Algorithmically Efficient Networks

Kyomin Jung Devavrat Shah

Massachusetts Institute of Technology

Abstract

In this paper, we are interested in identifying *algorithmically efficient networks*. That is, determine minimal structural properties of a network that are sufficient to design simple, distributed algorithms to operate a communication network efficiently. The key algorithmic tasks in a communication network pertain scheduling (physical layer) and congestion control (network layer). Recent exciting progress has led to the understanding that good algorithmic solution for joint scheduling and congestion control can be obtained via combination of the "maximum weight scheduling" of Tassiulas and Ephremides (1992) along with a rate controller inspired by flow-level resource allocation model (e.g. Kelly, Maullo and Tan (1998)). Since the rate controller, by design, is simple and distributed, the key algorithmic challenge lies in designing efficient implementation of "maximum weight scheduling" in the network. In general, this involves solving a network-wide hard combinatorial optimization problem (maximum weight independent set) every time. And this problem, in general is known to be computationally hard.

Motivated to find algorithmically efficient networks, we characterize a large class of network structure that allow for simple, distributed algorithm design for finding approximate maximum weight independent set (or schedule). Specifically, we establish that for any network structure with *geometry* (i.e. polynomial growth of network neighborhood), an algorithm, that determines schedule of each node based information obtained from *constant*-depth local neighborhood, achieves essentially optimal performance. Our algorithm allows for a smooth trade-off between overhead for computing schedule versus the performance. The key technical innovation is based on a novel graph decomposition algorithm, which we believe should be of interest in its own right.

It should be noted that our results immediately apply to wireless network deployed in a geographic area as they naturally posses *geometry*. In general, a network designer may engineer the network so as to induce geometry and therefore allow for efficient algorithm design.

I. Introduction

The primary purpose of a well engineered communication network is to satiate heterogeneous demands of various network users while utilizing network resources efficiently. This involves design of appropriate resource allocation mechanisms that can be implemented. Most of the modern communication networks are built using layered architecture. Therefore, they require designing simple, distributed algorithms for scheduling (physical layer) and congestion control (network layer).

The exciting progress over the past two decades has finally led to the following understanding (see Srikant and Shakkottai [20]): a good algorithmic architectural solution for joint scheduling and congestion control can be obtained based on the maximum weight scheduling by Tassiulas and Ephremides [23] along with a rate control algorithm that attempts to solve a utility maximization problem subject to network resource constraints (e.g. works by Kelly, Maullo and Tan [11], Low and Lapsley [14] and Srikant [22]). Therefore, a network can be operated in an algorithmically efficient manner with high performance if the maximum weight scheduling and rate control can be performed efficiently.

The rate control algorithm is simple and distributed because essentially it is an iterative algorithm to solve a concave maximization problem with linear constraints that have separable structure (a la monotropic program). On the other hand, the maximum weight scheduling algorithm is required to determine schedule of nodes subject to network scheduling constraints that are imposed by the physical layer. For example, in a wireless network this corresponds to determining which set of non-interfering nodes should transmit so that their collective weight (which is usually their queue-size) is maximized. In general, this can be abstracted as finding maximum weight independent set in the graph induced by network scheduling constraints. It is well known that finding maximum weight independent set in general graph is NP-hard [4] and even hard to approximate within $n^{1-o(1)}$ ($B/2^{O(\sqrt{\log B})}$

All authors are with Laboratory for Information and Decision Systems, MIT. This work was supported in parts by NSF projects HSD 0729361, CNS 0546590, TF 0728554 and DARPA ITMANET project. Authors' email addresses: {kmjung, devavrat}@mit.edu

for degree B graph) factor [25]. Therefore, algorithmic efficient design of joint scheduling and congestion control in a general is unlikely.
This leads to the following key question of this paper:

what are the minimal structural properties of the network that allow for efficient design of scheduling algorithm and thus lead to efficient communication network design?

A. Previous work

We present a brief summary of previous results about design of efficient network scheduling algorithms. The maximum weight independent set scheduling algorithm of Tassiulas and Ephremides that we discussed earlier provides excellent performance guarantees: it is throughput optimal and has low average delay (or average queue-size). However, finding such schedule (or maximum weight independent set) is computationally hard. This led researchers to consider various approximations of maximum weight independent set (or scheduling) algorithm.

Roughly speaking, there are two types of approximation algorithms. The first class of algorithms are very simple and greedy (like) (e.g. see [7], [2], [3], [13] and their adoptions to independent set model). These algorithms are not even throughput optimal and can be arbtirarily bad in terms of loss of throughput. The second class of algorithms are throughput optimal and very simple (e.g. natural adaptation of algorithm of Modiano, Shah and Zussman [16] – see [10]). They are based on randomization and distributed computation of summation. However, this algorithm suffers from possibly very huge average queue-size or delay. This raises the question about possibility of a throughput optimal and low delay scheduling algorithm. In a recent work, Shah, Tse and Tsitsiklis [19] established impossibility of such result under standard computational hypothesis.

This impossibility result is an *adversarial* or a *worst-case* result. Therefore, it is natural to inquire whether there is a large class of network structure for which such impossibility result does not hold. Implicitly, such line of query was pursued in some of the recent works for very specific graph structures. Specifically, work by Hunt et. al. [9] provides an approximation algorithm for maximum weight independent set for unit disk graphs. This was utilized by Sharma, Mazumdar and Shroff [21] in their work to identify that max-weight scheduling with $K-$hop matching. A similar approach was employed in another recent work by Sarkar and Ray [17] for a very restricted class of graph structures. However, identification of a broad class of graph structures that posses such property has remained unknown.

B. Our contributions

As the main result of this paper, we provide an intuitively pleasing characterization of a class of network graphs that allow for simple algorithmic design for finding maximum weight independent set. Specifically, we establish that networks with *geometry* are algorithmically efficient. By networks with geometry, we mean graphs in which the size of the local neighborhood of each node in the network grows polynomially. We call such graphs *polynomially growing graphs*. We show that many of the popular graph models are special instances of our thus identified structure. For example, popular wireless newtork model of geometric random graph (e.g. Gupta and Kumar [6]) or popular computational geometric graph model of finite doubling dimension graphs are special instances of our polynomially growing graphs.

Our result is indeed constructive in the sense that we provide a simple algorithm that obtains excellent approximation (formally, also known as Polynomial Time Approximation Scheme or PTAS) of maximum weight independent set for any polynomially growing graph. The basic version of the algorithm is randomized. It is based on a novel randomized graph decomposition scheme. Its distributed implementation can be obtained in a straightforward manner. This decomposition partitions the graph into *constant* size local partitions and our algorithm decides solution of each node only based on information of this partition. Interestingly enough, due to geometry of the graph and clever randomized decomposition, thus obtained solution is an excellent approximation of the maximum weight independent set. As part of this algorithm, each node requires only *constant* amount of information exchange with its local neighbors. This information exchange and quality of approximation exhibit an intuitively pleasing trade-off between them. It should be noted that, in a sense our algorithm can be thought of as a (randomized) generalized greedy algorithm.

Finally, we note that our algorithm can be made *deterministic*. We provide a *derandomization* of the randomized algorithm with minimal additional complexity. Indeed, our motivation for providing this derandomization is primarily theoretical in nature.

C. Organization

The paper is organized as follows. In section II, we introduce the basic notation and the model. In section III, we introduce the *polynomially growing graph* in detail. We provide examples of such graphs. In section IV, we present our main result. We describe our *randomized* PTAS for maximum weight independent set for polynomially growing graphs. Finally, we provide a derandomization of this algorithm in section V. Discussion and future works are described in section VI.

II. Notations and Model

Let $G = (V, E)$ be an undirected network graph with $|V| = n$. Let $\mathcal{N}(v) = \{u \in V : (u, v) \in E\}$ denote the set of all neighbors of $v \in V$. The time is assumed to be slotted, and let $\tau \in \mathbf{Z}_+$ denote the time. Each node $v \in V$ is capable of wireless transmission at the unit rate to any of its neighbors. We ignore the power control for simplicity, but as a reader may notice, it can be easily included in the model. At each node, packets (of unit size) arrive according to an external arrival process. Let $\bar{A}(\tau) = [\bar{A}_v(\tau)]$ denote the cumulative arrival process until time $\tau \in \mathbf{Z}_+$, i.e. $\bar{A}_v(\tau)$ be the total number of packets arrived at node v in the time interval $[0, \tau]$. We assume $\bar{A}(\tau) = \mathbf{0}$. Let $A_v(\tau) = \bar{A}_v(\tau) - \bar{A}_v(\tau - 1)$ be the number of packets arriving at node v in time slot τ. We assume that at most one packet can arrive at a node v in each time slot, i.e. $A_v(\tau) \in \{0, 1\}$. Finally, we assume that $A_v(\cdot)$ are Bernoulli i.i.d. random variables with $\Pr(A_v(\tau) = 1) = \lambda_v$. Let $\lambda = [\lambda_v]$ denote the arrival rate vector.

For simplicity and ease of explanation, we assume that network is single-hop, i.e. data arriving at a node v is to be sent to one of its neighbors. However, as explained in [24], for multi-hop network, the scheduling algorithm choosing the maximum weight independent set with weights being "difference of queue-sizes" is throughput optimal. Hence our result can be applied to the multi-hop network. Let $Q_v(\tau)$ denote the queue-size at node v at time τ with $Q(\tau) = [Q_v(\tau)]$. We assume the system starts empty, i.e. $Q(0) = \mathbf{0}$. Let $\bar{D}(\tau) = [\bar{D}_v(\tau)]$ denotes the cumulative departure process from $Q(\tau)$ and $D(\tau) = [D_v(\tau)]$ denote the number of departures in time slot τ. Then,

$$\begin{aligned} Q(\tau) &= Q(0) + \bar{A}(\tau) - \bar{D}(\tau) = \bar{A}(\tau) - \bar{D}(\tau) \\ &= Q(\tau - 1) + A(\tau) - D(\tau). \end{aligned} \tag{1}$$

Departure at each time happens according to the scheduling algorithm which needs to satisfy *interference constraint* that no two neighboring nodes transmit data in the same time slot. To this end, let $\mathcal{I}$ denote the set of all independent set of G. A subset I of nodes of G is called an independent set of G if no two neighboring nodes of G belongs to I. Then at each time, the scheduling algorithm schedules nodes of an independent set $I \in \mathcal{I}$ to transmit packets. In what follows, we will denote independent set I as vector $I = [I_v]$ with $I_v \in \{0, 1\}$ and $I_v = 1$ indicates that node v is in I.

