Link Scheduling and Channel Assignment with a Graph Spectral Clustering Approach

[†]Zhifeng He, [†]Shiwen Mao, [‡]Sastry Kompella, and [§]Ananthram Swami
[†]Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201 USA
[‡]Information Technology Division, Naval Research Laboratory, Washington, DC 20375 USA
[§]Army Research Laboratory, Adelphi, MD 20783 USA
Email: zzh0008@tigermail.auburn.edu, smao@ieee.org, sk@ieee.org, a.swami@ieee.org

***Abstract*—We tackle the challenging problem of link scheduling and channel assignment in multi-channel, multi-hop wireless networks, aiming to achieve high network throughput. We adopt the signed graph spectral clustering algorithm to solve this problem heuristically, by jointly considering channel gain and mutual interference. The basic idea is to cluster links with smaller mutual interference and allocate channels with a higher gain. Simulation results demonstrate the advantages of our proposed algorithm over a benchmark scheme in terms of minimum link throughput and sum throughput.**

***Index Terms*—Link scheduling; channel assignment; spectral graph theory; clustering.**

I. Introduction and Related Work

Due to the unprecedented growth of mobile data in recent years, improving wireless network capacity has become an urgent issue. There has been lot of work on analyzing wireless network capacity. In a seminal work [1], Gupta and Kumar derived the lower bound and upper bound for the wireless network capacity. Specifically, they derived bounds for random networks under a Protocol Model and a Physical model of wireless link interference. The authors of [2] and [3] also investigated analytic models for wireless network capacity, based on a simple Point-to-Point coding model, where a transmission involves only one transmitter and one receiver. In [4], the authors considered a scenario with multiple transmitters and receivers, and derived lower bounds on the wireless network throughput capacity under this scenario.

Improving network capacity with effective scheduling and resource allocation has attracted intense attention from researchers since the birth of wireless networks. The authors of [5] proposed to use Maximum RSS Opportunistic Routing, carrier sensing and probe link scheduling algorithms, so that the network capacity can be improved via optimal power. The authors in [6] studied the problem of optimal scheduling for concurrent transmissions from multiple users to maximize the overall capacity of all users. The formulated problem was shown to be NP-hard and the authors proposed a heuristic algorithm with polynomial time complexity. The basic idea of the proposed heuristic algorithm is to iteratively add a good link to the current link schedule at each iteration, so that the current network performance can be continuously improved. A similar problem is studied in [7], where the authors proposed a graph coloring scheme to color the transmission links sequentially, so that the SINR at each link was kept above a predefined threshold and links of the same color could transmit on the same channel concurrently.

Although great enhancements have been achieved and useful insights have been obtained, a major drawback of these existing solutions is that they do not consider frequency diversity at different links. Channels are treated as having the same capacity for different links in these works. Therefore channels are randomly allocated to links. However, in a real wireless network environment, a given channel may have different channel gains for different links due to varying path loss or shadow fading, and therefore allocating a channel to different links may result in different network capacities. Therefore, to improve the network throughput as much as possible, we need to consider the gain of a channel varying across different links.

In this paper, we present a spectral graph theory approach to link scheduling and channel assignment in multi-channel multi-link wireless networks. The main contributions of our proposed algorithm include the following:

1) We propose a novel approach of applying spectral graph clustering algorithm to link scheduling and channel assignment, aiming to achieve high wireless network capacity.
2) We design a similarity metric to evaluate the similarity between two link−channel tuples and construct the corresponding adjacency matrix for the purpose of clustering.
3) We study the case where each link can access only one channel as well as the case where each link can access multiple channels, and provide effective solutions to both cases.

The efficacy of the proposed approach is evaluated via simulations, and considerable margins over a benchmark scheme are demonstrated.

In the rest of this paper, we present our system model and problem statement in Section II. The spectral clustering algorithm is applied to the formulated problem in Section III. Section IV evaluates the performance of the proposed approach by comparison with a benchmark scheme. Section V concludes the paper.

