Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-16)

Robust Decision Making for Stochastic Network Design

Akshat Kumar,[†] Arambam James Singh,[†] Pradeep Varakantham,[†] Daniel Sheldon[‡]
[†]School of Information Systems, Singapore Management University
{akshatkumar,ajsingh,pradeepv}@smu.edu.sg
[‡]College of Information and Computer Sciences, University of Massachusetts Amherst
tsheldon@cs.umass.edu

Abstract

We address the problem of robust decision making for stochastic network design. Our work is motivated by *spatial conservation planning* where the goal is to take management decisions within a fixed budget to maximize the expected spread of a population of species over a network of land parcels. Most previous work for this problem assumes that accurate estimates of different network parameters (edge activation probabilities, habitat suitability scores) are available, which is an unrealistic assumption. To address this shortcoming, we assume that network parameters are only partially known, specified via interval bounds. We then develop a decision making approach that computes the solution with *minimax regret*. We provide new theoretical results regarding the structure of the minmax regret solution which help develop a computationally efficient approach. Empirically, we show that previous approaches that work on point estimates of network parameters result in high regret on several standard benchmarks, while our approach provides significantly more robust solutions.

1 Introduction

Several dynamic processes over a network such as spread of information and opinions, viral marketing (Domingos and Richardson 2001; Kempe, Kleinberg, and Tardos 2003; Leskovec, Adamic, and Huberman 2007), and disease propagation among humans (Anderson and May 2002) can be described as a diffusion or cascade over the network. The spread of wildlife over a network of land patches can be described using a similar diffusion process, also known as *metapopulation modeling* in ecology (Hanski 1999). Our work is motivated by the spatial conservation planning problem where goal is to find strategies to conserve land parcels to maximize the expected spread of the endangered wildlife. Conservation strategies, which correspond to *network design*, include deciding which land parcels to purchase for conservation within a fixed budget. This problem has recently received much attention in the AI community (Sheldon et al. 2010; Ahmadizadeh et al. 2010; Golovin et al. 2011; Kumar, Wu, and Zilberstein 2012; Wu et al. 2013; Wu, Sheldon, and Zilberstein 2014; Xue, Fern, and Sheldon 2014; 2015; Wu, Sheldon, and Zilberstein 2015).

A key assumption implicit in most such previous approaches is that accurate estimates of different parameters of the underlying network diffusion process, such as edge activation probabilities and habitat suitability scores, are known. However, this assumption is practically unrealistic. Even using learning approaches to estimate parameter values from observed data (Kumar, Wu, and Zilberstein 2012) may not lead to precise estimates due to missing, noisy data. Therefore, in our work we develop conservation strategies in the presence of partially specified diffusion process parameters. We assume the commonly used notion of interval bounds (Boutilier et al. 2003), where only upper and lower bounds on different parameter values are known. The decision problem in this case is that of computing a *robust* conservation strategy that minimizes the maximum regret (minimax regret) within the space of feasible parameter values.

The *independent cascade* (IC) model (Kempe, Kleinberg, and Tardos 2003) is the basic building block that describes the diffusion process in social networks as well as in metapopuplation modeling in ecology (Hanski 1994). Recently, there is increasing interest in modeling the effects of adding random noise to edge parameters of the network—probabilities p_{uv} that model the strength of influence from node u to v (Adiga et al. 2014; Goyal, Bonchi, and Lakshmanan 2011; He and Kempe 2014). However, He and Kempe show that most *independent random* noise models can be subsumed within the IC model and as such, do not add anything new to the IC model. From a practical perspective, the noise in estimating network parameters may not be random or independent due to systemic bias in algorithmic techniques and observed data. Therefore, it is better to consider an adversarial setting which provides a worst case analysis (He and Kempe 2014). This motivates our goal of computing the minimax regret solution.

While He and Kempe [2014] also consider an adversarial setting for the IC model, our work differs from it significantly. The analysis of He and Kempe is mainly focused on diagnosing instability in the presence of noise. If the network is deemed unstable, there is no algorithmic recourse presented in (He and Kempe 2014) to compute a robust solution. Our work addresses this issue by computing the minimax regret solution for network design in the presence of noise. Furthermore, directly computing the minimax regret solution is not tractable as the feasible parameter space is

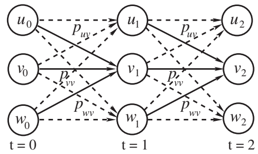


Figure 1: Time indexed layered graph. Solid edges are active, dashed are inactive for this example cascade.

continuous. We therefore develop new theoretical insights that characterize the properties of the minimax regret solution making its computation tractable. To improve tractability further, we use an iterative constraint generation procedure to minimize the maximum regret, and incorporate the sample average approximation (SAA) framework to address the stochasticity in the network design. Empirically, we show that previous approaches that work on point estimates of network parameters result in high regret on several standard benchmarks, while our approach provides significantly more robust solutions. We also provide operational insights showing how the decision by assuming point estimates of parameters can be disadvantageous in an adversarial setting as it may lead to isolation of wildlife habitats.

2 The Conservation Planning Problem

We first provide an overview of the stochastic network design problem for conservation planning (Sheldon et al. 2010). In this problem, the goal is to design conservation strategies to maximize the expected spread of a species through a network of habitat patches over a given time period. The spread of species is modeled as a stochastic cascade or diffusion over a network of *habitat patches* $\mathcal{H}$, which are analogous to nodes in a graph. The species can only survive within habitat patches that are conserved. The habitat patches are grouped into non-overlapping *land parcels* $1, \ldots, L$. A land parcel l is available for purchase at a cost c_l. A conservation strategy or the network design problem is to decide which land parcels to purchase and conserve within a fixed budget $\mathcal{B}$ over the time period T.

The metapopulation model used to describe the stochastic patch occupancy dynamics (Hanski and Ovaskainen 2000) is similar to the independent cascade (IC) model. The diffusion process starts from the initial source set $\mathcal{S}$ of occupied patches at time $t=0$. At each subsequent time step, the following stochastic events can take place. The population at an occupied patch u at time t can colonize an unoccupied patch v at time $t+1$ with probability p_{uv}. Another possibility is that the population at patch u becomes extinct at time $t+1$ with probability $1-p_{uu}$.

