Fall 2008 Theory Seminar

Fall 2008 Theory Seminar

Thursdays 2:30-3:30, in room SC 3405

Theory seminar gnome: Mike Rosulek (rosulek at uiuc dot edu)

To receive weekly reminders and announcements, please sign up for the theory-seminar mailing list.

As usual, every theory student is expected to speak at least once, but non-theory students are also welcome to volunteer. Email Mike for a speaking slot. Student speakers are encouraged to register for CS 591JE (CRN 35949).

Schedule

- August 28 - The faculty each gave an overview of their research interests.
- September 4 - Nitish Korula talked about algorithms for secretary problems in graphs & hypergraphs.

### Algorithms for Secretary Problems in Graphs and Hypergraphs.  
Nitish Korula

In the classical secretary problem, one is allowed to pick a single element from a sequence of elements appearing online in a random order. One must decide immediately after seeing any element (and before seeing the next element) whether to pick it or not; once made, a decision cannot be changed. The goal is to maximize the probability of picking the largest element. There is a well-known algorithm that has a 1/e probability of picking this element.

Several generalizations of this problem have been studied, where each element has a weight, and the goal is to pick a maximum-weight subset of independent elements, for a suitable definition of independence. For example, in the k-secretary problem, the goal is to pick a set of at most k elements from a randomly ordered sequence, to maximize their total weight. Two years ago, the matroid secretary problem was introduced, in which the elements form a matroid, and the goal is to pick a maximum-weight independent set in the matroid; Babaioff et al gave an O(log n)-competitive algorithm for this problem.

We study various problems related to finding maximum-weight matchings in graphs and hypergraphs where edges and vertices appear online, giving constant-factor approximations for all these problems in hypergraphs of bounded edge-size. In particular, we obtain an 8-approximation for the transversal matroid secretary problem, improving (and significantly simplifying) the 16-competitive algorithm due to Dimitrov and Plaxton. We also give a simple 2e-approximation for constructing a maximum-weight tree in a graph, where the edges appear online (that is, the secretary problem on graphic matroids). This improves the previously known 16-approximation.

This is joint work with Martin Pál.

- September 11 - Sariel Har-Peled talked about improved bounds for sampling

### A note on sampling and derandomization  
Sariel Har-Peled

In this talk, we will discuss improved bounds on epsilon-sampling (for spaces with bounded VC dimension), and point out how they can be constructed using discrepancy. The proof uses the chaining technique, which is surprising in this context. This leads to derandomization of some optimization problems, including the problem of a O(log opt) approximation algorithm for the problem of covering points with minimum cost disks. Previously, only a randomized algorithm was known for this problem.

This talk is based on the paper:


And some new observations by the speaker.

- September 18 - Ryan Stout talked about complete fairness in 2-party secure function evaluation
### Complete Fairness in Secure Two-Party Computation

S. Dov Gordon, Carmit Hazay, Jonathan Katz, Yehuda Lindell

Presentation by Ryan Stout

In the setting of secure two-party computation, two mutually distrusting parties wish to compute some function of their inputs while preserving, to the extent possible, various security properties such as privacy, correctness, and more. One desirable property is fairness, which guarantees that if either party receives its output, then the other party does too. Cleve (STOC 1986) showed that complete fairness cannot be achieved in general in the two-party setting; specifically, he showed (essentially) that it is impossible to compute Boolean XOR with complete fairness. Since his work, the accepted folklore has been that nothing non-trivial can be computed with complete fairness, and the question of complete fairness in secure two-party computation has been treated as closed since the late '80s. In this paper, we demonstrate that this widely held folklore belief is false by showing completely-fair secure protocols for various non-trivial two-party functions including Boolean AND/OR as well as Yao's "millionaires' problem".

Surprisingly, we show that it is even possible to construct completely-fair protocols for certain functions containing an "embedded XOR", although in this case we also prove a lower bound showing that a super-logarithmic number of rounds are necessary. Our results demonstrate that the question of completely-fair secure computation without an honest majority is far from closed.