II. System Model and Problem Statement

A. System Model

We consider a network consisting of M pairs of nodes (i.e., links) and N orthogonal channels, where each pair of nodes consists of a transmitter and a receiver. The objective is to maximize the sum throughput across all pairs. Denote the bandwidth of channel j by B_j, the transmit power by P_T, and the interference that link i' causes to link i, when link i uses channel j and link i' use channel j', by $\mathcal{I}_{ij,i'j'}$. The gain of channel j on link i is g_{ij} and the noise power is N_0. The throughput maximization problem can be formulated as

$$\text{max:} \sum_{i=1}^{M} B_j \log\left(1+\frac{P_T\sum_{j=1}^{N} y_{ij}g_{ij}}{N_0+\sum_{i'\neq i}^{M}\sum_{j'=1}^{N} y_{i'j'}\mathcal{I}_{ij,i'j'}}\right) \quad (1)$$

$$\text{s.t.} \sum_{j=1}^{N} y_{ij} = 1,\ \forall\, 1\leq i\leq M. \quad (2)$$

$$y_{ij} = \begin{cases} 1, \text{if link } i \text{ transmits on channel } j \\ 0, \text{otherwise,} \end{cases} \quad \forall\, i,j. \quad (3)$$

Note that each link uses only one channel and we do not enforce that every channel be allocated in this problem.

This is a NonLinear Integer Programming (NLIP) problem which is known to be NP-hard in [6]. We propose to use the spectral clustering algorithm for signed graphs proposed in [10] to solve this problem with reduced complexity in a heuristic approach.

B. Spectral Clustering Algorithm

Our proposed solution employs a spectral clustering algorithm to assign channels to links to improve the sum throughput. A spectral clustering algorithm forms clusters using eigenvectors of matrices derived from data. The input to the algorithm is a adjacency matrix consisting of the similarity score of each pair of points in the dataset, and the output are the clusters and points within each cluster, where points within the same cluster are better connected to each other than with those in other clusters.

The basic idea of our proposed algorithm is as follows: if we denote (i,j) as a link$-$channel tuple, meaning channel j is allocated to link i and different links may share the same channel, then there will be totally $M\times N$ different tuples. If we cluster the $M\times N$ tuples into N clusters, where each cluster has M tuples, then each cluster will be a *feasible* channel allocation for the M links and N channels; in each cluster, each link is allocated exactly one channel (see later discussions). Next, we compare the performance in terms of sum throughput of the M feasible schemes to find the cluster, i.e., the feasible channel allocation, with the best performance. The authors in [8] proposed a K-means based spectral clustering algorithm, which can be used to form the N clusters.

C. Determine the Adjacency Matrix

Intuitively, if some links have small mutual interference, we want to allocate the same channel to these links, which may result in a higher throughput for these links. Moreover, the throughput of a link is also dependent on the channel gain. Therefore, we also hope that these links have a high channel gain on the channel that is allocated to them. If we let such link$-$channel tuples form a cluster, i.e., a feasible channel allocation scheme for all channels and links, then it is expected that the feasible scheme will yield a comparatively higher system throughput.

In the adjacency matrix, each element represents the similarity score of a pair of data points. If some link$-$channel tuples have small mutual interference and high channel gain on a channel, then we assign high similarity scores for these tuples. Since the spectral clustering algorithm clusters the data points having high similarity scores into a cluster, while the data points in different clusters have low similarity scores, the tuples that have high similarity scores are supposed to be in the same cluster, i.e., forming a feasible channel allocation.

Since in a feasible channel allocation, i.e., a cluster, each link should be allocated exactly one channel, different channel allocations for a link should be put in different clusters. We assign negative similarity scores to these tuples that represent a link is allocated with different channels, so that these tuples will not be put in the same cluster. Furthermore, if two different links use different channels, then they should not cause any interference to each other. Therefore, their similarity scores should be 0.

