My primary research interest is on the design and analysis of algorithms that can handle uncertainty in the input, especially those in the online and stochastic optimization domains. Since many interesting optimization problems are computationally intractable (NP-Hard), we resort to designing approximation algorithms which provably output good solutions. However, a common assumption in traditional algorithms is that the exact input is known in advance. What if this is not the case? What if there is uncertainty in the input? For instance, the job sizes in a scheduling problem may be *revealed only when the job arrives* (online optimization). Or the algorithm has *some distributional information* about each job, but the exact size is revealed only on arrival (stochastic optimization).

With the growing size of input data and their typically distributed nature (cloud computing), it has become imperative for algorithms to handle varying forms of input uncertainty. For example, with new markets emerging across the globe, the long-term problem of placing new data centers (and linking them up fault-tolerantly) can be cast as an online network design problem. On the other hand, the day-to-day task of scheduling requests at these data centers is a stochastic optimization problem, since the algorithm has knowledge of prior request patterns.

Unfortunately, current techniques are not robust enough to deal with many of these problems, thus necessitating the need for new algorithmic tools. Answering such questions, and more generally identifying the tools for solving such problems, has been the driving force of much of my research. For example, we designed the first non-trivial online algorithms for the *fault-tolerant network design* problem in [9], near-optimal online algorithms for the well-studied *broadcast scheduling* problem in [1], and also the first constant-factor algorithms for the *general stochastic knapsack* and the celebrated *multi-armed bandits* problems in [4]. In the following sections, I will discuss the three major threads of my research related to optimization under uncertainty: online optimization, adaptive stochastic optimization and green computing. In each section, I will highlight my current work using a specific problem, and then discuss the future directions. In addition to these areas, I am also interested in, and have worked on other topics in theoretical computer science such as differential privacy [3], traditional approximation algorithms [2, 12, 11], as well as their applications in practice [13].

Adaptive Stochastic Optimization

Consider the following two celebrated classes of problems, *multi-armed bandits* and *restless bandits*, both of which are special cases of *partially observable Markov decision processes* (POMDPs). In these problems, there is a collection of arms (projects), each described by a probabilistic process. The algorithm can (adaptively) work on a particular project at each timestep. The project then makes a transition as dictated by its random process, and fetches the reward associated with its new state. The goal is to maximize the total reward over B timesteps. In the restless version, even the inactive projects can make random transitions, which the algorithm does not observe.

These problems model the fundamental issue in decision making under uncertainty, the *exploration-exploitation (or the risk-reward) tradeoff*. In addition, the restless variant also captures the notion of the projects evolving over time. As a result of their generality, they are being successfully applied in diverse fields like economics, machine learning, queuing theory, networking (adaptive routing), etc.

While these problems have been studied in the above-mentioned areas, the issue of their approximability is much less understood. In fact, prior to our work [4], we did not even know good algorithms for the very special case of the *(correlated) stochastic knapsack* problem[1]. Here, different projects correspond to different jobs, and their probabilistic processes corresponds to randomness over their actual size (and reward). The goal is to (adaptively) schedule the jobs to maximize the total reward of jobs completed within B timesteps. Correlations arise naturally in scheduling — e.g., if a job crashes immediately with some probability (thereby fetching no reward), then its

[1]Even this special case can serve as a good abstraction of problems encountered in real-time scheduling, data centers, OS schedulers, etc. where the algorithm has distributional information based on prior requests.

eventual reward and size are directly correlated.

My current research focuses on developing techniques to first handle such correlations, with the ultimate goal of understanding the approximability of the general variants of the bandit problems. In [4], we designed the first constant-approximations for the correlated stochastic knapsack problem. Our main contribution is in formulating a new *time-indexed LP relaxation*, and using it to capture the marginals of the optimal adaptive solution. Our strengthening is somewhat analogous to the "lift-and-project" ideas used in traditional optimization, and lets us handle pair-wise correlations. This approach is fairly general: using these ideas, we also obtained the first constant-approximation [4] to the optimal adaptive policy, for the general multi-armed bandits problem[2]. Building on this framework, we also designed algorithms [5] for the *stochastic orienteering problem* — here, the stochastic jobs are located at different vertices on a graph, and the algorithm needs to visit a vertex to process the job.

Future Directions. There are several concrete directions to pursue in this area. The restless bandits problem mentioned above still the main algorithmic challenge. While it is hard to approximate in its full generality, there are several interesting special cases that permit good algorithms (e.g., the problem of choosing between multiple bursty channels to transmit data, where the noise on each channel is modeled by a simple Markov chain). A tangible starting point in this direction would be to explore the applicability of our time-indexed LP formulations and adaptive rounding strategies, to simple settings involving restless bandits.

