# Wireless Mesh Network design by Optimization of Spectral Properties and Max-k-Cut

Veeranjaneyulu Sadhanala

Wednesday 29th November, 2017

**Abstract**

Topology control and channel assignment are key steps in the design of Wireless Mesh Networks (WMNs). Topology control consists of selecting a subset of communication links to set up the routing network, subject to certain degree constraints. Assigning non-overlapping frequency channels to the selected links in order to minimize the interference during communication is called the channel assignment problem. For topology control, we optimize spectral properties of the network graph such as effective resistance and second smallest eigenvalue of the graph Laplacian. We are not aware of usage of these quantities in WMN literature even though they are well-studied in other contexts. For channel assignment, we minimize total interference assuming all the links are active. Based on our simulations, we recommend methods to use depend on the network size. We further propose an alternating minimization scheme to jointly optimize topology and channel assignment and leave it for future work.

***DAP Committee members:***

Ryan Tibshirani ⟨*ryantibs@gmail.com*⟩ (Statistics & Machine Learning);
Javier Peña ⟨*jfp@andrew.cmu.edu*⟩ (Tepper School);
Andrew Price ⟨*adp162@gmail.com*⟩ (A Large Internet Company)

Approved by DAP Committee Chair: ______________________________

# 1 Introduction

Wireless Mesh Networks(WMNs) are communication networks consisting of radios relaying information in a wireless fashion. The radios can live in computers or in dedicated boxes for example. Because of their resilience and adaptability, WMNs are used for communication in military, satellite groups, municipalities and university campuses (Akyildiz et al., 2005; Pathak and Dutta, 2011). These are also suitable for deployment in densely populated urban areas where laying cables for wired networks is expensive.

We study two steps, namely topology control and channel assignment, that are key to designing a good WMN, in the following network setup. Suppose we are given a set of $n$ mesh routers (nodes) $V$, and a possible set of communication links (edges) $\overline{E}$. The nodes lie on a plane and the region around each node is divided into $S$ equiangular regions called sectors. Figure 1 shows a network of $n = 4$ nodes where the 4 sectors are shown for node $u$ with angle $\pi/2$. Routers have the following hardware constraint:

*Sector constraint:* Nodes can connect to at most $r = 1$ of its neighbors in each sector.

**Topology control** consists of choosing a subset $E$ of $\overline{E}$ that forms the routing network and satisfies the sector constraint. A typical design goal is a routing network that is well-connected and facilitates high throughput.

Reducing *interference* between simultaneous communications is another important competing goal. Two links $(u_1, v_1), (u_2, v_2) \in \overline{E}$ are said to interfere with each other if the signal from $u_1$ or $v_1$ has a strength greater than certain threshold at $u_2$ or $v_2$ or vice-versa. Figure 1 explains the phenomenon of interference with an example. Availability of multiple non-overlapping frequency channels mitigates the interference problem, in that, simultaneous communications can happen if they are on different channels. However there a limited number of such channels (say $K = 4$) in the frequency range that the routers operate in.

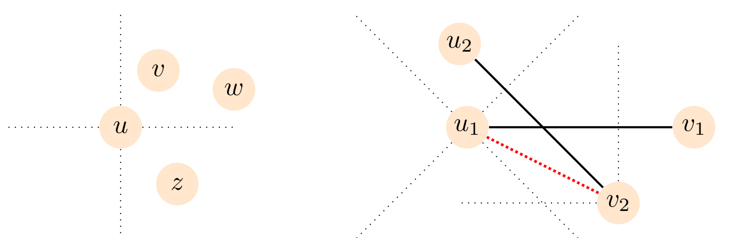


Figure 1: *(Left - Sector constraint) 4 sectors around node $u$ are indicated with dotted lines. Suppose $(u, v), (u, w) \in \overline{E}$. Sector constraint dictates that at most one of them can be in $E$. (Right - Interference): Communication on $(u_1, v_1)$ interferes with reception at node $v_2$ while node $u_2$ is transmitting to $v_2$ if both the communications are in the same frequency channel. Red dotted line indicates interference.*

**Channel assignment** consists of assigning frequency channels to links to reduce interference. It is non-trivial to optimally assign channels to links. In fact, minimization of total number of interfering pairs of links is equivalent to the well-known max-k-cut problem and hence is NP-complete (Frieze and Jerrum, 1997).

The eventual goal is to maximize quality of service in the network by optimal topology control and channel assignment. We take a step towards that by developing topology control algorithms that give well-connected networks and channel assignment algorithms that efficiently minimize the total interference, using a mixture of continuous and discrete optimization programs.

The rest of the report is organized as follows. In Section 2 we give details about the network model and data. In Section 3, we recall some graph spectral properties, connection between minimization of interference and max-k-cut and mention relevant works. We describe the topology and channel assignment algorithms in Sections 4 and 5 respectively. Then we empirically compare the algorithms in Section 6 on simulated networks.

## 2 Network model and quality measure

Networks are generated according to specifications given by Andrew Price. We are given $n$ mesh routers and their neighborhoods in terms of RSSI (received signal strength indicator). Each router has 4 directional antennas with their axes in the horizontal plane, transmitting and receiving in disjoint angular regions of size $\pi/2$. The directional antennas are not steerable. The radios operate in one of $K = 4$ available channels in the 5GHz band and follow IEEE 802.11s specification. More concrete details about the network generation such as RSSI calculation are given in Section 6.

In the following, we describe models for interference, link capacity and the approach we use to measure the quality of the networks given by topology control and channel assignment algorithms. A bit of notation: let $[m]$ denote the set $\{1, \ldots, m\}$ for positive integers $m$.

### 2.1 Interference model

For nodes $i, j \in [n]$ let $P(i, j)$ denote the RSSI at $i$ due to $j$. Two links $e = (u, v)$ and $e' = (u', v') \in E$ are said to interfere with each other if communication on one link creates a noise large enough to significantly affect the communication on the other link, that is, if

$$\max_{i \in \{u,v\}, j \in \{u',v'\}} P(i, j) \geq p_0 \quad \vee \quad \max_{i \in \{u',v'\}, j \in \{u,v\}} P(i, j) \geq p_0$$

where $p_0$ is a threshold depending on the hardware and $a \vee b = \max\{a, b\}$.

### 2.2 Capacity model

The capacity $\mathrm{cap}(e)$ of a link $e \in E$ is modeled as an non-decreasing function of its SNR (see Section 6 for a concrete function we use). When there is interference, for $e \in E$, let $d_I(e)$ denote the number of links $e' \in E$ in the selected topology, that interfere with $e$. A link is assumed to interfere with itself and so $d_I(e) \geq 1$. Considering CSMA/CA protocol that is used in 802.11s standard, we assume that the capacity of $e$ is reduced by a factor of $d_I(e)$ due to interference when all the links are active. That is, we assume that the capacity under interference is $\mathrm{cap}(e)/d_I(e)$.

