DIMAP Workshop on Advances in Modern Graph Algorithms, Venice
April 8-11, 2019

Day 1, Monday, April 8

Distributed Exact Shortest Paths Algorithm in Sublinear Time

Michael Elkin, Ben-Gurion University of the Negev

The distributed single-source shortest paths problem is one of the most fundamental and central problems in the message-passing distributed computing. Classical Bellman-Ford algorithm solves it in $O(n)$ time, where n is the number of vertices in the input graph G. Peleg and Rubinovich, FOCS'99 showed a lower bound of $\Omega(D + \sqrt{n})$ for this problem, where D is the hop-diameter of G.

Whether or not this problem can be solved in $o(n)$ time when D is relatively small is a major notorious open question. Despite intensive research that yielded near-optimal algorithms for the *approximate* variant of this problem, no progress was reported for the original problem.

We answer this question in the affirmative, and devise an algorithm that requires $O((n \log n)^{5/6})$ time, for $D = O(\sqrt{n \log n})$, and $O(D^{1/3} \cdot (n \log n)^{2/3})$ time, for larger D. This running time is sublinear in n in almost the entire range of parameters.

We also devise the first algorithm with non-trivial complexity guarantees for computing exact shortest paths in the *multipass semi-streaming* model of computation.

From the technical viewpoint, our algorithm computes a *hopset* G'' of a skeleton graph G' of G *without first computing* G' itself. We then conduct a Bellman-Ford exploration in $G' \cup G''$, while computing the required edges of G' *on the fly*. As a result, our algorithm computes *exactly* those edges of G' that it really needs, rather than computing approximately the entire G'.

The talk will be self-contained. It is based on a paper that appeared in STOC'17.

A Tale of 3 Matrices

Keren Censor-Hillel, Technion

I will discuss two recent papers in which we give fast algorithms for multiplying sparse matrices in the congested clique model and how we exploit them for various graph computations, even for non-sparse graphs. Our results significantly improve upon the state-of-the-art for approximate APSP, MSSP and diameter, as well as exact SSSP.

Based on joint works with Michal Dory, Janne Korhonen, Dean Leitersdorf and Elia Turner.

Distributed Shortest Paths, Exactly

Danupon Nanongkai, KTH Royal Institute of Technology

This talk concerns the problem of quickly computing distances and shortest paths on distributed networks (the CONGEST model). There have been many developments for this problem in the last few year, resulting in tight approximation schemes. This left open whether exact algorithms can perform equally well. In this talk, I will (i) discuss some recent progress in answering this question, especially the recent near-linear time algorithm (joint work with Aaron Bernstein to appear in STOC 2019), and (ii) reflect on developed and missing techniques for solving graph problems in the CONGEST model in general.

Approximating APSP without Scaling

Karl Bringmann, MPI

Zwick's $(1+\varepsilon)$-approximation algorithm for the All Pairs Shortest Path problem runs in time $\tilde{O}(\frac{n^\omega}{\varepsilon}\log W)$, where $\omega \leq 2.373$ is the exponent of matrix multiplication and W denotes the largest weight. This can be used to approximate several graph characteristics including the diameter, radius, median, minimum-weight triangle, and minimum-weight cycle in the same time bound.

We study whether the factor $\log W$ can be avoided, that is, whether APSP and related problems admit *strongly polynomial* approximation schemes, whose running time is independent of W. For APSP on undirected graph as well as for several graph characteristics on directed or undirected graphs we remove the $\log W$-factor from Zwick's running time. For APSP on directed graphs, we design a strongly polynomial approximation scheme in time $\tilde{O}(n^{\frac{\omega+3}{2}}\varepsilon^{-1})$. We also explain why this has a worse exponent than ω: Any improvement over our exponent $\frac{\omega+3}{2}$ would improve the best known algorithm for MinMaxProduct. In fact, we prove that approximating directed APSP and exactly computing the MinMaxProduct are equivalent. Our techniques yield a framework for approximation problems over the (min,+)-semiring that can be applied more generally, for example to MinPlusConvolution.

Joint work with Marvin Künnemann and Karol Węgrzycki.

Counting (with) homomorphisms

Radu Curticapean, Basic Algorithms Research Copenhagen (BARC) and ITU Copenhagen

We survey classical, recent, and upcoming results on counting graph homomorphisms, a simple yet comprehensive framework that unifies various graph-theoretic counting problems, sometimes in unexpected ways.