Based on the above discussions, we define our adjacency matrix $\mathbf{A}$ as in Fig. 1. In $\mathbf{A}$, a $*$ means that the value of the entry is greater than 0, and is given as

$$\mathbf{A}_{ij,i'j'} = \frac{g_{ij}}{\mathcal{I}_{ij,i'j'}}, \text{for } i\neq i', j=j'. \quad (4)$$

The general spectral clustering algorithm only deals with symmetric adjacency matrix. However, our adjacency matrix is not necessary symmetric, since $\frac{g_{ij}}{\mathcal{I}_{ij,i'j'}}$ is not necessarily equal to $\frac{g_{i'j'}}{\mathcal{I}_{i'j',ij}}$. To make the problem tractable, we further set $\mathbf{A}_{ij,i'j'} = \mathbf{A}_{i'j',ij} = \min\{\mathbf{A}_{ij,i'j'}, \mathbf{A}_{i'j',ij}\}$ in our adjacency matrix.

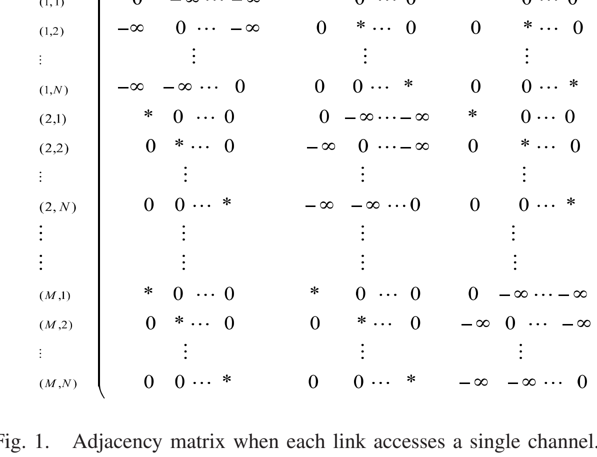

Fig. 1. Adjacency matrix when each link accesses a single channel.

Note that the adjacency matrix $\mathbf{A}$ has some negative entries. Specifically, for two tuples (i, j) and (i', j'), if $i = i', j \neq j'$, we set $\mathbf{A}_{ij,i'j'} = -\infty$. This is to make sure that tuples (i, j) and (i', j') will not be put in the same cluster, to prevent two different channel allocations for the same link being in the same cluster. In a feasible channel allocation, a link should not be allocated more than one channel, and a cluster should not contain negative elements.

III. Proposed Solution

A. Dealing with Negative Weights

The general spectral clustering algorithm only deals with graphs with non-negative weights, i.e., non-negative entries in the adjacency matrix. Since our adjacency matrix $\mathbf{A}$ has negative entries, we need a more general spectral clustering algorithm.

There is some literature dealing with signed graph spectral clustering algorithms. As proposed in [9], intuitively, a positive edge in a graph denotes proximity or similarity, and a negative edge denotes dissimilarity or distance. Negative edges are found in many types of networks: social networks may contain not just *friend* but also *foe* links, and connections between users and products may denote *like* and *dislike*. In other cases, negative edges may be explicitly introduced, as in the case of constrained clustering, where some pairs of points must or must not be in the same cluster. These problems can all be modeled with signed graphs.

Gallier in [10] proposed a new spectral clustering algorithm to deal with signed graphs with negative and positive weights. In the general spectral clustering algorithm, the degree matrix $\mathbf{D}$, with entries $\mathbf{D}_{ij}$, is defined as $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$ and $\mathbf{D}_{ij} = 0$ for $i \neq j$. The author define the *signed degree matrix* $\bar{\mathbf{D}}$, with entries $\bar{\mathbf{D}}_{ij}$, as $\bar{\mathbf{D}}_{ii} = \sum_j |\mathbf{A}_{ij}|$ and $\bar{\mathbf{D}}_{ij} = 0$ for $i \neq j$, in which absolute values of the weights of the adjacency matrix are used. The solution to signed graph spectral clustering is to replace the degree matrix $\mathbf{D}$ with the signed degree matrix $\bar{\mathbf{D}}$, and to replace the Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ by the signed Laplacian $\bar{\mathbf{L}} = \bar{\mathbf{D}} - \mathbf{A}$. This way, the signed Laplacian $\bar{\mathbf{L}}$ is always positive semidefinite, and it may be positive definite for unbalanced graphs. And then a generalization of the K-way normalized clustering algorithm is used to cluster the signed graphs using the signed Laplacian $\bar{\mathbf{L}}$, which is called K-Way Clustering Algorithm of Signed Graphs (termed as KSG algorithm in later discussions) in [10].