### 2.3 Network quality measure

To measure the quality of a routing network $E$ with a channel assignment $c$, we propose the following approach, similar to Wong and Gary Chan (2014). We simulate a set of source/sink traffic flow demands $\{(s_i, t_i, d_i) : i = 1, \ldots, n_d\}$, where $s_i, t_i$ are the source and sink and $d_i$ is the desired bandwidth for the $i$th demand. We calculate the maximum $\alpha$ such that the $i$th demand is satisfied with a flow of at least $\alpha \cdot d_i$ by solving the following linear program in (1). This is appropriate when a minimum quality of service needs to be met for each flow. Let $f_i$ denote the directed non-negative flow intended to satisfy the $i$th demand. We distinguish $f_i(u,v)$ from $f_i(v,u)$ for $u, v \in V$.

$$\begin{aligned}
&\underset{\alpha,\, f\geq 0}{\text{maximize}} \quad \alpha && (1)\\
&\text{subject to} \sum_{(u,v)\in E} f_i(u,v) = \sum_{(v,w)\in E} f_i(v,w) \quad \forall v \in V \setminus \{s_i, t_i\},\ \ \forall i \in [n_d] && (2)\\
&\sum_{(u,s)\in E} f_i(u,s_i) = \sum_{(t_i,u)\in E} f_i(t,u) = 0 \quad \forall i \in [n_d] && (3)\\
&\sum_i f_i(u,v) + f_i(v,u) \leq c(u,v) \quad \forall (u,v) \in E && (4)\\
&\sum_v f_i(s_i,v) \geq \alpha \cdot d_i \quad \forall i \in [n_d] && (5)
\end{aligned}$$

In the above, (2) is the flow constraint, (4) is the capacity constraint and (5) ensures that the demand is satisfied at least up to a factor of $\alpha$. (3) sets the flows to source and from sink to 0 because this is an undirected graph. Denote the amount of $i$th flow $\alpha_i = \sum_{(s_i,v)\in E} f_i(s_i,v)$ and the average flow $\overline{\alpha} = \frac{1}{n_d}\sum_{i=1}^{n_d} \alpha_i$. Note that $\alpha_i \geq \alpha \cdot d_i$ for all $i \in [n_d]$ and so $\overline{\alpha} \geq \alpha \cdot \frac{1}{n_d}\sum_{i=1}^{n_d} d_i$. When all the demands are unity, $\alpha, \overline{\alpha}$ are the minimum and average flows possible. We use $\alpha$ and $\overline{\alpha}$ as measures of quality of the network. The higher these quantities are, the better the network is.

## 3 Background and Related work

Before describing the topology control and channel assignment algorithms, we recall the definitions of effective resistance, Fiedler value and explain how channel assignment is reduced to max-k-cut. Then we mention relevant previous literature.

### 3.1 Effective resistance and Fiedler value

Let $N = (V, E, w)$ be the (routing) network where $w_{ij}$ denotes the bandwidth of link $(i,j)$. Let $L \in \mathbb{R}^{n\times n}$ be the graph Laplacian of $N$, given by

$$L_{ij} = \begin{cases} -w_{ij} & i \neq j \\ \sum_{i'\neq i} w_{ii'} & i = j. \end{cases}$$

See Spielman (2007) for the following properties of $L$. $L$ is positive semidefinite and $L\mathbb{1} = 0$ for any graph. That is, the smallest eigenvalue of $L$ is always 0. That second smallest eigenvalue $\lambda_2(L)$, also called the Fiedler value reveals some interesting properties of the graph. If $\lambda_2(L) = 0$, then the graph is disconnected. In fact, the multiplicity of eigenvalue 0 gives the number of connected components in the graph. If $E' \subseteq E$ with the same edge weights, then $\lambda_2(L(E')) \leq \lambda_2(L(E))$. Between two subgraphs of a graph, larger $\lambda_2(L)$ indicates better connectivity.

Now we recall the definition of effective resistance. Treating the bandwidths of network links as electrical conductances, the electrical network corresponding to $N$ has a total effective resistance given by

$$R = \sum_{i<j} R_{ij} = \begin{cases} n \cdot \mathrm{tr}\left(L^{+}\right) & \text{if } N \text{ is connected} \\ \infty & \text{otherwise} \end{cases}$$

where tr denotes the trace operator and $L^{+}$ is the pseudo-inverse of $L$ (Ghosh et al., 2008). $R_{ij}$ is known to be proportional to the average commute time for a random walker to go from $i$ to $j$ and return to $i$, where the random walker takes steps to its neighbors with a probability proportional to the edge weights. A smaller $R = \sum_{i<j} R_{ij}$ indicates a better connected graph. $R$ gets larger as network gets closer to getting disconnected and becomes $\infty$ for disconnected graphs. It gets smaller as we add edges. In other words, if $E \supseteq E'$ with the same edge weights, then $R(E) \leq R(E')$ with a slight abuse of notation.

### 3.2 Channel assignment and Max-k-Cut

Suppose the routing network $E$ is known and let $m = |E|$. One approach to solve the channel assignment problem is to find an assignment $\widehat{c}$ that minimizes the total number of interfering pairs of links:

$$\widehat{c} \in \operatorname*{argmin}_{c \in \{1,.,K\}^m} \quad \frac{1}{2} \sum_{e,e' \in E} I_{ee'} \mathbb{1}(c_e = c_{e'}). \tag{6}$$

where $I_{ee'} \in \{0, 1\}$ indicates whether the links $e, e' \in E$ interfere if they are on the same channel. Rewrite the objective in (6) to note that,

$$\operatorname*{argmin}_{c \in [K]^m} W - \frac{1}{2} \sum_{e,e' \in E} I_{ee'} \mathbb{1}(c_e \neq c_{e'}) \quad = \quad \operatorname*{argmax}_{c \in [K]^m} \frac{1}{2} \sum_{e,e' \in E} I_{ee'} \mathbb{1}(c_e \neq c_{e'})$$

where $W = \sum_{e,e' \in E} I_{ee'}/2$. This is the well-known max-k-cut problem on the *edge conflict graph* whose nodes are given by the edges $e \in E$ and $I_{ee'}$ indicates whether there is an edge between $e$ and $e'$ in the conflict graph. The problem is known to be NP-complete. Further, any polynomial time approximation scheme has an approximation ratio less than a fixed quantity that is less than 1, unless P=NP (Frieze and Jerrum, 1997). In Section 5, we borrow heuristics for max-k-cut from graph theory literature to solve our channel assignment problem.

### 3.3 Related works

Effective resistance or average commute time and Fiedler value are well-studied concepts in graph theory (Fiedler, 1973; von Luxburg et al., 2014; Chung, 1997). They have applications in several areas such as collaborative movie recommendation (Fouss et al., 2005), commute-time kernel based clustering (Yen et al., 2007), semi-supervised learning (Zhou and Schölkopf, 2004), image segmentation (Shi and Malik, 1997) and graph sparsification (Spielman and Srivastava, 2011).

Spielman and Srivastava (2011) give a graph sparsification algorithm which delivers a sparse subgraph while preserving the eigenvalues upto a constant multiplicative factor $\kappa$ arbitrarily close to 1, thus preserving the effective resistance and Fiedler value upto the same factor $\kappa$. However, with the sector constraint, we believe that their method will be ineffective in our setting because in practice, the constant factor that is hidden in the complexity of the number of edges that need to be sampled for sparsification is large.

Ghosh et al. (2008) give a fast interior-point method to minimize the effective resistance of a graph subject to the constraint that the nonnegative edge weights of the graph sum to 1. Ghosh and Boyd (2006) propose a computationally tractable heuristic to approximately maximize the second smallest eigenvalue of the Laplacian of the graph, if one needs to add a fixed number of edges to a give base graph. We compare our methods to this heuristic in the experiments.

