Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

Approximate Algorithms for Stochastic Network Design

Xiaojian Wu
College of Information and Computer Sciences
University of Massachusetts Amherst
xiaojian@cs.umass.edu

1 Overview

I study the problems of optimizing a range of stochastic processes occurring in networks, such as the information spreading process in a social network [Kempe *et al.*, 2003; Chen *et al.*, 2010], species migration processes in landscape network [Sheldon *et al.*, 2010; Wu *et al.*, 2013; 2014a; 2014b], virus spreading process in human contact network. In the typical network design framework, the planner can take a set of *management actions*, such as adding edges, removing nodes and so on, to construct a network in order to benefit the activities occurring in the networks. The set of actions being taken is called a policy. To characterize and measure the properties of the constructed network, a function is defined to map each possible policy to a real value. The goal is to select a least costly policy so that the value of the policy is greater or equal to certain value. Well known network design problems are Steiner tree problem [Hwang *et al.*, 1992], maximum spanning tree problem [Kruskal, 1956], survival network design [Nemhauser *et al.*, 1993] and so on. Over the last decades, substantial research has been done on both the theoretical and algorithmic aspects of these problems [Gupta and Könemann, 2011]. However, the standard network design framework fails to capture certain types of practical problems in which (1) the planner can take much richer set of management actions, such as raising the probabilities of certain edge being present or select the source that triggers the stochastic process; and (2) the function that maps each policy to a real value may be defined using probability and expectation. Some of these aspects have been addressed in recent applications, such as maximizing the spread of an endangered species called Red-Cockaded Woodpecker [Sheldon *et al.*, 2010; Kumar *et al.*, 2012]. However, existing solution techniques are not general-purpose frameworks and cannot model a broad class of network design problems. Moreover, the existing techniques that attempt to find high quality policies such as standard mixed integer program (MIP) solver, greedy algorithms or heuristic based methods suffer from limited scalability or poor performance.

My thesis contributes to both modeling and algorithm development. My first goal is to define a unifying network design framework called stochastic network design (SND) to model a broad class of network design problems under stochas- ticity. My second goal, which is my major focus, is to design effective and scalable general-purpose approximate algorithms to solve problems that can be formulated by the SND framework.

2 Stochastic Network Design Framework

The input is a (directed) graph $G = (V, E)$ where V is a set of vertices and E is a set of edges. Each edge is associated with a probability p_{uv} that the edge will be present in the network. Namely, it defines a distribution of networks $G' = (V, E')$ with $E' \subseteq E$. The probability of G' is simply equal to the product of probabilities of all edges being present or absent.

To define the management actions in my framework, I first define a set of non-overlapping domains $\Xi = \{\chi_1, \chi_2, ..., \chi_{|\Xi|}\}$ with $\chi_i \subseteq E$. For each domain, a finite set $A_\chi$of candidate management actions are available, each having certain cost, to change the probabilities of all edges in the domain. Namely, after taking an action $a \in A_\chi$, the new probability of any edge in χ becomes $p_{e|a}$. To unify notations, I let each action set to contain a zero cost "noop" action a_0 represents the case that no action is taken and probabilities remains unchanged. A *policy* π chooses exact one action for each domain. The *cost of a policy* is the total cost of chosen actions. A policy defines a new stochastic network with changed probabilities on edges.

To define the objective of the optimization framework, each pair of vertices is associated with a reward $r_{s,t}$ that encodes the importance to connect this pair. Then, we use $z(\pi) = \sum_{s,t \in V} P(s \rightsquigarrow t|\pi) r(s, t)$ to measure how well the vertices in the network are connected to each other in expectation, where $P(s \rightsquigarrow t|\pi)$ is the probability that s is connected to t under policy π. The decision making problem is to find a policy that maximizes $z(\pi)$ subject to a budget b limiting the total cost of actions being taken, that is, the optimal policy is $\pi^* \in \arg\max_{\{\pi|c(\pi) \leq b\}} z(\pi)$. It has been shown that the problem is at least $\#P$ in general graph [Valiant, 1979], so my goal is to develop efficient approximate algorithms.

3 Approximate Algorithms Overview

So far, I created algorithms for SND problem in two aspects.

1. For tree structured networks, I created fully polynomial-time approximation schemes for SND problem under certain assumptions [Wu *et al.*, 2014a; 2014b].