B. K-Way Clustering of Signed Graphs [10]

The graph clustering problem can be formalized as an optimization problem. Before we present the optimization problem, we first introduce three concepts: $volume$, $links$ and cut. For a subset of points $P \in V$, where V denotes the entire set of points in the graph, the $volume$ of P is defined as the weight of all edges adjacent to the points in P.

$$\text{vol}(P) = \sum_{v_i \in P} \sum_{j=1}^{m} \mathbf{A}_{ij}. \tag{5}$$

For two given subsets of points $P, Q \in V$, the $links$ between P and Q is defined as:

$$\text{links}(P, Q) = \sum_{v_i \in P} \sum_{v_j \in Q} \mathbf{A}_{ij}. \tag{6}$$

Denote by $\bar{P}$ the complement of P in V. Then links$(P, \bar{P})$ measures how many links escape from P. The cut of P can be expressed as

$$\text{cut}(P) = \text{links}(P, \bar{P}). \tag{7}$$

The optimization problem is formulated using the notion of cut. To partition V points into K clusters is equivalent finding a partition $(C_1, \dots, C_K)$ that minimizes the quantity cut$(C_1, \dots, C_K) = \frac{1}{2}\sum_{i=1}^{K} \text{cut}(C_i) = \frac{1}{2}\sum_{i=1}^{K} \text{links}(C_i, \bar{C}_i)$. To yield satisfactory results for $K > 2$ cases, the author proposes to normalize the cuts by dividing by some measure of each subset $\mathbf{A}_i$. The idea is to minimize the cost function Ncut$(C_1, \dots, C_K) = \sum_{i=1}^{K} \frac{\text{links}(C_i, \bar{C}_i)}{\text{vol}(C_i)} = \sum_{i=1}^{K} \frac{\text{cut}(C_i, \bar{C}_i)}{\text{vol}(C_i)}$.

The KSG algorithm using normalized cut can be formulated as [10]

$$\begin{aligned} \text{min: } \mu(X) &= \text{Ncut}(C_1, \dots, C_K) \\ &= \sum_{j=1}^{K} \frac{\text{cut}(C_j, \bar{C}_j)}{\text{vol}(C_j)} \\ &= \sum_{j=1}^{K} \frac{(X^j)^{\mathrm{T}} \bar{\mathbf{L}} X^j}{(X^j)^{\mathrm{T}} \bar{\mathbf{D}} X^j} \end{aligned} \tag{8}$$

$$\text{s.t. } (X^i)^{\mathrm{T}} \bar{\mathbf{D}} X^j = 0, \forall X \in \{0, 1\}, 1 \leq i \neq j \leq K, \tag{9}$$

Given a partition of V, where V represents the set of links to be clustered in our problem, into K clusters $(C_1, \dots, C_K)$, we represent the block j of this partition by a vector X^j (superscript T denotes transpose). $X^j = (x_1^j, \dots, x_N^j)$ where

$$x_i^j = \begin{cases} 1, & \text{if } v_i \in C_j \\ 0, & \text{otherwise,} \end{cases} \quad \text{for all } 1 \leq i \leq N, 1 \leq j \leq K. \tag{10}$$

$X \in \{0, 1\}$ means that X is binary valued.

It is NP-hard to find an optimal solution to this NLIP problem, and the author proposed to drop the integer constraints to solve a relaxed problem, to reduce the computational complexity. Specifically, the constraint $X \in \{0, 1\}$ is dropped from the original problem.

