming (Web Page and Software). $http://stanford.edu/~boyd/cvx$. September 2008.
18. M. Grant and S. Boyd. Graph Implementations for Nonsmooth Convex Programs, Recent Advances in Learning and Control (a tribute to M. Vidyasagar), V. Blondel, S. Boyd, and H. Kimura, editors, $http://stanford.edu/~boyd/graph_dcp.html$. *Lecture Notes in Control and Information Sciences, Springer*, 2008.
19. S. Boyd and L. Vandenberghe. Convex Optimization. *Cambridge University Press*, 2004.
20. L. Xiao and M. Johansson, and S. Boyd. Simultaneous Routing and Resource Allocation via Dual Decomposition. *IEEE Transactions on Communications*, vol. 52, no. 7, pages 1136-1144, July 2004.
21. A. Tizghadam and A. Leon-Garcia. On Random Walks in Direction-Aware Network Problems. *ACM SIGMETRICS Performance Evaluation Review (PER)*, vol. 38, no. 2, pages 9-11, September 2010.
22. JK. Johnson, M. Chertkov. A Majorization-Minimization Approach to Design of Power Transmission Networks. *arXiv:1004.2285*, September 2010.
23. DA. Spielman, N. Srivastava. Graph Sparsication by Effective Resistances. *Proceedings of the 40th annual ACM Symposium on Theory of Computing STOC'08*, pages 563-568, 2008.
24. D. P. Bertsekas, A. Nedic, and A. E. Ozdaglar. Convex Analysis and Optimization. *Athena Scientific*, April 2003.