There are several works addressing topology control and channel assignment in WMNs and so we do not attempt to mention all of them here. The survey paper by Pathak and Dutta (2011) gives a good overview. Subramanian et al. (2008) formulate the channel assignment problem as minimization of total interference and give a heuristic called Tabu search to solve the problem. Sridhar et al. (2009) considers its integer programming formulation and its Lagrangian relaxations. Liu et al. (2011) greedily chooses channels to minimize the maximum interference on any link. They further choose a topology by assuming that the directional antennae can be oriented in arbitrary directions. We do not have the flexibility to change the orientations in our setting. Wong and Gary Chan (2014) propose to jointly optimize topology and channel assignment by solving an integer program. They assume that if the SINR (Signal to Interference + Noise Ratio) is greater than a certain threshold on a link $e$, then the communication can happen on $e$. However, if there are links $e'$ interfering with $e$, then if the devices follow CSMA/CA protocol, it should not be possible for them to be active simultaneously even if SINR is good. Therefore, their method may be more realistic with a future MAC protocol.

## 4 Topology Control algorithms

The objective in topology control is to choose a a subset $E$ of $\overline{E}$ with good properties. Denote $\overline{m} = |\overline{E}|, m = |E|$. Let $z_e = \mathbb{1}(e \in E)$ be a binary variable for $e \in \overline{E}$. First, we claim that the sector constraint is linear in $z$. Let $N(i, s)$ denote the edges in $\overline{E}$ that are incident on the node $i$ in its sector $s$. The sector constraint is

$$\sum_{e \in N(i,s)} z_e \leq r \quad \text{for all nodes } i \text{ and sectors } s$$

where $r$ is the number of radios available on each sector. Let $A_s$ be the 0-1 matrix of size $nS \times \overline{m}$, with $A_s((i-1) \cdot S + s,\ e) = \mathbb{1}(e \in N(i, s))$ for $i \in [n], s \in [S], e \in \overline{E}$ where $n = |V|$ and $S$ is the number of sectors available per node. [1] Then the following is equivalent to the above constraint:

$$A_s z \leq r\mathbb{1}.$$

[1]Note that we abused the notation a bit here; we used $e$ as an element in both $\overline{E}$ and $[\overline{m}]$.

The first method we consider is a naive one for baseline comparison. The next two methods are to optimize effective resistance and Fiedler value respectively. The last method is a heuristic designed to be fast.

### 4.1 Choose Nearest Neighbors

In this naive method, we keep iterating through an arbitrarily ordered list of nodes and make connections to the nearest neighboring nodes whenever possible. In other words, at each node $i$, for all free sectors on $i$, pick the nearest neighbor on the sector that has a free radio on its sector and add this link to the topology. The distance is measured by power received; higher the power received, lower the distance.

### 4.2 Minimize Effective Resistance

For $z \in \{0,1\}^{\overline{m}}$, let $G(z)$ denote the graph formed by the edges $\{e \in \overline{E} : z_e = 1\}$. Let $R(z), L(z), L^+(z)$ denote the effective resistance of $G(z)$, the graph Laplacian of $G(z)$ and its pseudoinverse respectively. The problem is to

$$\underset{z\in\{0,1\}^{\overline{m}}}{\text{minimize}}\, R(z) \quad \text{subject to} \quad A_s z \leq r,\ G(z) \text{ is connected.} \tag{7}$$

Following Ghosh et al. (2008), by observing $R(z) = n \cdot \operatorname{tr}(L^+(z))$ when $G(z)$ is a connected graph, the above problem can be reduced to the following SDP with binary constraints as shown below.

$$\begin{aligned} \underset{z\in\{0,1\}^{\overline{m}},Y}{\text{minimize}} \quad & n \cdot \operatorname{tr}(Y) \\ \text{subject to} \quad & A_s z \leq r \\ & M(z,Y) := \begin{bmatrix} L(z) + \mathbb{1}\mathbb{1}^T/n & I_n \\ I_n & Y \end{bmatrix} \succeq 0. \end{aligned} \tag{8}$$

For (7), (8) the optimal value is assumed to be $\infty$ when they are infeasible. The SDP constraint in (8) enforces connectivity as argued now. If $G(z)$ is disconnected, then recalling that nullity$(L(z))$ is the number of components in $G(z)$, $L(z) + \mathbb{1}\mathbb{1}^T/n$ should have a nontrivial null space. For $a \in \text{null}(L(z) + \mathbb{1}\mathbb{1}^T/n), b \in \mathbb{R}^n$, such that $a \neq 0, a^T b \neq 0$,

$$\begin{bmatrix} a^T b^T \end{bmatrix} M(z,Y) \begin{bmatrix} a \\ b \end{bmatrix} = 2a^T b + b^T Y b.$$

For any fixed $Y \in \mathbb{R}^{n\times n}$, $a$ can be scaled by an arbitrarily large constant so that the above quantity becomes negative. Hence, if $G(z)$ is disconnected, then $M(z,Y)$ cannot be positive semidefinite. In other words, the constraint $M(z,Y) \succeq 0$ enforces connectivity of $G(z)$.

On the other hand, if $G(z)$ is connected, then $L(z) + \mathbb{1}\mathbb{1}^T/n \succ 0$ and hence $M(z,Y) \succeq 0$ if and only if its Schur complement of $Y$ is positive semidefinite, that is,

$$M(z,Y) \succeq 0 \quad \Longleftrightarrow \quad Y - (L(z) + \mathbb{1}\mathbb{1}^T/n)^{-1} \succeq 0.$$

Therefore (8) may be rewritten as

$$\begin{aligned} \underset{z\in\{0,1\}^{\overline{m}}}{\text{minimize}} \quad & n \cdot \mathrm{tr}(L(z) + \mathbb{1}\mathbb{1}^T/n)^{-1} \\ \text{subject to} \quad & A_s z \leq r,\ G(z) \text{ is connected} \end{aligned} \tag{9}$$

with the relation $\widehat{Y} = (L(\widehat{z}) + \mathbb{1}\mathbb{1}^T/n)^{-1}$ at solutions $\widehat{z}$ to (9) and $(\widehat{z}, \widehat{Y})$ to (8). Further, when $G(z)$ is connected, $n \cdot \mathrm{tr}(L(z) + \mathbb{1}\mathbb{1}^T/n)^{-1} = n\mathrm{tr}(L^+(z)) + n$. Therefore (7) and (8) are equivalent upto a constant difference in the objective.

### 4.3 Maximize the Fiedler value

As noted in Section 3, the second smallest eigenvalue of $L$, denoted by $\lambda_2(L)$ is a well-known connectivity metric of a graph, see Fiedler (1973). Maximizing $\lambda_2(L)$ is another approach to get a well-connected sub-graph. The problem can be again formulated as an integer SDP problem, see Ghosh and Boyd (2006):

$$\begin{aligned} \underset{z\in\{0,1\}^{\overline{m}},\lambda}{\text{maximize}} \quad & \lambda \\ \text{subject to} \quad & L(z) + \lambda(\mathbb{1}\mathbb{1}^T/n - I_n) \succeq 0 \\ & A_s z \leq r \end{aligned} \tag{10}$$

To see the validity of the formulation, observe that if the eigenvalues of $L$ are $0 = \lambda_1 \leq \lambda_2 \leq \cdots \lambda_n$, then the corresponding eigenvalues of $L + t(\mathbb{1}\mathbb{1}^T/n - I_n)$ are $0, \lambda_2 - t, \cdots, \lambda_n - t$ for any $t \in \mathbb{R}$.