Definition 1: We say that a system is *stable* for given λ under the particular scheduling policy if

$$\limsup_{\tau \to \infty} \mathbb{E}[Q_v(\tau)] < \infty, \quad \forall\, v \in V.$$

From [24], it is clear that the set of all λ for which there exists a scheduling policy so that the system is stable is given by $\Lambda = \mathsf{Co}(\mathcal{I})^o$, where $\mathsf{Co}(\mathcal{I})$ is the convex hull of $\mathcal{I}$ in $\mathbb{R}^n$, and S^o indicates the interior of a set S in $\mathbb{R}^n$. Hence, we call Λ the *throughput region* of the system, and such λ's are called *feasible arrival rate vectors*.

Given a feasible arrival rate vector $\lambda \in \Lambda$, define $\varepsilon_\lambda > 0$ by

$$\varepsilon_\lambda = \sup\ \{\alpha > 0 \mid \frac{\lambda}{1 - \alpha} \in \Lambda\}.$$

The 'maximum weight independent set' (MWIS) $I^*(\tau)$ at time τ with respect to the weight vector $Q(\tau - 1)$ is defined by

$$I^*(\tau) = \arg\max_{I \in \mathcal{I}} \langle I, Q(\tau - 1) \rangle,$$

with notation $\langle A, B \rangle = \sum_{v \in V} A_v B_v$.

Definition 2: An independent set $I(\tau) \in \mathcal{I}$ is called an *ε-approximation MWIS* at time τ if it satisfies that

$$\langle I, Q(\tau - 1) \rangle \geq (1 - \varepsilon) \langle I^*(\tau), Q(\tau - 1) \rangle.$$

In [18], it was shown that the system under an ε_λ-approximation MWIS scheduling algorithm is stable for λ. As discussed before, finding ε-approximation MWIS for general network graph is computationally hard [4], [25].

We present an algorithm that computes an ε-approximation MWIS for any polynomially growing graphs, which runs in time $O(n)$ for any constants $\varepsilon > 0$. Our algorithm can be implemented in a distributed manner with $O(1)$ message exchanges per node to compute an ε-approximation MWIS schedule at each time τ.

A. Relation to other models

The above model ignores the multi-hop setup. However, we have done so to keep the exposition simple. The scheduling algorithm of interest with the independent set interference constraint remains the same as ε-approximation MWIS with weights being some-what different. To explain this, we give two well-known scenarios as follows.

1. Multi-hop queuing network. As considered in [24], for given network G, let S be set of data-flows with arrival rate λ_s for flow $s \in S$. Let f_s, d_s denote source and destination node respectively for a flow $s \in S$. The routing is assumed to be pre-determined in the network. If s passes through $v \in V$, then let $h(v,s) \in V$ denote its next hop unless $v = d_s$, in which case its data departs from the system. Let $Q_{vs}(\tau)$ denote the queue-size of flow s at node v at time τ. Define

$$W_{vs}(\tau) = \begin{cases} Q_{vs}(\tau) - Q_{h(v,s)s}(\tau) & \text{if } \ v \neq d_s \\ 0 & \text{if } \ v = d_s. \end{cases}$$

Now define the weight at node v at time τ by

$$W_v(\tau) = \max_{s \in S} W_{vs}(\tau),$$

and $W(\tau) = [W_v(\tau)]$. Then by the argument in [24], the ε-approximation MWIS scheduling algorithm for appropriate constant $\varepsilon > 0$ makes the system stable for any feasible arrival rate vector.

2. Joint resource allocation & scheduling. In [12], it is well explained that the problem of congestion control and scheduling decomposes into weakly coupled two sub-problems: (i) the congestion control, and (ii) the scheduling. We describe the link-level scheduling problem. We urge an interested reader to go through [12] for details. The setup of the problem is the same as that in the *example 1* described above with a difference that the routing is not pre-determined. In this case, the coupling of congestion control and scheduling happens via Lagrange multipliers

$$\mathbf{q}(\tau) = [q_e(\tau)]_{e \in E}.$$

With the interference model of this paper, the scheduling problem boils down to the selection of MWIS $I^*(\tau)$ with respect to the weight

$$W(\tau - 1) = [W_v(\tau - 1)],$$

where

$$W_v(\tau - 1) = \max_{e: e=(u,v) \in E} q_e(\tau - 1).$$

In this paper, we will restrict our discussion to the single-hop model described earlier. However, it should be clear that our algorithms can be applied to more general setups.

B. Technical preliminaries

Here we present some useful technical preliminaries. Consider a discrete time Markov chain on countable state space $S = \mathbb{N}^M$ for some finite integer M. Let $X(\tau)$ denote the random state of Markov chain at time $\tau \in \mathbf{Z}_+$. Let $X(0) = \mathbf{0}$ (In fact, it can be any finite state). Let $L : S \to [0, \infty)$ and $f : S \to [0, \infty)$ be any non-negative valued functions with $L(\mathbf{0}) = 0$. Then the following is well-known [15].

Proposition 1: Let a Markov chain be aperiodic and irreducible. Let there exists a closed and bounded set $C \subset S$ such that the Markov chain satisfies the following condition: $\forall\ \tau \in \mathbf{Z}_+$,

$$\mathbb{E}[L(X(\tau + 1))|X(\tau)] \leq L(X(\tau)) - f(X(\tau)) + B \cdot \mathbf{1}_{\{X(\tau) \in C\}},$$

with $B > 0$ a constant, and $\sup_{x \in C} L(x) < \infty$. Then,

(a) The Markov chain is recurrent with a unique stationary distribution $\pi = [\pi(x)]_{x\in S}$ such that

$$\pi(f) = \sum_{x\in S} \pi(x) f(x) < \infty.$$

(b) Further,

$$\lim_{\tau\to\infty} \mathbb{E}[f(X(\tau)] \to \pi(f).$$

Similar result (also known as Foster's criteria) is used in most of the previous works to prove stability of the system, and to obtain a bound on average queue-sizes.

III. Polynomially Growing Graph

In this section, we define the class of polynomially growing network graphs and its property.

Definition 3: Let $\mathbf{d}_G$ be the shortest path distance metric of a given graph G, and let $\mathbf{B}_G(v,r) = \{w \in V | \mathbf{d}_G(w,v) < r\}$. If there are constants $C > 0$ and $\rho > 0$ so that for any $v \in V$ and $r \in \mathbb{N}$,

$$|\mathbf{B}_G(v,r)| \leq C \cdot r^{\rho},$$

then we say G is *polynomially growing* with *growth rate* ρ and corresponding constant C.

Practical network graphs including the following geometric network graphs and doubling dimensional graphs, satisfy the above property.

Example 1 : Geometric Graph Consider a wireless network with n nodes represented by the vertices $V = \{1,\ldots,n\}$ placed in a 2-dimensional geographic region given by the $\sqrt{n}\times\sqrt{n}$ square of area n[1] in an *arbitrary* manner (not necessarily random). Let E be the set of edges between nodes indicating which pair of nodes can communicate. Let $\mathbf{d}_E(\cdot,\cdot)$ be the Euclidean distance of the Euclidean space. Given a vertex $v \in V$, let $\mathbf{B}_E(v,r) = \{u \in V :\ \mathbf{d}_E(u,v) < r\}$. We assume that the wireless network satisfies the following simple assumptions.

1) There is an $R > 0$ such that no two nodes having distance larger than R can establish a communication edge with each other[2] where R is called the transmission radius.
2) Graph G has bounded density $D > 0$, i.e. for all $v \in V$, $\frac{|\mathbf{B}_E(v,R)|}{R^2} \leq D$.

A geometric random graph obtained by placing n nodes in the $\sqrt{n}\times\sqrt{n}$ square uniformly at random and connecting two nodes that are within distance $R = \Theta(\sqrt{\log n})$ of each other satisfies the previous assumptions with high probability.

Lemma 2: Any geometric graphs satisfying the above two assumptions are polynomially growing with growth rate 2.

Proof: Let G be a geometric graph with a transmission radius R and a bounded density D. First, note that in the Euclidean space, for any $r \geq R$, $\mathbf{B}_E(v,r)$ can be covered by $\Theta\left(\left(\frac{r}{R}\right)^2\right)$ many balls of radius R. Hence, together with the definition of the bounded density D, there is a constant $D' > 0$ so that for all $v \in V$ and $r \geq R$,

$$\frac{|\mathbf{B}_E(v,r)|}{r^2} \leq D'. \tag{2}$$

Now, for a given two connected vertices $v, w \in V$ of G, let $v = v_0, v_1, v_2 \ldots, v_\ell = w$ be a shortest path in G. By the definition of the transmission radius, for all $i = 0, 1 \ldots, (\ell - 1)$,

$$\mathbf{d}_E(v_i, v_{i+1}) \leq R.$$

[1]Placing the nodes in the specified square is for simple presentation. The same result holds when the nodes are placed in any Euclidean rectangle, and when the nodes are place in any region of k-dimensional Euclidean space.

[2]It does not imply that nodes within distance R must communicate.