In the generic problem for this framework, there are some objects on the left and some objects on the right. The objects on the left stand in relationships, as specified by a graph L. Likewise, the objects on the right share relationships specified by a graph R. Now, a homomorphism f from L to R is a function that maps the left objects to the right objects in a way that is compatible with their relationships: If uv is an edge in L, then $f(u)f(v)$ is required to be an edge in R.

Given L and R as input, counting homomorphisms from L to R is #P-complete, and thus at least NP-hard: The problem can express counting proper 3-colorings of graphs (by taking L as input and fixing R to a triangle) or counting cliques of a desired size (by fixing L to a complete graph and taking R as the input, assuming R has no self-loops). We study how the complexity of this problem varies under more general restrictions on L and R.

Concerning restrictions on R, Dyer and Greenhill proved a dichotomy for the complexity of counting homomorphisms to fixed graphs R almost 20 years ago: The problem is #P-complete unless R has a very simple structure. We revisit their result, sketching a new and simplified proof that also gives tight lower bounds under the exponential-time hypothesis ETH.

Concerning restrictions on L, it is clear that counting homomorphisms is polynomial-time solvable for every fixed L. In this setting, we wish to understand the exponent of optimal algorithms. A celebrated result by Marx shows that, assuming ETH, this exponent is the treewidth of L up to logarithmic factors. We survey how the complexity of counting general small subgraph patterns can be understood by expressing them as homomorphism counting problems and using bounds such as Marx's on the complexity of the relevant homomorphism problems.

While the settings of fixed L or fixed R may seem very different, we observe that they are both governed by the same connections between complexity-theoretic and algebraic properties of graph homomorphisms.

Based on joint works with Hubie Chen, Holger Dell, and Dániel Marx.

Some Connections Between the Conjectures in Fine-Grained Complexity

Amir Abboud, IBM Almaden Research Center

Fine-grained complexity utilizes a small set of conjectures to derive conditional lower bounds for a large collection of problems. These conjectures concern the time complexity of a few core problems such as k-SAT, Orthogonal Vectors, 3SUM, All Pairs Shortest Paths, and k-Clique. The relationships between these conjectures are poorly understood.

This talk will discuss some connections between the conjectures, including a tight reduction from Weighted-k-Clique to Orthogonal Vectors (joint work with Karl Bringmann, Holger Dell, and Jesper Nederlof) and new findings about the Set Cover Conjecture.

Day 2, Tuesday, April 9

Improved Approximation for Tree Augmentation: Saving by Rewiring

Fabrizio Grandoni, IDSIA, University of Lugano

In the Tree Augmentation Problem (TAP) we are given a tree $T = (V, E)$ and a set L of additional edges (links). Our goal is to find a minimum-cardinality subset of links L' such that the graph $(V, E \cup L')$ is 2-edge-connected. A long line of results on TAP culminated in the previously best known approximation guarantee of 1.5 achieved by a combinatorial approach due to Kortsarz and Nutov [ACM Transactions on Algorithms 2016], and also by an SDP-based approach by Cheriyan and Gao [Algorithmica 2017]. Moreover, an elegant LP-based $(1.5 + \epsilon)$-approximation has also been found very recently by Fiorini, Gross, Koenemann, and Sanità [SODA 2018]. In this talk, we show that an approximation factor below 1.5 can be achieved, by presenting a 1.458-approximation that is based on a novel rewiring Technique.

Joint work with Christos Kalaitzis and Rico Zenklusen.

Distributed Connectivity and a New Sampling Lemma

Valerie King, University of Victoria

We consider the following problem. Each node in a graph has a distinct ID and each knows only the ID's of its neighbors. Suppose it can send one message to a referee who must determine the graph's connected components. The graph sketching technique described by Ahn, Guha and McGregor in 2012 gives a method which requires only $O(\log^3 n)$ bits to be sent by each node, to compute the solution with high probability, and this is tight, according to a recent result of Nelson and Yu. However this method requires public randomness.

We began by investigating the one-way communication cost of this problem when there is private randomness, and ended up proving a surprising lemma about sampling in graphs and connectivity. This is joint work with Jacob Holm, Mikkel Thorup, Or Zamir, and Uri Zwick.

