Grant Schoenebeck
Research Statement

In my research, I apply computational approaches and insights to better understand the world we live in. My research often explores how the local and the global interact. In many complex systems global effects arise from local manipulations. I will highlight examples of this in two settings: 1) Social and economic networks, where each node's actions only directly impacts those around it; 2) Algorithms for NP-complete optimization problems, where linear and semidefinite program relaxation hierarchies impose additional integrality constraints on every small set of variables with the hope that this will force global integrality.

Theoretical Computer Science and Social and Economic Networks

Networks provide a medium in which the local and global can interact. In fact, sociologists initiated the study of social networks to help understand the interaction between more easily observed "local" phenomenon and the larger society scale effects, for which knowledge beyond statistics is difficult [Gra73].

Processes over Social Networks

Social networks can be viewed as a distributed system where each node operates in a local, autonomous, and, possibly, self-interested way. What (global) goals can be accomplished by a network?

Models for specific processes that are both realistic and analyzable main elusive, in part, because our current mathematical techniques fail on even (very) simple models. Building up mathematical understandings of these very simple models is an import step to creating better models.

- In [MS10], with Elchanan Mossel, I formalize a computational model of social networks, and study how quickly consensus—deciding between two choices which are otherwise indistinguishable—can be reached within networks. I also study how quickly a consensus can be reached which is identical to the majority of the original signals. I parameterized our model by the number of bits of memory that each agent has access to, and show how this parameter affects the rate of convergence to consensus.

Processes such as aggregating knowledge distributed across the nodes and the spreading of information, diseases, technologies, and rumors have been studied, but many problems remain unaddressed due to mathematical impasses. For example, the simplest model in [MS10] has no nontrivial lower bound even though it is oversimplified for most settings, in part, because all agents are indifferent between choices. What if each agent has some private information and some preference?

Strategic Considerations Besides assuming that the networks are distributed, one may assume that different nodes act toward different goals. A system is said to be in equilibrium if each agent is acting toward his goal, and this imposes (motivates) certain local rules—the effects of which can be studied on the system. For example, the "price of anarchy" measures how much the equilibrium strategy underperforms the global optimum. Will the system converge to a global equilibrium via some local best response dynamics? How sophisticated do the agent's strategies need to be to realize equilibrium?

- In [FST11], with Raphael Frongillo and Omer Tamuz, I propose a model of learning on social networks in dynamic environments which describes a group of agents who are each trying to estimate an underlying state that varies over time, given access to weak signals and the estimates of their social network neighbors. I formulate three models of agent behavior and study when they converge and the quality of estimators they produce.

- In [DSVV09], with Konstantinos Daskalakis, Gregory Valiant, and Paul Valiant, I study and classify the complexity of computing Nash equilibrium in action graph games, a generalization of graphical games.

- In [SV06], with Salil Vadhan, I determine the computational complexity of four problems related to Nash equilibria in various models of concisely represented games including graphical games (where the players are vertices of graph, and each player's payoff depends only her neighbors' strategies) and circuit games (where the input is a circuit which computes the payoffs).

Structure of Social Networks: What do social networks look like?

While we have unprecedented amounts of digital data on social interactions it is often unclear how to make inferences from them. Most of the data is culled from networks that people join (be they Verizon, Facebook, or arXiv).

- In [Sch11] I use simulations to show that data from digital social networks may differ substantially from the actual social networks on top of which the digital networks are constructed. I model people adopting a social technology as it spreads on some underlying social network, and show that the network of adopters contains three basic properties—all of which have been observed in data mining (for example a power-law type degree distribution)—even though the underlying network has none of these properties.

While digital data gives us a selective global view, the selection of data into the data set is a local process—each person adopts only based on his view of the world, and thus this adoption process can be more easily studied. Can we analyze these two notions together to recover the underlying social network? If not, what properties can we hope to recover from this selectively sampled data? What techniques are required for this reconstruction?

One place to hope to find answers to these questions is in the property testing literature. However, these results usually make "worst-case" assumptions and so do not supply a lot of direct guidance to this problem.