Similar to (8), the above integer SDP quickly becomes computationally intractable. Ghosh and Boyd (2006) propose the following greedy heuristic to maximize $\lambda_2(L)$, given a budget of edges to be added: greedily pick an edge $(i, j)$ with the highest $|v_i - v_j|^2$ where $v$ is the Fiedler vector and add it to the graph. If $\lambda_2$ is not a repeated eigenvalue, then $|v_i - v_j|^2$ is the increment in $\lambda_2(L)$ due to the addition of $(i, j)$ up to a first order approximation. See Ghosh and Boyd (2006) for more details. We adapt this heuristic to our setting as follows: We start from an empty graph and until the graph remains disconnected, we keep adding edges between different components while satisfying the sector constraint. Once it is connected, we follow their heuristic while satisfying the sector constraint. Whenever adding an edge selected according to the highest difference $|v_i - v_j|$ violates the sector constraint, we don't add it and simply go to the edge with the next highest difference and so on.

### 4.4 Maximize $\mathrm{tr}(L)$

The cost of solving the above integer SDPs scales poorly with $n$; it takes several tens of minutes for $n > 50$. We develop a heuristic with much lower computational complexity here. Instead of minimizing $\mathrm{tr}(L^+)$ to minimize effective resistance, we maximize $\mathrm{tr}(L)$. $\mathrm{tr}(L)$ is simply the sum of weights of the edges selected subject to the sector constraint. In symbols,

$$\underset{z\in\{0,1\}^{\overline{m}}}{\text{maximize}} \quad \sum_e z_e w_e \quad \text{subject to} \quad A_s z \leq r\mathbb{1},\ G(z) \text{ is connected} \tag{11}$$

Let $w_{\min} = \min_{e\in\overline{E}} w_e$. The connectivity constraint can be formulated by asserting that for the $n-1$ source sinks pairs $\{(u_1, u_i) : u_1 \in V, u_i \in V \setminus \{u_1\}\}$, there is a flow of at least $w_{\min}$.

$$
\begin{aligned}
\underset{z\in\{0,1\}^{\overline{m}}, f\geq 0}{\text{maximize}} \quad & \sum_e z_e w_e \\
\text{subject to} \quad & A_s z \leq r\mathbb{1}, \\
& \sum_{(v',v)\in\overline{E}} f_i(u,v) = \sum_{(v,v')\in\overline{E}} f_i(v,v') \quad \forall v \in V \setminus \{u_1, u_i\}, \;\; 2 \leq i \leq n, \\
& \sum_{(u,u_1)\in\overline{E}} f_i(u,u_1) = \sum_{(u_i,u)\in\overline{E}} f_i(u_i,u) = 0 \quad 2 \leq i \leq n, \\
& f_i(u,v) + f_i(v,u) \leq c(u,v)w(u,v) \quad \forall (u,v) \in \overline{E}, \\
& \sum_v f_i(u_1,v) \geq w_{\min}.
\end{aligned} \tag{12}
$$

This is an integer linear program (ILP). The number of constraints and nonzeros in the connectivity constraint is $\Theta(n\overline{m})$. For $n$ upto a 1000, it can be solved in reasonable time using branch-and-bound and cutting planes. For larger $n$, bundle methods which carefully add constraints may be used, but this is beyond the scope of this report.

### 4.5 Implementation details

The problems (8),(10) are computationally expensive to solve even for small problems. We use a simple branch-and-bound algorithm BNB from YALMIP toolbox (Löfberg, 2004). In branch-and-bound, lower bounds are updated by solving the SDP obtained by relaxing the binary variables which are not fixed at the branching node to $[0, 1]$. We use the standard SDPT3 (Toh et al., 1999) solver to solve the SDP relaxations.

Even for small networks, the BNB solver takes several tens of minutes to solve the problem to optimality. On the other hand, the ILP (11) can be solved pretty quickly with Gurobi software (Gu et al., 2010) which uses a parallel branch-and-bound algorithm with cutting planes. We also use this solution to warm-start the BNB solver.

For effective resistance minimization method in (8), the relaxed SDP may be solved by a potentially faster method that is obtained by adapting the clever interior point method given by Ghosh et al. (2008) for a slightly different problem. We leave this to future work.

# 5 Channel Assignment algorithms

In this section we assume that the topology $E \subseteq \overline{E}$ is known and seek to find a channel assignment. We rewrite (6) as an ILP with binary variables $c, C$ of sizes $m \times K$ and $m \times m \times K$ respectively:

$$\underset{c,C}{\text{minimize}} \sum_{e,e' \in E} \frac{1}{2} I_{ee'} \sum_{k=1}^{K} C_{ee'k} \tag{13}$$

$$\text{subject to } \; c_{ek} + c_{e'k} - 1 \leq C_{ee'k}, \qquad e, e' \in E, k \in [K] \tag{14}$$

$$\sum_k c_{ek} = 1, \qquad e \in E, k \in [K] \tag{15}$$

Here $c_{ek} = 1$ means that the edge $e \in E$ is assigned channel $k$. (15) ensures that a channel is assigned to all edges. The triangle inequality in (14) helps linearize the optimization criterion. Further, if $e, e' \in E$ interfere on single channel and $C_{ee'k} = 1$ for some $k \in [K]$ at the optima, then it must happen that $c_{ek} = c_{e'k} = 1$ and hence $e, e'$ interfere after channel assignment as well.

For small networks, the above ILP above can be solved to optimality using branch-and-bound techniques. But as the network gets larger, the computation quickly becomes intractable. Here we consider a few methods that work in practice.

## 5.1 Random channel assignment

In this naive algorithm, links are assigned channels drawn uniformly randomly from $\{1, \cdots, K\}$. The probability of two interfering links $e, e'$ to be on the same channel $P(c_e = c_{e'}) = 1/K$. Therefore, expected total interference

$$\mathbb{E}\widehat{I} = \sum_{e,e'} \frac{1}{2} I_{ee'} P(c_e = c_{e'}) = \frac{W}{K}$$

where $W = 1/2 \sum_{e,e'} I_{ee'}$ is the sum of weights of all edges in the conflict graph. In the single channel scenario, the interference is $W$. That means, with $K$ channels, this simple heuristic can reduce the interference by a factor of $K$ in expectation.

## 5.2 Local Greedy algorithm

For an edge $e \in E$, if we switch $c_e$ to the channel that is least used among the edges it conflicts with, that is, if we choose

$$c_e \in \underset{c}{\text{argmin}} \sum_{e' \in E} I_{ee'} \mathbb{1}(c = c_{e'}), \tag{16}$$

then the total interference can only decrease or stay put. Scanning through the list of edges reveals at least one such edge where a channel switch reduces interference or we must be at a local optimum. Note that the total interference only decreases during the course of the algorithm. Therefore, if the conflict weights $I_{ee'}$ are integers bounded by a constant, then the algorithm converges to a local optimum in polynomial time. A local optimum $\widehat{c}$ satisfies $\hat{I} \leq \frac{W}{K}$ because

$$\hat{I} = \frac{1}{2}\sum_{e}\sum_{e' \in E} I_{ee'}\mathbb{1}(\widehat{c}_e = \widehat{c}_{e'}) \leq \frac{1}{2}\sum_{e}\frac{1}{K}\sum_{e' \in E} I_{ee'} = \frac{W}{K}.$$

The inequality above is due to the fact that at a local minimum, the minimum interference involving $e \in E$ is $\leq$ average interference involving $e$.