By the triangular inequality in the Euclidean metric,

$$\mathbf{d}_E(v,w) \le \sum_{i=0}^{\ell} \mathbf{d}_E(v_i, v_{i+1}) \le R \cdot \ell.$$

So we obtain

$$\mathbf{d}_E(v,w) \le R\ell = R \cdot \mathbf{d}_G(v,w).$$

Hence, for any $v \in V$ and $r \in \mathbb{N}$,

$$\mathbf{B}_G(v,r) \subset \mathbf{B}_E(v,Rr).$$

From (2),

$$|\mathbf{B}_G(v,r)| \le |\mathbf{B}_E(v,Rr)| \le \left(D'R^2\right) r^2,$$

which shows that the growth rate of G is 2. ■

Example 2 : Doubling dimensional graph A graph is said to have a *doubling dimension* $\varphi > 0$ if any ball of radius $2r$ (w.r.t. the shortest path metric) in G can be covered by at most 2^{φ} many balls of radius r for any $r \in \mathbb{N}$. A graph with a constant doubling dimension is called *a doubling dimensional graph*. The notion of doubling dimensional graphs was first introduced in [1], [8], [5]. It is easy to check that a grid graph $\mathbb{Z}^d$ has doubling dimension d. Clearly, any graph with n nodes has doubling dimension at most $O(\log_2 n)$. The network graph structure arising in practice, such as that in wireless network or peer-to-peer network is likely to have a constant doubling dimension. The following Lemma shows that any doubling dimensional graph is polynomially growing.

Lemma 3: A graph with a constant doubling dimension φ is polynomially growing with growth rate φ.

Proof: First, we will show that for any $x \in V$ and any $t \in \mathbb{Z}_+$,

$$|\mathbf{B}_G(x, 2^t)| \le 2^{t\varphi}. \tag{3}$$

The proof of (3) is by induction on $t \in \mathbb{Z}_+$. For the base case, consider $t = 0$. Now, $\mathbf{B}_G(x, 2^0)$ is essentially the set of all points which are at distance less than 1 from x by the definition. Since it is metric with distance being integer, this means that $\mathbf{B}_G(x,1) = \{x\}$. Hence, $|\mathbf{B}(x,1)| = 1 \le 2^{0 \times \rho(\mathcal{M})}$ for all $x \in \mathcal{X}$.

Now suppose that the claim of Lemma is true for all $t \le k$ and all $x \in \mathcal{X}$. Consider $t = k+1$ and any $x \in \mathcal{X}$. By definition of the doubling dimension, there exists $\ell \le 2^{\varphi}$ balls of radius 2^k, say $\mathbf{B}_G(y_j, 2^k)$ with $y_j \in \mathcal{X}$ for $1 \le j \le \ell$, such that

$$\mathbf{B}_G(x, 2^{k+1}) \subset \cup_{j=1}^{\ell} \mathbf{B}_G(y_j, 2^k).$$

Therefore,

$$|\mathbf{B}_G(x, 2^{k+1})| \le \sum_{j=1}^{\ell} |\mathbf{B}_G(y_j, 2^k)|.$$

By inductive hypothesis, for $1 \le j \le \ell$,

$$|\mathbf{B}_G(y_j, 2^k)| \le 2^{k\varphi}.$$

Since we have $\ell \le 2^{\varphi}$, we obtain

$$|\mathbf{B}_G(x, 2^{k+1})| \le \ell \; 2^{k\varphi} \;\; \le \;\; 2^{(k+1)\varphi}.$$

This completes the proof of inductive step, and that of (3).

Now, for any $r \in \mathbb{N}$, and any $x \in V$, let $2^t \le r < 2^{t+1}$ for $t \in \mathbb{Z}_+$. From (3), we obtain that

$$|\mathbf{B}_G(x,r)| \le |\mathbf{B}_G(x, 2^{t+1})| \le 2^{(t+1)\varphi} = 2^{\varphi} \cdot (2^t)^{\varphi} \le 2^{\varphi} \cdot r^{\varphi}, \tag{4}$$

which shows the Lemma. ■

Property of polynomially growing graphs The following Lemma shows that any subgraph of a polynomially growing graph is also a polynomially growing graph.

Lemma 4: If G is polynomially growing with growth rate ρ, any subgraph $\hat{G} = (\hat{V}, \hat{E})$ of G obtained by

removing some edges and vertices of G is also polynomially growing with growth rate at most ρ.

Proof: For any vertex $v, w \in \hat{V}$, note that

$$\mathbf{d}_{\hat{G}}(v, w) \geq \mathbf{d}_G(v, w),$$

since any path in $\hat{G}$ from v to w is also a path in G. Hence, for any $v \in \hat{V}$ and $r \in \mathbb{N}$,

$$\mathbf{B}_{\hat{G}}(v, r) \subset \mathbf{B}_G(v, r).$$

Hence,

$$|\mathbf{B}_{\hat{G}}(v, r)| \leq |\mathbf{B}_G(v, r)| \leq r^\rho,$$

which shows the Lemma from the definition 3. ■

For example, any subgraph of a grid graph is a polynomially growing graph.

IV. ε-APPROXIMATION OF MWIS FOR POLYNOMIALLY GROWING GRAPHS

Recall that the problem we want to solve is as follows. Given a polynomially growing graph $G = (V, E)$ with constant growth rate ρ, and with nonnegative node weights $w = [w_i] \in \mathbb{R}^n_+$. Let $\epsilon > 0$ be a give constant, and let MWIS be defined as

$$x^* = \text{argmax}_{\underline{\mathrm{x}} \in \mathcal{I}} w^T \underline{\mathrm{x}}.$$

We want to find an independent set $\hat{\underline{\mathrm{x}}} = (x_i) \in \{0, 1\}^n$ such that

$$w^T \hat{\underline{\mathrm{x}}} \geq (1 - \epsilon) w^T x^*,$$

which we call an ε-approximation of MWIS. In this section, we present a randomized algorithm that computes an ε-approximation of MWIS in expectation for any graph with polynomial growth ρ, which runs in time $O(n)$ for any constant $\varepsilon > 0$ and $\rho > 0$. The algorithm consists of the following two steps.

1) Obtain a graph decomposition of G into small components by removing some vertices of G.
2) Produce a global solution by merging the local solutions of each connected components.

In section IV-A, we will present a randomized graph decomposition for the first step, and in section IV-B, we present how to obtain a global solution from this randomized decomposition and show that the global solution achieves a good approximation in expectation. Then in section V, we establish a derandomization of this algorithm.

A. Step 1: Randomized Graph Decomposition Algorithm

First, given constants $\varepsilon > 0$ and $\Delta \in \mathbb{N}$, we define a notion (ε, Δ)-*decomposition* of a graph $G = (V, E)$.

Definition 4: We call a random subset of vertices $\mathcal{B} \subset V$ as (ε, Δ)-*decomposition* of G if the followings hold:

- (a) For any $v \in V$, $\Pr(v \in \mathcal{B}) \leq \varepsilon$.
- (b) Let $S_1, \ldots, S_\ell$ be the connected components of graph $G' = (V', E')$ where $V' = V \backslash \mathcal{B}$ and $E' = \{(u, v) \in E : u, v \in V'\}$. Then, $\max_{1 \leq k \leq \ell} |S_k| \leq \Delta$ with probability 1.

Note that the (ε, Δ)-decomposition $\mathcal{B}$ forms the union of boundary vertices of each connected components. Now we describe our graph decomposition algorithm for any $\varepsilon > 0$ and an operational parameter K, which will depend on ε and the growth rate ρ and the corresponding constant C of the graph. We will show that our algorithm will output (ε, Δ)-decomposition where Δ will depend on ε and K.

Given ε and K, define a random variable $\mathbf{Q}$ over $\{1, \ldots, K\}$ as

$$\Pr[\mathbf{Q} = i] = \begin{cases} \varepsilon(1 - \varepsilon)^{i-1} & \text{if } 1 \leq i < K \\ (1 - \varepsilon)^{K-1} & \text{if } i = K \end{cases}.$$

The **Decomposition Algorithm** (ε, K) described next essentially does the following. The algorithm performs iteratively. Initially, all vertices are colored *white*. If there is any *white* vertex, choose any of them *arbitrarily*. Let u be the chosen vertex. Draw an independent random number Q as per distribution $\mathbf{Q}$. Select all *white* vertices

that are at distance Q from u in $\mathcal{B}$ and color them *blue*; color all *white* vertices at distance $< Q$ from u (including u itself) as *red*. Repeat this process until no more *white* vertices are left. Output $\mathcal{B}$ (i.e. *blue* nodes).

Precise description of the algorithm is as follows. We will set

$$K = K(\varepsilon, \rho, C) = \frac{8\rho}{\varepsilon} \log\left(\frac{8\rho}{\varepsilon}\right) + \frac{4}{\varepsilon} \log C + \frac{4}{\varepsilon} \log \frac{1}{\varepsilon} + 2.$$

This definition is exploited in the proof of Lemma 5.

Decomposition Algorithm (ε, K)

(1) Initially, set $\mathcal{W} = V$, $\mathcal{B} = \emptyset$ and $\mathcal{R} = \emptyset$.
(2) Repeat the following till $\mathcal{W} \neq \emptyset$:
  (a) Choose an element $u \in \mathcal{W}$ uniformly at random.
  (b) Draw a random number Q independently according to the distribution $\mathbf{Q}$.
  (c) Update
    (i) $\mathcal{B} \leftarrow \mathcal{B} \cup \{w | \mathbf{d_G}(u, w) = Q \text{ and } w \in \mathcal{W}\}$,
    (ii) $\mathcal{R} \leftarrow \mathcal{R} \cup \{w | \mathbf{d_G}(u, w) < Q \text{ and } w \in \mathcal{W}\}$,
    (iii) $\mathcal{W} \leftarrow \mathcal{W} \cap (\mathcal{B} \cup \mathcal{R})^c$.
(3) Output $\mathcal{B}$.