After obtaining the continuous, infeasible solution, the next step is to derive an integer solution from the continuous solution. The author used approximation techniques to obtain an integer solution in the neighborhood of the continuous solution. Denote the continuous solution as Z, and $\mathbf{\Lambda}$ as a $K \times K$ matrix with nonzero and pairwise orthogonal columns. We need to find a discrete matrix closest to one of these $Z\mathbf{\Lambda}$, which means that $\mathbf{\Lambda}$ is also an unknown matrix. To derive a discrete solution X closest (in a suitable sense) to $Z\mathbf{\Lambda}$, we can solve the following problem, where F denotes the Frobenius norm. This is also a hard problem because it is a nonlinear optimization problem involving two unknown matrices X and $Z\mathbf{\Lambda}$. To simplify the problem, the author proceeded by alternating steps during which $\|X - Z\mathbf{\Lambda}\|_F$ is minimized with respect to X while keeping $\mathbf{\Lambda}$ fixed, and steps during which $\|X - Z\mathbf{\Lambda}\|_F$ is minimized with respect to $\mathbf{\Lambda}$ while keeping X fixed. The proposed heuristic algorithm to solve this problem is complicated and the solution procedure is lengthy, interested readers can refer to [10] for more details.

C. Finding the Best Cluster

The signed graph spectral clustering algorithm partitions the $M \times N$ tuples into N clusters, and a link cannot use more than one channel in a cluster. Therefore, in each cluster, there are M tuples, where the link in each tuple is different from each other and each link has exactly one allocated channel. Thus each cluster represents a feasible channel assignment.

To obtain the channel assignment yielding the highest sum throughput, we compare the expected sum throughput of the N feasible assignments, and then choose the feasible assignment with the best performance.

D. Computational complexity Analysis

The computational complexity of our proposed algorithm mainly consists of three parts: the computation of matrix A, $\bar{D}$, and $\bar{L}$, and KSGalgorithm to solve the clustering problem, and finding the best cluster using the clustering result. From the structure of A, $\bar{D}$, and $\bar{L}$ we know that the overall computational complexity for all of A, $\bar{D}$, and $\bar{L}$ is $\mathcal{O}(NM^2)$, while finding the best cluster by comparing the expected sum throughput of the N feasible assignments is also $\mathcal{O}(NM^2)$ according to Eq.(1).

We refer the reader to [10] for an analysis of the computational complexity of the KSG algorithm.

E. The Case of Multi-channel Access

If a link can use multiple channels simultaneously with, say, some channel bonding and aggregation techniques, a link can be allocated more than one channel in a cluster (assuming up to N channels), then the problem can be formulated as

$$\text{max:} \sum_{i=1}^{M}\sum_{j=1}^{N} y_{ij} B_j \log\left(1 + \frac{P_T g_{ij}}{N_0 + \sum_{i' \neq i}^{M}\sum_{j'=1}^{N} y_{i'j'}\mathcal{I}_{ij,i'j'}}\right) \tag{11}$$

$$\text{s.t.} \sum_{j=1}^{N} y_{ij} \le N, \forall\, 1 \le i \le M. \tag{12}$$

Let's set the number of clusters to be also N. There will be no constraint on how many channels a link can be assigned in a cluster. Therefore the adjacency matrix can be expressed as in Fig. 2 below.

In this adjacency matrix, as in the previous case, $\mathbf{A}_{ij,i'j'}$ is not necessarily equal to $\mathbf{A}_{i'j',ij}$, since the mutual interference does not necessarily reciprocal. However, a difference when compared with the previous case is that, in this adjacency matrix, all elements are no less than zero, i.e., there is no negative element in this adjacency matrix. The authors in [11] proposed a spectral clustering algorithm to deal with asymmetric adjacency matrices with non-negative weights.

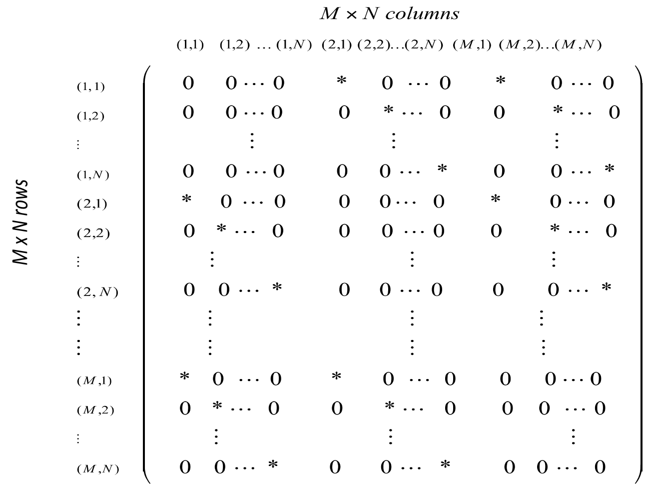

Fig. 2. Adjacency matrix when each link can access multiple channels.