Longest Cycle above degeneracy

Fedor Fomin, University of Bergen

An undirected graph G is d-degenerate if every subgraph of G has a vertex of degree at most d. By the classical theorem of Erdös and Gallai from 1959, every graph of degeneracy $d > 1$ contains a cycle of length at least $d + 1$. The proof of Erdös and Gallai is constructive and can be turned into a polynomial time algorithm constructing a cycle of length at least $d + 1$. We prove that when the input graph is 2-connected, then deciding whether G contains a cycle of length at least $d + k$ can be done in time $2^{O(k)}|V(G)|^{O(1)}$.

Based on a joint work with Petr A. Golovach, Daniel Lokshtanov, Fahad Panolan, Saket Saurabh, and Meirav Zehavi.

Near-Optimal Approximate Decremental All Pairs Shortest Paths

Shiri Chechik, Tel Aviv University

Computing shortest paths is one of the fundamental problems of graph algorithms. The goal of *dynamic* all pairs shortest paths (APSP) is to maintain shortest path trees from all vertices as the edges of the graph change over time. The algorithm is said to be decremental if it handles only deletions, incremental if it handles only insertions and fully dynamic if it handles both deletions and insertions. In this talk I will present a near optimal decremental algorithm that maintains approximate all pairs shortest paths.

Planar Diameter via Metric Compression

Merav Parter, The Weizmann Institute of Science

We develop a new approach for distributed distance computation in planar graphs that is based on a variant of the metric compression problem recently introduced by Abboud et al. [SODA'18]. In our variant of the Planar Graph Metric Compression Problem, one is given an n-vertex planar graph $G = (V, E)$, a set of $S \subseteq V$ source terminals lying on a single face, and a subset of target terminals $T \subseteq V$. The goal is to compactly encode the $S \times T$ distances.

One of our key technical contributions is in providing a compression scheme that encodes all $S \times T$ distances using $\tilde{O}(|S| \cdot D^{O(1)} + |T|)$ bits, for unweighted graphs with diameter D. This significantly improves the state of the art of $\tilde{O}(|S| \cdot 2^D + |T| \cdot D)$ bits. We also consider an approximate version of the problem for *weighted* graphs, where the goal is to encode $(1+\epsilon)$ approximation of the $S \times T$ distances, for a given input parameter $\epsilon \in (0,1]$. Here, our compression scheme uses $\tilde{O}(\mathrm{poly}(|S|/\epsilon) + |T|)$ bits. In addition, we describe how these compression schemes can be computed in near-linear time. At the heart of this compact compression scheme lies a VC-dimension type argument on planar graphs, using the well-known Sauer's lemma.

This efficient compression scheme leads to several improvements and simplifications in the setting of diameter computation, most notably in the distributed setting: - There is an $\tilde{O}(D^5)$-round randomized distributed algorithm for computing the diameter in planar graphs, w.h.p. -There is an $\tilde{O}(D^3) + \mathrm{poly}(\log n/\epsilon) \cdot D^2$-round randomized distributed algorithm for computing an $(1+\epsilon)$ approximation of the diameter in weighted graphs with polynomially bounded weights, w.h.p.

No sublinear round algorithms were known for these problems before. These distributed constructions are based on a new recursive graph decomposition that preserves the (unweighted) diameter of each of the subgraphs up to a logarithmic term. Using this decomposition, we also get an *exact* SSSP tree computation within $\tilde{O}(D^2)$ rounds.

Joint work with Jason Li, to appear in STOC'19

Day 3, Wednesday, April 10

Online Algorithms via Projections

Seffi Naor, Technion

We consider the k-server problem on trees and HSTs and give an online algorithm based on Bregman projections. This algorithm has a competitive ratio that matches some of the recent results given by Bubeck et al. (STOC 2018), whose algorithm was based on mirror-descent-based continuous dynamics prescribed via a differential inclusion. We discuss the connections between this work and previous work on online primal-dual algorithms for several problems.

Joint work with Niv Buchbinder, Anupam Gupta, and Marco Molinaro.

