A Clustering Approach to Wireless Scheduling

Wei Cui and Wei Yu
Electrical and Computer Engineering Department
University of Toronto, Toronto, Ontario M5S 3G4, Canada
Email: {cuiwei2, weiyu}@ece.utoronto.ca

***Abstract*—Scheduling is an important task for interference avoidance and for quality-of-service provisioning in dense wireless networks. While most existing works frame the scheduling task as an optimization problem, due to the non-convex structure of the problem, the existing solutions can reach local optima only and tend to have high computational complexity. This paper explores an alternative perspective to scheduling. Recognizing the importance of interference management, we experiment with the use of various clustering techniques for the scheduling task in a wireless device-to-device (D2D) network. Specifically, we construct a representation of interference in the wireless network, form clusters of highly interfering D2D links, then schedule only one link in each cluster. We compare different clustering strategies and show the promising potential of very low complexity scheduling algorithms based on this clustering approach to the wireless link scheduling problem.**

I. Introduction

Scheduling over wireless networks with densely deployed device-to-device (D2D) links has long been a well-studied but challenging task. There are a large number of works exploring various different scheduling strategies, such as greedy heuristic search [1], iterative computations with local convergence guarantee [2], [3], methods based on information theory considerations [4], [5], or methods for achieving the global optimum but with exponential complexity such as polyblock-based optimization [6] or nonlinear column generation [7]. These traditional mathematical optimization algorithms always require complete channel state information (CSI) and tend to have high algorithmic complexities. For maximizing long-term network utility (such as proportional fairness), where repeated optimization is needed over the time slots, the optimization approach would incur a large computation time overhead.

There are also a parallel set of works addressing the resources allocation problem through modeling and resolving conflicts between neighbouring links. In [8]–[11], *conflict graphs* are constructed for hypergraph coloring in time slots or frequency bands allocations, where conflicts are modeled as edges in graphs. Further in [12], *clusters* of non-interfering mobile stations are identified through the *max-K-Cut* algorithm for the problem of subchannel allocation. In addition, [13] focuses on subchannels allocations for the uplink in femtocells via forming clusters of interfering mobile users, with orthogonal subchannels assigned within each cluster.

This paper explores opportunities for developing scheduling algorithms for wireless D2D networks through *graph clustering*. Our main idea is to recognize that the clustering strategy is a good fit for the wireless scheduling problem, because by forming clusters consisting D2D links with strong mutual interferences and by scheduling only one link from each cluster at the time, interference can be naturally avoided.

In this respect, our approach is related to that of [14], which applies the graph spectral clustering technique to the link scheduling and channel assignment problem. In [14], by considering tuples of link-and-channel combinations, clusters that consist of compatible combinations are formed, and scheduling is done by searching for the best cluster. It is worth emphasizing that the approach of [14] is actually the exact opposite of the proposed technique of this paper. Instead of putting compatible links into clusters as in [14], this paper proposes to put incompatible (i.e., mutually interfering) links into the same cluster, then to schedule one link from each cluster. The main advantage of the proposed approach as compared to [14] is that our approach allows a natural extension to optimizing network utilities with long-term fairness. Further, this paper considers various different clustering techniques, while [14] only explores graph spectral clustering. As shown in this work, there are different clustering techniques that provide comparable performances yet with much reduced complexities.

Specifically, for a given wireless network, this paper proposes to construct a graph with the proximity measures (either channel state information (CSI) or geographical location information) reflecting the interference strength between the D2D links. We then explore the following clustering techniques: (i) Spectral Clustering, (ii) Hierarchical Clustering, (iii) K-Means Clustering; then proceed with two simple scheduling heuristics: scheduling the link closest to the cluster center for the sum-rate maximization; and scheduling all links in a within-cluster round-robin fashion for proportional fairness optimization. Despite the simplicity of these heuristics, the simulation results are competitive to the state-of-art optimization algorithms, while having advantageous time complexities especially for optimizing the long-term network utility.