Algorithm 1: Spectral Clustering Algorithm for the Case of Multi-channel Access [8], [11]

Input : a set of points $S = \{s_1, \dots, s_n\}$ in $\Re^l$
Output : k clusters

Form the adjacency matrix $\mathbf{A} \in \Re^{n\times n}$ with elements $\mathbf{A}_{ij}$;
Define a diagonal matrix $\mathbf{D}$ whose (i,i)-element is the sum of $\mathbf{A}$'s i-th row, and compute $\mathbf{L}_{AS} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{A})$;
Find $x_1, x_2, ..., x_k$, the k largest eigenvectors of $\mathbf{L}_{AS}$, and form matrix $\mathbf{X} = [x_1 x_2 \dots x_k] \in \Re^{n\times k}$ by stacking the eigenvectors in columns, and matrix $\mathbf{Y}$ with $Y_{ij} = X_{ij}/\sqrt{\sum_j X_{ij}^2}$;
Treating each row of $\mathbf{Y}$ as a point in $\Re^k$, cluster them into k clusters with the K-means algorithm ;
Assign the original point s_i to cluster j if and only if row i of the matrix $\mathbf{Y}$ is assigned to cluster j.

The Laplacian matrix of the asymmetric adjacency matrix is defined as

$$\mathbf{L}_{AS} = \mathbf{D}^{-1}\mathbf{L} = \mathbf{D}^{-1}(\mathbf{D} - \mathbf{A}), \tag{13}$$

where $\mathbf{L}$ is the Laplacian matrix corresponding to the symmetric adjacency matrix. In [11], a spectral clustering algorithm is combined with the K-means algorithm to cluster the data points and the detailed procedure is described in Alg. 1.

IV. Simulation Results

In this section we present the simulation results of the signed graph spectral clustering algorithm on the channel allocation problem. The main simulation parameters and their values are presented in Table I. The error bars in the simulation figures represent the 95% confidence interval obtained by repeating each simulation 50 times with different random seeds. Here we choose the algorithm proposed in [7], which uses graph coloring to assign channels to communication links without violating the SINR constraint, as a benchmark scheme for performance comparison. To make our proposed scheme comparable to the benchmark scheme, we also adopt the concept of SINR threshold in our proposed scheme: if the

TABLE I
SIMULATION PARAMETERS AND VALUES

Parameter	*Value*	*Parameter*	*Value*
M	30	N	3
g_{ij}	$(0, 1)$	$\mathcal{I}_{ij,i'j'}$	$(0, 1)$
B	10 kb/s	N_0	0.1
P_T	1	γ	0.1

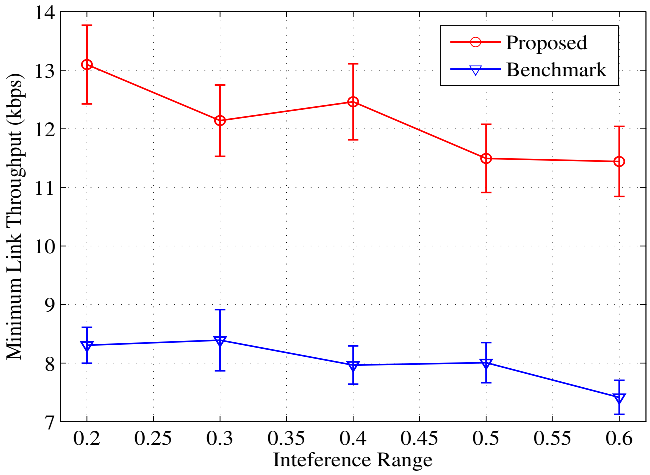

Fig. 3. Minimum link capacity among all links for single channel access.