### Effcient Distributed Approximation Algorithms

Gopal Pandurangan, Department of Computer Science, Purdue University

The emerging area of distributed approximation algorithms lies at the intersection of two well-established theoretical computer science areas: distributed computing and approximation algorithms. Distributed approximation algorithms tradeoff optimality of the solution for the amount of resources consumed by the distributed algorithm. Besides a fundamental theoretical interest in understanding the algorithmic complexity of distributed approximation, there is also a practical motivation in studying distributed approximation algorithms. Emerging networking technologies such as ad hoc wireless sensor networks and peer-to-peer networks operate under inherent resource constraints such as energy, bandwidth etc. A distributed algorithm which exchanges a large number of messages and takes a lot of time can consume a relatively large amount of resources, and is not suitable in a resource-constrained network. Hence it becomes necessary to design efficient distributed algorithms for various network optimization problems that have low communication and time complexity, even possibly at the cost of a reduced quality of solution.

This talk will focus on some recent results on the design and analysis of efficient distributed approximation algorithms for important network optimization problems including the minimum spanning tree and other spanning substructures, the minimum Steiner tree and related problems, the shortest paths problem etc. These are fundamental problems in distributed computing and are widely used primitives in distributed communication networks. In particular, I will talk about our recent result on using probabilistic tree embeddings to design the first-known time-optimal distributed logarithmic-approximation algorithms for the generalized Steiner forest problem, the shortest paths problem, and the optimum routing cost tree problem.

Bio:

Gopal Pandurangan is an assistant professor of Computer Science at Purdue University. He has a Ph.D. in Computer Science from Brown University. His research interests are broadly in design and analysis of algorithms, in particular, randomized algorithms and probabilistic analysis of algorithms, with applications to distributed computing, network algorithms, and computational biology.
Approximating Min-max Integer Programs
Kyle Fox

Abstract: I'll talk a little about what min-max integer programs (MIPs) are, give a brief overview of a generalized way to approximate them, and show how that result allows for a good approximation for the low congestion routing problem and the vector scheduling problem. For vector scheduling there's some new results from last Spring using results from the generalized MIP techniques and some work Chandra did previously.

An exact (exponential) algorithm for the Minimum Biclique Cover problem
Alina Ene

I will present a very simple exact algorithm for the Minimum Biclique Cover problem that is exponential in the worst case but works very well in some settings that arise in practice. I will also give a brief overview of some classical results on minimum cardinality biclique covers.

Shrinking Graphs while Preserving Element-Connectivity
Nitish Korula

Abstract: Given a graph G(V,E) with a specified set of terminals T \subseteq V(G), we say that 2 paths are element-disjoint if they share no edges or non-terminal vertices. The element-connectivity of terminals s and t is the maximum number of pairwise element-disjoint paths between s and t. It is easy to see that the element-connectivity of s and t is at least their vertex-connectivity, and at most their edge-connectivity. Also, element-connectivity problems are often an intermediate step between their edge- and vertex-connectivity counterparts; in some cases, element-connectivity problems are similar to the easier edge-connectivity versions, and in others, they resemble the more difficult vertex-connectivity problems.

We describe a simple step to eliminate an edge between non-terminals while preserving the pairwise element-connectivity of all terminals. Repeating this step, we can eliminate all edges between non-terminals, so we are left with a bipartite (or quasi-bipartite) graph; many structural theorems can be more easily proved in these reduced bipartite graphs. Applications of this reduction include algorithms for packing element-disjoint Steiner Trees and Forests, and a much simpler analysis of algorithms for single-sink vertex-connectivity problems.

This is joint work with Chandra Chekuri.

Stopping Sets: Hardness of Approximation and New Automorphism Group Decoding Algorithms
Olgica Milenkovic

Codes on graphs have in the recent past attracted significant attention due to the fact that they can be efficiently decoded via iterative decoding algorithms, collectively known as message passing schemes. For each transmission channel, message passing algorithms tend to fail when encountering special noise configurations that are otherwise correctable by maximum-likelihood decoders. One simple example includes the binary erasure channel, for which erasure-patterns known as stopping sets completely characterize the bit error rate performance of message passing at large signal-to-noise ratios.

The size of the smallest stopping set in a code graph and the distribution of stopping set sizes is influenced by the choice of the dual basis of the code. Unfortunately, as will be shown, finding the smallest stopping set of a graph is NP-hard, so that testing for small stopping sets in a given code is computationally intractable. We therefore focus on finding optimal or near-optimal "redundant representations" of the code which allow for eliminating the influence of stopping sets on the decoder's performance. We also propose a new modification of messages passing, termed automorphism group decoding, that virtually resolves most drawbacks of message passing decoders that arise due to the existence of small stopping sets.