II. Wireless Link Scheduling

Consider a wireless network consists of N independent D2D links with full frequency reuse over the bandwidth W. We use p_i to denote the transmit power level at the transmitter of the ith link, $g_{ij} \in \mathbb{R}$ to denote the *channel gain* from the transmitter of the jth link to the receiver of the ith link, and σ^2 to denote the background noise power level. A set of scheduling decisions is to select a subset of links to be activated, using a set of *binary* optimization variables $\{x_i\}_{i\in[1,N]}$, denoting whether the ith transmitter will transmit at its full power p_i ($x_i = 1$); or get turned off ($x_i = 0$). Correspondingly, the achievable rate for link i is:

$$R_i = W \log\left(1 + \frac{g_{ii} p_i x_i}{\Gamma(\sum_{j \neq i} g_{ij} p_j x_j + \sigma^2)}\right), \quad (1)$$

where Γ denotes the SNR gap to the information theoretical channel capacity, introduced by practical coding and modulation for the linear Gaussian channel [15].

A. Sum-Rate Maximization

The sum-rate maximization scheduling problem is as follows:

$$\begin{aligned} &\underset{\mathbf{x}}{\text{maximize}} && \sum_{i=1}^{N} R_i && \text{(2a)} \\ &\text{subject to} && x_i \in \{0, 1\},\ \forall i. && \text{(2b)} \end{aligned}$$

This optimization is challenging due to the highly non-linear interaction between variables $\{x_i\}$ in the signal-to-interference-and-noise (SINR) expressions. One state-of-the-art scheduling algorithm that can provide locally optimal solutions is *FPLinQ* [2], which uses the fractional programming optimization technique, requiring accurate $\{g_{ij}\}$ as inputs. We use this method as a benchmark to test the competitiveness of the new approaches.

B. Proportional Fairness Scheduling

While Problem (2) has been extensively studied, it does not incorporate the fairness among all links. To provide fairness, we can maximize the *proportional fairness utility* over the multiple time slots. The overall network utility is defined as the sum of certain utility function of each link's long-term average rate. The long-term average rate up to some time t, denoted by $\bar{R}^t$, is computed iteratively in an exponentially weighted fashion:

$$\bar{R}_i^t = (1 - \alpha)\bar{R}_i^{t-1} + \alpha R_i^t \quad t \leq T \quad (3)$$

where R_i^t is the ith link's instantaneous rate at time-slot t (computed by (1)). The proportional fairness utility uses the logarithm function as the utility function. In this case, the network utility becomes:

$$\sum_{i=1}^{N} \log(\bar{R}_i) \quad (4)$$

To maximize (4), traditional optimization algorithms (e.g. FPLinQ) perform *incremental* optimization [16], [17], in which the utility optimization problem is equivalently solved by maximizing a *weighted sum-rate* problem in each time-slot as follows:

$$\sum_{i=1}^{N} w_i R_i^t \quad (5)$$

with the weights being computed as:

$$w_i = \left.\frac{\partial U(\bar{R}_i^t)}{\partial R}\right|_{\bar{R}_i^t} = \left.\frac{\partial \log(\bar{R}_i^t)}{\partial R}\right|_{\bar{R}_i^t} = \frac{1}{\bar{R}_i^t}. \quad (6)$$

III. Representing Interference

This paper proposes to partition the links into clusters with heavily interfering links, then avoid multiple links being simultaneously scheduled within each cluster in order to effectively enforce *interference avoidance*. The first step is to represent the structure of interference either as a graph or in the geographic spatial domain.

A. Graph Representation of Interference

The structure of interference in a wireless network can be represented by a weighted graph. We consider undirected graph in this paper, and construct the graph representation for the wireless D2D network as follows. Each node represents a D2D link, while each edge represents the *maximum* interference level between the two D2D links it connects. Specifically, we

- **Step 1:** Assign one node for each D2D link;
- **Step 2:** Draw an undirected edge between every pair of nodes to form a fully-connected graph;
- **Step 3:** Set the edge weight connecting the ith and the jth links, e_{ij}, to be:

$$\text{weight}(e_{ij}) \doteq \max(g_{ij}, g_{ji}) \quad (7)$$

The rationale for using the maximum interference level is that as long as one of the two cross-link channels creates strong interference, the corresponding two links should not be scheduled at the same time. Naturally, a cluster of tightly-connected nodes represents a set of heavily-interfering D2D links that should be avoided being scheduled simultaneously. We perform clustering on graphs with spectral clustering and hierarchical clustering (and its equal cluster size variant), then schedule only one link per cluster to avoid interference.