The goal of the conservation planning problem is to select a set of land parcels to purchase that maximize the expected number of patches occupied at time T. Let the conservation strategy be denoted using a binary vector $\boldsymbol{y}=\langle y_1, \ldots, y_L\rangle$. Each binary variable y_l denotes whether the corresponding land parcel l is purchased or not. Let binary random variables $X_{v_t}(\boldsymbol{y})$ denote whether the patch v is occupied or unoccupied at time t under a given strategy $\boldsymbol{y}$. Let $\mathcal{B}$ denote the given budget and T denote the plan horizon. The stochastic network design problem is:

$$\max_{\boldsymbol{y}} \sum_{v\in\mathcal{H}} \mathbb{E}\big[X_{v_T}(\boldsymbol{y})\big] \quad \text{s.t. } \sum_{l=1}^{L} c_l y_l \leq \mathcal{B} \tag{1}$$

Notice that problem (1) is a stochastic optimization problem (Kleywegt, Shapiro, and Homem-de-Mello 2002) as the objective function is an expectation over all possible cascades. A principled approach is to solve this problem approximately by generating N independent cascade scenarios from the underlying stochastic diffusion model. This results in a deterministic approximation of the given stochastic problem, which is easier to solve. This approach is also known as sample average approximation (SAA) (Kleywegt, Shapiro, and Homem-de-Mello 2002), and has been used previously for this problem (Sheldon et al. 2010; Kumar, Wu, and Zilberstein 2012).

SAA Sampling We provide a brief overview of the SAA scheme for the conservation planning problem; for details we refer to (Sheldon et al. 2010). We first define a *time-indexed* layered graph $G=(V,E)$ as shown in figure 1. In this graph, there is a node u_t for each habitat patch $u\in\mathcal{H}$ and time step t. An edge (u_t, v_{t+1}) is present if the associated probability $p_{u_t v_{t+1}}>0$.

An SAA cascade k is generated by sampling from a biased coin with probability $p_{u_t v_{t+1}}$ independently for each edge (u_t, v_{t+1}) in this graph. If the toss outcome is *heads*, the edge is *active* and is included in the cascade, else it is inactive. Thus, a cascade k defines a subgraph $G^k=(V,E^k)$ where only active edges are included in E^k. Given a conservation strategy $\boldsymbol{y}$, we can determine if a node v_t is occupied in cascade k by checking if there exists a *valid* path from some node u_0 to v_t in G^k such that 1) patch u is occupied at time 0, 2) all the patches along the path are purchased.

Let the activation of node v_t under a cascade k and strategy $\boldsymbol{y}$ be denoted as $X^k_{v_t}(\boldsymbol{y})$. The SAA approximation for N training cascades is given as:

$$\max_{\boldsymbol{y}} \sum_{v\in\mathcal{H}} \mathbb{E}\big[X_{v_T}(\boldsymbol{y})\big] \;\approx\; \max_{\boldsymbol{y}} \frac{1}{N}\sum_{k=1}^{N}\sum_{v\in\mathcal{H}} X^k_{v_T}(\boldsymbol{y}) \tag{2}$$

The SAA scheme provides good convergence guarantees—as N goes to infinity, the SAA objective converges to the original expected objective (Kleywegt, Shapiro, and Homem-de-Mello 2002).

3 Regret Based Network Design

Most previous approaches assume that point estimates of colonization probabilities p_{uv} and other parameters such as suitability scores of habitat patches, are know a priori (Sheldon et al. 2010; Kumar, Wu, and Zilberstein 2012; Wu, Sheldon, and Zilberstein 2014). As highlighted previously, this is an unrealistic assumption. Therefore, our goal in this section is to develop the theory and a practical approach to compute robust solutions when network parameters are only partially known.

Similar to previous works (He and Kempe 2014; Boutilier et al. 2003), we assume only the knowledge of upper and lower bounds on different parameters. That is, for edge probabilities, we have $p^{\text{true}}_{uv} \in [\underline{p}_{uv}, \overline{p}_{uv}]$, where p^{true} is the true (but unknown) parameter. In the presence of such uncertainty, a natural approach is to compute a solution $\boldsymbol{y}$ that obtains minimum max-regret, where max-regret is the largest quantity by which the decision maker could regret taking the decision $\boldsymbol{y}$ while allowing different parameters to vary within the bounds (Boutilier et al. 2003). That is, *nature* acts as an adversary and chooses parameters (within the allowed bounds) that maximize the regret of decision $\boldsymbol{y}$. We next formalize these notions for conservation planning.

3.1 The MMR Formulation

We now describe the minimax regret (MMR) formulation for our network design problem. Let $\boldsymbol{p} = \{p_{uv}\ \forall (u,v)\}$ denote the edge probabilities for the entire time-indexed graph G (as shown in figure 1). For ease of exposition, we drop dependence on time; edge (u,v) always implies if node u is in time slice t, then node v belongs to slice $t+1$. Let $\mathcal{P} = \times_{(u,v)}\mathcal{P}_{uv}$ denote the entire possible space of edge probabilities that are allowed as per the known upper and lower bounds. Each set $\mathcal{P}_{uv}$ is defined as: $\mathcal{P}_{uv} = \{p_{uv}\,|\,\underline{p}_{uv} \le p_{uv} \le \overline{p}_{uv}\}$. The *pairwise regret* of a decision $\boldsymbol{y}$ w.r.t. $\boldsymbol{y}'$ over $\mathcal{P}$ is:

$$R(\boldsymbol{y},\boldsymbol{y}';\mathcal{P}) = \max_{\boldsymbol{p}\in\mathcal{P}}\Big[\sum_{v\in\mathcal{H}}\mathbb{E}\big[X_{v_T}(\boldsymbol{y}');\boldsymbol{p}\big] - \sum_{v\in\mathcal{H}}\mathbb{E}\big[X_{v_T}(\boldsymbol{y});\boldsymbol{p}\big]\Big] \quad (3)$$

Using the regret function defined above, we define the *maximum regret* of a decision $\boldsymbol{y}$ as follows:

$$\mathrm{MR}(\boldsymbol{y},\mathcal{P}) = \max_{\boldsymbol{y}'} R(\boldsymbol{y},\boldsymbol{y}';\mathcal{P}) \quad (4)$$

Using the above formulation, the final MMR criterion is:

$$\mathrm{MMR}(\mathcal{P}) = \min_{\boldsymbol{y}} \mathrm{MR}(\boldsymbol{y},\mathcal{P}) \quad (5)$$

Our goal in this work is to solve the problem (5) for the conservation planning problem.

