

# Research Statement

Varsha Dani

March 2011

My primary interest is in the applicability of probabilistic methods and ideas to a wide range of problems in theoretical computer science. My main research focus until 2008 was in the areas of Learning Theory and Online Optimization, and the major part of my work in these areas was the basis of my Ph.D. thesis. I have also, in the past, done some work in the areas of Fair Division algorithms and understanding Information Markets, (the latter as a summer intern at Yahoo! Research Labs) both areas which I still find interesting, but in which I am not currently active.

After receiving my Ph.D. I took a two year long break from academic research to spend time with my daughter, who was born in Fall 2008. Although I tried to keep abreast, by attending seminars and talks whenever possible, I did not participate in any research activities during this time.

Starting in Fall 2010, I have once again started to actively think about research problems. In particular, in addition to revisiting my interest in Learning Theory, I have also become interested in a couple of new areas, and have some ongoing collaborations on questions in these fields. The first of these is Rational Secret Sharing, which is a problem in distributed computing in a game theoretic setting, on which I have some recent results with Prof. Jared Saia and a couple of his students at the University of New Mexico. The other area that I have recently taken an interest in is phase transitions for random instances of combinatorial problems. Here too, I have some recent results in collaboration with Prof. Cris Moore.

In the following sections I will describe my work in these different areas in some detail, and I will also mention some ongoing work and ideas for future projects.

## Rational Secret Sharing

Secret sharing refers to a scheme whereby each member of a group of  $n$  agents has a share of a secret, which can be reconstructed if and only if a sufficient number  $m$  of the shares are pooled together. This fundamental problem in security was introduced independently by Blakley [11] and Shamir [26] and is an important primitive in many cryptographic applications, including secure multiparty computation. Shamir's elegant solution to the problem is based on the fact that there is a unique polynomial of degree  $m - 1$  that passes through  $m$  points in the plane. This scheme encodes the secret as the value at zero of a degree  $m - 1$  polynomial, and the shares are some  $n$  points on the graph of the polynomial (for example its value at  $1, 2, \dots, n$ ).

In recent years there has been interest in studying the secret sharing problem in a game theoretic framework where the agents are self-interested rational players, who want to learn the secret, but would prefer that the others remain in the dark. Under this model, agents may be unwilling to reveal their own shares to others, if they are able to gather enough shares to learn the secret without doing so. The problem then becomes one of mechanism design: we would like to design a protocol that ensures that any  $m$  or more players participating in the protocol will all learn the secret, while fewer than  $m$  players cannot reconstruct it. Moreover, it should be a Nash equilibrium strategy for players to follow the protocol rather than deviating from it. Early work on the rational secret sharing problem (Halpern and Teague [20], Abraham et al. [1], Lysyanskaya and Triandopoulos [23], Gordon and Katz [19]) assumed that communication between the players was via broadcast channels. When the players are at nodes of a large computer network however, it is more reasonable to assume that communication between them will be point-to-point. This poses some interesting challenges for a protocol designer, because while in broadcast channels all broadcast messages are common knowledge (everyone knows the message, everyone knows that everyone knows the message etc.) with point-to-point channels, nobody knows of any communications that are not communicated to them directly. This makes deviation from the protocol potentially very hard to detect. Nevertheless, Kol and Naor [21] were able to demonstrate a protocol for the rational secret sharing problem with point-to-point

communication, which is an  $\epsilon$ -Nash equilibrium. (They also showed that there can be no Nash equilibrium in this setting.)

Unfortunately, all previous solutions to this problem require each agent to send  $O(n)$  messages in expectation, with an expected latency of  $O(n)$ , and so these do not scale to large networks. Rational secret sharing is a primitive for rational multiparty computation, which can be used to compute an arbitrary function in a completely decentralized manner, without a trusted external party. A typical application of rational multiparty computation might be to either run an auction, or to hold a lottery to assign resources in a network. It is easy to imagine such applications where the number of players is large, and where it is important to have algorithms whose bandwidth and latency costs scale well with the number of players. Moreover, in a game theoretic setting, standard tricks to circumvent scalability issues, like running the protocol only on a small subset of the players, may be undesirable since they could lead to increased likelihood of bribery attacks.