### 5.3 Simulated Annealing

The algorithm is similar to the local greedy algorithm above. The difference is that, here we do not always choose a channel that minimizes the interference; we sometimes choose a locally suboptimal channel in order to escape out of local optima. Let $T(t)$ be a positive sequence that decreases such that $T(t) \to 0$, but not too fast – it should satisfy $\sum_{t=1}^{\infty} e^{-b/T(t)} = \infty$ for a constant $b$ (Aarts and Korst, 1989). For example, the sequence $T(t) = b/(1 + \ln(t))$ satisfies these constraints. The key step is to sample an edge $e$ and a channel $k$, calculate $\Delta(t)$, the change in interference if $c_e$ is set to $k$ and set

$$c_e = k \text{ with probability } \min\left\{1, e^{-\Delta(t)/T(t)}\right\}$$

at the $t$th iteration. In practice, the sequence $T(t) = b/(1 + \ln(t))$ decreases too slowly to make the procedure computationally tractable. We use a $T(t)$ that decreases at a rate of $1/t$.

### 5.4 SDP Relaxation and Randomized rounding

On a graph $G(V, E, w)$, inspired by the seminal work of Goemans and Williamson (1995), Frieze and Jerrum (1997) relax the max-k-cut problem

$$\frac{1}{2}\max_{c \in [K]^{|V|}} \sum_{(i,j) \in E} w_{ij}\mathbb{1}(c_i \neq c_j)$$

to the following SDP:

$$\begin{aligned} \max_{X \succeq 0} \quad & \frac{K-1}{2K}\sum_{(i,j) \in E} w_{ij}(1 - X_{ij}) \\ \text{subject to} \quad & X_{ii} = 1 \text{ for } i \in [|V|] \\ & X_{ij} \geq -1/(K-1) \text{ for } i \neq j. \end{aligned} \tag{17}$$

Once the above SDP is solved, the solution $\widehat{X}$ can be used to get channels as follows. Draw random vectors $z_1, \ldots, z_K \in \mathbb{R}^{|V|}$ where the entries are i.i.d standard normal distributed. Then simply set

$$\widehat{c}_i = \underset{j}{\operatorname{argmax}}\langle \widehat{X}_i, z_j \rangle.$$

Frieze and Jerrum (1997) show that the expected max-k-cut objective at $\widehat{c}$ is at least $\alpha_K$ times the optimal objective, where $\alpha_K > 1 - \frac{1}{K}$. This inequality is loose for small $K$; they show $\alpha_2 \geq$

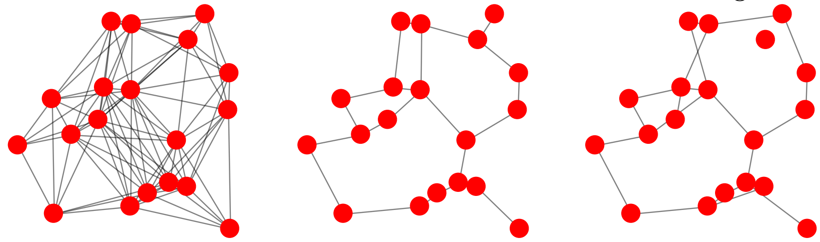


Figure 2: *A network with $n = 20$ nodes over a 200m x 200m region. Edges in the left graph are between nodes within interference range. The middle graph shows the edges selected by minimizing the effective resistance subject to the sector constraint. The right graph shows the topology selected by nearest neighbors algorithm. Note the isolated node in the top right region.*

$0.878, \alpha_3 \geq 0.800, \alpha_4 \geq 0.850$. This means, $W - \mathbb{E}\widehat{I} \geq \alpha_K(W - I^*)$, where $\widehat{I}$ is the interference resulting from the channel assignment $\widehat{c}$ and $I^*$ is the minimum interference. In other words,

$$\mathbb{E}\widehat{I} \leq (1 - \alpha_K)W + \alpha_K I^*.$$

Observe that when $I^*$ is small, the upper bound is smaller than $W/K$ obtained from random channel assignment or the local greedy method. Further, the optimal value $\widehat{W}_{\text{sdp}}$ of the SDP relaxation (17) satisfies $\widehat{W}_{\text{sdp}} \geq W^*$ where $W^*$ is the optimal max-k-cut value. Noting that $I^* = W - W^*$ provides the following lower bound on $I^* : I^* \geq W - \hat{W}_{\text{sdp}}$.

# 6 Experiments

Networks are generated as follows. Router locations are sampled uniformly from a square planar region. Orientations of the routers are sampled uniformly from $[0, 2\pi)$. The following hardware settings determine the pairwise RSSI. Radios on the routers transmit with a power of 11dB. Noise floor is $-85$dB and sensitivity is at $-79$dB. The directional antenna gain is $10 \log K$dB. The transmit power and antenna gain are chosen to conform with Wi-Fi Effective Isotropic Radiated Power (EIRP) regulatory limits. Noise floor and sensitivity are dictated by hardware and these numbers represent a typical consumer device. For two nodes $i, j$, the power received by one node due to the other is calculated using the Friis tranmission formula

$$P_j = P_i \left(\frac{\lambda}{4\pi d}\right)^\beta G_i G_j$$

where $\beta$ is the path loss exponent, $G_i, G_j$ are antenna gains and $d$ is the distance between $i$ and $j$. In typical environments path loss exponents vary from 2 to 4 (2 is for free space). We assume $\beta = 3$ which is pretty standard for an indoor environment with rich scattering like an office. With these settings, the transmission range is 71m and the interference range is 113m.

The capacity of a link $e$, given its SNR is modeled as $\text{cap}(e) = \min(90, 15 + 6 \cdot (\text{SNR} - 7)_+))$Mb/s where $x_+ := \max(0, x)$. This is an approximate continuous model of the data rates specified in IEEE standard 802.11ac, assuming 40 MHz channels with SGI.

### 6.1 Results

We simulated 20 networks with $n = 20$ nodes each in a 200m $\times$ 200m region and ran the following algorithms for topology control:

- ER: minimize effective resistance by solving the integer SDP (8)
- L2: maximize second smallest eigenvalue of the Laplacian by solving the integer SDP (10)
- LG: maximize second smallest eigenvalue of the Laplacian using the greedy heuristic adapted from Ghosh and Boyd (2006), as described in Section 4.3
- MC: maximize the sum of capacities of the links selected by solving the ILP (12).
- NN: the nearest neighbor heuristic described in Section 4.1.