Lemma 5: Given graph G with growth rate $\rho = \rho(G)$ and the corresponding constant C, and $\varepsilon \in (0, 1)$, the output of the **Decomposition algorithm**(ε, K) becomes a $(2\varepsilon, CK^\rho)$-decomposition of G.

Proof: To prove that the random output set $\mathcal{B} \subset V$ of the algorithm with parameters $(\varepsilon, K(\varepsilon, \rho))$ we need to establish properties (a) and (b) of Definition 4.

Proof of (a). To prove (a), we state and prove the following Claim.

Claim 6: Consider metric space $\mathcal{G} = (V, \mathbf{d_G})$ with $|V| = n$. Let $\mathcal{B} \subset V$ be the random set that is output of **Decomposition algorithm** with parameter (ε, K) applied to $\mathcal{G}$. Then, for any $v \in V$,

$$\Pr[v \in \mathcal{B}] \leq \varepsilon + P_K |\mathbf{B}_G(v, K)|,$$

where $\mathbf{B}_G(v, K)$ is the ball of radius K in $\mathcal{G}$ with respect to the $\mathbf{d_G}$, and $P_K = (1 - \varepsilon)^{K-1}$.

Proof of Claim 6: The proof is by induction on the number of points n over which the metric space is defined. When $n = 1$, the algorithm chooses only point as u_0 in the initial iteration and hence it can not be part of the output set $\mathcal{B}$. That is, for this only point, say v,

$$\Pr[v \in \mathcal{B}] = 0 \leq \varepsilon + P_K |\mathbf{B}(v, K)|.$$

Thus, we have verified the base case for induction ($n = 1$).

As induction hypothesis, suppose that the Claim 6 is true for any metric space on n points with $n < N$ for some $N \geq 2$. As the induction step, we wish to establish that for a metric space $\mathcal{G} = (V, \mathbf{d_G})$ with $|V| = N$, the Claim 6 is true. For this, consider any point $v \in V$. Now consider the first iteration of the **Decomposition algorithm** applied to $\mathcal{G}$. The algorithm picks $u_0 \in V$ uniformly at random in the first iteration. Given v, depending on the choice of u_0 we consider four different cases (or events). We will show that in these four cases,

$$\Pr[v \in \mathcal{B}] \leq \varepsilon + P_K |\mathbf{B}_G(v, K)|$$

holds.

Case 1. This case corresponds to event E_1 where the chosen random u_0 is equal to point v of our interest. By definition of the algorithm, under the event E_1, v will never be part of output set $\mathcal{B}$. That is,

$$\Pr[v \in \mathcal{B} | E_1] = 0 \ \leq \ \varepsilon + P_K |\mathbf{B}(v, K)|.$$

Case 2. Now, suppose u_0 is such that $v \neq u_0$ and $\mathbf{d_G}(u_0, v) < K$. Call this event E_2. Further, depending on choice of random number Q_0, define the following events

$$E_{21} = \{\mathbf{d_G}(u_0, v) < Q_0\},\ E_{22} = \{\mathbf{d_G}(u_0, v) = Q_0\},\ \text{and}\ E_{23} = \{\mathbf{d_G}(u_0, v) > Q_0\}.$$

By definition of the algorithm, when E_{21} happens, v is selected as part of $\mathcal{R}_1$ and hence v can never be a part of output $\mathcal{B}$. When E_{22} happens, v is selected as part of $\mathcal{B}_1$ and hence it is definitely a part of output set $\mathcal{B}$. When E_{23} happens, v is neither selected in set $\mathcal{R}_1$ nor selected in set $\mathcal{B}_1$. It is left as an element of the set $\mathcal{W}_1$. This new set $\mathcal{W}_1$ has points less than N. The original metric $\mathbf{d_G}$ is still the metric on the points[3] of $\mathcal{W}_1$. By definition, the algorithm only cares about $(\mathcal{W}_1, \mathbf{d_G})$ in the future and it is not affected by its decisions in past. Therefore, we can invoke induction hypothesis which implies that if event E_{23} happens then the probability of $v \in \mathcal{B}$ is bounded above by $\varepsilon + P_K|\mathbf{B}(v, K)|$. Finally, let us relate the $\Pr[E_{21}|E_2]$ with $\Pr[E_{22}|E_2]$. Suppose $\mathbf{d_G}(u_0, v) = \ell < K$. By definition of probability distribution of $\mathbf{Q}$, we have

$$\Pr[E_{22}|E_2] \;=\; \varepsilon(1-\varepsilon)^{\ell-1}, \tag{5}$$

$$\begin{aligned}\Pr[E_{21}|E_2] &= (1-\varepsilon)^{K-1} + \sum_{j=\ell+1}^{K-1} \varepsilon(1-\varepsilon)^{j-1} \\ &= (1-\varepsilon)^{\ell}. \end{aligned} \tag{6}$$

That is,

$$\Pr[E_{22}|E_2] = \frac{\varepsilon}{1-\varepsilon}\Pr[E_{21}|E_2].$$

Let $q \stackrel{\triangle}{=} \Pr[E_{21}|E_2]$. Then,

$$\begin{aligned}\Pr[v \in \mathcal{B}|E_2] &= \Pr[v \in \mathcal{B}|E_{21} \cap E_2]\Pr[E_{21}|E_2] + \Pr[v \in \mathcal{B}|E_{22} \cap E_2]\Pr[E_{22}|E_2] \\ &\quad + \Pr[v \in \mathcal{B}|E_{23} \cap E_2]\Pr[E_{23}|E_2] \\ &= 0 \times q + 1 \times \frac{\varepsilon q}{1-\varepsilon} + (\varepsilon + P_K|\mathbf{B}(v, K)|)\left(1 - \frac{q}{1-\varepsilon}\right) \\ &= \varepsilon + P_K|\mathbf{B}(v, K)| + \frac{q}{1-\varepsilon}(\varepsilon - \varepsilon - P_K|\mathbf{B}(v, K)|) \\ &= \varepsilon + P_K|\mathbf{B}(v, K)| - \frac{qP_K|\mathbf{B}(v, K)|}{1-\varepsilon} \\ &\leq \varepsilon + P_K|\mathbf{B}(v, K)|. \end{aligned} \tag{7}$$

Case 3. Now, suppose $u_0 \neq v$ is such that $\mathbf{d_G}(u_0, v) = K$. We will call this event E_3. Further, define the event $E_{31} = \{Q_0 = K\}$. Due to the independence of selection of Q_0, $\Pr[E_{31}|E_3] = P_K$. Under the event $E_{31} \cap E_3$, $v \in \mathcal{B}$ with probability 1. Therefore,

$$\begin{aligned}\Pr[v \in \mathcal{B}|E_3] &= \Pr[v \in \mathcal{B}|E_{31} \cap E_3]\Pr[E_{31}|E_3] + \Pr[v \in \mathcal{B}|E_{31}^c \cap E_3]\Pr[E_{31}^c|E_3] \\ &= 1 \times P_K + \Pr[v \in \mathcal{B}|E_{31}^c \cap E_3](1 - P_K). \end{aligned} \tag{8}$$

Under the event $E_{31}^c \cap E_3$, we have $v \in \mathcal{W}_1$, and the remaining metric space $(\mathcal{W}_1, \mathbf{d_G})$. This metric space has $< N$ points. Further, the ball of radius K around v with respect to this new metric space has at most $|\mathbf{B}(v, K)| - 1$ points (this ball is with respect to the original metric space $\mathcal{G}$ on N points). Now we can invoke induction hypothesis for this new metric space (because of similar justification as in the previous case) to obtain

$$\Pr[v \in \mathcal{B}|E_{31}^c \cap E_3] \;\leq\; \varepsilon + P_K(|\mathbf{B}(v, K)| - 1). \tag{9}$$

[3]Note the following subtle but crucial point. We are not changing the metric $\mathbf{d_G}$ after we remove points from original set of points as part of the **Decomposition algorithm**.

From (8) and (9), we have

$$\begin{aligned}\Pr[v \in \mathcal{B}|E_3] &\leq P_K + (1-P_K)(\varepsilon + P_K(|\mathbf{B}(v,K)|-1)) \\ &= \varepsilon(1-P_K) + P_K|\mathbf{B}(v,K)| + P_K^2(1-|\mathbf{B}(v,K)|) \\ &\leq \varepsilon + P_K|\mathbf{B}(v,K)|. \end{aligned} \tag{10}$$

Case 4. Finally, let E_4 be the event that $\mathbf{d_G}(u_0, v) > K$. Then, at the end of the first iteration of the algorithm, we again have the remaining metric space $(\mathcal{W}_1, \mathbf{d_G})$ such that $|\mathcal{W}_1| < N$. Hence, as before, by induction hypothesis we have

$$\Pr[v \in \mathcal{B}|E_4] \leq \varepsilon + P_K|\mathbf{B}(v,K)|.$$

Now, the four cases are exhaustive and disjoint. That is, $\cup_{i=1}^4 E_i$ is the universe. Based on the above discussion, we obtain the following.

$$\begin{aligned}\Pr[v \in \mathcal{B}] &= \sum_{i=1}^4 \Pr[v \in \mathcal{B}|E_i]\Pr[E_i] \\ &\leq \left(\max_{i=1}^4 \Pr[v \in \mathcal{B}|E_i]\right)\left(\sum_{i=1}^4 \Pr[E_i]\right) \\ &\leq \varepsilon + P_K|\mathbf{B}(v,K)|. \end{aligned} \tag{11}$$

This completes the proof of Claim 6. ■

Now, we will use Claim 6 to complete the proof of (a). The definition of growth rate implies that,

$$|\mathbf{B}(v,K)| \leq C \cdot K^\rho.$$

From the definition $P_K = (1-\varepsilon)^{K-1}$, we have

$$P_K\,|\mathbf{B}(v,K)| \leq C(1-\varepsilon)^{K-1}K^\rho.$$

Therefore, to show (a) of Definition 4, it is sufficient to show that our definition of K satisfies the following Lemma.