In joint work with Yamel Rodriguez, Mahnush Movahedi and Jared Saia [17], we designed a scalable protocol for  $n$ -out-of- $n$  rational secret sharing in point-to-point (and broadcast) networks, which is an  $\epsilon$ -Nash equilibrium. Our protocol requires each player to only send a constant number of messages, and has a latency of  $O(\log n)$ . There are a few ingredients in the design of this protocol. The first is to arrange the players in a complete binary tree, so that each communicates only with his neighbors in the tree, ensuring scalability. We use iterated 2-out-of-2 Shamir shares to create the shares of the secret; the “secret” is at the root, and at each level of the tree, shares are created for the two children which together determine the value at their parent node according to Shamir’s scheme. Shares are transmitted up the tree, and each node reconstructs a value from the messages of its children which it uses as a share to transmit to the next level. The value reconstructed at the root is transmitted back down the tree. Tag and hash verification schemes are used to ensure that the players do not fake messages. And finally, as in previous works, the protocol consists of a number of fake rounds, which are designed only to identify and punish players who are disinclined to reveal their information. The reconstruction of the true secret occurs in a random round, so players do not know in which round to fail to transmit their share. See figure 1 for an illustration of the flow of messages in the algorithm.

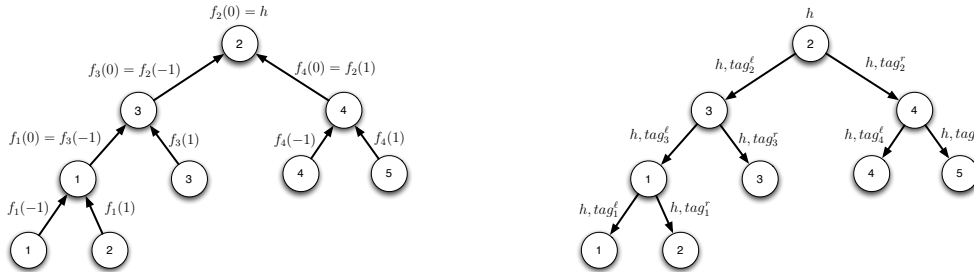


Figure 1: On the left is the up stage of each round of the algorithm; on the right is the down stage.

The  $n$ -out-of- $n$  case for rational secret sharing is a critical component of rational multiparty computation. Our result for  $n$ -out-of- $n$  rational secret sharing enables a protocol for rational multiparty computation that is an  $\epsilon$ -Nash equilibrium in the point-to-point, synchronous, non-simultaneous communication model. Moreover, it reduces worst case bandwidth by a multiplicative factor of  $n$ , and latency by a multiplicative factor of  $\Theta(n/\log n)$  over the rational multiparty protocol in this communication model from [21].

The real goal is to design a protocol for  $m$ -out-of- $n$  secret sharing where  $m < n$ . This is a harder problem because for any communication structure other than a clique, if the “missing” players are adversarially selected to be a cut of the graph, then there can be no communication between the two sides. Nevertheless, we show that by introducing redundancy into our binary tree structure, so that there are  $\Theta(\log n)$  players at each node, we can achieve an  $\epsilon$ -Nash equilibrium for a restricted setting in which  $m = \Theta(n)$  and the missing players are a random subset (independent of all the randomness in the protocol) rather than being chosen adversarially.

An extended abstract of our work [17] will appear in the proceedings of the upcoming conference on the Principles of Distributed Computing (PODC 2011). We also plan to submit a full version of the paper to

the Journal of Distributed Computing within the next few weeks.

Several open problems remain, which I would like to tackle in the near future. Can anything be done for the restricted case where the number of missing players is small, but missing players are selected adversarially? What about if the missing players are selected randomly, but only  $o(n)$  players are present? Can we design mechanisms that are resistant to collusion among the players? Also, while our algorithms lead to a  $\Theta(n)$  multiplicative reduction in communication costs for rational secure multiparty computation (SMPC), the overall bandwidth for this problem is still very high. It is known that there are certain functions, for which rational SMPC can not be performed in a scalable manner (for example, the parity function). However, we ask: Can we find classes of well-motivated functions for which scalable SMPC is possible? This is related to our second open problem which is: Can we design scalable algorithms for simulating a class of well-motivated mediators? In some sense, this problem may be harder than the SMPC problem, since some types of mediators offer different advice to different players. In other words, the problem is easier: a simple global coin toss is an effective mediator for many games. Another direction is the situation where there are a small number of adversarial or “Byzantine” players in with the rational players. Here things become complicated by the fact that these players cannot be assumed to play according to Nash equilibrium strategies, and therefore usual protocols fail. Designing protocols for this situation is another interesting open problem.