3.2 Pairwise Regret Computation

The first step for computing the MMR solution is to be able to compute the pairwise regret (3) between any two decisions $\boldsymbol{y}$ and $\boldsymbol{y}'$. We first establish some results about pairwise regret that make the computation tractable. We start with the following proposition for the time indexed graph G.

Proposition 1. *Assume that all the network parameters $\boldsymbol{p}_{-e}$ are fixed except the edge probability p_e for a particular edge e. The expected objective $\sum_v \mathbb{E}\big[X_{v_T}(\boldsymbol{y});\boldsymbol{p}\big]$ is linear in the edge probability p_e.*

Proof. We consider the following objective function:

$$F(p_e;\boldsymbol{p}_{-e},\boldsymbol{y}) = \sum_{v\in\mathcal{H}}\mathbb{E}\big[X_{v_T}(\boldsymbol{y});\boldsymbol{p}\big] = \sum_{\xi}\Pr(\xi)\sum_{v\in\mathcal{H}}\xi_{v_T}(\boldsymbol{y}) \quad (6)$$

where ξ denotes a particular outcome in the probability space of the underlying distribution $\boldsymbol{p}$, also referred to as a cascade. That is, ξ includes a binary random variable $\xi_{e'}$ for each edge e' in the time indexed graph. The random variable $\xi_{e'}$ is set to the outcome of the biased coin toss (with success probability $p_{e'}$). Notice that coin toss is independent for each edge. Based on the complete cascade ξ, we can define a subgraph $G_\xi = (V, E_\xi)$ of the original time indexed graph where an edge $e \in E_\xi$ iff $\xi_e = 1$.

Binary variable ξ_{v_T} denotes whether the node v_T is active or inactive as per the cascade ξ and decision $\boldsymbol{y}$. We expand the above expression further as:

$$\sum_{\xi}\Pr(\xi)\sum_{v\in\mathcal{H}}\xi_{v_T}(\boldsymbol{y}) = \sum_{\xi}\prod_{f\in E_\xi}p_f\prod_{f\in E\setminus E_\xi}(1-p_f)\big[\sum_{v\in\mathcal{H}}\xi_{v_T}(\boldsymbol{y})\big]$$

Notice that in the above expression, the probability p_e for the edge e appears *exactly* once in each term of the sum ($\sum_\xi$). Therefore, the above expression, and hence the expected objective is linear in the probability p_e. □

Based on prop. 1, we can write the expected objective as a function of probability p_e as below:

$$\begin{aligned}\sum_{v\in\mathcal{H}}\mathbb{E}\big[X_{v_T}(\boldsymbol{y});\boldsymbol{p}\big] &= F(p_e;\boldsymbol{p}_{-e},\boldsymbol{y})\\ &= p_e F_1(\boldsymbol{p}_{-e};\boldsymbol{y}) + F_2(\boldsymbol{p}_{-e};\boldsymbol{y}) \quad (7)\end{aligned}$$

where functions F_1 and F_2 depend only on parameters $\boldsymbol{p_{-e}}$. Based on this proposition, we show the following result.

Proposition 2. *There exists an edge-probability vector $\boldsymbol{p}^\star$ that provides the pairwise regret $R(\boldsymbol{y},\boldsymbol{y}';\mathcal{P})$ by maximizing the right-hand-side of* (3) *such that each probability $p^\star_e$ is at one of the extremes of the allowed range:*

$$p^\star_e \in \big\{\underline{p}_e, \overline{p}_e\big\}\ \forall e$$

Proof. We prove by contradiction. Let us assume that the probability $\boldsymbol{p}^\star$ has at least one edge p_e that is not at the extreme point of its range. The pairwise regret of Eq. (3) is:

$$\begin{aligned}R(\boldsymbol{y},\boldsymbol{y}';\mathcal{P}) &= \Big[\sum_{v\in\mathcal{H}}E\big[X_{v_T}(\boldsymbol{y}');\boldsymbol{p}^\star\big] - \sum_{v\in\mathcal{H}}E\big[X_{v_T}(\boldsymbol{y});\boldsymbol{p}^\star\big]\Big]\\ &= \Big[p^\star_e F_1(\boldsymbol{p}^\star_{-e};\boldsymbol{y}') + F_2(\boldsymbol{p}^\star_{-e};\boldsymbol{y}') - p^\star_e F_1(\boldsymbol{p}^\star_{-e};\boldsymbol{y}) - F_2(\boldsymbol{p}^\star_{-e};\boldsymbol{y})\Big]\\ &= p^\star_e\Big[F_1(\boldsymbol{p}^\star_{-e};\boldsymbol{y}') - F_1(\boldsymbol{p}^\star_{-e};\boldsymbol{y})\Big] + F_2(\boldsymbol{p}^\star_{-e};\boldsymbol{y}') - F_2(\boldsymbol{p}^\star_{-e};\boldsymbol{y})\end{aligned}$$

Only the first term in the above expression depends on $p^\star_e$. If we have $F_1(\boldsymbol{p}^\star_{-e};\boldsymbol{y}') > F_1(\boldsymbol{p}^\star_{-e};\boldsymbol{y})$, then we set $p^\star_e$ to the upper bound $\overline{p}_e$ to increase the regret. If we have $F_1(\boldsymbol{p}^\star_{-e};\boldsymbol{y}') < F_1(\boldsymbol{p}^\star_{-e};\boldsymbol{y})$, then we set $p^\star_e$ to the lower bound $\underline{p}_e$ to increase the regret. Both these situations are a contradiction as we assumed that $\boldsymbol{p}^\star$ was the optimal probability that maximized the right hand side of Eq. (3). In case $F_1(\boldsymbol{p}^\star_{-e};\boldsymbol{y}') = F_1(\boldsymbol{p}^\star_{-e};\boldsymbol{y})$, we can always set $p^\star_e$ to any extreme of its allowed range without affecting the regret.