B. Geographic Spatial Representation of Interference

The above-mentioned graph construction requires full CSI information, which are not always readily available. Alternatively, for wireless D2D networks, the $O(N)$ geographic information (i.e., the locations of the transmitters and receivers) provides a much more concise representation of approximate channel conditions. In this paper, we further explore the option of clustering on 2-D Euclidean space. Specifically, we represent each link by a 2-D point with its coordinates determined by the mid-point location between its transmitter and receiver. The distance of any two points, corresponding to the Euclidean distance between the mid-points of two links, provides a close approximation to the interference level between these two links. We then perform clustering on the 2-D Euclidean space with K-Means and its equal cluster size variant to avoid interference. Although using mid-point location for the 2-D representation of a link is a heuristic, simulations suggest that such representation provides close-to-optimal scheduling performances, even for wireless networks with long D2D links.

IV. Clustering Techniques

This paper explores three clustering techniques: spectral clustering and hierarchical clustering and its equal cluster size variant on graphs; also K-Means clustering and its equal cluster size variant in the geographic spatial domain.

A. Spectral Clustering

Spectral clustering is a classic algorithm operating on graphs with edge weights representing the affinities or adjacencies between any two nodes [18]. By finding the eigenvalues and eigenvectors from the Laplacian matrix of the graph (constructed from the adjacency matrix), the spectral clustering technique performs dimension reduction on the original graph, followed by basic clustering techniques (e.g. k-means [19]) on the obtained low dimensional representations. For our exploration, we use the version of the spectral clustering algorithm from [20] for obtaining multiple clusters on input graphs.

B. Hierarchical Clustering and its Equal Cluster Size Variant

Hierarchical clustering also operate on graphs with pairwise adjacency measures between the nodes. Compared with spectral clustering, it enjoys better time complexity, while having several shortcomings: the lack of theoretical justification, the need for ambiguous inter-cluster adjacency measure, and the lack of regularization on cluster sizes. In this paper, the last shortcoming is resolved via its equal cluster size variant.

The algorithm runs as follows: starting with each node being its own cluster, the hierarchical clustering proceeds by a recursive merging the neighbouring clusters: it finds the closest pair of clusters (corresponding to the edge with the largest weight) and merges the two clusters into one. The process stops until the desired number of clusters are formed. For the adjacency measure between two clusters with more than one node, we use the average of the pairwise distances among all connections between these clusters.

The equal cluster size variant operates on the same graphs, with a small change within its recursive merging process: once a cluster reaches (or potentially exceeding) the desired size, the entire cluster is fixed and removed from the graph, then the process continues for the remaining clusters.

C. K-Means Clustering and its Equal Cluster Size Variant

We also experiment with K-means clustering [19] based on geographic location inputs. The wireless links are represented by 2-D data-points in the Euclidean space. Clustering using geographic input is attractive because it only requires $O(N)$ inputs, as opposed to $O(N^2)$ CSI inputs.

The classic K-Means algorithm is well known. We skip the algorithm description and elaborate on its equal cluster size variant used in this paper, shown in Algorithm 1, which is a simplification of the modification proposed in [21].

D. Optimal Number of Clusters

One intrinsic difficulty in clustering is to determine the optimal number of clusters. As this paper mainly focuses on the potential of clustering-based scheduling methods, we adopt the link activation ratios from classic optimization: we compute the single time slot activation ratio from FPLinQ in sum-rate optimization and the average activation ratio over multiple time slots from FPLinQ in proportional fairness optimization to determine the number of clusters to form on each wireless network. In practice, this ratio can be estimated in advance.

Algorithm 1: K-Means Variant with Equal Cluster Size

INPUT: data-points, number of clusters; **while** *Not All Points are Assigned* **do**
    for *each unassigned data-point i* **do**
        Compute d_i as the difference between its distance to the furthest centroid and the closest centroid;
    end
    Rank d_i in the descending order into the ordered list l;
    while *None of the clusters reach their desired sizes* **do**
        Take one data-point from the top of l and assign it to the cluster of its closest centroid;
    end
    One cluster with the desired size formed as C_j;
    Remove the centroid and its data-points in C_j;
end
OUTPUT: A set of equal-sized clusters $\{C_j\}$.

V. Clustering Based Scheduling

A. Sum-Rate Maximization

If the objective is to maximize the sum rate, we can avoid interference by preventing scheduling links near cluster edges and to schedule only one single link per cluster close to the cluster center. For spectral clustering and hierarchical clustering, we propose to schedule the link with the highest sum of adjacency values to all other nodes within the same cluster. For K-means clustering, we schedule the link closest in the Euclidean distance to the centroid of its cluster.