Beyond worst-case analysis The current digital data on social interactions are so large that we need algorithms to interpret them. Even for rather basic questions, such as understanding community structure, computational barriers arise; the related computation problems of DenseSubset or Clique are notoriously computationally difficult, even to approximate. However, this computational hardness applies only to the "worst-case" where an algorithm is considered to fail if it incorrectly answers on even one hypothetical input. Analyzing non-"worst-case" settings has been a central challenge for theoretical computer scientists since the founding of the field. This is also an impediment to importing ideas from theoretical computer science to analyze social and economic networks where the "worst-case" setting is not interesting because social networks all seem to share certain properties.

One way forward is to show that heuristic solutions work in the average-case. However, to prove average-case results, we must know what the "average" network looks like, and that is exactly what we are using these algorithms to discover.

- In [AGSS11], with Sanjeev Arora, Rong Ge, and Sushant Sachdeva, I examine sociology research and distill several "local" observations about community structure on social networks. I design algorithms that provably and efficiently find the "global" community structure if the inputs conform to these observations. The algorithms are novel in the space of social networks, and creatively employ ideas from dense graph algorithms and property testing which were not previously known to apply to this area.

Future work on computationally hard social network analysis, should try to explore what assumptions are required for provably correct solutions and what happens when these assumptions fail to be met. Can we find assumptions that are testable or are well-documented in the literature (so that the results are not predetermined by the assumptions)? What kinds of features currently lie beyond our algorithm techniques, even in realistic settings (so that we better know what our current analyses may be missing)?

Duality of Structure and Process

The question of what social networks look like, and modeling processes over social networks are intimately connected even where the processes do not change the network. What does it means for two networks to look "alike?" One definition is that certain processes, if run on similar networks, ought to produce similar results.

The eigenvalue gap is one successful measure for what networks look like. It can be efficiently computed and says a lot about how certain processes that depend on "mixing" run on networks. For slightly more complicated processes—a node becomes infected when two of its neighbors are infected—much less is known. What properties of a network are important to say that two networks look alike (in certain situations)? What properties of a network are important to say that two networks look alike (in certain situations)? We must build up a better vocabulary to describe networks by identifying relevant (and efficiently computable) metrics that capture when two networks to "look alike".

While I, and many others, have contributed to the endeavor of importing theoretical computer science ideas into social and economic networks in the past, I think that we are at the tip of the iceberg. Many ideas have been explored in isolation, but getting them to work together in the same model is challenging.

These research goals necessarily draw on ideas from many fields including sociology, economics, and other areas of computer science and present many opportunities for collaboration. Theoretical computer science will continue to contribute to social network research by 1) bringing tools of mathematical sophistication to bear on traditional problems; 2) asking new questions and framing problems through a computational lens; and 3) modifying notions of (computational) hardness to be more appropriate to the context. Additionally, social networks research will have a positive contribution on theoretical computer science, in particular, by challenging it to reach beyond "worst-case" analysis.

Hierarchies, Heuristics, and Hardness

NP-complete combinatorial optimization problems (e.g. 3SAT, VertexCover, Clique) are important and well-studied, but remain largely enigmatic in fundamental ways. They are not believed to be efficiently solvable in all cases unless the famous P≠NP conjecture is false.

While P≠NP currently seems out of reach, one way to buildup understanding of these problems is to look at how restricted classes of algorithms perform and study for what cases they success or fail. Better understanding the existing algorithms promises insights into what makes these problems hard and what hurdles new algorithmic techniques must overcome.

Approximation Algorithms and the Power of Semidefinite and Linear Programs

Instead of finding the optimal solution to NP-complete combinatorial optimization problems, we can try to find approximately optimal solutions. To date, the most promising approach for approximating many combinatorial optimization problems has been linear programming relaxations (LPs) and a less understood generalization, semidefinite programming relaxations (SDPs).