Lemma 7: We have that

$$C(1-\varepsilon)^{K-1}K^\rho \leq \varepsilon.$$

Proof of Lemma 7:

We will show the following equivalent inequality.

$$(K-1)\log(1-\varepsilon)^{-1} \geq \rho\log K + \log C + \log\frac{1}{\varepsilon}. \tag{12}$$

First, note that for all $\varepsilon \in (0,1)$,

$$\log(1-\varepsilon)^{-1} \geq \log(1+\varepsilon) \geq \frac{\varepsilon}{2}.$$

Hence to prove (12), it is sufficient to show that

$$K \geq \frac{2\rho}{\varepsilon}\log K + \frac{2}{\varepsilon}\log C + \frac{2}{\varepsilon}\log\frac{1}{\varepsilon} + 1. \tag{13}$$

Recall that

$$K = K(\varepsilon,\rho) = \frac{8\rho}{\varepsilon}\log\left(\frac{8\rho}{\varepsilon}\right) + \frac{4}{\varepsilon}\log C + \frac{4}{\varepsilon}\log\frac{1}{\varepsilon} + 2.$$

From the definition of K, we will show that

$$\frac{K}{2} \geq \frac{2\rho}{\varepsilon}\log K$$

and

$$\frac{K}{2} \geq \frac{2}{\varepsilon}\log C + \frac{2}{\varepsilon}\log\frac{1}{\varepsilon} + 1,$$

which will prove (13).

The following is trivial.

$$\frac{K}{2} \geq \frac{2}{\varepsilon}\log C + \frac{2}{\varepsilon}\log\frac{1}{\varepsilon} + 1. \tag{14}$$

Now, let $\hat{K} = \frac{8\rho}{\varepsilon}\log\left(\frac{8\rho}{\varepsilon}\right)$. Then

$$\frac{\hat{K}}{2} = \frac{4\rho}{\varepsilon}\log\left(\frac{8\rho}{\varepsilon}\right) \geq \frac{2\rho}{\varepsilon}\left(\log\left(\frac{8\rho}{\varepsilon}\right) + \log\log\left(\frac{8\rho}{\varepsilon}\right)\right) = \frac{2\rho}{\varepsilon}\log\hat{K}.$$

I.e., $\frac{\hat{K}}{2} - \frac{2\rho}{\varepsilon}\log\hat{K} \geq 0$. Since the function $\phi(x) = \frac{x}{2} - \frac{2\rho}{\varepsilon}\log x$ is an increasing function of x when $x \geq \frac{4\rho}{\varepsilon}$, and from the fact that $K \geq \hat{K} \geq \frac{4\rho}{\varepsilon}$, we have

$$\frac{K}{2} \geq \frac{2\rho}{\varepsilon}\log K. \tag{15}$$

From (14) and (15), we have (13), which completes the proof of Lemma 7. ■

Proof of (b). First we give some notations. Define $R_t = \mathcal{R}_t - \mathcal{R}_{t-1}$, $B_t = \mathcal{B}_t - \mathcal{B}_{t-1}$ and

$$\partial R_t = \{v \in V : v \notin R_t \text{ and } \exists\ v' \in R_t \text{ s.t. } \mathbf{d_G}(v, v') = 1\}.$$

The followings are straightforward observations implied by the **Decomposition algorithm**: for any $t \geq 0$,

(i) $R_t \cap \mathcal{R}_{t-1} = \emptyset$,
(ii) $B_t \cap \mathcal{B}_{t-1} = \emptyset$,
(iii) $R_t \subset \mathbf{B}(u_{t-1}, Q_{t-1})$,
(iv) $B_t \subset \mathbf{B}(u_{t-1}, Q_{t-1}+1) - \mathbf{B}(u_{t-1}, Q_{t-1})$.

Now, we state and prove a crucial claim for proving (b).

Claim 8: For all $t \geq 0$, $\partial R_t \subset \mathcal{B}_t$.

Proof: We prove the Claim 8 by induction. Initially, $\partial R_0 = \mathcal{B}_0 = \emptyset$ and hence the claim is trivial. At the end of the first iteration, by the definition of the algorithm,

$$R_1 = \mathcal{R}_1 = \mathbf{B}(u_0, Q_0), \text{ and } B_1 = \mathcal{B}_1 = \mathbf{B}(u_0, Q_0+1) - \mathbf{B}(u_0, Q_0).$$

Therefore, by definition, $\partial R_1 = \mathcal{B}_1$. Thus, the base case of induction is verified. Now, as the hypothesis for induction, suppose that $\partial R_t \subset \mathcal{B}_t$ for all $t \leq \ell$, for some $\ell \geq 1$. As induction step, we will establish that $\partial R_{\ell+1} \subset \mathcal{B}_{\ell+1}$.

Suppose to the contrary, that $\partial R_{\ell+1} \not\subset \mathcal{B}_{\ell+1}$. That is, there exists $v \in \partial R_{\ell+1}$ such that $v \notin \mathcal{B}_\ell$. By definition of the algorithm, we have

$$R_{\ell+1} = \mathbf{B}(u_\ell, Q_\ell) - (\mathcal{R}_\ell \cup \mathcal{B}_\ell).$$

Therefore,

$$\partial R_{\ell+1} \subset (\mathbf{B}(u_\ell, Q_\ell+1) - \mathbf{B}(u_\ell, Q_\ell)) \cup \mathcal{R}_\ell \cup \mathcal{B}_\ell.$$

Again, by the definition of the algorithm we have

$$B_{\ell+1} = \mathbf{B}(u_\ell, Q_\ell+1) - \mathbf{B}(u_\ell, Q_\ell) - \mathcal{R}_\ell - \mathcal{B}_\ell.$$

Therefore, $v \in B_{\ell+1}$ or $v \in \mathcal{R}_\ell \cup \mathcal{B}_\ell$. Recall that by the definition of algorithm $\mathcal{B}_\ell \cap \mathcal{R}_\ell = \emptyset$. Since we have assumed that $v \notin \mathcal{B}_{\ell+1}$, it must be that $v \in \mathcal{R}_\ell$. That is, there exists $\ell' \leq \ell$ such that $v \in R_{\ell'}$. Now since $v \in \partial R_{\ell+1}$ by assumption, it must be that there exists $v' \in R_{\ell+1}$ such that $\mathbf{d_G}(v, v') = 1$. Since by definition $R_{\ell+1} \cap R_{\ell'} = \emptyset$, we have $v' \in \partial R_{\ell'}$. By induction hypothesis, this implies that $v' \in \mathcal{B}_{\ell'} \subset \mathcal{B}_\ell$. That is, $\mathcal{B}_\ell \cap R_{\ell+1} \neq \emptyset$, which is

a contradiction to the definition of our algorithm. That is, our assumption that $\partial R_{\ell+1} \not\subset \mathcal{B}_{\ell+1}$ is false. Thus, we have established the inductive step. This completes the induction argument and proof of the Claim 8. ■

Now when the algorithm terminates (which must happen within n iterations), say the output set is $\mathcal{B}_T$ and $V - \mathcal{B}_T = \mathcal{R}_T$ for some T. As noted above, $\mathcal{R}_T$ is a union of disjoint sets $R_1, \ldots, R_T$. We want to show that R_i, R_j are disconnected for any $1 \le i < j \le T$ using Claim 8. Suppose to the contrary that they are connected. That is, there exists $v \in R_i$ and $v' \in R_j$ such that $\mathbf{d_G}(v, v') = 1$. Since $R_i \cap R_j = \emptyset$, it must be that $v' \in \partial R_i, v \in \partial R_j$. From Claim 8 and fact that $\mathcal{B}_t \subset \mathcal{B}_{t+1}$ for all t, we have that $R_i \cap \mathcal{B} \neq \emptyset, R_j \cap \mathcal{B} \neq \emptyset$. This is contrary to the definition of the algorithm. Thus, we have established that $R_1, \ldots, R_T$ are disconnected components whose union is $V - \mathcal{B}_T$. By definition, each of $R_i \subset \mathbf{B}(u_{i-1}, K)$. Thus, we have established that $V - \mathcal{B}_T$ is made of connected components, each of which is contained inside balls of radius K with respect to $\mathbf{d_G}$. From the definition of the growth rate of a graph, this completes the proof of (b) and that of Lemma 5.

■

B. Step 2: Approximate MWIS

Now, the following randomized algorithm outputs a solution which is an ε-approximation of MWIS in expectation for any graph whenever its graph decomposition subroutine is an (ε, Δ)-decomposition for some constant $\Delta > 0$. Its running time depends on Δ, and it is $O(n)$ for any graphs with constant growth ratio ρ, and constant ε.

MWIS Approximation (ε, K)

(1) For the given graph G, use the **Decomposition algorithm** with parameters (ε, K) to a obtain decomposition of G.
  (a) Let $\mathcal{B}$ be the output of the decomposition algorithm.
  (b) $V - \mathcal{B}$ is divided into connected components with vertex sets $\mathcal{R}_1, \ldots, \mathcal{R}_L$ (L is some integer).
  (c) Let $G_1 = (\mathcal{R}_1, E_1), \ldots, G_L = (\mathcal{R}_L, E_L)$ be the corresponding disjoint subgraphs of G.
  (d) Let $\mathcal{I}(G_1), \ldots, \mathcal{I}(G_L)$ be set of independent sets of $G_1, \ldots, G_L$ respectively.

(2) For $\ell = 1, \ldots, L$ find
$$\mathbf{x}^*(G_\ell) \in \arg\max \left\{ \mathbf{w}^T \mathbf{x} : \mathbf{x} \in \mathcal{I}(G_\ell) \right\}.$$
  (a) The above computation can be done by dynamic programming in $O(2^{|\mathcal{R}_\ell|})$ operations for graph G_ℓ.