## Phase Transitions for Combinatorial Problems

Solving worst-case instances of combinatorial problems has been a long-time focus of theoretical computer science. Indeed the concept of NP-Completeness is based on the existence of hard instances which seem not to admit efficient solutions. In the last couple of decades however, there has been an interest in studying and understanding properties of *random* instances of combinatorial problems. One such model for random instances is that of Erdos-Renyi random graphs  $G(n, p)$ .

A random graph in  $G(n, p)$  is a graph on  $n$  vertices in which each of the  $\binom{n}{2}$  possible edges is included in the graph independently with probability  $p$ . Graph properties such as connectedness,  $k$ -colorability, existence of large independent sets, and many others, undergo a *phase transition*: there is a critical density of edges such that the property goes from being almost certainly true, for sparser graphs, to almost certainly false, for denser graphs. (In physics, “phase transition” refers to phenomena such as the freezing or boiling of liquids, in which macroscopic changes suddenly occur, as a result of temperature, or another parameter, crossing a critical threshold.) Part of the motivation for understanding phase transitions comes from the observation that the most difficult instances of the problem tend to arise at the critical threshold.

Many other combinatorial problems seem to have similar phase transitions. It is believed that locating these critical thresholds and studying the structure of the space of solutions near them will contribute to a better understanding of what makes these combinatorial problems intractable.

Unfortunately, even in many of the cases when it is known that a phase transition occurs, we do not know *where* it happens, due to the non-constructive nature of the proofs of existence. Explicit bounds, both upper and lower, on these phase transitions typically come from the analysis of algorithms.

Another approach is to construct a random variable that counts the number of solutions of the problem, and analyze its distribution, for instance, by computing its first and second moments. Alternatively, the variable may count some other quantity, which is nonzero exactly when the problem has solutions; some ingenuity may be needed to define this random variable so as to lead to the best possible analysis.

In [16], Cris Moore and I studied the critical threshold for independent sets in random graphs. Specifically the question is, given a size parameter  $\alpha$  what is the critical edge density (measured in terms of the average degree of the vertices) below which the random graph almost surely has independent sets containing an  $\alpha$  fraction of the vertices? Or turning the problem around, given a fixed edge density, what is the critical value of  $\alpha$  above which there are probably no independent sets of that size? Inspired by ideas from Achlioptas and Peres [3] and Achlioptas and Moore [2] for  $k$ -SAT, we analyzed a random variable that weights each independent set according to the total degree of its vertices. Applying the second moment method to this, we obtained a lower bound on the critical threshold for independent sets of a fixed size. Inverting this, we were able to prove new bounds on the probable size of the largest independent set in random graphs of a fixed average degree which are a slight improvement on the best previously known bounds from the work of Frieze [18].

Our paper [16] is available on the ArXiv, <http://arxiv.org/abs/1011.0180> and we plan to submit it to RANDOM-APPROX 2011 in April. We are also planning to submit an extended version to a special issue of the journal Random Structures and Algorithms later this summer.

We are still exploring the applicability of these techniques and other ideas to other combinatorial problems, including improved bounds for satisfiability and independent sets in random graphs with more structure, such as 3-regular graphs. Ultimately, as a long term goal, I would like to study the connections of these and similar phenomena in other more complicated models of random graphs to models of social networks.

## Online Optimization

The multi-armed bandit problem is a foundational model for sequential decision-making with missing information. First introduced by Robbins [25] in the context of the sequential design of statistical experiments such as clinical drug trials, it has subsequently been applied in such diverse areas as adaptive routing in networks, reinforcement learning and boosting, prediction markets, and financial portfolio rebalancing.

In the traditional multi-armed bandit problem, over a sequence of  $T$  rounds, a decision maker must choose a decision from a set of  $K$  options with unknown outcomes. Simultaneously, the environment assigns a cost to each outcome in an unknown way. After each such round, the algorithm gets the cost of its chosen decision as feedback which may help it make better decisions in future rounds. The canonical measure of performance is the “regret,” defined as the difference between the cost incurred by the decision maker and the cost of the single best decision which could have been made with full knowledge of the environment. This model has been rather well-studied over the years and is now essentially completely solved (see Auer et al [6]). The best regret possible is  $\sqrt{KT}$ , up to a sublogarithmic factor.

