Stochastic Network Design for River Networks

Xiaojian Wu and **Daniel Sheldon** and **Shlomo Zilberstein**
School of Computer Science
University of Massachusetts Amherst
Amherst, MA 01003
{xiaojian,sheldon,shlomo}@cs.umass.edu

Abstract

Stochastic network design techniques can be used effectively to solve a wide range of planning problems in ecological sustainability. We propose a novel approximate algorithm based on the sample average approximation (SAA) and mixed integer programming (MIP) to efficiently address the problem of using a limited budget to remove instream barriers, which prevent fish from accessing their natural habitat. In comparison with a dynamic programming (DP) benchmark algorithm, the advantage of our algorithm is the ability to produce a near optimal solution much faster, particularly when the budget is large and the DP based algorithm becomes intractable. Furthermore, while the DP based algorithm can only solve tree-structured stream networks, our algorithm is applicable to networks with a more general directed acyclic graph structure.

1 Introduction

Many ecological sustainability problems can be modeled as *stochastic network design* problems, which can then be converted to deterministic network design problems using the sample average approximation (SAA) technique [2]. This approach has facilitated the design of effective solutions for several decision problems in spatial conservation planning such as how to maximize the population spread of an endangered species over a period of time [7, 1, 3]. The stochastic optimization problem in [7] was described as a way to maximize the population size under a specific dynamics model, but it can be viewed more generally as *stochastic network design*: edges of a directed host network are present or absent stochastically according to an arbitrary joint probability distribution (in the simplest case, edges fail independently with a specified probability), and actions can be taken to increase the probability of certain edges being present. A budget constraint limits the total cost of actions that can be taken. The objective is to maximize the expected number (or total reward) of nodes that are reachable from a designated source vertex in the realized stochastic network.

In this paper, we adapt the stochastic network design framework to the problem of removing barriers such as dams and culverts to improve habitat connectivity in river networks [6]. We contribute a novel technical extension that allows actions to increase edge probabilities from a base value to one of an arbitrary number of pre-specified values; in previous work actions could only change the probabilities of certain edges from zero to one. The barrier removal problem is an interesting special case of stochastic network design where the network is a directed tree oriented away from the root vertex. Because of the special structure, it can be solved by a psuedo-polynomial time dynamic programming (DP) algorithm [6]. While DP is often practical, it has the downside that it scales quadratically with the available budget, which may be large. We show that the more general solution methodology of SAA and mixed integer programming (MIP) can often find near-optimal solutions in much less time than DP by adjusting the number of samples to trade off accuracy vs. running time.

A major advantage in treating barrier removal under the more general framework of stochastic network design is that it becomes easy to accommodate more general formulations of the problem. Indeed, most river networks are actually directed acyclic graphs (DAGs) and not directed trees, due to "braided" streams which have multiple flow channels. SAA+MIP can handle these networks directly, while DP needs to first apply a heuristic to convert the network to a tree, which discards all guarantees of optimality. A promising line of future work possible using SAA+MIP is to model fish that move in both directions through the network; in the current formulation, fish only move upstream from the ocean, which only makes sense for a limited number of species.

2 The Barrier Removal Problem

Wild fish have shown dramatic population declines over the past two centuries due to the presence of river barriers, such as dams, culvert, floodgates and weirs, which prevent fish from accessing or moving between parts of their historical range [6]. The way to resolve this problem is to retrofit existing barriers or to replace them by new instream structures that could make it easier for fish to get through barriers. O'Hanley and Tomberlin formulated an optimization problem to decide which subset of barriers to repair or remove to maximize available *upstream* habitat for anadromous fish (species like salmon that live part of the year in oceans and streams but travel up rivers and streams to spawn), and gave a pseudo-polynomial time dynamic programming (DP) algorithm to solve the problem [6]. O'Hanley and Tomberlin assumed the network was a directed tree. Here, we describe the more general case where the network is a DAG.

The *fish barrier removal problem* models fish swimming upstream in a DAG $G = (V, E)$ from a unique root vertex r (the ocean) that has a path to every other vertex. Edges represents stream segments and vertices represent either barriers ($v \in B \subseteq V$) or junctions of multiple streams ($v \in V \setminus B$). Fish are able to pass barrier $v \in B$ with probability p_v. A finite set of repair actions A_v are possible. Action $a \in A_v$ has cost c_{va} and, if taken, increases the barrier-passing probability to $p_{v|a}$. We assume A_v includes a default zero-cost "noop" action a_0 such that $p_{v|a_0} := p_v$. A policy π selects a single action $\pi(v)$—either a repair or noop—for each barrier. Given a fixed policy π, the *accessibility* $q_{v|\pi}$ of node v is the probability a fish swimming upstream from r can reach v by any path, assuming that it only attempts to cross each barrier once. If G is a directed tree, there is a unique path from r to v, and $q_{v|\pi}$ is the product of the barrier probabilities on this path. Upon passing v, fish are able to use the amount of habitat h_v between v and its nearest upstream neighbors. The goal is to find a policy π that maximizes the accessibility-weighted habitat $EH(\pi) = \sum_{v \in V} q_{v|\pi} \cdot h_v$, without exceeding a total budget b on the cost of the actions. This model is an instance of stochastic network design where barrier nodes are present or absent stochastically with probability $p_{v|\pi(v)}$ and reward h_v is collected for every node reachable from r in the realized network.[1]

3 Our Method

We present a novel approach to the barrier removal problem based on SAA and MIP. Despite maximizing a very different objective (accessibility-weighted available habitat, which is a structural property of the network), our problem and solution approach are very similar to previous work that maximized population size under a specific population dynamics model [7, 3]. We contribute a novel technical extension that allows multiple actions with different probability values.