For channel assignment, we consider three methods here: the ILP in (13) solved using Gurobi, the local greedy algorithm (GR) from Section 5.2 and simulated annealing (SA). Simulated annealing is implemented with an annealing schedule $T(t) = 2e7\,\widehat{W}(t)/W_{tot} \cdot 1/t$ where $W_{tot}$ is the sum of all weights in the conflict graph and $\widehat{W}(t)$ is the max-k-cut value at iteration $t$. We do not compare the SDP relaxation method because, as we later find out, it does not work as well as others.

| | $\alpha$ | | | $\overline{\alpha}$ | | | time (s) | | |
|---|---|---|---|---|---|---|---|---|---|
| | ILP | GR | SA | ILP | GR | SA | ILP | GR | SA |
| ER | **2.22** | 2.19 | 2.20 | **2.37** | 2.34 | 2.34 | 21 | 17 | 17 |
| L2 | 2.14 | 2.03 | 2.09 | 2.28 | 2.18 | 2.22 | 30 | 27 | 27 |
| LG | 1.04 | 1.07 | 1.07 | 1.30 | 1.34 | 1.34 | 2.9 | 0.027 | 0.04 |
| MC | 2.18 | 2.17 | 2.17 | 2.33 | 2.33 | 2.32 | 2.2 | 0.055 | 0.073 |
| NN | 0.91 | 0.90 | 0.92 | 1.25 | 1.24 | 1.25 | 2.2 | 0.014 | 0.028 |

Table 1: *$\alpha$, $\overline{\alpha}$ and time taken to solve, averaged over 20 simulated networks with $n = 20$ nodes and 10 sets of $n/2$ demand flows per network. The higher the $\alpha$ or $\overline{\alpha}$, the better the solution is.*

| A | $\alpha_{\text{ER+ILP}} - \alpha_{\text{A+ILP}}$ | $\alpha_{\text{A+ILP}} - \alpha_{\text{A+GR}}$ | $\alpha_{\text{A+ILP}} - \alpha_{\text{A+SA}}$ |
|---|---|---|---|
| ER | 0 | -0.00 $\pm$ 0.12 | 0.02 $\pm$ 0.09 |
| L2 | -0.03 $\pm$ 0.08 | 0.09 $\pm$ 0.10 | 0.04 $\pm$ 0.09 |
| LG | 0.43 $\pm$ 0.21 | 0.00 $\pm$ 0.03 | -0.01 $\pm$ 0.05 |
| MC | -0.00 $\pm$ 0.10 | 0.04 $\pm$ 0.06 | 0.05 $\pm$ 0.08 |
| NN | 0.62 $\pm$ 0.22 | 0.01 $\pm$ 0.02 | 0.00 $\pm$ 0.01 |

Table 2: *Relative performance of the topology control and channel assignment algorithms on simulated networks with $n = 20$ nodes. The second column shows mean and standard deviation of the relative differences $(\alpha_{ER+ILP} - \alpha_{A+ILP})/\alpha_{best}$ taken over the 20 simulations, for the topology algorithms A. Smaller value indicates better performance. The third and fourth columns shows same quantities for the differences due to the choice of channel assignment algorithms.*

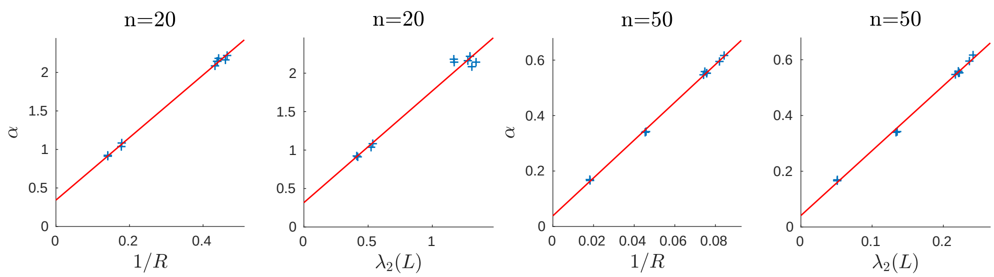


Figure 3: *$\alpha$ averaged over all simulations is plotted against the mean $1/R$ and $\lambda_2(L)$ over all simulations when the edges are weighted with capacities after taking interference into account. Each scatter point corresponds to an algorithm. The left two plots are for networks with $n = 20$ nodes and the right ones are for $n = 50$ nodes. Correlation coefficients are 0.9990, 0.9879, 0.9994, 0.9988.*

Sample $n_d = \min\{25, n/2\}$ distinct pairs of nodes $(s_i, t_i)$ and form unit demands $d_i = 1$ between $s_i$ and $t_i$ for $i \in [n_d]$. Minimum flow $\alpha$ for the simultaneous demands is obtained by solving the linear program (1). The average flow $\overline{\alpha}$ is computed as described in Section 2.3. This procedure is repeated 10 times and $\alpha$ and $\overline{\alpha}$ are averaged. The results are given in Table 1.

We observe that $\alpha$ and the average flow $\overline{\alpha}$ behave similarly. So we focus on $\alpha$ with the understanding that the remarks about $\alpha$ apply to $\overline{\alpha}$ as well. The effective resistance based method ER+ILP gives the best $\alpha$. ER, L2 and MC deliver a larger $\alpha$ than LG and NN. $\alpha$ does not vary significantly between the two channel assignment algorithms for a given topology.

Table 2 compares the algorithms in more detail. We fix the channel assignment algorithm to ILP and compare the performance of the other topology control algorithms to ER. In each simulation, we normalize the $\alpha$'s obtained by all algorithms w.r.t. the largest $\alpha$. The second column shows by how much ER is better. The difference and standard deviations strengthen our earlier observation that ER, L2 and MC are better than LG and NN. Similar behavior is observed when we switch to local greedy channel assignment. The third and fourth columns show that the performance does not degrade much if we use GR or SA channel assignment algorithms instead of exactly minimizing the total interference by ILP in these cases. Results are similar for larger networks of size $n = 50$ simulated in the same fashion; see Appendix B.

Computationally, ER and L2 based methods are at least an order of magnitude slower because the SDP problems that need to be solved at the branch-and-bound nodes are expensive. We in fact limit the number of branch-and-bound iterations to 500. Also we set a time limit of 600 seconds to solve the interference minimization ILP. For $n = 20$ nodes, over 20 simulations, the duality gap remained $> 1\%$ in one case for ER and 5 cases for L2. We warm-start ER and L2 branch-and-bound algorithms with the solution from MC. Without warm-starting, they take much longer.

**Variation of $\alpha$ with $R$ and $\lambda_2(L)$**

In Figure 3, we plot mean $\alpha$ versus mean inverse effective resistance and mean $\lambda_2(L)$ for all the algorithms considered. For both network sizes, we clearly see a high correlation between $\alpha$ and the two spectral properties. If the traffic demands and capacity models considered are applicable, then we seem to be justified in using ER and L2.

**Larger networks**

ER and L2 are computationally impractical for larger networks ($n > 100$) if we solve the integer SDPs using branch-and-bound and off the shelf SDP solvers. We use MC in such cases as we have seen that it gives good topologies for small networks. For channel assignment also, exact solving of (13) does not scale well. We compare GR, SA and the SDP relaxation methods for large graphs. SDP relaxation of max-k-cut (17) is solved using SDPNAL+ (Yang et al., 2014). Simulated annealing is implemented with an annealing schedule $T(t) = 2e7\,\hat{W}(t)/W_{tot} \cdot 1/t$ where $W_{tot}$ is the sum of all weights in the conflict graph and $\hat{W}(t)$ is the max-k-cut value at iteration $t$.