(3) Output $\hat{\mathbf{x}} = \cup_{\ell=1}^{L} \mathbf{x}^*(G_\ell)$ as an approximate maximum weight independent set of G.

Here we give some definition before we state our Lemma. Given a graph with growth rate ρ, and a constant $\delta \in (0, 1)$, let
$$\varepsilon(\delta) = \frac{\delta}{2} \qquad \text{and} \qquad K(\delta) = \frac{16\rho}{\delta} \log\left(\frac{16\rho}{\delta}\right) + \frac{8}{\delta} \log C + \frac{8}{\delta} \log \frac{2}{\delta} + 2.$$

Lemma 9: Let $\delta \in (0, 1)$ be a fixed constant. If the growth rate ρ is a constant, **MWIS Approximation** with parameters $(\varepsilon(\delta), K(\delta))$ takes $O(n)$ time and it generates independent set with weight at least $(1 - \delta)\mathbf{w}^T \mathbf{x}^*$ in expectation.

Proof: First, we state and prove a Lemma that evaluates the performance of the **MWIS Approximation** for any graph.

Lemma 10: Given weighted graph G, ε and K, the output $\hat{\mathbf{x}}$ of the **MWIS Approximation** has the following property:
$$\mathbb{E}\left[\mathbf{w}^T \hat{\mathbf{x}}\right] \ge \mathbf{w}^T \mathbf{x}^* \left[1 - \varepsilon - P_K\left(\max_{v \in V} |\mathbf{B}(v, K)|\right)\right].$$

Proof: Let the **Decomposition algorithm** applied to G produce a random subset $\mathcal{B} \subset V$ as its output and $V - \mathcal{B}$ be divided into connected components, $\mathcal{R}_1, \ldots, \mathcal{R}_L$ (as per Lemma 5). As described in the **Decomposition algorithm**, we have induced (disconnected) subgraphs $G_\ell = (\mathcal{R}_\ell, E_\ell)$, $1 \le \ell \le L$. The output of the **MWIS**

Approximation is

$$\hat{\mathbf{x}} = \cup_{\ell=1}^{L} \mathbf{x}^*(G_\ell),$$

where $\mathbf{x}^*(G_\ell)$ is the solution of maximum weight independent set in the subgraph G_ℓ. In the above expression, the $\cup$ operation of $\mathbf{x}^*(G_\ell)$'s means the standard union of sets. Note that $\mathbf{x}^*(G_\ell)$ are disjoint sets since G_ℓ are disconnected graphs. Therefore,

$$\mathbf{w}^T\hat{\mathbf{x}} \;\; = \;\; \sum_{\ell=1}^{L} \mathbf{w}^T\mathbf{x}^*(G_\ell). \tag{16}$$

Now consider any maximum weight independent set $\mathbf{x}^*$, i.e. $\mathbf{x}^* \in \arg\max \left\{\mathbf{w}^T\mathbf{x} : \mathbf{x} \in \mathcal{I}(G)\right\}$. The $\mathbf{x}^*$ corresponds to a subset of vertices V of G. The intersection of $\mathbf{x}^*$ and $\mathcal{R}_\ell$ induces an independent set in G_ℓ. Let

$$\mathbf{x}^*_\ell = \mathbf{x}^* \cap \mathcal{R}_\ell.$$

Since $\mathbf{x}^*(G_\ell)$ is a maximum weight independent set in G_ℓ, we have

$$\mathbf{w}^T\mathbf{x}^*(G_\ell) \;\; \geq \;\; \mathbf{w}^T\mathbf{x}^*_\ell. \tag{17}$$

Finally, denote $\mathbf{x}^*_{\mathcal{B}} = \mathbf{x}^* \cap \mathcal{B}$. Since $V = \mathcal{B} \cup \left(\cup_{\ell=1}^{L} \mathcal{R}_\ell\right)$ and all sets $\mathcal{B}, \mathcal{R}_1, \ldots, \mathcal{R}_L$ are disjoint, we have

$$\mathbf{w}^T\mathbf{x}^* \;\; = \;\; \mathbf{w}^T\mathbf{x}^*_{\mathcal{B}} + \sum_{\ell=1}^{L} \mathbf{w}^T\mathbf{x}^*_\ell. \tag{18}$$

From (16)-(18), we obtain

$$\mathbf{w}^T\hat{\mathbf{x}} \geq \mathbf{w}^T\mathbf{x}^* - \mathbf{w}^T\mathbf{x}^*_{\mathcal{B}}.$$

Therefore, with respect to the randomness in the **Decomposition algorithm** (using Lemma 5), we have

$$\begin{aligned}
\mathbb{E}\left[\mathbf{w}^T\hat{\mathbf{x}}\right] &\geq \mathbf{w}^T\mathbf{x}^* - \sum_{v\in\mathbf{x}^*} w_v \Pr[v \in \mathcal{B}] \geq \mathbf{w}^T\mathbf{x}^* - \left(\sum_{v\in\mathbf{x}^*} w_v \left(\varepsilon + P_K\left(\max_{v\in V} |\mathbf{B}(v,K)|\right)\right)\right) \\
&= \mathbf{w}^T\mathbf{x}^* - \left(\varepsilon + P_K\left(\max_{v\in V} |\mathbf{B}(v,K)|\right)\right)\left(\sum_{v\in\mathbf{x}^*} w_v\right) \\
&= \mathbf{w}^T\mathbf{x}^*\left[1 - \varepsilon - P_K\left(\max_{v\in V} |\mathbf{B}(v,K)|\right)\right].
\end{aligned} \tag{19}$$

This completes the proof of Lemma 10. ■

From Lemma 10, we obtain that our algorithm outputs a random independent set $\mathbf{x}$ such that

$$\mathbb{E}\left[\mathbf{w}^T\hat{\mathbf{x}}\right] \geq \mathbf{w}^T\mathbf{x}^*(1-\phi),$$

where $\phi = \varepsilon + P_K\left(\max_{v\in V} |\mathbf{B}(v,K)|\right)$.

For a graph G with growth ratio ρ and corresponding constant C, from the definition $K = K(\delta) = \frac{16\rho}{\delta}\log\left(\frac{16\rho}{\delta}\right) + \frac{8}{\delta}\log C + \frac{8}{\delta}\log\frac{2}{\delta} + 2$, and Lemma 7, we have

$$\phi = \varepsilon + P_K\left(\max_{v\in V} |\mathbf{B}(v,K)|\right) \leq 2\varepsilon.$$

Hence $\phi \leq 2\varepsilon = \delta$, which shows that our algorithm with parameter $(\varepsilon(\delta), K(\delta))$ generates independent set with average weight at least $(1-\delta)\mathbf{w}^T\mathbf{x}^*$.

To show that our algorithm runs in $O(n)$ time, note that by Definition 3 and Lemma 5, $(CK)^\rho$ is an upper bound on the size of each connected component obtained by the **Decomposition algorithm**. Then the running time for each connected component is at most $O\left(2^{(CK)^\rho}\right)$. Since K is a constant when ρ, C and δ are constants, and the total running time of the **MWIS Approximation** is $O\left(n2^{(CK)^\rho}\right) = O(n)$. This completes the proof of Lemma 9. ■

Here we note that use of the standard Markov's inequality will imply that by running **MWIS Approximation**

for $O(\log n)$ many times and taking the best solution, we can convert the average guarantee of the approximation to the high probability guarantee.

V. Derandomization

The **MWIS Approximation** presented so far is a randomized algorithm and provides performance guarantee in expectation. The randomness in the algorithm stems from the **Decomposition algorithm**. Therefore, to obtain a deterministic algorithm, we will de-randomized the decomposition algorithm. Precisely, given $\varepsilon > 0$ and a graph $G = (V, E)$ with constant growth ratio ρ, we will deterministically construct subsets $\mathcal{B}_1, \ldots, \mathcal{B}_m$ of V, where m is $poly(n)$, and assign a distribution μ on these subsets so that when a random subset, say $\mathcal{B}$, is chosen from $\mathcal{B}_1, \ldots, \mathcal{B}_m$ at random according to μ, the $\mathcal{B}$ satisfies properties (a) and (b) of Definition 4. Then we can apply Lemma 9 to this probability distribution μ, and we obtain that the output of **MWIS Approximation** with respect to μ provides ε-approximation of MWIS in expectation. Hence, we know that there is at least one $\mathcal{B}_i$ among $\mathcal{B}_1, \ldots, \mathcal{B}_m$, so that when we compute **MWIS Approximation** with this graph decomposition $\mathcal{B}_i$, the output independent set has weight at least $(1-\varepsilon)\mathbf{w}^T\mathbf{x}^*$. Hence, by computing **MWIS Approximation** for each of $\mathcal{B}_1, \ldots, \mathcal{B}_m$ one by one by and taking the best solution, i.e. an independent set with highest weight, we obtain ε-approximation of MWIS with probability 1 in $\mathsf{poly}(n)$ time.

A. Construction of Derandomization

In this section, we explain the construction of desired $\mathcal{B}_i$'s and then μ. To this end, first we introduce the following definition.

Definition 5: For a given graph $G = (V, E)$, and $r \in \mathbb{N}$, we say a subset Y of V is an *r-net* of G if Y satisfies

1) for every $y_1, y_2 \in Y$ with $y_1 \neq y_2$, $\mathbf{d_G}(y_1, y_2) \geq r$,
2) for every $x \in V$, there is $y \in Y$ such that $\mathbf{d_G}(x, y) < r$.

It can be easily check that one can construct an r-net of a graph $G = (V, E)$ in a greedy manner.