Sampling The SAA method converts the stochastic optimization problem of maximizing the expected reward into a deterministic one through sampling [2]. The core idea of converting from a stochastic model to a deterministic model is very general and can be applied to any MDP [5]. An important aspect of this work is that the resulting deterministic problem has a special network design structure that can be formulated as a MIP. Roughly speaking, the key property is that all possible policies can be efficiently explored in a MIP framework using the same randomly sampled values.

Our new construction for multiple different actions at each barrier is designed to retain this property. It works as follows: in the sth sampled network, for each barrier v, we generate an independent uniform random variable $U_v^s \in [0, 1]$. Then, we declare each action $a \in A_v$ to be *successful* for

[1] The same model can be viewed in terms of a network with stochastic *edges* instead of nodes, by declaring all edges leaving v to be present or absent stochastically with probability $p_{v|a}$.

(a) Complete stream network with the river highlighted (b) The river with barriers

Figure 1: Stream network of Massachusetts

sample s if $U_v^s \le p_{a|v}$. Note that the value U_v^s can be thought of as how unlucky the fish will be in passing barrier v in the sth trial: if the value is high, then a high-probability action must be taken to allow the fish to pass the barrier. Let $\mathcal{A}_v^s$ be the set of successful actions. For any policy π, node v is considered present (i.e. barrier v is passable) in the sampled network if any only if the selected action $a = \pi(v)$ belongs to $\mathcal{A}_v^s$. This event has probability $p_{v|a}$. The significance of this construction is that we can evaluate the passability of barrier v under *any* policy π simply by checking whether the selected action belongs to the successful set of actions. It is then easy to verify that the expected value of the total habitat $H_s(\pi)$ reachable from r under policy π in the sampled network is exactly $EH(\pi)$.

It is obvious that, for any fixed policy π, the sample average $\frac{1}{S}\sum_{s=1}^{S} H_s(\pi)$ converges to $EH(\pi)$. Basic results surrounding SAA also guarantee that the value of the optimal sample-average policy $\max_{\pi\in\Pi} \frac{1}{S}\sum_{s=1}^{S} H_s(\pi)$ over some feasible set Π also converges to the value of the optimal policy $\max_{\pi\in\Pi} EH(\pi)$ with respect to the true expected value [2]. The SAA technique optimizes this sample average instead of the true expected value.

Mixed Integer Program The following MIP finds a policy that maximizes the average accessible habitat $H_s(\pi)$ of the sampled network while respecting the budget constraint.

Let X_v^s be a binary variable indicating whether node v is reachable in the sampled network s, and let Y_{va} be a binary variable indicating whether or not the policy takes action a at barrier v. Let $Pa^s(v)$ be the parent set of v in network s. The first constraint indicates that the root vertex is always reachable, while the second dictates that node u can only be reachable if at least one of its parents is reachable. The third constraint further dictates that barrier node v is reachable only if at least one of the successful actions from $\mathcal{A}_v^s$ is taken. The final constraint enforces the budget limitation. There is always an optimal solution that takes at most one action per barrier—in practice, it can be helpful to enforce this as a constraint. Also, the binary variables X_v^s can be relaxed so that $0 \le X_v^s \le 1$ (e.g., see [7]).

$$
\begin{aligned}
\max \quad & \frac{1}{S}\sum_{s=1}^{S}\sum_{v\in V} X_v^s \cdot h_v & \\
\text{s.t.} \quad & X_r^s = 1 & \forall s \in S \\
& X_v^s \le \sum_{u\in Pa(v)} X_u^s & \forall v \in V, s \in S \\
& X_v^s \le \sum_{a\in\mathcal{A}_v^s} Y_{va} & \forall v \in B, \forall s \in S \\
& \sum_{v\in B}\sum_{a\in A_v} c_{va}\cdot Y_{va} \le b &
\end{aligned}
$$

4 Experimental Results

We ran experiments using river network and barrier data for the state of Massachusetts from the Conservation Assessment and Prioritization System (CAPS) project [4]. See Fig. 1b.