SINR on a link with respect to an allocated channel is below the SINR threshold γ, then the throughput on the link is 0.

We choose throughput as the performance metric. In Fig. 3, we compare the minimum link throughput, i.e., the throughput of the link whose throughput is the lowest among all the links. After comparing the minimum link throughput of each cluster, i.e., each feasible channel allocation scheme, we choose the cluster that yields the largest minimum link throughput as the final channel allocation scheme. We can see from the figure that our proposed algorithm outperforms the benchmark scheme in terms of minimum link throughput. The main reason is that in the benchmark scheme, all channels are supposed to have the same channel gain and thus the same channel capacity for all links. Then a channel is randomly selected from the set of available channels as long as the SINR constraint is not violated for these links. However, our proposed algorithm not only considers reducing the interference among links sharing the same channel, but also the channel gain for the links, both are important factors to improving SINR. Thus a higher link throughput is achieved.

Next we compare the sum throughput of our proposed algorithm and the benchmark scheme in Fig. 4. For this experiment, we choose the cluster that has the highest sum throughput as the final channel allocation. Simulation results in the figure show that our proposed algorithm outperforms the benchmark scheme in terms of sum throughput by a considerable margin.

In Figs. 5 and 6, we present the minimum link throughput and the sum throughput for the multiple channel access case.

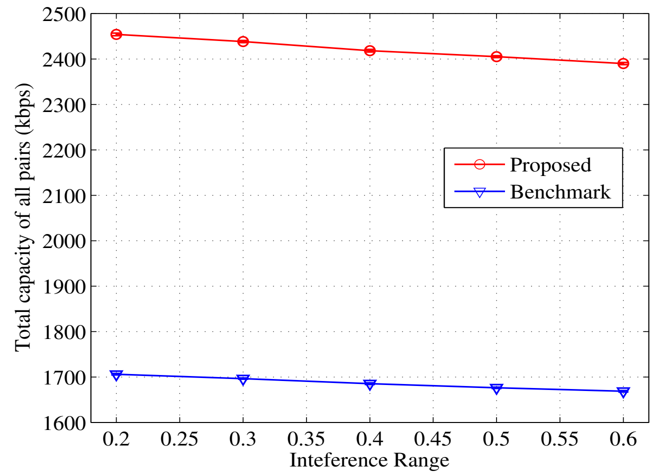

Fig. 4. System throughput of all links for single channel access.

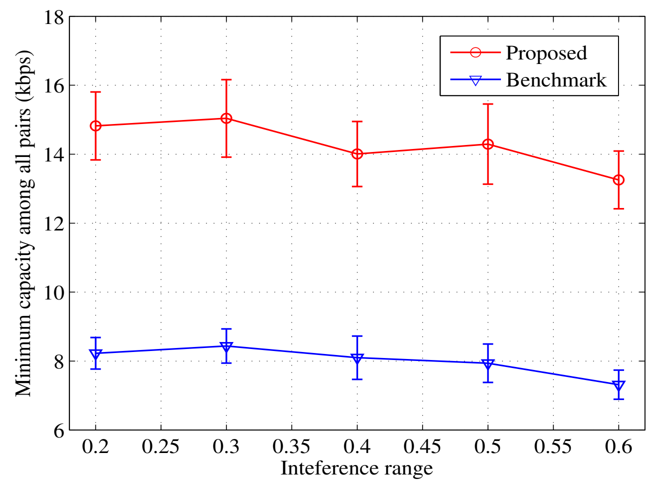

Fig. 5. Minimum link capacity among all links for multi-channel access.