Often we are confronted with decision sets  $D$  of very large size (possibly even infinite); if the environment can choose arbitrary cost functions, then, as noted, the above regret bound is the best that can be achieved. However in many settings, the large decision sets have some structure, and can be (isometrically) embedded into a low-dimensional Euclidean space so that the cost functions chosen by the environment are linear functions of the decisions. A case in point is the online network routing problem, also known as the “Drive to Work” problem. Here, on every round a path must be chosen from the set of all paths between two fixed nodes in the network. Although this decision set may be exponential in the size of the network, since the cost (or latency) of a path is the sum of the latencies on each of its edges, its inherent dimension is just the number of edges in the network.

In this setting, one hopes for a better regret bound than the  $\sqrt{|D|T}$  bound guaranteed by the results of Auer et al. [6], one that depends nicely on the inherent dimensionality of the problem, rather than the size of the decision set, which may be exponential in the dimension. A simple approach to this, used by Awerbuch and Kleinberg [7] and McMahan and Blum [24], separates the timeline into exploration rounds and exploitation rounds. During the exploration rounds, their algorithms play random decisions from a basis for the decision space and use the observed costs to construct estimators for the true costs. During the exploitation rounds, these estimators are used to select what seems to be the best decision. On these rounds the feedback is discarded. Using this approach and optimizing the fraction of exploration rounds, they were able to prove expected regret bounds with a polynomial dependence on the dimension,  $n$ , at the expense of the dependence on the time horizon  $T$  ( $O(\text{poly}(n)T^{2/3})$  in [7],  $O(\text{poly}(n)T^{3/4})$  in [24] against a more powerful model for the environment). In joint work with Tom Hayes, [14] we improved the regret bounds of McMahan and Blum [24] to  $O(nT^{2/3})$  and showed that this was the best that could be achieved by any separated-timeline approach. In subsequent work with Tom Hayes and Sham Kakade [12] we used estimation techniques inspired by linear regression to construct an algorithm whose expected regret is at most  $O(n^{3/2}\sqrt{T})$ . Here again we use an exploration vs. exploitation tradeoff. However, unlike the aforementioned separated-timeline approach, here the feedback is never discarded, leading to the improved bound. In more recent work, with Bartlett *et al.* [8], we combined the above technique with a “dynamic variance compensation” approach used by Auer *et al.* [6], thereby obtaining high-probability regret bounds of  $O(n^{3/2}\sqrt{T})$ . The best known lower bound for this problem is  $\Omega(n\sqrt{T})$ .

The original 1952 paper of Robbins [25] introduced the multi-armed bandit problem for independent identically distributed costs drawn from a fixed but unknown underlying distribution. In this case it is reasonable to seek *deterministic* algorithms achieving low regret. To solve this problem, Lai and Robbins [22]

proposed the following elegant idea (described here for the 2-arm bandit case): Most of the time, choose the apparently better decision, based on past observations, choosing the apparently worse decision just often enough to be very confident that its true mean is in fact worse. To this end, they proposed the idea of keeping track of an “upper confidence bound” for each population. Combining this idea with the power of Chernoff’s bounds leads to the following very clean formulation (see, e.g., Agrawal [4]). For each population, keep track of the empirical mean of observations, as well as the number of times it has been sampled,  $k$ . By Chernoff’s bound, the empirical mean plus  $\sqrt{2\ln(1/\delta)/k}$  is an upper bound for the true mean with probability at least  $1 - \delta$ . Now simply, in each round, choose the population with the highest upper confidence bound. This algorithm has the desirable property of being *self-correcting*: each time the truly worse population is selected, we expect its upper confidence bound to decrease, thereby reducing our tendency to select it again. This algorithm achieves a regret of  $O(\log T)$ .

The above algorithm extends to any finite decision set with essentially no modifications. However the bounds obtained scale linearly with the size of the decision set, making them impractical for stochastic online linear optimization, when the cardinality of the decision set is large compared to its inherent dimension  $n$ . Auer [5] gave an algorithm achieving  $O^*(\text{poly}(n)\sqrt{T})$  regret for finite decision sets. At the core of his algorithm is a natural generalization of the UCB algorithm described above, which, at every step, chooses the decision for which a certain upper confidence bound is maximized.

One may note that there is a substantial difference ( $\log T$  vs.  $\sqrt{T}$ ) between the bound obtained by Auer and those of his predecessors. One reason for this difference is a simple reversal of quantifiers. Lai and Robbins and Agrawal studied asymptotic regret in a model where the unknown distribution was fixed and the game was run for an arbitrarily long time. By contrast, in Auer’s work, the time horizon is fixed first, and the distribution of costs may depend on it. In this setting, there are lower bounds showing that  $\Omega(\sqrt{T})$  regret can be forced.