Linear programming relaxations start with an integer program formulation of the problem to be solved (where some linear objective function is to be minimized subject to linear constraints and variables may be required to be integral). While solving this program would yield the optimal solution, we do not know how to do so efficiently because it is NP-hard. The program is "relaxed" so that values are no longer restricted to be integers. This enlarges the search space, guaranteeing a better optimal value, but renders the optimization tractable. While the solution now may be fractional, it can usually be rounded back to a integral solution with some loss in the objective function. The integrality gap is the quotient of the true optimum by the relaxation's optimal solution (which is always at least as good), and measures the quality of the approximation. The integrality gap is sensitive to the original integer programming formulation, and an important question is when modifications to the integer program improve the algorithms of this framework. Several processes—Lovász Schrijver+ (LS+) [LS91] and the stronger Lasserre hierarchy [Las01] for semidefinite programs, and Lovász Schrijver (LS) [LS91], and the stronger Sherali-Adams hierarchy [SA90] for linear programs—were create to systematically improve semidefinite and linear programs at the cost of additional runtime. These systems proceed in rounds and, intuitively, after r rounds the solutions are required to be "locally" integral on all neighborhoods of up to r vertices but require time $n^{O(r)}$ to run. After n rounds, they are guaranteed to produce the exact optimal solution. If local integrality implies global integrality, then these hierarchies will provide a good approximation after only a few rounds. Indeed, these hierarchies provably imply some of the most celebrated approximation algorithms for NP-complete problems even after a few rounds. For example, the first round of LS+ (and hence also Lasserre) for the IndependentSet problem implies the Lovász θ-function [Lov79] and for the MaxCut problem gives the Goemans-Williamson relaxation [GW95]. The ARV relaxation of the SparsestCut [ARV04] problem is no stronger than the relaxation given in the third round of LS+ (and hence also Lasserre), and most recently the subexponential time algorithm for Unique Games [ABS10] is implied by a sublinear number of rounds of Lasserre.

These hierarchies give us a framework to study the power and limitations of linear and semidefinite programming by asking: what is the tradeoff between the running time and the guaranteed approximation in these hierarchies?

■ In [Sch08], I prove that the Lasserre hierarchy cannot refute a random 3XOR formula in $\Omega(n)$ rounds and thus requires exponential time. This is the first non-trivial integrality gap for the Lasserre hierarchy, the strongest of all the aforementioned hierarchies. Many corollaries follow, such as a similar integrality gap of $7/6 - \varepsilon$ for VertexCover. The techniques in the paper remain the only known way of obtaining integrality gaps for Lasserre.

■ In [STT07b], along with Madhur Tulsiani and Luca Trevisan, I prove an optimal integrality gap of $2 - \varepsilon$ for $\Omega(n)$ rounds in the LS hierarchy relaxation of the VertexCover and MaxCut problems. This result implies that a very large class of linear programs require exponential time to solve Vertex Cover (or Maximum Cut) to within a factor less than 2, even on random graphs. The previously best known $2-\varepsilon$ integrality gap for Vertex Cover [ABLT06] only survived $\Omega(\log(n))$ round of LS. These results were the first to illustrate the stark difference between linear program relaxations and semidefinite program relaxations (because MaxCut is better approximated after just one round by LS+).

■ In [STT07a], again with Madhur Tulsiani and Luca Trevisan, I prove an integrality gap of $7/6 - \varepsilon$ for Vertex Cover even after $\Omega(n)$ rounds of LS+. Prior to this result, there were no known integrality gaps for Vertex Cover that survived more than two rounds of LS+.

While already results along these lines have yielded key insights, much more work remains to be done to understand the power and limitations of linear and semidefinite programs. For example, we currently do not know what the integrality gap is for MaxCut, UniqueGames, VertexCover, or SparsestCut after even 2 rounds of Lasserre are applied. Recent algorithmic results (for example the subexponential time algorithm for UniqueGames [ABS10, BRS11]) are based on rounding a sublinear number of rounds of Lasserre. New semidefinite program rounding techniques could be discovered that apply to even lower rounds. This is a rich field with many exciting recent developments and the promise for many more in the near future.

While on the surface, we are simply studying a restricted (but powerful) class of algorithms, the implications of these results runs deeper. The aforementioned integrality gaps translate into rank lower bounds of corresponding proof systems. A recent result of Prasad Raghavendra [Rag08] shows that if the Unique Games Conjecture is true, then semidefinite programs provide optimal algorithms for a certain large class of problems. Work in this field has also been leveraged to show lower-bounds for other restricted classes of algorithms including metric embeddings [CMM07] and certain types of dynamic programming.