Recent Advances in Massive Parallel Algorithms for Matchings and Vertex Cover

Krzysztof Onak, IBM T.J. Watson Research Center

Maximum Matching and Vertex Cover belong to most fundamental and well studied graph problems. In recent years, they both have received a significant amount of attention in the Massive Parallel Computation model, which is inspired by practical data processing systems such as MapReduce. An important parameter of the model is the amount of space per machine, which specifies the extent to which the computation can be distributed across several machines. I will survey recent advances in the number of processing rounds needed for approximate algorithms for Maximum Matching and Vertex Cover when the space per machine is linear or strongly sublinear in the number of vertices.

On Solving Linear Systems in Sublinear Time

Robert Krauthgamer, The Weizmann Institute of Science

I will discuss sublinear algorithms that solve linear systems locally. In the classical version of this problem, the input is a matrix S and a vector b in the range of S, and the goal is to output a vector x satisfying $Sx = b$.

We focus on computing one coordinate of x (approximately), which potentially allows for sublinear algorithms. Our results show that there is a qualitative gap between symmetric diagonally dominant (SDD) and the more general class of positive semidefinite (PSD) matrices. For SDD matrices, we develop an algorithm that approximates that runs in polylogarithmic time, provided that S is sparse and has a small condition number (e.g., Laplacian of an expander graph). In contrast, for certain PSD matrices with analgous assumptions, the running time must be at least polynomial.

Joint work with Alexandr Andoni and Yosef Pogrow.

Set Cover and Vertex Cover with Delay

Yossi Azar, Tel Aviv University

We consider the problem of online set cover with delay (SCD) suggested by Carrasco, Pruhs, Stein and Verschae and resolve various of their open problems. A family of sets with costs and a universe of elements are known in advance. Requests arrive over time on the elements, and each request accumulates delay cost until served by the algorithm through buying a containing set. A request can only be served by sets that are bought after the request's arrival, and thus a set may be bought an unbounded number of times over the course of the algorithm.

Carrasco et al. showed an algorithm for SCD whose competitive ratio grows with the number of requests. Their algorithm runs in exponential time (no polynomial time logarithmic approximation algorithm is known even for the offline case). In addition, they show a lower bound on the competitive ratio which is exponentially smaller than their upper bound (for the case of known sets). We resolve various of their open problems. Specifically, we show an $O(\log k \log n)$-competitive randomized algorithm, where n is the number of elements and k is the maximum number of sets containing any single element. Our algorithm runs in polynomial time and achieves its competitive ratio for arbitrary long (possible infinite) sequence (In particular it can be used as an offline algorithm as well). We also show lower bounds of $\Omega(\sqrt{\log k})$ and $\Omega(\sqrt{\log n})$ on the competitiveness of any algorithm for SCD improving exponentially over their lower bound for known sets. Our $O(\log k \log n)$-competitive algorithm is based on exponential weights combined with the max operator (in contrast to most algorithms employing exponential weights, which use summation). The lower bounds are described by a recursive construction. As a side result, we also consider the special case of Vertex Cover with Delay (VCD) and show a simple 3-competitive deterministic algorithm.

Joint work with A. Chiplunkar, S. Kutten and N. Touitou

Approximate Matchings in Massive Graphs via Local Structure

Cliff Stein, Columbia University

Finding a maximum matching is a fundamental algorithmic problem and is fairly well understood in traditional sequential computing models. Some modern applications require that we handle massive graphs and hence we need to consider algorithms in models that do not allow the entire input graph to be held in the memory of one computer, or models in which the graph is evolving over time.

A few years ago, we introduced a new concept called an "Edge Degree Constrained Subgraph (EDCS)", which is a subgraph that is guaranteed to contain a large matching, and which can be identified via local conditions. We will explain the EDCS and survey its uses for finding 3/2-approximate matching in several different environments, including dynamic graphs, Map Reduce, streaming and distributed computing.

This work is joint with Sepehr Asadi, Aaron Bernstein, MohammadHossein Bateni and Vahab Marrokni.

$(1+\epsilon)$-Approximate Incremental Matching in Constant Amortized Update Time

Chris Schwiegelshohn, Sapienza, University of Rome

We present an algorithm for matching in incremental graphs, where edges arrive one by one. Our main result is a $1+\epsilon$ approximation for the maximum matching problem using linear work overall. A particular noteworthy aspect of our work is an algorithm for approximate matching in general (non-bipartite) graphs, that avoids blossum contraction, that may be of independent interest.