Therefore, optimal probability must be the either extreme end of the allowed range. □

Prop. 2 makes pairwise regret computation R, which is a stochastic optimization problem, significantly more tractable as it allows us to integrate the SAA procedure for pairwise regret. We utilize this fact for tractability in the next sections.

3.3 Maximum Regret (MR) Computation

In this section, we develop a scalable computation approach to compute the max-regret MR (4). Notice that computing the pairwise regret R (3) (a sub-step of MR) is a stochastic optimization problem. Therefore, we plan to use the SAA scheme as highlighted in section 2. However, applying SAA to pairwise regret (3) is not straightforward as the underlying distribution $\boldsymbol{p}$ itself is a variable. Therefore, how to generate samples before solving the optimization problem itself is not clear. We propose a novel solution to this problem. Our strategy will be to simultaneously optimize over the distribution $\boldsymbol{p}$ in (3) and generate samples from $\boldsymbol{p}$ required for the SAA approximation on-the-fly by embedding the inverse transform sampling (ITS) procedure for a Bernoulli distribution (or a biased coin) within a single mathematical program. We detail this process below.

Consider the sampling procedure for an edge e with probability p_e. We first generate a uniform random number r_e. If $r_e \leq p_e$, then edge activation $x_e = 1$, else $x_e = 0$. We cannot apply directly this procedure to pairwise regret computation as the probability p_e is not known. However, according to prop. 2, we know that p_e must be the either extreme. We exploit this fact to write a set of linear constraints that encode the sampling procedure. Let I_e denote the binary variable for the edge e. If $I_e = 1$, then $p_e = \overline{p}_e$, else $p_e = \underline{p}_e$. For any value of I_e, the sampling procedure is encoded via following linear constraints:

$$x_e = I_e \quad \text{If } \underline{p}_e < r_e \leq \overline{p}_e \tag{8}$$

$$x_e = 0 \quad \text{If } r_e > \overline{p}_e \tag{9}$$

$$x_e = 1 \quad \text{If } r_e \leq \underline{p}_e \tag{10}$$

The validity of above constraints can be easily checked for different values of I_e ($\in \{0,1\}$) and the random number r_e. Therefore, to simulate N SAA samples, we generate apriori N uniform random numbers $r_e^k,\ k \in \{1,\dots,N\}$ for each edge e in the time indexed graph. The SAA approximation for the MR (4) can then be written as the following program:

$$\max_{\boldsymbol{y}', \boldsymbol{I}, \{\boldsymbol{x}_{\boldsymbol{y}}^{\boldsymbol{k}}\}, \{\boldsymbol{x}_{\boldsymbol{y}'}^{\boldsymbol{k}}\}} \frac{1}{N}\Big[\sum_{k=1}^{N}\sum_{v\in\mathcal{H}} x_{v,y'}^k - \sum_{k=1}^{N}\sum_{v\in\mathcal{H}} x_{v,y}^k\Big] \tag{11}$$

$$\text{s.t. } \boldsymbol{x}_{\boldsymbol{y}}^{\boldsymbol{k}} \in \Omega(\boldsymbol{x}_{\boldsymbol{y}}^{\boldsymbol{k}}, \boldsymbol{y}, \boldsymbol{I}, \boldsymbol{r}^k) \quad \forall k \in \{1,\dots,N\} \tag{12}$$

$$\boldsymbol{x}_{\boldsymbol{y}'}^{\boldsymbol{k}} \in \Omega(\boldsymbol{x}_{\boldsymbol{y}'}^{\boldsymbol{k}}, \boldsymbol{y}', \boldsymbol{I}, \boldsymbol{r}^k) \quad \forall k \in \{1,\dots,N\} \tag{13}$$

where $\boldsymbol{y}'$ is the decision (to be optimized) that provides the max-regret for the given decision $\boldsymbol{y}$; $\boldsymbol{I} = \{I_e\ \forall e \in E\}$ is the set of binary variables for each edge denoting whether the corresponding p_e is $\underline{p}_e$ or $\overline{p}_e$; $\boldsymbol{x}_{\boldsymbol{y}}^{\boldsymbol{k}} = \{x_{v,y}^k \forall v\} \cup \{x_{e,y}^k \forall e\}$ is the set of binary variables for each node v and edge e in the time-indexed graph. $x_{v,\boldsymbol{y}}^k$ denotes whether the node v becomes occupied for the SAA sample k and the decision $\boldsymbol{y}$; $x_{e,y}^k$ denotes whether the edge e becomes *active* for sample k and the decision $\boldsymbol{y}$; $\boldsymbol{x}_{\boldsymbol{y}'}^{\boldsymbol{k}}$ denotes the same for the decision $\boldsymbol{y}'$. The parameters $\boldsymbol{r}^k = \{r_e^k\ \forall e \in E\}$ are uniform random numbers for the SAA sample k.