In a different direction, a limitation of most current techniques is that they use LPs to bound the optimal value. What if the underlying problem does not have a good LP relaxation? We handle this issue for the stochastic orienteering problem [5] , but I am interested in the broader goal, which is to identify a general technique.

Looking ahead, my long term objective is to study and understand problems which lie at the *interface of stochastic approximation algorithms and areas like machine learning and stochastic processes*. E.g., consider the following basic problem in machine learning: given a set of classifiers (along with their likelihoods), it is well known that the classifier with minimum average error is the *Bayes optimal classifier*, which outputs the most likely answer at each point. Now, suppose we can (adaptively) query the true label of a few points. What should our query strategy be, in order to minimize the average error of the resultant Bayes optimal classifier? Can our strategy compare favorably with the error of the optimal strategy? We can thus view this, and many other learning problems, from the stochastic approximation algorithms framework. The advantage that this framework offers is that it lets us capture the distance from optimality of our algorithm, *on an instance by instance basis*. Contrast this with the machine learning view, which only give absolute upper and lower bounds *for a class of instances*.

Online Network Design

Consider the following network design problem: we are given a graph, and a collection of source-sink requests (s_i, t_i) arrive over time. The goal is to always maintain the cheapest network, in which each source s_i has *two edge-disjoint paths* (more generally r_i edge-disjoint paths) to the corresponding sink t_i. Once a link is bought, it cannot be deleted subsequently. The objective is to minimize the *competitive ratio*, i.e., the ratio of the cost of the online algorithm's network to that of the optimal offline solution.

These problems are well-motivated from a practical perspective, since they model the task of incrementally building a fault-tolerant network, when the connectivity requirements are only known over time. E.g., new service areas might emerge, and we can only augment our existing network to satisfy the new requirements (as re-solving the problem would correspond to re-laying the network from scratch). While the offline version of this problem has been well-understood, we don't understand the online versions as well. In fact, all previous online algorithms suffer from a major drawback: they only work for the single-connectivity setting.

My research focuses on designing techniques to handle higher connectivity requirements online. In [9], we design the first non-trivial online algorithm for the *fault-tolerant network design* problem. The main technical ingredient is a reduction from the network design problem to a *compact set cover* instance, for which we know good online

[2]Earlier constant-factor algorithms need some form of independence, known as the martingale assumption for the individual arms.

algorithms[3]. Somewhat surprisingly, our reduction uses tree-embeddings, which are one-connected objects, in a novel way to simplify the structure of higher-connectivity requirements.

This result illustrates an interesting consequence of studying alternate computational models: *our online algorithm led to an entirely new approach for solving the offline problem.* Using these ideas, we subsequently designed algorithms for many other offline graph-connectivity problems [10], for which there were no known guarantees.

Future Directions. There are many classical problems in network design for which we do not know of algorithms that can handle input uncertainty. Intuitively, the reason is that natural online algorithms (like greedy) typically have poor gaurantees for these problems, and the known offline solution techniques seem to inherently use the global knowledge of the input. I am interested in exploring our new technique from [9] to understand fault-tolerant versions of other network design problems like facility location, group Steiner connectivity, etc.

In addition to online network design, I have also worked on other online optimization problems such as broadcast scheduling [1] and scheduling with speed-scaling [8, 6]. Looking ahead, I am interested in understanding the class of problems which combine scheduling elements and network-design elements: for example, what if the broadcast servers are located on a network, and can only serve requests up to a particular radius? This would introduce newer challenges, such as network interference, which we must address.

More generally, I am interested in the design of algorithmic techniques that easily permit migrating between offline algorithms and online algorithms. For example, greedy algorithms and the use of tree embeddings fall into this category. In this direction, we have been looking at solving convex programs online (where constraints arrive online), using natural greedy-style update rules [7], and I would like to explore this area further.

Green Computing Algorithmics

This field deals with the design and analysis of algorithms that also take energy into consideration. There are several motivations for studying such problems. One key reason has been the alarming rise in the energy consumed by data centers[4]. Another driving force has been the increasing need for energy efficiency on heterogeneous multi-core chips, in part stemming from the explosion of mobile computing devices.

These issues open a number of algorithmic challenges to address. For example, one approach to manage energy on multi-core chips is through the use of *dynamic speed-scaling* techniques. The main question then is: how do we adjust the speed at which the processors are run, so as to achieve good tradeoff between the energy energy dissipation and the QoS of jobs. In [8], we show that a simple greedy scheduling algorithm (equipped with an equally simply speed-scaling policy) is near optimal for the online version of this problem on a heterogeneous multi-core system (when the QoS is measured as the average response time for the jobs). Subsequently, we also designed *non-clairvoyant algorithms* [6] which don't even know the sizes of jobs until they are completed.