Parts of the results are joint work with T. Hehn, S. Laendner, A. McGregor, and J. Huber.

Making Sense of Making Change
Tracy Grauman

Given a set of (positive integer) coins, what's the best way to make change from its denominations? We investigate some classes of "greedy" coin sets; that is, coin sets on which the greedy algorithm is actually optimal. We will also look into my results thus far with regards to characterizing greedy obstructions of length four, where an obstruction is defined as a non-greedy coin set which – even after adding any number or denomination of coins (though, finitely many) of greater value – never becomes a greedy coin set.

Making Sense of Making Change
Tracy Grauman

Given a set of (positive integer) coins, what's the best way to make change from its denominations? We investigate some classes of "greedy" coin sets; that is, coin sets on which the greedy algorithm is actually optimal. We will also look into my results thus far with regards to characterizing greedy obstructions of length four, where an obstruction is defined as a non-greedy coin set which – even after adding any number or denomination of coins (though, finitely many) of greater value – never becomes a greedy coin set.

How to solve the change making problem and greedy coin sets
Tracy Grauman

November 13 - Tracy Grauman talked about the change making problem and greedy coin sets

Making Sense of Making Change
Tracy Grauman

Given a set of (positive integer) coins, what's the best way to make change from its denominations? We investigate some classes of "greedy" coin sets; that is, coin sets on which the greedy algorithm is actually optimal. We will also look into my results thus far with regards to characterizing greedy obstructions of length four, where an obstruction is defined as a non-greedy coin set which – even after adding any number or denomination of coins (though, finitely many) of greater value – never becomes a greedy coin set.

November 20 - Hemanta Maji talked about the complexity of secure function evaluation problems
Complexity of Multi-Party Computation
Hemanta Maji

In symmetric secure function evaluation, Alice has an input $x$, Bob has an input $y$, and both parties wish to securely compute $f(x,y)$ using a protocol. We classify these functions $f$ according to their "cryptographic complexities," and show that the landscape of complexity among these functions is surprisingly rich.

For two different models of security, we give new combinatorial characterizations of the functions $f$ that have secure protocols. We also show that there exists a strict, infinite hierarchy of complexity among these functions, with respect to their cryptographic difficulty.

Joint work with Manoj Prabhakaran and Mike Rosulek

• December 4 - Ben Moseley talked about algorithms for unicast and broadcast online scheduling

Online Scheduling to Minimize the Maximum Delay Factor
Ben Moseley

We consider two scheduling models: First is the standard model (unicast) where requests (or jobs) are independent. The other is the broadcast model where broadcasting a page can satisfy multiple outstanding requests for that page. We consider online scheduling of requests when they have deadlines. Unlike previous models, which mainly consider the objective of maximizing throughput while respecting deadlines, here we focus on scheduling all the given requests with the goal of minimizing the maximum delay factor. The delay factor of a schedule is defined to be the minimum alpha >= 1 such that each request $i$ is completed by time $a_i + alpha(d_i - a_i)$ where $a_i$ is the arrival time of request $i$ and $d_i$ is its deadline. Delay factor generalizes the previously defined measure of maximum stretch which is based only on the processing times of requests.

We prove strong lower bounds on the achievable competitive ratios for delay factor scheduling even with unit-time requests. Motivated by this, we consider resource augmentation analysis and prove the following positive results. For the unicast model we give algorithms that are $(1 + \eps)$-speed $O(1/\eps)$-competitive in both the single machine and multiple machine settings. In the broadcast model we give an algorithm for same-sized pages that is $(2 + \eps)$-speed $O(1/(\eps^2))$-competitive. For arbitrary page sizes we give an algorithm that is $(4 + \eps)$-speed $O(1/(\eps^2))$-competitive.

Joint work with Chandra Chekuri

• December 11 - Md. Abul Hassan Samee's talk will be rescheduled for the Spring 2009 theory seminar.

Previous Semesters

• Spring 2008 Theory Seminar
• Fall 2007 Theory Seminar
• Spring 2007 Theory Seminar
• Fall 2006 Theory Seminar