B. Proportional Fairness Scheduling

For scheduling with a fairness objective, we adopt the *round-robin* approach to schedule links within each cluster, which ensures equal opportunity of being scheduled for all links in each cluster. This fairness further extends to the entire wireless network when the cluster sizes are equal. (This is the reason for exploring the equal size variants of the clustering algorithms.) We note that the equal cluster size constraint cannot be enforced by spectral clustering. Nonetheless, as spectral clustering minimizes the notion of *normalized cut* among formed clusters, the output cluster sizes are implicitly restricted to not differ greatly.

C. Time Complexity Analysis

One of the main advantages of the proposed clustering approach to scheduling is its low complexity, especially for proportional fairness scheduling. Table I shows the complexity of various algorithms. With the proposed round-robin scheduling scheme, we only need to compute the clusters once at the beginning (assuming no drastic changes in channel conditions

TABLE I: Time Complexity Analysis

Method	Sum-Rate	Proportional Fairness
Spectral Clustering	$O(N^3)$	$O(N^3) + TO(N)$
Hierarchical Clustering	$O(N^2)$	$O(N^2) + TO(N)$
K-Means	$O(N^2)$	$O(N^2) + TO(N)$
FPLinQ	$O(N^2)$	$TO(N^2)$
(Weighted) Greedy	$O(N^2)$	$TO(N^2)$

within the time slots), then the scheduling decisions reduce to switching among links in each cluster. With typical settings of $N < 100$ and T surpassing hundreds or even thousands, the complexity saving of clustering based approach is significant. We emphasize the crucial benefits of such time complexity saving, since most practical wireless applications require the scheduling operation to be repeated in each time-slot.

VI. Experimental Validation

A. Wireless Network Settings

We consider wireless networks over square regions with various sizes. Equal-length wireless D2D links are uniformly deployed within the region. We restrict a minimum of 5-meter distance between any transmitter and receiver within the network. For channel modeling, we consider only the path-loss component[1], under the short-range outdoor model ITU-1411, with 5MHz bandwidth at the carrier frequency of 2.4GHz. All antennas are with 2.5dBi antenna gain and 1.5 meters in height. The transmit power is 40dBm for every link. We assume -169dBm/Hz background noise level, and 6dB SNR gap. For all simulations, we test over 500 randomly generated wireless networks, over various settings on the number of D2D links, the region size, and the direct-channel distance of each link.

B. Visualization of Clusters

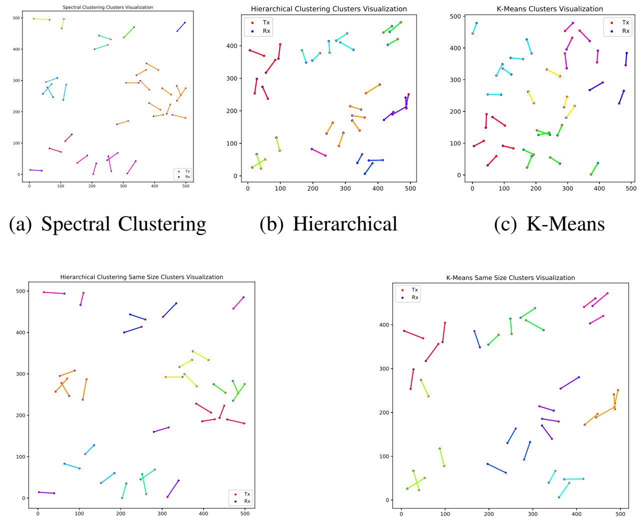
With each clustering method, we randomly sample a wireless network and visualize the resulting clusters of D2D links, as shown in Fig. 1. For the most parts, the clustering results are satisfying, as groups of crowded links are identified. However, there are pitfalls in the equal-size variants of the clustering algorithms: After one cluster is formed, all its data-points are removed from the pool of data-points, which would occasionally lead to far-away data-points being clustered together in later stage, as shown in the examples in Fig. 1(d) and Fig. 1(e). This problem with equal cluster size variants can be a limiting factor to their performances as shown in Section VI-D.

C. Sum Rate Maximization

We perform sum-rate maximization with the clustering based scheduling approaches, along with following benchmarks:

- **Greedy:** Following descending order of direct-link channel strength, schedule a link only if it increases sum-rate.
- **All Active:** Schedule all links.

[1]Experiments suggest clustering-based scheduling methods are less competitive when shadowing and fast-fading are included. With random channel fluctuations added to each channel, the clustering structures of wireless networks are blurred, rendering clusters being less representative.