Future Directions. This field is relatively new, and there is great potential for making an impact. I believe that our first goal should be to identify the right models of energy efficiency for various problem situations. For instance, the common models used for energy efficiency are speed-scaling and dynamic re-sizing. However, there are many other techniques for achieving better energy efficiency, like sharing redundant computations and using renewable energy resources. We need good theoretical abstractions to capture these concepts in problem settings like multi-core scheduling, data center management, and network routing. The next challenge would then be to design good algorithms which achieve useful tradeoffs between the resources like energy, QoS, etc. A concrete problem in this direction is that of *energy efficient routing*: here, routers expend energy depending on the speed (or bandwidth) they operate at, and we would need to design algorithms that achieve good tradeoff between the routing delays and the total energy consumed.

[3]Notice that most natural ways to get such a mapping (like viewing graphs cuts as 'elements' and edges crossing them as 'sets') would result in *exponentially large* set systems.

[4]The electricity consumed at data centers is growing at a rate that is ten times faster than that of the total electricity consumed by the US as a whole.

References

[1] Nikhil Bansal, Ravishankar Krishnaswamy, and Viswanath Nagarajan. Better scalable algorithms for broadcast scheduling. ICALP 2010. `http://www.cs.cmu.edu/~ravishan/broadcast.pdf`.

[2] Nikhil Bansal, Ravishankar Krishnaswamy, and Barna Saha. On capacitated set cover problems. APPROX 2011. `http://www.cs.cmu.edu/~ravishan/column-res.pdf`.

[3] Aditya Bhaskara, Ravishankar Krishnaswamy, and Kunal Talwar. Unconditional differentially private mechanism for linear queries. Submitted to STOC 2012. `http://www.cs.cmu.edu/~ravishan/privacy.pdf`.

[4] Anupam Gupta, Ravishankar Krishnaswamy, Marco Molinaro, and R. Ravi. Approximation algorithms for correlated knapsack and non-martingale bandits. FOCS 2011. `http://www.cs.cmu.edu/~ravishan/stocK.pdf`.

[5] Anupam Gupta, Ravishankar Krishnaswamy, Viswanath Nagarajan, and R. Ravi. Approximation algorithms for stochastic orienteering. SODA 2012. `http://www.cs.cmu.edu/~ravishan/stoc-orient.pdf`.

[6] Anupam Gupta, Ravishankar Krishnaswamy, and Kirk Pruhs. Nonclairvoyantly scheduling power-heterogeneous processors. Sustainable Computing: Informatics and Systems, 2011. `http://www.cs.cmu.edu/~ravishan/green.pdf`.

[7] Anupam Gupta, Ravishankar Krishnaswamy, and Kirk Pruhs. Online primal-dual for non-linear optimization with applications to speed scaling. Submitted to STOC 2012. `http://www.cs.cmu.edu/~ravishan/convexpd.pdf`.

[8] Anupam Gupta, Ravishankar Krishnaswamy, and Kirk Pruhs. Scalably scheduling power-heterogeneous processors. ICALP 2010. `http://www.cs.cmu.edu/~ravishan/power.pdf`.

[9] Anupam Gupta, Ravishankar Krishnaswamy, and R. Ravi. Online and stochastic survivable network design. STOC 2009. Invited to Special Issue of SIAM Journal of Computing. `http://www.cs.cmu.edu/~ravishan/cost-shares.pdf`.

[10] Anupam Gupta, Ravishankar Krishnaswamy, and R. Ravi. Tree embeddings for two-edge-connected network design. SODA 2010. `http://www.cs.cmu.edu/~ravishan/2-ecnd.pdf`.

[11] Ravishankar Krishnaswamy, Amit Kumar, Viswanath Nagarajan, Yogish Sabharwal, and Barna Saha. The matroid median problem. SODA 2011. `http://www.cs.cmu.edu/~ravishan/matroid-median.pdf`.

[12] Ravishankar Krishnaswamy and Maxim Sviridenko. Inapproximability results for multi-level facility location. SODA 2012. `http://www.cs.cmu.edu/~ravishan/facloc.pdf`.

[13] Vyas Sekar, Ravishankar Krishnaswamy, Anupam Gupta, and Michael K Reiter. Network-wide deployment of intrusion detection and prevention systems. CoNEXT 2010. `http://www.cs.cmu.edu/~ravishan/conext.pdf`.