Nevertheless, there really is a fundamental difference between the stochastic  $K$ -armed bandit problem and the stochastic bandit linear optimization problem. In joint work with Tom Hayes and Sham Kakade [13] we analyze a simpler version of Auer’s algorithm and show that it gets  $O^*(n\sqrt{T})$  regret with high probability, in the model where the time horizon is selected first. Moreover, we show that this is optimal up to polylogarithmic factors, that is, we present a lower bound that shows that we have the correct dependence on  $n$  as well. We then show that if the distribution is fixed first, and independently of  $T$  then under certain special circumstances, and certainly when the decision set is a finite set or a fixed polytope, we get similar asymptotic behaviour to that of Lai and Robbins and Agrawal, *i.e.*, we get  $O(n^2 \log^3 T)$  regret. On the other hand, we show that this is not possible in general: there are decision sets for which  $\Omega(n\sqrt{T})$  can be forced, even if the distribution is set independently of  $T$ .

After a two-year break, I am starting to once again work actively on a number of new directions within Online Optimization. For instance, I would like to extend our result for online linear optimization for i.i.d. random variables to handle more complicated stochastic models, such as exchangeable random variables, or hidden Markov models. In these settings, because the environment is somewhat more restricted in setting its cost functions, we may hope to prove improved bounds with stronger notions of regret. I am also interested in investigating techniques for handling more complex types of loss functions, such as convex functions.

## Machine Learning Reductions

The goal in supervised learning is to use a “training set” of labelled examples to produce a classifier which can (correctly) predict labels of previously unseen (and unlabelled) examples. In *binary classification*, the classifier must distinguish between two kinds of data (*e.g.*, is this a cancerous growth or not? is this mushroom edible or not? is this an image of a face or not?) Other examples of supervised learning problems include

- multiclass classification: correctly distinguish between multiple kinds of example, *e.g.*, which digit is represented by this handwritten character,
- importance-weighted classification: some examples are more important than others,
- cost-sensitive classification: some types of mistake are more costly than others, *e.g.* stopping at a green light is less costly than running a red light, and

- regression: fitting a numerical function to the data.

Considerable effort has been spent on each of these tasks, but especially on binary classification, which is ostensibly the easiest. But is this really true!?

If one can learn the binary classifications, “is this an A or not,” “is this a B or not,” and “is this a C or not,” then these classifiers can be combined to solve the 4-way classification problem “which letter is this, A,B,C or none of the above?”

A *reduction* between supervised learning tasks is a procedure allowing an algorithm for one of the tasks to be converted into a learning algorithm for the other task. Although there were already a number of reductions known in the literature, a coherent theory was lacking. In joint work with Beygelzimer, Hayes, Langford and Zadrozny [9], we introduced a formal notion of error limiting reductions, and constructed such a reduction from cost sensitive classification to binary classification. We also analyzed the error rates of several existing reductions under our new model.

## The Wisdom of Crowds

In recent years information markets have attracted attention for their purported ability to predict such future events as election results, sporting events and terrorist activity. Several studies have suggested that such markets may be more accurate than traditional polls. But so far economics and game theory have been able to give no rigorous analysis of the extent to which markets efficiently aggregate information.

In joint work, with Omid Madani, David Pennock and Sumit Sanghai and Brian Galebach [15], using data from an on-line gaming site called ProbabilitySports, we did several experiments to compare the predictive power of several online and offline prediction algorithms, including a simulated information market. Our results suggest that it is hard to do much better than simple averaging algorithms on this data set. (This probably says more about sports fans than it does about learning algorithms and information markets in general.)

This work was done while I was a summer intern at Yahoo! Research Labs.

## Fair Allocation

In the generalized cake-cutting problem, a cake is to be divided among  $n$  players, according to some notion of fair division. Two such notions are equitable division, where each player gets at least a  $1/n$  fraction of the cake according to his or her own valuation (which is an arbitrary measure on the cake), and envy-free division, where no player prefers the piece allocated to someone else to their own piece. This may be different from equitable division since the players may have different valuations.