Now note that in our **Decomposition Algorithm**, there are two random choices: (i) selection of center points u_t, and (ii) selection of the random radius $Q_t \in \{1, \cdots, K\}$ for u_t. It can be checked that in the proof of Lemma 5 we never use the randomness of (i), i.e. any choice of u_t will make the Lemma remain true. So the main issue to tackle here is the choice of Q_t so that

1) the resulting support of the output sets $\mathcal{B}$ according this random distribution μ is $\mathsf{poly}(n)$, and
2) Lemma 5 is satisfied.

For this, we essentially reduce the overall randomness by *coupling* choice of random radius for *far away* vertices in V. Such approach is likely to work as one may observe that in our **Decomposition Algorithm**, if two vertices are farther from each other then $4K$ and they are chosen as centers, then their choice of radius does not affect each other for their immediate neighborhood selection. Next, we formalize this intuition.

Remind that given a constant $\varepsilon > 0$ and a graph G with growth rate ρ, K is defined as

$$K = K(\varepsilon, \rho, C) = \frac{8\rho}{\varepsilon} \log\left(\frac{8\rho}{\varepsilon}\right) + \frac{4}{\varepsilon} \log C + \frac{4}{\varepsilon} \log \frac{1}{\varepsilon} + 2,$$

and probability distribution $\mathbf{Q}$ over $\{1, \ldots, K\}$ is defined as as

$$\Pr[\mathbf{Q} = i] = \begin{cases} \varepsilon(1-\varepsilon)^{i-1} & \text{if } 1 \leq i < K \\ (1-\varepsilon)^{K-1} & \text{if } i = K \end{cases}.$$

First, divide V into disjoint set of vertices $V_1, \ldots, V_\ell$ so that each V_i becomes a $4K$-*net* of G. We can do this process in $O\left(n^2\right)$ time by choosing $V_1, \ldots V_\ell$ one by one (in the original metric space $\mathbf{d_G}$). In the next section, we will show that $\ell \leq (4K)^\rho$ for any partition of V into $4K$-nets. Then we fix an ordering $\mathbf{O}$ of vertices V so that we choose all the vertices in V_1 first, then choose all the vertices of V_2, and so on.

Now, consider $\bar{Q} = (Q_1, \ldots, Q_\ell) \in \{1, \ldots, K\}^\ell$. We will define a decomposition $\mathcal{B}(\bar{Q})$ for each $\bar{Q}$. They are the decompositions we wanted to obtain. Here, Q_i will do the role of radius for all the vertices of V_i in the process

of **Decomposition Algorithm**. We define the probability distribution μ on those $(Q_1, \ldots, Q_\ell) \in \{1, \ldots, K\}^\ell$ as follows.

$$\Pr[Q_1 = i_1, \ldots, Q_\ell = i_\ell] \propto \prod_{j=1}^{\ell} \Pr(\mathbf{Q} = i_j).$$

Now, for each $\bar{Q} = (Q_1, \ldots, Q_\ell) \in \{1, \ldots, K\}^\ell$, we will obtain a corresponding decomposition $\mathcal{B}(\bar{Q})$ by running the following modified version of the **Decomposition Algorithm**.

Modified Decomposition Algorithm $(\varepsilon, K, \bar{Q})$

(1) Initially, set $\mathcal{W} = V$, $\mathcal{B} = \emptyset$ and $\mathcal{R} = \emptyset$.
(2) Repeat the following till $\mathcal{W} \neq \emptyset$:
  (a) **Choose the smallest element** $u \in \mathcal{W}$ **by the order O.**
  (b) **If** $u \in V_i$**, let** $Q = Q_i$**.**
  (c) Update
    (i) $\mathcal{B} \leftarrow \mathcal{B} \cup \{w | \mathbf{d_G}(u, w) = Q \text{ and } w \in \mathcal{W}\}$,
    (ii) $\mathcal{R} \leftarrow \mathcal{R} \cup \{w | \mathbf{d_G}(u, w) < Q \text{ and } w \in \mathcal{W}\}$,
    (iii) $\mathcal{W} \leftarrow \mathcal{W} \cap (\mathcal{B} \cup \mathcal{R})^c$.
(3) Output $\mathcal{B}(\bar{Q})$.

B. Correctness of Derandomization

Now we show that the number of graph decompositions we have obtained is dependent only on ε and ρ.

Lemma 11: We have

$$\ell \leq (4K)^\rho,$$

which implies that the number of graph decompositions obtained by the modified process is $K^\ell \leq K^{(4K)^\rho}$, which is independent of n.

Proof: To prove that $\ell \leq (4K)^{\rho+1}$, fix any vertex w in V_ℓ. Since V_j is a $4K$-net of the metric space $(V^{(j-1)}, \mathbf{d_G})$ for any $1 \leq j \leq \ell$, and $w \in V_\ell \subset V^{(j-1)}$, from the definition 5 we have

$$\mathbf{B}_G(w, 4K) \cap V_j \neq \emptyset.$$

Hence we have

$$\ell \leq |\mathbf{B}_G(w, 4K)| \leq (4K)^\rho$$

by the definition of the growth rate. ■

Now, we claim the following Lemma. It essentially states that if we consider the new probability distribution μ on the graph decomposition, the Lemma 5 still holds. Therefore, the above discussion will immediately imply that all our approximation guarantee can be achieved deterministically in $\mathsf{poly}(n)$ time.

Lemma 12: Given a graph G with growth rate $\rho = \rho(G)$ and the corresponding constant C, and $\varepsilon \in (0, 1)$, the random graph decomposition according to the distribution μ becomes a $(2\varepsilon, CK^\rho)$-decomposition of G.

Proof: To prove that the random graph decomposition $\mathcal{B} \subset V$ according to μ becomes a a $(2\varepsilon, CK^\rho)$-decomposition, we need to establish properties (a) and (b) of Definition 4.

Proof of (b) is essentially the same as that of Lemma 5(b). Now, to prove (a) we will use argument similar to that in the proof of Lemma 5(a). Next, we state and prove Claim 13, which will imply the proof of Lemma 12(a) by Lemma 7.

Claim 13: Consider a metric space $\mathcal{G} = (V, \mathbf{d_G})$ with $|V| = n$. Let $K > 1$ be an integer and $\mathcal{B} \subset V$ be the random decomposition chosen randomly among $\mathcal{B}_1, \mathcal{B}_2 \ldots, \mathcal{B}_m$ according to μ. Then, for any $v \in V$,

$$\Pr[v \in \mathcal{B}] \leq \varepsilon + P_K |\mathbf{B}_G(v, K)|,$$

where $\mathbf{B}_G(v, K)$ is the ball of radius K with respect to $\mathbf{d_G}$, and $P_K = (1-\varepsilon)^{K-1}$.

Proof: Let $K > 1$ be fixed. As per the algorithm description, the output set $\mathcal{B}$ is generated according to μ by choosing centers in the **Decomposition Algorithm** from sets $V_1, \ldots, V_\ell$ and using the random radius $(Q_1, \ldots, Q_\ell)$. First, we have that the sets $V_i \subset V$ are disjoint, $V = \cup_{i=1}^{\ell} V_i$, and for any $x, y \in V_j$, $\mathbf{d_G}(x, y) \geq 4K$. In what follows, we will call a subset $U \subset V$ as $4K$-*sparse* if for $x, y \in U$, $\mathbf{d}(x, y) \geq 4K$. Thus, all $V_i, 1 \leq i \leq \ell$ are $4K$-sparse. We prove the Claim 13 for any finite metric space (note that the **Modified Decomposition Algorithm** can be applied for any finite metric space) by induction over ℓ, the number of disjoint $4K$-spare subsets the metric space $\mathcal{G} = (V, \mathbf{d_G})$ decomposes itself into.

Base case. Suppose $\ell = 1$. That is, the metric space is such that V, the set of all nodes, is $4K$-sparse. Then, as per algorithm, no vertex will become a part of the output set $\mathcal{B}$ since the radius will be at most K, we have that for any $v \in V$,

$$\Pr[v \in \mathcal{B}] = 0 \leq \varepsilon + P_K |\mathbf{B}(v, K)|.$$

Thus, we have verified the base case for induction with $\ell = 1$.

Induction hypothesis. Now, suppose that the Claim 13 holds for any metric space that can be decomposed into $\ell \leq N$ $4K$-sparse sets, for some $N \geq 1$.

Induction step. We wish to show that the Claim 13 holds for any metric space that can be decomposed into $N+1$ $4K$-spare sets. To this end, let $\mathcal{G} = (V, \mathbf{d_G})$ with $V = \cup_{i=1}^{N+1} V_i$, with V_i be $4K$-sparse for $1 \leq i \leq N+1$.

Now, recall that the **Modified Decomposition Algorithm** operates by first selecting vertices of V_1 as centers before choosing vertices from any $V_i, 1 < i \leq N+1$. Since, the randomly chosen radius $Q_1 \leq K$ and all the vertices in V_1 are at distance $4K$, we can essentially assume that all vertices of V_1 are chosen as centers simultaneously. At the end of this step, let $\mathcal{B}_1$, $\mathcal{R}_1$ and $\mathcal{W}_1$ be the resulting sets in the **Modified Decomposition Algorithm** (see the pseudo-code in Section V-A). Now consider a fixed vertex $v \in V$, and let u_1 be a vertex in V_1 that is closest to v (with arbitrary tie-breaking). Note that since V_1 is $4K$-sparse, there is at most one $w \in V_1$ so that $\mathbf{d_G}(v, w) \leq K$. Depending on $\mathbf{d_G}(u_1, v)$ we consider the following four different cases (or events).