We generate 6 networks of increasing size from $n = 50$ to $n = 2000$ nodes. See Table 3. Simulated annealing takes a few minutes for the largest network we considered with $n = 2000$ nodes while SDP relaxation takes more than two hours. Further, even though SDP method has a good theoretical bound, its minimization performance is not as good as the simpler GR and SA methods. The relationship between $\alpha$ and interference is not clear from here, see Appendix C.

| Graph | | | | $\alpha$ | | | Interference (% of $W_{tot}$) | | | | time (s) | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$ | $\overline{m}$ | $m$ | M | GR | SA | SDP | GR | SA | SDP | BND | GR | SA | SDP |
| 50 | 111 | 63 | 618 | 0.24 | 0.18 | 0.18 | 4.3 | 3.4 | 7.3 | 3.0 | 0.109 | 0.435 | 7.43 |
| 105 | 261 | 140 | 1756 | 0.79 | 0.59 | 0.72 | 7.9 | 6.7 | 13.3 | 4.9 | 0.341 | 0.601 | 9.36 |
| 219 | 602 | 305 | 4282 | 1.31 | 1.34 | 1.23 | 7.7 | 7.2 | 13.6 | 4.9 | 1.33 | 10.8 | 32.8 |
| 457 | 1320 | 652 | 10127 | 1.22 | 1.33 | 1.23 | 9.2 | 7.9 | 14.5 | 5.6 | 5.6 | 25.9 | 133 |
| 956 | 2931 | 1365 | 22777 | 0.42 | 0.54 | 0.41 | 9.5 | 8.5 | 15.5 | 5.7 | 23.1 | 139 | 747 |
| 2000 | 6151 | 2909 | 50128 | 1.87 | 1.89 | 1.80 | 9.7 | 8.5 | 15.5 | 5.9 | 38.6 | 272 | 8778 |

Table 3: *Performance of local Greedy(GR), Simulated Annealing(SA), SDP relaxation followed by randomized rounding on networks of increasing sizes, with $K = 4$. $M$ denotes the number of the edge conflicts. The BND column shows the lower bound on interference obtained by solving the relaxed SDP. $W_{tot}$ is the sum of all weights in the conflict graph.*

## 7 Discussion and Future Work

Even though graph spectral properties such as effective resistance and Fiedler value are well-studied in graph theory, to our knowledge, they are not used in the wireless mesh network community. Under the traffic demand model, capacity under interference model and routing assumptions, we have empirically seen that inverse effective resistance and the Fiedler value are highly correlated with the eventual quality of the network. Therefore looking for the subgraph with high values for these spectral properties is justified in this setting. For large networks, when optimizing the above quantities is computationally impractical, one may use the faster maximum capacity heuristic that we proposed. However, the relation between interference and $\alpha$ is not clear from the experiments. Running the simulations on a discrete-event simulator such as NS-3 will help us verify these ideas further. If estimates of link-wise loads are available (say, via metrics collected during network operation), they can be incorporated into edge weights during both topology control and channel assignment. Ideas for joint optimization of topology control and channel assignment are discussed in Appendix A.

## Acknowledgements

I would like to thank all my advisors for their guidance and valuable feedback. Andrew Price and Varada Gopalakrishnan introduced the problem to me and taught about wireless mesh networks. Javier Pena helped me improve key problem formulations of topology control and channel assignment problems. Ryan Tibshirani guided me throughout the project in developing the methodology.

## References

Aarts, E. and Korst, J. (1989). *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. John Wiley & Sons, Inc., New York, NY, USA.

Akyildiz, I. F., Wang, X., and Wang, W. (2005). Wireless mesh networks: A survey. *Comput. Netw. ISDN Syst.*, 47(4):445–487.

Chung, F. R. K. (1997). *Spectral Graph Theory*. American Mathematical Society.

Fiedler, M. (1973). Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(2):298–305.

Fouss, F., Pirotte, A., and Saerens, M. (2005). A novel way of computing similarities between nodes of a graph, with application to collaborative recommendation. In *Proceedings of the 2005 IEEE/WIC/ACM International Conference on Web Intelligence*, WI '05, pages 550–556, Washington, DC, USA. IEEE Computer Society.

Frieze, A. and Jerrum, M. (1997). Improved approximation algorithms for maxk-cut and max bisection. *Algorithmica*, 18(1):67–81.

Ghosh, A. and Boyd, S. (2006). Growing well-connected graphs. In *Decision and Control, 2006 45th IEEE Conference on*, pages 6605–6611. IEEE.

Ghosh, A., Boyd, S., and Saberi, A. (2008). Minimizing effective resistance of a graph. *SIAM Rev.*, 50(1):37–66.

Goemans, M. X. and Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42(6):1115–1145.

Gu, Z., Rothberg, E., and Bixby, R. (2010). Gurobi 4.0.2. software.

IEEE (2013). IEEE Std 802.11ac-2013 - 22.5 parameters for vht-mcss.

Liu, Q., Jia, X., and Zhou, Y. (2011). Topology control for multi-channel multi-radio wireless mesh networks using directional antennas. *Wirel. Netw.*, 17(1):41–51.

Löfberg, J. (2004). Yalmip : A toolbox for modeling and optimization in matlab. In *In Proceedings of the CACSD Conference*, Taipei, Taiwan.

Pathak, P. H. and Dutta, R. (2011). A survey of network design problems and joint design approaches in wireless mesh networks. *IEEE Communications Surveys & Tutorials*, 13(3).

Shi, J. and Malik, J. (1997). Normalized cuts and image segmentation. In *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, CVPR '97, pages 731–, Washington, DC, USA. IEEE Computer Society.

Spielman, D. A. (2007). Spectral graph theory and its applications. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, pages 29–38, Washington, DC, USA. IEEE Computer Society.

Spielman, D. A. and Srivastava, N. (2011). Graph sparsification by effective resistances. *SIAM J. Comput.*, 40(6):1913–1926.

Sridhar, S., Guo, J., and Jha, S. (2009). Channel assignment in multi-radio wireless mesh networks: A graph-theoretic approach. In *Proceedings of the First International Conference on COMmunication Systems And NETworks*, COMSNETS'09, pages 180–189. IEEE Press.

Subramanian, A. P., Gupta, H., Das, S. R., and Cao, J. (2008). Minimum interference channel assignment in multiradio wireless mesh networks. *IEEE Trans. Mob. Comput.*, 7(12):1459–1473.

Toh, K. C., Todd, M., and Ttnc, R. H. (1999). Sdpt3 – a matlab software package for semidefinite programming. *OPTIMIZATION METHODS AND SOFTWARE*, 11:545–581.

von Luxburg, U., Radl, A., and Hein, M. (2014). Hitting and commute times in large random neighborhood graphs. *Journal of Machine Learning Research*, 15:1751–1798.

Wong, W. and Gary Chan, S.-H. (2014). Topology optimization for wireless mesh with directional antennas. In *2014 IEEE International Conference on Communications, ICC 2014*, pages 2791–2796.

Yang, L., Sun, D., and Toh, K.-C. (2014). Sdpnal+: A majorized semismooth newton-cg augmented lagrangian method for semidefinite programming with nonnegative constraints.

Yen, L., Fouss, F., Decaestecker, C., Francq, P., and Saerens, M. (2007). Graph nodes clustering based on the commute-time kernel. In *Proceedings of the 11th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'07, pages 1037–1045, Berlin, Heidelberg. Springer-Verlag.

Zhou, D. and Schölkopf, B. (2004). *Learning from Labeled and Unlabeled Data Using Random Walks*, pages 237–244. Springer Berlin Heidelberg, Berlin, Heidelberg.