(a) Spectral Clustering (b) Hierarchical (c) K-Means

(d) Hierarchical Equal Size (e) K-Means Equal Size

Fig. 1: Spectral Clustering Results Visualization

TABLE II: Average Sum Rate Performance

Sum Rate (% of FPLinQ)	50 Links length: $30m$ 500×500 (m^2)	30 Links length: $70m$ 1000×1000 (m^2)	70 Links length: $20m$ 1000×1000 (m^2)
Spectral Clustering	86.3	90.6	95.0
Hierarchical Clustering	84.8	90.9	95.1
K-Means Clustering	81.8	89.4	94.0
All Active	43.7	58.4	76.6
Greedy	83.2	96.0	97.3
FPLinQ	100	100	100

- **FPLinQ:** We run the state-of-the-art scheduling algorithm FPLinQ [2] with 100 iterations for outputs.

The sum-rate results are presented in Table II, where each entry is the average, over all testing networks, of the percentages compared with the sum-rate results achieved by FPLinQ.

D. Proportional Fairness Optimization Results

We further evaluate the clustering-based scheduling methods under the proportional fairness objective. Modifications on the benchmarks are as following:

- **Weighted Greedy:** Following descending order of the weight (by (6)) times the link's rate without interference, schedule a link only if it increases weighted sum-rate.
- **FPLinQ:** We run FPLinQ with 100 iterations, on weighted sum-rate optimization for sequential scheduling outputs.

Over the 500 wireless networks, we schedule for 500 time-slots. Instead of computing the exponentially weighted average rate as in (3), we compute the arithmetic average rate, which is conceptually similar yet faster to compute. We present the results as computed by (4) in Table III. We include the cumulative distribution curves of the average rates of all D2D

TABLE III: Average Log Utility Performance

Log Utility (Mbps)	50 Links length: $30m$ 500×500 (m^2)	30 Links length: $70m$ 1000×1000 (m^2)	70 Links length: $20m$ 1000×1000 (m^2)
Spectral Clustering	53.4	59.2	158.2
Hierarchical Clustering	55.9	59.8	159.2
K-Means Clustering	57.6	60.1	159.1
Hierarchical Clustering Equal-Size	55.5	56.0	155.3
K-Means Clustering Equal-Size	50.5	53.0	153.8
Weighted Greedy	47.8	58.1	152.0
FPLinQ [2]	63.0	61.8	159.4

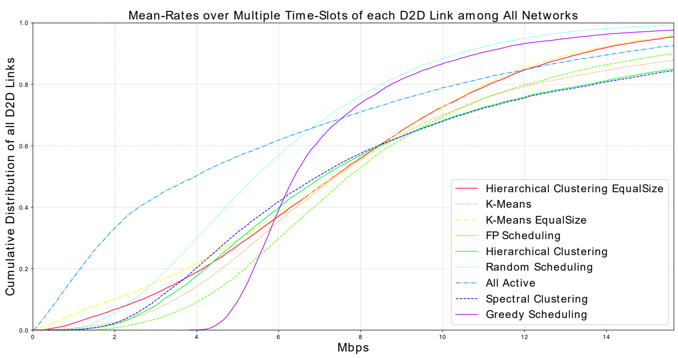


Fig. 2: CDF for mean-rates by D2D links over 500 wireless networks.

links, aggregated over all wireless networks at one test setting in Fig. 2.

E. Analysis

For both sum-rate and proportional fairness optimization, our clustering-based scheduling methods with simple scheduling heuristics produce consistently competitive performances as compared to FPLinQ over various network settings. Comparing among the five clustering techniques, the three original versions of clustering methods enjoy the best performance. For K-Means, it performs surprisingly well despite the crude geographic representation of each D2D link (by taking the mid-point location), even for long link length of 70 meters. In contrast, the equal-size versions of the clustering algorithms do not perform as well, as explained in Section VI-B.

Overall, these results are surprisingly good given the simplicity and heuristic nature of the proposed methods. Our results point to the potential role of learning-based algorithms for performing network-level scheduling and resource allocation tasks. Instead of modeling the problem in an optimization formulation, exploiting the intrinsic underlying structure and performing pattern matching may already suffice to achieve excellent performance at very low computational complexity.

[2]FPLinQ Scheduling is occasionally stuck at all links being scheduled off. To get these results, we schedule the link with the highest proportional fairness weight (computed as (6)) when this happens.