Table 1 shows different constraints that define the constraint set Ω. Constraint (**??**) enforces the budget constraint that the cost of land parcels purchased should be within the given budget. Constraints (**??**)–(**??**) encode the sampling procedure for each edge e and the given random number r_e. If $x_e = 1$, then the edge is considered active, else inactive. *Edge transmission constraints* use the variable t_e to encode that an edge $e = (u, v)$ is able to colonize the node v iff (1) node u is occupied (encoded by x_u) (2) edge (u, v) is active (encoded by x_{uv}). Constraints (**??**)–(**??**) encode whether there is at least one incoming edge (u, v) for node v that can successfully colonize node v. Constraints (**??**)–(**??**) encode that a node v becomes occupied iff 1) there is at least one incoming edge that can colonize v (denoted by variable n_v), 2) the land parcel $\mathcal{A}(v)$ corresponding to node v is purchased (denoted by variable $y_{\mathcal{A}(v)}$). Finally, constraints (**??**)–(**??**) denote which nodes are initially occupied/unoccupied. Notice that only the variables $\boldsymbol{I}$ that denote whether the probability p_e should be min or max and decision variables $\boldsymbol{y}$ are binary, the rest are continuous.

Constraint set $\Omega(\boldsymbol{x_y}, \boldsymbol{y}, \boldsymbol{I}, \boldsymbol{r})$		
`Budget constraint:`		
$\sum_{l=1}^{L} c_l y_l \leq \mathcal{B}$		(14)
`Edge Sampling Constraint:`		
$x_e = I_{uv}$ If $\underline{p}_e < r_e \leq \overline{p}_e$	$\forall e \in E$	(15)
$x_e = 0$ If $r_e > \overline{p}_e$	$\forall e \in E$	(16)
$x_e = 1$ If $r_{uv} \leq \underline{p}_e$	$\forall e \in E$	(17)
`Edge transmission constraints:`		
$t_e \geq x_u + x_{uv} - 1$	$\forall e = (u,v) \in E$	(18)
$t_e \leq x_u$	$\forall e = (u,v) \in E$	(19)
$t_e \leq x_{uv}$	$\forall e = (u,v) \in E$	(20)
`Atleast One Neighbor Active Constraint:`		
$n_v \leq \sum_{(u,v)\in E} t_{uv}$	$\forall v \in V$	(21)
$n_v \geq t_{uv}$	$\forall (u,v) \in E, \forall v \in V$	(22)
`Node Activation Constraint:`		
$x_v \geq n_v + y_{\mathcal{A}(v)} - 1$	$\forall v \in V$	(23)
$x_v \leq n_v$	$\forall v \in V$	(24)
$x_v \leq y_{\mathcal{A}(v)}$	$\forall v \in V$	(25)
`Initially occupied nodes:`		
$x_v = 1$	$\forall v \in \mathcal{S}$	(26)
$x_v = 0$	$\forall v \in V_0 \setminus \mathcal{S}$	(27)
`Continous/Binary variables:`		
$x_v, x_e, n_v, t_e \in [0,1]$		(28)
$I_e \in \{0,1\},\ y_v \in \{0,1\}$		(29)

Table 1: Constraint set Ω for max-regret mixed-integer program

3.4 Minimizing the Maximum Regret

In this section, we present an iterative constraint generation approach to compute the MMR solution for the network design problem. We can simplify the MMR problem as below:

$$\mathrm{MMR}(\mathcal{P}) = \min_{\boldsymbol{y}} \mathrm{MR}(\boldsymbol{y}, \mathcal{P}) \tag{14}$$

$$= \min_{\boldsymbol{y}} \max_{\boldsymbol{y}', \boldsymbol{p}} \big[U(\boldsymbol{y}', \boldsymbol{p}) - U(\boldsymbol{y}, \boldsymbol{p})\big] \tag{15}$$

where we defined $U(\boldsymbol{y}, \boldsymbol{p}) = \sum_{v\in\mathcal{H}} \mathbb{E}\big[X_{v_T}(\boldsymbol{y}); \boldsymbol{p}\big]$. Notice that for a fixed $\boldsymbol{y}$, the quantities $\boldsymbol{y}'^{*}$ and $\boldsymbol{p}^{\star}$ that maximize $\big[U(\boldsymbol{y}', \boldsymbol{p}) - U(\boldsymbol{y}, \boldsymbol{p})\big]$ should also satisfy the relation that $\boldsymbol{y}'^{*} = \arg\max_{y'} U(\boldsymbol{y}', \boldsymbol{p}^{\star})$. We can use this fact to replace the inner maximization in (15) as a (possibly large) set of constraints below. Let $\mathcal{P}^{\text{ext}}$ denote the space of allowed probability vectors $\boldsymbol{p}$ such that each edge probability p_e is $\underline{p}_e$ or $\overline{p}_e$. Let $\mathcal{D}(\boldsymbol{p})$ denote the best decision $\boldsymbol{y}'^{\star} = \arg\max_{\boldsymbol{y}'} U(\boldsymbol{y}', \boldsymbol{p})$. We can rewrite the MMR optimization problem as below:

$$\min_{\boldsymbol{y}} M \tag{16}$$

$$M \geq U\big(\mathcal{D}(\boldsymbol{p}), \boldsymbol{p}\big) - U\big(\boldsymbol{y}, \boldsymbol{p}\big) \;\; \forall \boldsymbol{p} \in \mathcal{P}^{\text{ext}} \tag{17}$$

As the function U is an expectation, we can use the SAA scheme similar to the one used for the problem (11) to approximate this function. Wang and Ahmed [2008] show that the solution of such an SAA approximation approaches the true optimum of the stochastic program (having expected value constraints) with probability approaching one exponentially fast with the increasing number of samples N. The SAA problem is:

$$\min_{\boldsymbol{y}} M \tag{18}$$

$$M \geq \frac{1}{N}\sum_{k=1}^{N}\sum_{v\in\mathcal{H}} x^{k}_{v,\mathcal{D}(\boldsymbol{p})} - \frac{1}{N}\sum_{k=1}^{N}\sum_{v\in\mathcal{H}} x^{k}_{v,y} \;\; \forall \boldsymbol{p} \in \mathcal{P}^{\text{ext}} \tag{19}$$

$$\boldsymbol{x}^{k}_{\mathcal{D}(\boldsymbol{p})} \in \Omega(\boldsymbol{x}^{k}_{\mathcal{D}(\boldsymbol{p})}, \mathcal{D}(\boldsymbol{p}), \boldsymbol{I}_{\boldsymbol{p}}, \boldsymbol{r}^{k}) \quad \forall k \in \{1,\ldots,N\} \tag{20}$$