In the discrete version, instead of a cake, a collection of indivisible goods is to be shared by  $n$  players, who may have different valuations for them. Real-life examples include the redistribution of property in a divorce settlement or an inheritance, or negotiating the terms of a treaty or business contract. In such cases, the indivisibility of the goods can make it impossible to achieve perfectly equitable allocations (indeed, when there is just one good, it can only be assigned to one player.) In joint work with Ivona Bezakova [10], we studied a notion of fairness called max-min fairness, in which the goal is to maximize the value to the player getting the smallest value, and subject to that, maximize the value to the player getting the second smallest value, and so on. Although we showed that the problem of finding an optimal such allocation is NP-hard, we were able to efficiently compute an approximately optimal allocation for the algorithmic problem where all the valuations are known.

## References

- [1] I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed computing meets game theory: robust mechanisms for rational secret sharing and multiparty computation. In *Proceedings of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pages 53–62. ACM, 2006.

- [2] D. Achlioptas and C Moore. Two moments suffice to cross a sharp threshold. *SIAM Journal on Computing*, 36:740 – 762, 2006.
- [3] D. Achlioptas and Y Peres. The threshold for random  $k$ -sat is  $2k \log 2 - o(k)$ . *Journal of the American Mathematical Society*, 17:947–973, 2004.
- [4] R. Agrawal. Sample mean based index policies with  $O(\log n)$  regret for the multi-armed bandit problem. *Advances in Applied Probability*, 27:1054–1078, 1995.
- [5] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, 2003.
- [6] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2003.
- [7] B. Awerbuch and R. Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the 36th ACM Symposium on Theory of Computing (STOC)*, 2004.
- [8] P. Bartlett, V. Dani, T. P. Hayes, S. M. Kakade, A. Rakhlin, and A. Tewari. High probability regret bounds for online linear optimization. *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, 2008.
- [9] Alina Beygelzimer, Varsha Dani, Thomas P. Hayes, John Langford, and Bianca Zadrozny. Error limiting reductions between classification tasks. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 49–56, New York, NY, USA, 2005. ACM.
- [10] Ivona Bezakova and Varsha Dani. Allocating indivisible goods. Technical Report 20, University of Chicago, Department of Computer Science Technicacel Report TR-2004-10, 2004.
- [11] G.R. Blakley. Safeguarding cryptographic keys. In *Proceedings of the National Computer Conference 48*, page 313317, 1979.
- [12] V. Dani, T. P. Hayes, and S. M. Kakade. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems 20 (NIPS 2007)*. 2007.
- [13] V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, 2008.
- [14] Varsha Dani and Thomas P. Hayes. Robbing the bandit: Less regret in online geometric optimization against an adaptive adversary. In *Proceedings of the 17th ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2006.
- [15] Varsha Dani, Omid Madani, David M. Pennock, Sumit K. Sanghai, and Brian Galebach. An empirical comparison of algorithms for aggregating expert predictions. In *UAI*. AUAI Press, 2006.
- [16] Varsha Dani and Cristopher Moore. Independent sets in random graphs from the weighted second moment method. *Manuscript*, 2010. arXiv:1011.0180v1 [cs.CC].
- [17] Varsha Dani, Mahnush Movahedi, Yamel Rodriguez, and Jared Saia. Scalable rational secret sharing. In *Proceedings of the thirtieth annual ACM symposium on Principles of distributed computing (PODC)*, 2011.
- [18] A Frieze. On the independence number of random graphs. *Discrete Mathematics*, 81:171–175, 1990.
- [19] S. Gordon and J. Katz. Rational secret sharing, revisited. *Security and Cryptography for Networks*, pages 229–241, 2006.
- [20] J. Halpern and V. Teague. Rational secret sharing and multiparty computation: extended abstract. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, page 632. ACM, 2004.

- [21] G. Kol and M. Naor. Games for exchanging information. In *Proceedings of the 40th annual ACM symposium on Theory of computing*, pages 423–432. ACM, 2008.
- [22] T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6:4, 1985.
- [23] A. Lysyanskaya and N. Triandopoulos. Rationality and adversarial behavior in multi-party computation. *Advances in Cryptology-CRYPTO 2006*, pages 180–197, 2006.
- [24] H.B. McMahan and A. Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *Proceedings of the 17th Annual Conference on Learning Theory (COLT)*, 2004.
- [25] H. Robbins. Some aspects of the sequential design of experiments. In *Bulletin of the American Mathematical Society*, volume 55, 1952.
- [26] Adi Shamir. How to share a secret. *Communications of the ACM 22 (11)*, pages 612–613, 1979.