Day 4, Thursday, April 11

Dynamic Primal-dual Algorithms for Maximum Matching

Sayan Bhattacharya, University of Warwick

Consider a dynamic graph $G = (V, E)$ that keeps changing via a sequence of updates, where each update inserts/deletes an edge in G. We want to maintain a large matching in G with small update time. This problem has received significant attention in recent years in the dynamic algorithms community.

In this talk, we will present a natural primal-dual algorithm deterministic algorithm for this problem that has approximation ratio $(2 + \epsilon)$ and an amortized update time of $O(1/\epsilon^2)$.

Fully Dynamic Maximal Independent Set with Sublinear Update Time

Shay Solomon, Tel Aviv University

The Maximal Independent Set (MIS) problem is a fundamental one, but surprisingly little was known about it in the dynamic setting until recently. In this talk I will present the first dynamic algorithm for the problem with sublinear in n update time.

Joint work with Sepehr Assadi, Krzysztof Onak and Baruch Schieber.

Breaking Quadratic Time for Small Vertex Connectivity

Thatchaphol Saranurak, Toyota Technological Institute, Chicago

Vertex connectivity is a classic extensively-studied problem. Given an integer k, its goal is to decide if an n-node m-edge graph can be disconnected by removing k vertices. Although a $O(m)$-time algorithm was postulated since 1974 [Aho, Hopcroft, and Ullman], and despite its sibling problem of edge connectivity being resolved over two decades ago [Karger STOC'96], so far no vertex connectivity algorithms are faster than $O(n^2)$ time even for $k = 4$ and $m = O(n)$. In the simplest case where $m = O(n)$ and $k = O(1)$, the $O(n^2)$ bound dates five decades back to [Kleitman IEEE Trans. Circuit Theory'69]. For the general case, the best bound is $\tilde{O}(min\{kn^2, n^\omega + nk^\omega\})$ [Henzinger, Rao, Gabow FOCS'96; Linial, Lovász, Wigderson FOCS'86].

In this talk, I will present a randomized Monte Carlo algorithm with $\tilde{O}(m + k^3 n)$ time. This algorithm proves the conjecture by Aho, Hopcroft, and Ullman when $k = O(1)$ up to a polylog factor, breaks the 50-year-old bound by Kleitman, is fastest for $4 < k < n^{0.456}$. The story is similar for the directed graphs where we obtain an algorithm running time at most $\tilde{O}(k^2 m)$.

The key to our results is to avoid computing single-source connectivity, which was needed by all previous exact algorithms and is not known to admit $o(n^2)$ time. Instead, we design the first local algorithm for computing vertex connectivity; without reading the whole graph, our algorithm can find a separator of size at most k or certify that there is no separator of size at most k "near" a given seed node.

This talk is based on joint works with Danupon Nanongkai and Sorrachai Yingchareonthawornchai.

Lossy Kernels for some Connected Deletion Problems

Ramanujan Sridharan, University of Warwick

One of the most frequently investigated subgraph hitting problems in the literature is the $\mathcal{F}$-Deletion problem which generalises numerous well-studied NP-complete problems. In this problem, $\mathcal{F}$ is a fixed finite family of graphs and one is given a graph G and an integer k as input. The objective is to determine whether at most k vertices can be deleted from G so that the resulting graph is $\mathcal{F}$-minor free (does not contain a minor isomorphic to a graph in $\mathcal{F}$).

Already when $\mathcal{F}$ is as simple as $\{K_2\}$, this problem is the same as the classic Connected Vertex Cover problem and hence is NP-hard to approximate below a factor of 2 and does not admit a polynomial kernel under standard hypotheses. Moreover, no non-trivial approximation is known even in the special case of $F = \{K_3\}$ (Connected Feedback Vertex Set).

In this talk, we will consider the case of $\mathcal{F} = \{\mathcal{K}_{\ni}\}$ and describe a *Polynomial Size Approximate Kernelization Scheme*, which is an efficient preprocessing algorithm whose output has size bounded polynomially in a parameter of the input and whose error can be made arbitrarily close to 0. As a consequence of our techniques, we obtain a factor $min\{\mathsf{OPT}^c, n^{1-\delta}\}$ (for some δ) approximation for this problem.