$$\boldsymbol{x}^{k}_{\boldsymbol{y}} \quad \in \Omega(\boldsymbol{x}^{k}_{\boldsymbol{y}}, \boldsymbol{y}, \boldsymbol{I}_{\boldsymbol{p}}, \boldsymbol{r}^{k}) \quad \forall k \in \{1,\ldots,N\} \tag{21}$$

where $\boldsymbol{I}_{\boldsymbol{p}}$ denotes that different edge variables I_e are fixed as per the given probability p_e (if $p_e = \overline{p}_e$, then $I_e = 1$, zero otherwise). Notice that even though the problem (18) is a deterministic one, the number of constraints (19) are prohibitively large. Therefore, it is not tractable to solve this problem a single large program. Fortunately, there is a well known technique in operations research known as *constraint generation* to solve such problems (Boutilier et al. 2003). In this technique, we iteratively solve the program (18) using a mixed-integer solver by using only a subset of constraints on the variable M. Initially, we start with the pair $\langle \mathcal{D}(\boldsymbol{p}^{\text{base}}), \boldsymbol{p}^{\text{base}} \rangle$, where $\boldsymbol{p}^{\text{base}}$ is the initial estimate of the true probability. We then get a decision $\boldsymbol{y}$ from solving (18). However, this may not be the true MMR solution. Thus, we look for the currently missing constraint on M that is maximally violated by the decision $\boldsymbol{y}$. This violated constraint is found by finding the decision $\boldsymbol{y}'$ and the associated probability vector $\boldsymbol{p}_{\boldsymbol{y}'}$ that maximizes the regret w.r.t. the current MMR decision $\boldsymbol{y}$ by solving the MR problem (11). Once such a solution $\langle \boldsymbol{y}', \boldsymbol{p}_{\boldsymbol{y}'} \rangle$ is found, it is added to the MMR program, and the whole process repeats. Such an iterative constraint generation procedure is guaranteed to terminate and converge to the true MMR solution (to the accuracy afforded by the SAA scheme) as shown in (Boutilier et al. 2003). In practice, we found that ≈ 30 iterations of constraint generation were sufficient for convergence in most cases.

Accuracy of SAA Approximation The SAA approach converges to the true optimum of the given stochastic optimization problem as number of samples $N \to \infty$. Fortunately, we can get stochastic bounds on the quality of the SAA solution compared with the true optimum when using a finite number of samples N (Sheldon et al. 2010; Kleywegt, Shapiro, and Homem-de-Mello 2002). As the MR problem (11) is repeatedly solved for the constraint generation, we apply SAA analysis to it. Let $\boldsymbol{p}^{\star}$ be the probability vector corresponding to the variables $\boldsymbol{I}$ found by solving the MR problem. For this vector $\boldsymbol{p}^{\star}$, we have the following SAA approximation:

$$\max_{\boldsymbol{y}'} \Big[\sum_{v\in\mathcal{H}} \mathbb{E}\big[X_{v_T}(\boldsymbol{y}'); \boldsymbol{p}^{\star}\big] - \sum_{v\in\mathcal{H}} \mathbb{E}\big[X_{v_T}(\boldsymbol{y}); \boldsymbol{p}^{\star}\big] \Big] \approx$$

$$\max_{\boldsymbol{y}'} \frac{1}{N} \Big[\sum_{k=1}^{N}\sum_{v\in\mathcal{H}} \big[X^{k}_{v_T}(\boldsymbol{y}'); \boldsymbol{p}^{\star}\big] - \sum_{k=1}^{N}\sum_{v\in\mathcal{H}} \big[X^{k}_{v_T}(\boldsymbol{y}); \boldsymbol{p}^{\star}\big] \Big] \tag{22}$$

Empirically, we compute the stochastic upper and lower bounds for the above approximate max-regret for the decision $\boldsymbol{y}'$ using the same settings as in (Sheldon et al. 2010), which come out to be fairly close to each other.

4 Experiments

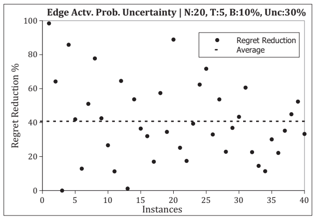
We used a publicly available conservation planning benchmark which represents a geographical region on the coast of North Carolina (Ahmadizadeh et al. 2010). This benchmark has about 80 different instances of the conservation planning problem. The endangered species is the Red-cockaded Woodpecker (RCW). The network consists of 411 territories or habitat patches grouped into 146 parcels. Each parcel can be purchased at a certain cost, establishing 146 decision variables $\boldsymbol{y}$. Each patch also has a habitat suitability score in the range $[0, 9]$.

In this work, we consider two types of parameter uncertainty—uncertainty in edge activation probabilities p_e and in suitability scores s_v for each node v. To evaluate the impact of suitability score uncertainty, we used a modified objective function $\sum_{v\in\mathcal{H}} \mathbb{E}\big[s_v X_{v_T}(\boldsymbol{y})\big]$. That is, the goal is to maximize the number of high suitability score habitats. For suitability score uncertainty, the same result holds—the suitability that maximizes the pairwise regret occurs at the either extreme points of the allowed range (proof is similar to the one we presented earlier).

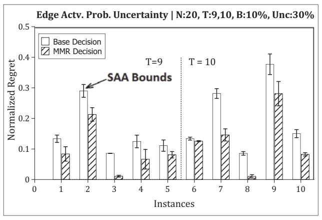
Parameter Setting We first compute p^{base}_e for each edge e using the equations provided in (Sheldon et al. 2010). Then we add uncertainty ϵ to each parameter to get the range $[p^{\text{base}}_e - \epsilon \cdot p^{\text{base}}_e, p^{\text{base}}_e + \epsilon \cdot p^{\text{base}}_e]$. The range for suitability scores is also computed in a similar manner.