# A Joint Optimization

In this section, we propose a method to jointly optimize topology control and channel assignment. Let $g(z, c)$ be a function which takes the set of selected edges $z \in \{0, 1\}^{\overline{m}}$ and a channel assignment $c$ and gives the effective capacities for the selected edges, considering interference and a traffic pattern. For example, in section 2.3, we assumed that all edges in $E$ are active and the capacity of a link is assumed to be the capacity without interference (but with a standard background noise) divided by the number of links it interferes with. With the capacities given by $g$, the desired properties of the network such as good connectivity and high quality of service may be characterized in terms of graph spectral properties such as effective resistance and Fiedler value. In other words, we would like to minimize a spectral property $\sigma$ of the graph:

$$\underset{z\in\{0,1\}^{\overline{m}},\ c\in[K]^{\overline{m}}}{\text{minimize}} \quad \sigma(g(z, c)) \quad \text{subject to} \quad A_s z \leq r. \tag{18}$$

We may employ an alternating minimization strategy to solve this. For concreteness, let $\sigma$ be the effective resistance. For the model described in Section 2.3, we have

$$g_e(z, c) = \frac{z_e w_e}{\text{num. of edges conflicting with e}} = \frac{z_e w_e}{\sum_{e'\in\overline{E}} I_{ee'} \mathbb{1}(c_e = c_{e'}) z_{e'}}$$

where $w_e$ is the capacity of edge $e$ at base noise level without interference from other edges. Given $c$, defining $\widetilde{I}_{ee'} = I_{ee'}\mathbb{1}(c_e = c_{e'})$, we have $g_e(z, c) = z_e w_e/(\widetilde{I}z)_e$, or compactly $g(z, c) = (z \cdot w)./(\tilde{I}z)$ where $\cdot$ denotes element-wise operations. Given $c$, we need to solve

$$\underset{z}{\text{minimize}} \quad R\left((z \cdot w)./(\tilde{I}z)\right) \quad \text{subject to} \quad A_s z \leq r. \tag{19}$$

The objective is still a convex function of $z$. We approximate the objective with $R((z \cdot w)./\tilde{I}1))$. In other words, to count $I(e)$ = the number of edges conflicting with $e$, we assume all the edges are selected. This is the ER topology control algorithm with edge weights reduced by a factor of $I(e)$.

Given $z$, that is, a connected subgraph, channels are assigned to the subgraph's edges by minimizing its effective resistance where the edge weights are $w_e/I(e)$ with $I(e)$ being the number of edges conflicting with $e$ under the channel assignment.

# B Results for $n = 50$

We show the results for larger networks with $n = 50$ nodes generated over a 300m × 300m planar area. See Table 4 and Table 5. ER, L2 and MC are again better than the other two methods by a significant margin. There is not much difference between the performances ER, L2 and MC. Again, the MC+GR combination seems to be very favorable: it is computationally fast and $\alpha$ is the best.

On a closer look though, ER and L2 methods do not converge in the budget of 500 branch-and-bound iterations. For ER, the duality gap (difference of upper and lower bounds divided by their sum) is greater than 1% in all 10 simulations and the average duality gap is 3.19%. For L2 the duality gap is larger with a minimum of 5%, maximum of 28% and an average of 14.7%. Given more iterations, the duality gap should go down for both ER and L2. MC is used to initialize ER and L2, and that explains why ER, L2 and MC perform very similarly.

| | $\alpha$ | | | $\overline{\alpha}$ | | | time (s) | | |
|---|---|---|---|---|---|---|---|---|---|
| | ILP | GR | SA | ILP | GR | SA | ILP | GR | SA |
| ER | 0.99 | 1.01 | 1.04 | 1.01 | 1.03 | 1.06 | 1491.7 | 891.7 | 891.9 |
| L2 | 0.99 | 1.03 | 1.03 | 1.01 | 1.05 | 1.05 | 702.7 | 102.4 | 102.5 |
| LG | 0.56 | 0.56 | 0.58 | 0.61 | 0.60 | 0.62 | 600.1 | 0.1 | 0.2 |
| MC | 0.99 | **1.07** | 1.01 | 1.01 | **1.09** | 1.03 | 600.1 | 0.1 | 0.3 |
| NN | 0.25 | 0.24 | 0.24 | 0.36 | 0.35 | 0.35 | 600.1 | 0.1 | 0.2 |

Table 4: *$\alpha$, $\overline{\alpha}$ and time taken to solve, averaged over 10 simulated networks with $n = 50$ nodes and 10 sets of $n/2$ demand flows per network. The higher the $\alpha$ or $\overline{\alpha}$, the better the solution is.*

| A | $\alpha_{\text{ER+ILP}} - \alpha_{\text{A+ILP}}$ | $\alpha_{\text{A+ILP}} - \alpha_{\text{A+GR}}$ | $\alpha_{\text{A+ILP}} - \alpha_{\text{A+SA}}$ |
|---|---|---|---|
| ER | 0 | -0.02 $\pm$ 0.08 | -0.04 $\pm$ 0.13 |
| L2 | 0.00 $\pm$ 0.00 | -0.01 $\pm$ 0.12 | -0.03 $\pm$ 0.09 |
| LG | 0.37 $\pm$ 0.23 | 0.01 $\pm$ 0.04 | -0.01 $\pm$ 0.03 |
| MC | 0.00 $\pm$ 0.00 | -0.07 $\pm$ 0.10 | -0.01 $\pm$ 0.11 |
| NN | 0.64 $\pm$ 0.23 | 0.01 $\pm$ 0.03 | 0.01 $\pm$ 0.03 |

Table 5: *Relative performance of the topology control and channel assignment algorithms on simulated networks with $n = 50$ nodes. The second column shows mean and standard deviation of the relative differences $(\alpha_{ER+ILP} - \alpha_{A+ILP})/\alpha_{best}$ taken over the 10 simulations, for the topology algorithms A. Smaller value indicates better performance. The third and fourth columns shows same quantities for the differences due to the choice of channel assignment algorithms.*

## C Variation of $\alpha$ with interference

We examine the variation of $\alpha$ with interference in this section. Simulate 20 networks each of size $n = 50, 55, 60, \ldots, 100$. After topology selection by MC, we run GR and SA channel assignment algorithms. Then we plot the difference in average $\alpha$ between GR and SA versus difference in average interference (divided by $W_{tot}$) between GR and SA. See Figure 4. The correlation coefficient is $-0.52$ but the p-value is 0.10 indicating that this relationship requires further investigation.

## More Future work

The MC algorithm is fast and is a good alternative to ER and L2 in our settings. But if the graph has more than say a thousand nodes, the number of flow constraints grows to millions. It will be interesting to find algorithms that add only $O(\overline{m})$ constraints perhaps incrementally.

Effective resistance $R_{ij}$ may be approximated by $1/d_i + 1/d_j$ for the purpose of optimization as it is known to be a good approximation in large graphs under certain conditions (von Luxburg et al., 2014). This may make the optimization more tractable.

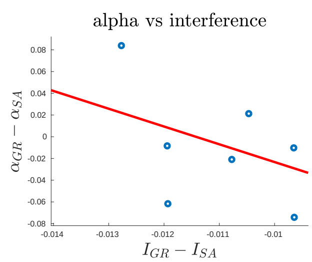


Figure 4: Variation of (difference in) $\alpha$ with (difference in) interference for networks of sizes $n = 50, 55, \ldots, 100$ over 20 simulations.