Other Classes of Algorithms

Other algorithmic classes–greedy algorithms, genetic algorithms, local search algorithms, simulated annealing, metric embeddings, and well as techniques from machine learning–also show promise for approximating NP-complete combinatorial optimization problems or solving them on "easy" instances. The theoretical understanding of when and why these algorithms perform well is needed but underdeveloped. Better understanding properties of "hard" and "easy" instances may give us a broader understanding across algorithmic techniques.

Hard Instances Even if we do not hope to solve NP-complete problems over all instances, some can be solved over many instances that arrive in practice. Can we distinguish the "hard" instances of a problem from the "easy" instances? What properties of instances make them hard or easy? Can we find instances that are hard for all current algorithmic techniques? The aforementioned semidefinite programming relaxation integrality gaps have started to give us insights into these questions. For example the 3-XOR results of [Sch08] showed that instances that have a lot of "expansion" are "hard". Conversely, the results of [AKK$^+$08] show that a similar "expansion" property makes instances of UniqueGames "easy."

Research studying specific classes of algorithms has forged deep relations between approximation algorithms, proof complexity, metric embeddings, average case hardness, and local algorithms. This a rich field with many exciting recent developments and the promise for many more in the near future.

This work connects to social and economic networks research in three ways: 1) Social networks can be viewed as a certain type of distributed algorithm. 2) Social network analysis often employs unproven heuristics, and understanding the strengths and limitations of these analyses requires understanding the strengths and limitations of the algorithms employed. 3) Understanding how the local and global interact is vital.

Serendipitous Collaborations

Lastly, I enjoy collaborating with researchers from different institutions and backgrounds. This has led to several papers in areas that I would not consider my core competencies, but nonetheless have been very fruitful and rewarding.

- Property Testing: [BKS+10] Along with Arnab Bhattacharyya, Swastik Kopparty, Madhu Sudan, and David Zuckerman, I show an optimal analysis for testing low degree polynomials (Reed-Muller codes). This is a fundamental problem in property testing and coding theory. In addition, it has deep relations to additive combinatorics, and recently, this work was employed to create novel new codes with applications to complexity theory including the Unique Games Conjecture.

- Online Algorithms: [GRST10] Along with Anupam Gupta, Aaron Roth, and Kunal Talwar, I show a constant factor approximation for non-monotone submodular optimization problems with matroid constraints. Additionally, this extends to the online "secretary" setting for partition matroid constraints, and to general k-matroid constraints (however in this case, the approximations is only $O(log(k))$.

- Cryptography: [HS11] Along with Thomas Holenstein, I proved hardness amplification results for weakly verifiable puzzles that apply to any monotone function, and reprove a generalized version of Yao's XOR lemma (with additional properties) in a novel and simple way. Using the aforementioned results, I show that any weak cryptographic protocol whose security is given by the unpredictability of single bits can be strengthened with a natural information theoretic protocol.

I am excited to continue to meet engaging people with different research backgrounds who are keen to work on exciting new problems with me.

References

[ABLT06] Sanjeev Arora, Béla Bollobás, László Lovász, and Iannis Tourlakis. Proving integrality gaps without knowing the linear program. *Theory of Computing*, 2(2):19–51, 2006. 4

[ABS10] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *Proceedings of the 51th IEEE Symposium on Foundations of Computer Science*, 2010. 3, 4

[AGSS11] Sanjeev Arora, Rong Ge, Sushant Sachdeva, and Grant Schoenebeck. Finding overlapping communities in social networks: Toward a rigorous approach. In Submission, 2011. 2

[AKK+08] Sanjeev Arora, Subhash Khot, Alexandra Kolla, David Steurer, Madhur Tulsiani, and Nisheeth K. Vishnoi. Unique games on expanding constraint graphs are easy: extended abstract. In *Proceedings of the 40th ACM Symposium on Theory of Computing*, pages 21–28, 2008. 4