We evaluated both our SAA+MIP algorithm as well as the dynamic programming (DP) algorithm of [6] on a single watershed in southwestern MA (Fig. 1a) for a variety of different budgets. To be able to run the DP algorithm, we extracted a tree from the braided river network using a minimum spanning tree algorithm. The resulting network has 4045 vertices and 1523 barriers (Fig. 1b). We assumed that each barrier where streams crossed roads (e.g., culverts) had default passability of 0.1, and could be raised to a random number within either $[0.2, 0.5]$ or $[0.7, 1.0]$ by actions of cost within $[10, 15]$ and $[20, 28]$, respectively. Dams had default passability of 0, and could be raised to a random number within either $[0.2, 0.5]$ or $[0.7, 1.0]$ at costs within $[100, 120]$ and $[200, 230]$, respectively.

To compare runtime vs. solution quality, Fig. 2 shows the runtime of SAA+MIP as a fraction of the DP runtime (horizontal axis), and the value of SAA+MIP as a fraction of the DP value (vertical axis). Each line corresponds to a different budget, and shows the range of performance SAA+MIP can achieve with different numbers of samples S ranging from 5 to 200. The total cost of a policy that takes the most expensive action at each barrier is about 40K; we test budgets up to 30K. The figure shows that, with the same time as DP, SAA+MIP achieves 95% of the optimal value over a range of budgets. In all cases, SAA+MIP can achieve a reasonably good solution (say, within 80% of optimal) in a small fraction of the DP running time. Large budgets slow down the psuedo-polynomial DP algorithm considerably but lead to easier MIPs—in these cases, SAA+MIP can achieve performance within 5% or 10% of optimal much faster than DP. For example, with budget of 30K, DP takes about 600 seconds to find an optimal solution while SAA+MIP only takes 30 seconds to achieve 95% optimality.

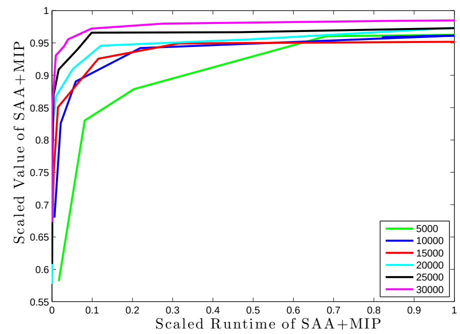

Figure 2: Comparing DP with SAA+MIP

5 Discussion

Another significant advantage of SAA+MIP is the ability to deal with more general problem formulations. One such example is a DAG corresponding to braided river networks. One can apply DP in this situation only by applying a heuristic to first convert the DAG into a tree (e.g., by generating a spanning tree), solving the problem, and then applying the resulting barrier treatment to the original DAG. We found that this approach worked well in our example network, but it provides no guarantee on the quality of the solution. Running SAA+MIP in this case is still worthwhile because it provides a guarantee on distance from optimality (which is why we know that DP is nearly optimal). Interestingly, the solution quality of DP on a subtree deteriorates with lower budgets to the point where SAA+MIP on the original DAG produces a better value (for example, with a budget of 1000 or 2000). Intuitively, with a limited budget, it is more critical not to remove barriers where ignored edges could help improve connectivity. Furthermore, examples such as the one in Fig. 3 underscore the fact that applying DP on a subtree can be arbitrarily worse than the optimal solution for the original DAG. In the example, there are two barriers vertices, 2 and 3, of which one can be repaired. The rewards are integer multiples of x, and the dashed edge is removed to obtain a tree. Vertex 2 will be repaired because it gives value $4x$ in the subtree (and also the DAG), while repairing 3 only gives value $3x$. However, the optimal policy for the DAG is to remove 3, which gives value $5x$. By increasing x, the gap can be made arbitrarily large. More complex examples can be constructed; we suspect it is difficult in general to devise a heuristic for extracting subtrees that can avoid all bad cases.

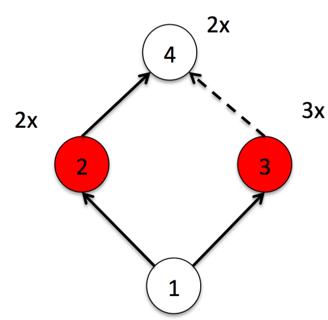

Figure 3: Sample DAG

Acknowledgments

We thank Brad Compton and Scott Jackson for providing the river network and barrier data and helping with the interpretation of the data.

References

[1] Daniel Golovin, Andreas Krause, Beth Gardner, Sarah J. Converse, and Steve Morey. Dynamic resource allocation in conservation planning. In *Proceedings of the 25th Conference on Artificial Intelligence*, pages 1331–1336, 2011.

[2] Anton J. Kleywegt, Alexander Shapiro, and Tito Homem-de-Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12:479–502, 2002.

[3] 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.

[4] Kevin McGarigal, Scott Jackson, and Brad Compton. The conservation assessment and prioritization system. `www.umasscaps.org`, 2013.

[5] Andrew Y. Ng and Michael Jordan. Pegasus: A policy search method for large MDPs and POMDPs. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 406–415, 2000.

[6] Jesse Rush OHanley and David Tomberlin. Optimizing the removal of small fish passage barriers. *Environmental Modeling & Assessment*, 10(2):85–98, 2005.

[7] 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.