Case 1. This case corresponds to event E_1 where u_1 is equal to point v of our interest. By the definition of the **Modified Decomposition Algorithm**, under the event E_1, v will never be part of output set $\mathcal{B}$. That is,

$$\Pr[v \in \mathcal{B} | E_1] = 0 \ \leq \ \varepsilon + P_K |\mathbf{B}(v, K)|.$$

Case 2. Now, suppose u_1 is such that $v \neq u_1$ and $\mathbf{d_G}(u_1, v) < K$. Call this event E_2. Further, depending on the choice of random number Q_1, define events

$$E_{21} = \{\mathbf{d_G}(u_1, v) < Q_1\}, \ E_{22} = \{\mathbf{d_G}(u_1, v) = Q_1\}, \text{ and } E_{23} = \{\mathbf{d_G}(u_1, v) > Q_1\}.$$

Description of the **Modified Decomposition Algorithm** implies that under E_{21}, v is selected as a part of $\mathcal{R}_1$. Hence it can never be a part of $\mathcal{B}$. When E_{22} happens, v is selected as part of $\mathcal{B}_1$ and hence it is definitely a part of $\mathcal{B}$. When E_{23} happens, v is neither selected in the set $\mathcal{R}_1$ nor selected in the set $\mathcal{B}_1$. It is left as an element of the set $\mathcal{W}_1$. Now, the original metric $\mathbf{d_G}$ is still a metric on points of $\mathcal{W}_1$. And for any radius value of Q_1, $\mathcal{W}_1$ is a subset of $\cup_{i=2}^{N+1} V_i$. That is, $\mathcal{W}_1$ is partitioned into N disjoint $4K$-sparse subsets. Given *Markovian* nature of the **Modified Decomposition Algorithm** and inductive hypothesis, it follows that conditional on event E_{23}, the probability that node $v \in \mathcal{B}$ is bounded above by $\varepsilon + P_K |\mathbf{B}(v, K)|$. Finally, the argument similar to that in the proof of Lemma 5 implies

$$\Pr[E_{22} | E_2] = \frac{\varepsilon}{1-\varepsilon} \Pr[E_{21} | E_2].$$

Let $q \stackrel{\triangle}{=} \Pr[E_{21}|E_2]$. Then,

$$\begin{aligned}
\Pr[v \in \mathcal{B}|E_2] &= \Pr[v \in \mathcal{B}|E_{21} \cap E_2] \Pr[E_{21}|E_2] + \Pr[v \in \mathcal{B}|E_{22} \cap E_2] \Pr[E_{22}|E_2] \\
&\quad + \Pr[v \in \mathcal{B}|E_{23} \cap E_2] \Pr[E_{23}|E_2] \\
&= 0 \times q + 1 \times \frac{\varepsilon q}{1-\varepsilon} + (\varepsilon + P_K|\mathbf{B}(v,K)|) \left(1 - \frac{q}{1-\varepsilon}\right) \\
&= \varepsilon + P_K|\mathbf{B}(v,K)| + \frac{q}{1-\varepsilon} (\varepsilon - \varepsilon - P_K|\mathbf{B}(v,K)|) \\
&= \varepsilon + P_K|\mathbf{B}(v,K)| - \frac{qP_K|\mathbf{B}(v,K)|}{1-\varepsilon} \\
&\leq \varepsilon + P_K|\mathbf{B}(v,K)|. \qquad (20)
\end{aligned}$$

Case 3. Now, suppose u_1 is such that $\mathbf{d_G}(u_1, v) = K$. Call this event E_3. Further, let event $E_{31} = \{Q_1 = K\}$. Due to the definition of μ, $\Pr[E_{31}|E_3] = P_K$. Under the event $E_{31} \cap E_3$, $v \in \mathcal{B}$ with probability 1. Therefore,

$$\begin{aligned}
\Pr[v \in \mathcal{B}|E_3] &= \Pr[v \in \mathcal{B}|E_{31} \cap E_3] \Pr[E_{31}|E_3] + \Pr[v \in \mathcal{B}|E_{31}^c \cap E_3] \Pr[E_{31}^c|E_3] \\
&= 1 \times P_K + \Pr[v \in \mathcal{B}|E_{31}^c \cap E_3](1 - P_K). \qquad (21)
\end{aligned}$$

Under the event $E_{31}^c \cap E_3$, we have $v \in \mathcal{W}_1$. As before, the remaining metric space $(\mathcal{W}_1, \mathbf{d_G})$ can be divided into N disjoint $4K$-sparse subsets. Therefore, applying induction hypothesis we obtain that

$$\Pr[v \in \mathcal{B}|E_{31}^c \cap E_3] \leq \varepsilon + P_K(|\mathbf{B}(v,K)| - 1). \qquad (22)$$

From (21) and (22), we have

$$\begin{aligned}
\Pr[v \in \mathcal{B}|E_3] &\leq P_K + (1 - P_K)(\varepsilon + P_K(|\mathbf{B}(v,K)| - 1)) \\
&= \varepsilon(1 - P_K) + P_K|\mathbf{B}(v,K)| + P_K^2(1 - |\mathbf{B}(v,K)|) \\
&\leq \varepsilon + P_K|\mathbf{B}(v,K)|. \qquad (23)
\end{aligned}$$

Case 4. Finally, let E_4 be the event that $\mathbf{d_G}(u_1, v) > K$. Then, clearly v is in $\mathcal{W}_1$ at the end of iterations of choosing all the nodes in V_1 as center vertices in the **Modified Decomposition Algorithm**. Again, invoking the induction hypothesis as before we obtain

$$\Pr[v \in \mathcal{B}|E_4] \leq \varepsilon + P_K|\mathbf{B}(v,K)|.$$

Now, the four cases are exhaustive and disjoint. That is, $\cup_{i=1}^4 E_i$ is the universe. Hence, based on the above discussion, we obtain the following.

$$\Pr[v \in \mathcal{B}] \leq \varepsilon + P_K|\mathbf{B}(v,K)|. \qquad (24)$$

This completes the proof of Claim 13. ■

■

VI. Discussion

As the main result of this paper, we established that graphs that are polynomially growing are algorithmically efficient as they allow for designing simple approximation schemes for maximum weight independent set. There are two natural future directions.

The first direction concerns identification of finer characterization of such efficient graph structure. Our result suggests that all the inefficient (or hard) network graph structures must be super-polynomially growing. However, if a network graph is a tree with constant degree, say 3, then it is exponentially (and hence super-polynomially) growing. But trivially, it has simple efficient algorithm (e.g. Belief Propagation) for solving the maximum weight independent set exactly ! Naturally, this demands a non-trivial investigation of the super-polynomially growing network graphs for finding efficient structures.

The second direction is about iterative implementation of our algorithm. Specifically, our algorithm as explained is two-stage mechanism: first, find a good decomposition and then utilize it to obtain a good solution. Ideally, we

would like to have a "merged" version of this two stage approach that has message-passing implementation with in-built randomization.

REFERENCES

[1] P. Assouad. Plongements lipschitziens dans $\mathbb{R}^n$. *Bull. Soc. Math. France*, 111(4):429–448, 1983.

[2] P. Chaporkar, K. Kar, and S. Sarkar. Throughput guarantees through maximal scheduling in wireless networks. In *43rd Allerton conference on Comm. Control and computing*, 2005.

[3] L. Chen, S. H. Low, M. Chang, and J. C. Doyle. Optimal cross-layer congestion control, routing and scheduling design in ad-hoc wireless networks. In *IEEE INFOCOM*, 2006.

[4] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*.

[5] Anupam Gupta, Robert Krauthgamer, and James R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *FOCS '03: Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, page 534, Washington, DC, USA, 2003. IEEE Computer Society.

[6] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[7] B Hajek and G. Sasaki. Link scheduling in polynomial time. *IEEE Trans. Inf. Theory*, 34, 1988.

[8] Juha Heinonen, editor. *Lectures on Analysis on Metric Spaces*. Springer, 2001.

[9] H. Hunt-III, M. Marathe, V. Radhakrishnan, S. Ravi, D. Rosenkrantz, and R. Stearns. Nc-approximation schemes for np- and pspace-hard problems for geometric graphs. *J. Algorithms*, 26(2):238–274, 1998.

[10] Kyomin Jung and Devavrat Shah. Low delay scheduling in wireless network. *IEEE International Symposium on Information Theory (ISIT)*, 2007.

[11] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 1998.

[12] X. Lin, N. Shroff, and R. Srikant. A tutorial on cross-layer optimization in wireless networks. *Submitted, available through* `csl.uiuc.edu/rsrikant`, 2006.

[13] X. Lin and N. B. Shroff. Impact of imperfect scheduling in wireless networks. In *IEEE INFOCOM*, 2005.

[14] Steven H. Low, Senior Member, and David E. Lapsley. Optimization flow control, i: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7:861–874, 1999.

[15] S. P. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Springer-Verlag, London, 1993.

[16] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless network via gossiping. In *ACM SIGMETRICS/Performance*, 2006.

[17] Saswati Sarkar and Saikat Ray. Arbitrary throughput versus complexity tradeoffs in wireless networks using graph partitioning. *IEEE Transactions on Automatic Control*, 53(10):2307–2323, 2008.

[18] Devavrat Shah. Randomization and heavy traffic theory: New approaches to the design and analysis of switch algorithms. *Ph.D Thesis*, 2004.

[19] Devavrat Shah, David Tse, and John N. Tsitsiklis. Hardness of low delay scheduling. *personal conversation*.

[20] Srinivas Shakkottai and R. Srikant. Network optimization and control. *Foundations and Trends in Networking*, 2(3):271–379, 2007.

[21] G. Sharma, R. Mazumdar, and N. Shroff. On the complexity of scheduling in wireless networks. In *ACM Mobicom*, 2006.

[22] Rayadurgam Srikant. *The Mathematics of Internet Congestion Control (Systems and Control: Foundations and Applications)*. SpringerVerlag, 2004.

[23] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37:1936–1949, 1992.

[24] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37:1936–1948, 1992.

[25] L. Trevisan. Non-approximability results for optimization problems on bounded degree instances. In *ACM STOC*, 2001.