[ARV04] Sanjeev Arora, Satish Rao, and Umesh Vazirani. Expander flows and a $\sqrt{\log n}$-approximation to sparsest cut. In *Proceedings of the 36th ACM Symposium on Theory of Computing*, 2004. 3

[BKS+10] Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *Proceedings of the 51th IEEE Symposium on Foundations of Computer Science*, 2010. Eariler Version appeared as Technical Report TR09-086 on ECCC in October, 2009. 5

[BRS11] Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *Proceedings of the 52th IEEE Symposium on Foundations of Computer Science*, 2011. 4

[CMM07] Moses Charikar, Konstantin Makarychev, and Yury Makarychev. Local global tradeoffs in metric embeddings. *Foundations of Computer Science, Annual IEEE Symposium on*, 0:713–723, 2007. 4

[DSVV09] Constantinos Daskalakis, Grant Schoenebeck, Gregory Valiant, and Paul Valiant. On the complexity of Nash equilibria of action-graph games. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms*, pages 710–719, 2009. 1

[FST11] Rafael M. Frongillo, Grant Schoenebeck, and Omer Tamuz. Social learning in a changing world. In *The Seventh Annual Workshop on Internet and Network Economics (WINE 2011)*, 2011. 1

[Gra73] M.S. Granovetter. The Strength of Weak Ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973. 1

[GRST10] Anupam Gupta, Aaron Roth, Grant Schoenebeck, and Kunal Talwar. Constrained non-monotone submodular maximization: Offline and secretary algoritms. In *The 6th Workshop on Internet and Network Economics (WINE 2010)*, December 2010. 5

[GW95] M.X. Goemans and D.P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. Preliminary version in *Proc. of STOC'94*. 3

[HS11] Thomas Hollenstein and Grant Schoenebeck. General hardness amplification of predicates and puzzles. In *Theory of Cryptography Conference*, March 2011. 5

[Las01] Jean B. Lasserre. An explicit exact sdp relaxation for nonlinear 0-1 programs. In *Proceedings of the 8th International IPCO Conference on Integer Programming and Combinatorial Optimization*, pages 293–303, London, UK, 2001. Springer-Verlag. 3

[Lov79] László Lovász. On the Shannon capacity of a graph. *IEEE Transactions on Information Theory*, 25:1–7, January 1979. 3

[LS91] L. Lovasz and A. Schrijver. Cones of matrices and set-functions and 0-1 optimization. *SIAM J. on Optimization*, 1(12):166–190, 1991. 3

[MS10] Elchanan Mossel and Grant Schoenebeck. Arriving at consensus in social networks. In *The First Symposium on Innovations in Computer Science (ICS 2010)*, 2010. 1

[Rag08] Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the 40th ACM Symposium on Theory of Computing*, pages 245–254, 2008. 4

[SA90] Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxation between the continuous and convex hull representations. *SIAM J. Discret. Math.*, 3(3):411–430, 1990. 3

[Sch08] Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science*, pages 593–692, 2008. 4

[Sch11] Grant Schoenebeck. Potential networks, contagious communities, and social network structure. 2011. 2

[STT07a] Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. A linear round lower bound for Lovasz-Schrijver SDP relaxations of Vertex Cover. In *Proceedings of the 39th ACM Symposium on Theory of Computing (STOC07)*, 2007. Earlier version appeared as Technical Report TR06-098 on Electronic Colloquium on Computational Complexity. 4

[STT07b] Grant Schoenebeck, Luca Trevisan, and Madhur Tulsiani. Tight integrality gaps for Lovasz-Schrijver LP relaxations of Vertex Cover and Max Cut. In *Proceedings of the 39th ACM Symposium on Theory of Computing*, 2007. Earlier version appeared as Technical Report TR06-132 on Electronic Colloquium on Computational Complexity. 4

[SV06] Grant Schoenebeck and Salil Vadhan. The computational complexity of Nash equilibria in concisely represented games. In *Proceedings of the 7th ACM conference on Electronic commerce (EC 2006)*, pages 270–279, 2006. Originally appeared as Technical Report TR05-052 on ECCC in May 2005. Subsequent version to appear in Transactions on Computational Theory. 1