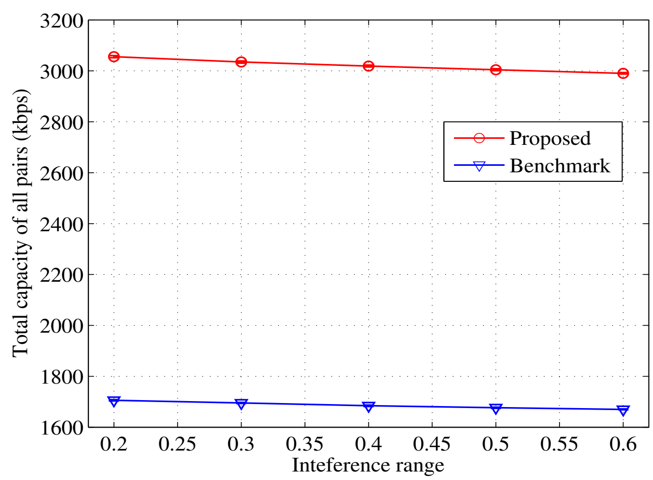

Fig. 6. System throughput of all links for multi-channel access.

We can see that the multiple channel access algorithm outperforms the single channel access algorithm in terms of both minimum link throughput and system throughput. The main reason is that a link is now allowed to access multiple channels at the same time in the multiple channel access algorithm. There is more flexibility in assigning the channels and thus the throughput performance is improved.

V. Conclusions

We studied the problem of channel assignment in multi-channel, multi-link wireless networks. We considered both channel gains and mutual interference and employed a signed graph clustering algorithm to solve this problem, where link−channel tuples were clustered into multiple clusters so that links with smaller mutual interference were clustered and the channel with higher channel gains were allocated. Simulation results showed that the proposed scheme outperforms a benchmark scheme in terms of minimum link capacity and sum throughput. We will investigate how to develop a distributed version of this approach in our future work.

Acknowledgment

This work is supported in part by the US National Science Foundation (NSF) under Grant CNS-1320664 and by the Wireless Engineering Research and Education Center (WEREC) at Auburn University.

References

[1] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol.46, no.2, pp.388–404, Mar. 2000.

[2] L. Xie and P. Kumar, "A network information theory for wireless communication: scaling laws and optimal operation," *IEEE Trans. Inf. Theory*, vol.50, no.5, pp.748–767, May 2004.

[3] M. Franceschetti, O. Dousse, D.N.C. Tse, and P. Thiran, "Closing the gap in the capacity of wireless networks via percolation theory," *IEEE Trans. Inf. Theory*, vol.53, no.3, pp.1009–1018, Mar. 2007.

[4] T. Zhang, K. Lu, S. Fu, Y. Qian, W. Liu, and J. Wang "Improving the capacity of large-scale wireless networks with network-assisted coding schemes," *IEEE Trans. Wireless Commun.*, vol.11, no.1, pp.88–96, Oct. 2012.

[5] Y. Wang, "Improving wireless network capacity under high transmission power with adaptive carrier sensing and routing," *Proc. IEEE/ACM IWQoS'13*, Montreal, Canada, June 2013, pp.1–6.

[6] Z. Yang, L. Cai, and W.S. Lu, "Practical scheduling algorithms for concurrent transmissions in rate-adaptive wireless networks," *Proc. IEEE INFOCOM'10*, San Diego, CA, Mar. 2010, pp.120–128.

[7] A.D. Gore, S. Jagabathula, and A. Karandikar, "On high spatial reuse link scheduling in STDMA wireless ad hoc networks," *Proc. IEEE GLOBECOM'07*, Washington, DC, Nov. 2007, pp.736–741.

[8] A.Y. Ng, M.I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in Neural Information Processing Systems*, pp.849–856, 2002.

[9] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E.W. De Luca, and S. Albayrak. "Spectral analysis of signed graphs for clustering, prediction and visualization," *Proc. 10th SIAM Int. Conf. Data Mining*, Columbus, OH, Apr./May 2010, pp.559–559.

[10] J. Gallier, "Spectral theory of unsigned and signed graphs applications to graph clustering: A survey," *arXiv:1311.2492*, Jan. 2015.

[11] L. Liu, X. Chen, D. Luo, Y. Lu, G. Xu, and M. Liu, "HSC: A spectral clustering algorithm combined with hierarchical method," *Neural Network World*, vol.23, no.6, pp.499–521, Jan. 2013.