2. For networks being directed graphs, I created sampling based approximate algorithms which perform well in practice.

3.1 For Tree Structured Networks

In this case, I create FPTAS for two problems:

1. barrier removal in river network to maximize the spread of fish from a unique location, for instance the entrance of the river network, which is similar to the influence maximization problem in social network [Kempe *et al.*, 2003] except that the actions here are to remove barriers but not select sources.
2. barrier removal to maximize the connectivity of river segments, in which the difficulty is the need to consider all pairs of vertices while optimizing the connectivity.

These two problems essentially are correlated such that I can derive FPTAS for both of them using the same techniques. The basic idea is to first create a pseudocode polynomial-time algorithm using dynamic programming technique. Then, a rounding strategy is used to convert it into an FPTAS. Theoretically, the runtime of first problem is $O(\frac{n^2}{\epsilon^2})$ and of the second problem is $O(\frac{n^8}{\epsilon^6})$. Empirically, two algorithms can be implemented very efficiently (especially for the second problem) which outperform the existing techniques both in solution quality and runtime.

3.2 For Directed Graphs

For directed graph, the dynamic program technique is not applicable, so we use the sample average approximation (SAA) scheme to convert the stochastic optimization problem into a discrete optimization problem and then develop efficient approximate algorithms to solve it. Previously, the SAA scheme has been used to solve a spatial conservation planning problem called Red-Cockaded Woodpecker (RCW) problem [Sheldon *et al.*, 2010] where the management actions are to purchase land parcels or vertices in the network. I extended it to solve barrier removal problem where the actions can be taken to increase the passage probabilities of barriers [Wu *et al.*, 2013]. However, the existings methods to solve the converted discrete optimization problem are standard integer programming solver or greedy algorithms, which either appear unscalable on large networks or perform poorly. Recently, to solve the resulted discrete problem, we develped a much faster combinatorial algorithm based on Lagrangian relaxation and primal-dual techniques. We apply the algorithm to the RCW problem and show that the algorithm produces near optimal solution efficiently.

4 Futher Work

One future work will extend my combinatorial algorithm to solve more complex problems, for example the predisaster preparation for road network. Since SAA algorithm usually requires many samples to guarantee the near optimality and the complexity of the discrete problem increases rapidly with the number of samples, another interesting futuer work is to create efficient sequential sampling algorithms, where only a few samples are used in each iteration to update the solution and eventually the solution converges to a near optimal one.

References

[Chen *et al.*, 2010] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM, 2010.

[Gupta and Könemann, 2011] Anupam Gupta and Jochen Könemann. Approximation algorithms for network design: A survey. *Surveys in Operations Research and Management Science*, 16(1):3–20, 2011.

[Hwang *et al.*, 1992] Frank K Hwang, Dana S Richards, and Pawel Winter. *The Steiner tree problem*. Elsevier, 1992.

[Kempe *et al.*, 2003] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth international conference on Knowledge discovery and data mining*, pages 137–146, 2003.

[Kruskal, 1956] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical society*, 7(1):48–50, 1956.

[Kumar *et al.*, 2012] Akshat Kumar, Xiaojian Wu, and Shlomo Zilberstein. Lagrangian relaxation techniques for scalable spatial conservation planning. In *Proceedings of the 26th Conference on Artificial Intelligence*, pages 309–315, 2012.

[Nemhauser *et al.*, 1993] GL Nemhauser, A Rinnooy Kan, S Graves, and P Zipkin. Handbooks in operation research and management science, chapter design of survivable networks. *Optimization*, 1, 1993.

[Sheldon *et al.*, 2010] Daniel Sheldon, Bistra Dilkina, Adam Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla Gomes, David Shmoys, William Allen, Ole Amundsen, and William Vaughan. Maximizing the spread of cascades using network design. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 517–526, 2010.

[Valiant, 1979] Leslie G. Valiant. The complexity of enumeration and reliability problems. *SIAM Journal on Computing*, 8(2):410–421, 1979.

[Wu *et al.*, 2013] Xiaojian Wu, Daniel Sheldon, and Shlomo Zilberstein. Stochastic network design for river networks. In *NIPS Workshop on Machine Learning for Sustainability*, 2013.

[Wu *et al.*, 2014a] Xiaojian Wu, Daniel Sheldon, and Shlomo Zilberstein. Rounded dynamic programming for tree-structured stochastic network design. In *Proceedings of the 28th Conference on Artificial Intelligence*, 2014.

[Wu *et al.*, 2014b] Xiaojian Wu, Daniel Sheldon, and Shlomo Zilberstein. Stochastic network design in bidirected trees. In *Advances in Neural Information Processing Systems*, 2014.