Edge Activation Regret Analysis For this analysis, we used a time horizon of $T = 5, 9, 10$, number of SAA samples $N = 20$, and the budget $\mathcal{B} = 10\%$. We set uncertainty level $\epsilon = 30\%$ as it provided pronounced effects of the MMR solution while also being a realistic level of noise.

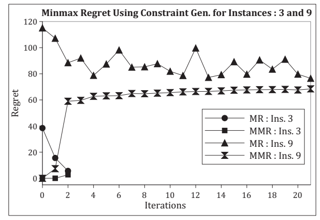
Fig 2(a) shows the percentage reduction in regret ($= (\text{MR}(\boldsymbol{y}^{\text{base}}) - \text{MMR}) / \text{MR}(\boldsymbol{y}^{\text{base}})$) by the MMR decision $\boldsymbol{y}^{\text{mmr}}$ over the best decision $\boldsymbol{y}^{\text{base}}$ for the base probability for different instances for horizon 5. The decision $\boldsymbol{y}^{\text{base}}$ does not take into account the uncertainty in edge activation parameters, thus, provides significantly higher regret than $\boldsymbol{y}^{\text{mmr}}$. We can see that, on-an-average, the MMR solution provides about 40% reduction in regret. Let Q^{base} denote the objective value of the problem (2) for the decision $\boldsymbol{y}^{\text{base}}$ and probabilities $\boldsymbol{p}^{\text{base}}$. The max-regret for decision $\boldsymbol{y}^{\text{base}}$ was reasonably



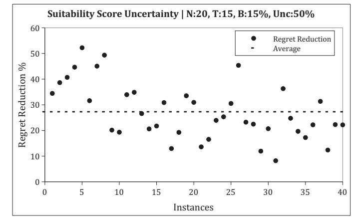
(a) Regret reduction for T=5



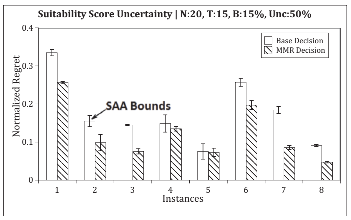
(b) Regret for T=9,10; SAA bounds



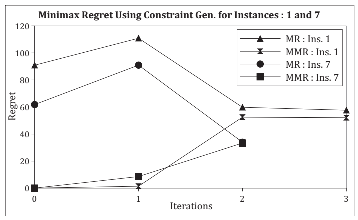
(c) Constraint generation iterations



(d) Regret reduction for suit. score



(e) Regret for T=15, SAA bounds



(f) Constraint generation iterations

(g) — (g_1) Base decision; (g_2) adversary's decision; (g_3) isolated habitats

Figure 2: Regret analysis for edge activation probability and suitability score uncertainty. Figure (g) best viewed electronically

high ($\mathrm{MR}(\boldsymbol{y}^{\text{base}})/Q^{\text{base}} \approx 5\%$) even for horizon 5. Thus, these set of results show that our approach can provide significant reduction in the max-regret over the base solution.

Fig 2(b) shows the same set of results for increased horizon $T = 9$ and $T = 10$. We show results for 10 instances where regret was quite pronounced. For each instance on the x-axis, we show the MR for the base decision $\boldsymbol{y}^{\text{base}}$ and the MMR for the decision upon convergence of the constraint generation procedure. To highlight the fact that regret was relatively high w.r.t.to the base objective Q^{base}, we normalize the regret for each instance by dividing it by Q^{base}. We can clearly see that the regret for the base decision can be quite high (e.g., more than 35% for instance '9'). These results also show that the MMR decision significantly reduces the regret validating the usefulness of our approach. The error bars for each column in 2(b) show the accuracy of the SAA approximation by showing the upper and lower bounds for respective decisions. We observe that these bounds are fairly close confirming the accuracy of the SAA approximation.

Fig 2(c) shows the iterations of the constraint generation procedure for two instances ('3' and '9'). The 'MR' curve shows the max-regret (or upper bound on 'MMR'). As the constraint generation procedure proceeds, we can see that 'MR' decreases and 'MMR' increases until they meet at the convergence point, which is our solution. On an average, even for time horizon 10, the constraint generation procedure converged in about 30 iterations.

We highlight that it was challenging to scale our MMR approach for much more than horizon 10. The reason is that as the horizon increases, so do the number of binary $\boldsymbol{I}$ variables (see constraints (8)–(10)). This increases the complexity of solving the MIP for MR computation. In our future work, we would explore the usage of combinatorial and dynamic programming based approaches (Wu, Sheldon, and Zilberstein 2014; 2015), and approaches such as Lagrangian relaxation that can make mixed-integer programming scalable (Kumar, Wu, and Zilberstein 2012).

Fig 2(d)–(f) show the same set of results for the uncertainty in the suitability score of different nodes. For this case too, the average reduction in regret was around 29% as shown in fig. 2(d). Fig. 2(e) shows the normalized regret for the base decision and the MMR decision. This figure confirms that the MMR decision significantly reduces the regret. The SAA bound analysis for this figure also confirms the accuracy of SAA approximation.

