University of Michigan–Ann Arbor

Department of Electrical Engineering and Computer Science

EECS 498 004 **Advanced Graph Algorithms**, Fall 2021

Lecture 1: Introduction to expanders

October 5, 2021

Instructor: Thatchaphol Saranurak

Scribe: Aditya Anand

1 Expanders: an introduction

Expanders are central objects in TCS and Math with a lot of applications. In this course, we will see how expanders are so powerful in the area of efficient graph algorithms. Expanders find a lot of applications in graph algorithms - many problems can be solved faster on expanders. We will see *expander decomposition* - the result that allows us to think of any graph as a collection of expanders connected by a small fraction of edges. This will allow us to transfer our study of expanders to general graphs - this nice structure can often be helpful in designing algorithms.

There are other areas where expanders appear a lot: coding theory, pseudo-randomness, PCP (probabilistic checkable proofs). Before we look at a formal definition of expanders, let us try to understand what an expander is "morally" - what are the properties, and what do expander graphs look like.

2 Expanders - sparse but well-connected!

Very simply put, an expander is a **sparse complete graph**! This sounds contradictory - but let us try to understand the context of this statement. What we mean to say is that expanders behave like complete graphs with respect to many properties, but they can be really sparse. Here are some views of expanders: each view captures "connectivity" in its own sense.

- **Cut view**: Robust against deletions
- **Flow view**: Allow all-to-all routing
- **Probabilistic view**: Random walks mix rapidly
- **Algebraic view**: Large spectral gap

Each of the above is useful, but it turns out that they are equivalent! They all characterize the same thing, expanders. In the first half of the course, we will study each of these views and understand how they are all equivalent in charactetrizing expanders. Along the way we will learn a lot of concepts like the flow-cut gap, metric embeddings, Laplacian of graphs, spectral gaps, random

Table 1: Characteristics of a *good* algorithm

Correct					
Fast					
Deterministic					
Parallel					
Distributed					
Dynamic					
Low space					
Simple to implement					
Easy to understand					
What else ???					

walks, etc. Some of the algorithmic applications which we will look at along the way include Multi-commodity flow, multicut, sparsest cut, vertex and edge connectivity. In the second half of the course, we will try to develop *good* algorithms based on expanders. For instance, we will look at graph sparsification from cut, flow and spectral perspectives. We will develop fast near-linear time algorithms for solving many graph problems based on our study of expanders using the key tool of *expander decomposition*, which says that every graph looks like a collection of expanders, with a small fraction of edges between the expanders.

What is a *good* algorithm though? What is the goal of an algorithm designer? Table 1 tries to list some of the properties that a *good* algorithm might hope to achieve. Discuss graph algorithms what you have learned before and try to rate them in terms of the parameters in Table 1.

We will look at some state of the art algorithms for solving fundamental graph problems in the second half of the course. The tentative goal in the second half of the course will include:

- $(1+\epsilon)$-approximate max flow in **undirected** graphs in near-linear time. [All exact max flow algorithms are still much slower.]
- $(1+\epsilon)$-approximate shortest path in **undirected** graphs in near-linear work, polylog depth. [Dijkstra's is not parallel]
- other things along the way... sparsifiers, oblivious routing, dynamic algorithms, etc.

3 Background

I will assume some background knowledge. You can always ask question but I might sometimes tell you where you can look up about these topics instead.

- Basic algorithms and data structures (sorting, binary search trees, graph searching, shortest path, max flow, etc).
- Basic graph theory

- Basic probability and concentration bounds (Markov, Chernoff)
- Linear programming and duality