References

[1] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic, "FlashLinQ: A synchronous distributed scheduler for peer-to-peer ad hoc networks," *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1215–1228, Aug. 2013.

[2] K. Shen and W. Yu, "FPLinQ: A cooperative spectrum sharing strategy for device-to-device communications," in *IEEE Int. Symp. Inf. Theory (ISIT)*, Jun. 2017, pp. 2323–2327.

[3] Q. Shi, M. Razaviyayn, Z.-Q. Luo, and C. He, "An iteratively weighted MMSE approach to distributed sum-utility maximization for a MIMO interfering broadcast channel," *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4331–4340, Apr. 2011.

[4] N. Naderializadeh and A. S. Avestimehr, "ITLinQ: A new approach for spectrum sharing in device-to-device communication systems," *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1139–1151, Jun. 2014.

[5] X. Yi and G. Caire, "Optimality of treating interference as noise: A combinatorial perspective," *IEEE Trans. Inf. Theory*, vol. 62, no. 8, pp. 4654–4673, Jun. 2016.

[6] L. P. Qian and Y. J. Zhang, "S-MAPEL: Monotonic optimization for non-convex joint power control and scheduling problems," *IEEE Trans. Wireless Commun.*, vol. 9, no. 5, pp. 1708–1719, May 2010.

[7] M. Johansson and L. Xiao, "Cross-layer optimization of wireless networks using nonlinear column generation," *IEEE Trans. Wireless Commun.*, vol. 5, no. 2, pp. 435–445, Feb. 2006.

[8] B. Zhuang, D. Guo, E. Wei, and M. L. Honig, "Scalable spectrum allocation and user association in networks with many small cells," *IEEE Trans. Commun.*, vol. 65, no. 7, pp. 2931–2942, Jul. 2017.

[9] I. Rhee, A. Warrier, J. Min, and L. Xu, "DRAN: Distributed randomized TDMA scheduling for wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 8, no. 10, pp. 1384–1396, Oct. 2009.

[10] M. Necker and M. C. Necker, "A graph-based scheme for distributed interference coordination in cellular ofdma networks," in *IEEE Veh. Tech. Conf. (VTC Spring)*, May 2008.

[11] Y. Yang, B. Bai, and W. Chen, "Spectrum reuse ratio in 5g cellular networks: A matrix graph approach," *IEEE Trans. Mobile Comput.*, vol. 16, pp. 3541–3553, Dec. 2017.

[12] R. Y. Chang, Z. Tao, J. Zhang, and C. J. Kuo, "Multicell ofdma downlink resource allocation using a graphic framework," *IEEE Trans. Veh. Technol.*, vol. 58, pp. 3494–3507, Feb. 2009.

[13] J. Cao, T. Peng, X. Liu, W. Dong, R. Duan, Y. Yuan, W. Wang, and S. Cui, "Resource allocation for ultra-dense networks with machine learning based interference graph construction," *IEEE Internet Things J.*, pp. 3494–3507, Dec. 2019, early access.

[14] Z. He, S. Mao, S. Kompella, and A. Swami, "Link scheduling and channel assignment with a graph spectral clustering approach," in *IEEE MILCOM*, Nov. 2016.

[15] J. G. D. Forney and G. Ungerboeck, "Modulation and coding for linear Gaussian channels," *IEEE Trans. Inf. Theory*, vol. 44, no. 6, Oct. 1998.

[16] E. F. Chaponniere, P. J. Black, J. M. Holtzman, and D. N. C. Tse, "Transmitter directed code division multiple access system using path diversity to equitably maximize throughput," U.S. Patent 345 700, Jun. 30, 1999.

[17] J. Huang, R. Berry, and M. Honig, "Distributed interference compensation for wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 5, pp. 1074–1084, May 2006.

[18] J. Cheeger, "A lower bound for the smallest eigenvalue of the laplacian," in *Proceedings of the Princeton conference in honor of Professor S. Bochner*, 1969.

[19] S. Lloyd, "Least squares quantization in pcm," *IEEE Trans. Inf. Theory*, vol. 28, pp. 129–137, March 1982.

[20] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. MIT Press, 2001, pp. 849–856.

[21] E. Schubert and A. Zimek, "ELKI: A large open-source library for data analysis - ELKI release 0.7.5 "heidelberg"," *CoRR*, vol. abs/1902.03616, 2019. [Online]. Available: http://arxiv.org/abs/1902.03616