Fig. 2(g_1)–(g_3) show the operational insights behind regret-based decision making for a particular instance with horizon $T = 10$. These figures show the relevant parts of the underlying network with each node being a habitat patch. Fig 2(g_1) shows the base decision $\boldsymbol{y}^{\text{base}}$ which is the best decision for initial parameter estimate $\boldsymbol{p}^{\text{base}}$. Nodes belonging to the same land parcel are shown in same color; different colors show different parcels purchased. Fig. 2(g_2) shows the *adversary's* decision $\boldsymbol{y}'$ that maximizes the regret $\mathrm{MR}(\boldsymbol{y}^{\text{base}})$. We can observe that both these decision's are quite different (in $\boldsymbol{y}'$, parcels at the bottom right corner are not purchased). The reason is that the adversary choses probability vector $\boldsymbol{p}'$ in such a way that the *connectivity* of nodes shown in black colors in fig 2(g_3) is significantly reduced if the decision taken was $\boldsymbol{y}^{\text{base}}$. We measure connectivity by doing monte-carlo sampling and computing the expected number of times a node is occupied for the pair $\langle \boldsymbol{y}^{\text{base}}, \boldsymbol{p}^{\text{base}} \rangle$ and the pair $\langle \boldsymbol{y}^{\text{base}}, \boldsymbol{p}' \rangle$. If the drop in connectivity is more than 20%, then such a node is shown using black color in 2(g_3). Intuitivey, such nodes represent regions of the network that are highly likely to be disconnected from the initial seed nodes if the decision taken was $\boldsymbol{y}^{\text{base}}$ (which assumed the point estimate $\boldsymbol{p}^{\text{base}}$). The adversary exacerbates the losses for the decision $\boldsymbol{y}^{\text{base}}$ by not purchasing the parcels corresponding to black-colored nodes in its chosen decision $\boldsymbol{y}'$. While such clean separation between the base decision and the adversary's decision was not seen in every tested instance, the general pattern was that the adversary would chose a distribution $\boldsymbol{p}'$ such that some regions lose connectivity to the initial seed nodes under the base decision $\boldsymbol{y}^{\text{base}}$. Such an analysis provides further operational insights into the dynamics of regret-based decision making that can be used by policy makers to fine-tune their decisions.

5 Conclusion

We addressed the key issue of robust decision making using regret minimization for a spatial conservation planning problem. Our work addresses the realistic setting when the knowledge of network parameters is uncertain. We provided new theoretical results regarding the structure of the minimax regret which formed the basis for a scalable constraint generation based solution approach. Empirically, we showed that our minimax regret decision provided significantly more robust solutions than the previous approach that assumes point estimates of parameters. We also provided insights suggesting that ignoring network parameter uncertainty can lead to poor quality decisions risking isolation of habitats.

Acknowledgments

Support for this work was provided in part by the research center at the School of Information Systems at the Singapore Management University, and the National Science Foundation under Grant No. 1125228.

References

Adiga, A.; Kuhlman, C. J.; Mortveit, H. S.; and Vullikanti, A. 2014. Sensitivity of diffusion dynamics to network uncertainty. *Journal of Artificial Intelligence Research* 51:207–226.

Ahmadizadeh, K.; Dilkina, B.; Gomes, C. P.; and Sabharwal, A. 2010. An empirical study of optimization for maximizing diffusion in networks. In *International Conference on Principles and Practice of Constraint Programming*, 514–521.

Anderson, R. M., and May, R. M., eds. 2002. *Infectious diseases of humans: dynamics and control*. Oxford press.

Boutilier, C.; Patrascu, R.; Poupart, P.; and Schuurmans, D. 2003. Constraint-based optimization with the minimax decision criterion. In *Principles and Practice of Constraint Programming*, 168–182.

Domingos, P., and Richardson, M. 2001. Mining the network value of customers. In *International Conference on Knowledge Discovery and Data Mining*, 57–66.

Golovin, D.; Krause, A.; Gardner, B.; Converse, S.; and Morey, S. 2011. Dynamic resource allocation in conservation planning. In *AAAI Conference on Artificial Intelligence*, 1331–1336.

Goyal, A.; Bonchi, F.; and Lakshmanan, L. V. S. 2011. A data-based approach to social influence maximization. *VLDB Endow.* 5(1):73–84.

Hanski, I., and Ovaskainen, O. 2000. The metapopulation capacity of a fragmented landscape. *Nature* 48(2):755–758.

Hanski, I. 1994. A practical model of metapopulation dynamics. *Journal of Animal Ecology* 151–162.

Hanski, I., ed. 1999. *Metapopulation ecology*. Oxford University Press.

He, X., and Kempe, D. 2014. Stability of influence maximization. In *International Conference on Knowledge Discovery and Data Mining*, 1256–1265.

Kempe, D.; Kleinberg, J.; and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *International conference on Knowledge discovery and data mining*, 137–146.

Kleywegt, A. J.; Shapiro, A.; and Homem-de-Mello, T. 2002. The sample average approximation method for stochastic discrete optimization. *Journal on Optimization* 12:479–502.

Kumar, A.; Wu, X.; and Zilberstein, S. 2012. Lagrangian relaxation techniques for scalable spatial conservation planning. In *AAAI Conference on Artificial Intelligence*, 309–315.

Leskovec, J.; Adamic, L. A.; and Huberman, B. A. 2007. The dynamics of viral marketing. *ACM Trans. Web* 1(1).

Sheldon, D.; Dilkina, B.; Elmachtoub, A.; Finseth, R.; Sabharwal, A.; Conrad, J.; Gomes, C.; Shmoys, D.; Allen, W.; Amundsen, O.; and Vaughan, W. 2010. Maximizing the spread of cascades using network design. In *International Conference on Uncertainty in Artificial Intelligence*, 517–526.

Wang, W., and Ahmed, S. 2008. Sample average approximation of expected value constrained stochastic programs. *Operations Research Letters* 36(5):515 – 519.

Wu, X.; Kumar, A.; Sheldon, D.; and Zilberstein, S. 2013. Parameter learning for latent network diffusion. In *International Joint Conference on Artificial Intelligence*, 2923–2930.

Wu, X.; Sheldon, D.; and Zilberstein, S. 2014. Rounded dynamic programming for tree-structured stochastic network design. In *AAAI Conference on Artificial Intelligence*, 479–485.

Wu, X.; Sheldon, D.; and Zilberstein, S. 2015. Fast combinatorial algorithm for optimizing the spread of cascades. In *International Joint Conference on Artificial Intelligence*, 2655–2661.

Xue, S.; Fern, A.; and Sheldon, D. 2014. Dynamic resource allocation for optimizing population diffusion. In *International Conference on Artificial Intelligence and Statistics*, 1033–1041.

Xue, S.; Fern, A.; and Sheldon, D. 2015. Scheduling conservation designs for maximum flexibility via network cascade optimization. *Journal of Artificial Intelligence Research* 52:331–360.