Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Scheduling Conservation Designs via Network Cascade Optimization

Shan Xue and **Alan Fern** and **Daniel Sheldon**
{xue, afern, sheldon}@eecs.oregonstate.edu
School of EECS, Oregon State University, Corvallis, OR, USA, 97331

Abstract

We introduce the problem of scheduling land purchases to conserve an endangered species in a way that achieves maximum population spread but delays purchases as long as possible, so that conservation planners retain maximum flexibility and use available budgets in the most efficient way. We develop the problem formally as a stochastic optimization problem over a *network cascade* model describing the population spread, and present a solution approach that reduces the stochastic problem to a novel variant of a Steiner tree problem. We give a primal-dual algorithm for the problem that computes both a feasible solution and a bound on the quality of an optimal solution. Our experiments, using actual conservation data and a standard diffusion model, show that the approach produces near optimal results and is much more scalable than more generic off-the-shelf optimizers.

1 Introduction

Reserve site selection is a key problem in conservation planning in which planners select land to be designated as nature reserves, either to achieve general conservation goals such as preserving biodiversity, or to achieve specific goals such as supporting the recovery of an endangered species. Many different algorithms have been proposed to select reserve sites by formulating a numerical measure of reserve quality (together with the possible addition of constraints the reserve must satisfy) and then solving for the optimal set of sites under the proposed model (e.g., see the review article by Williams, Revelle, and Levin (2005)).

One crucial aspect of reserves is spatial configuration: characteristic such as size, shape, and connectivity all have important effects on the health of future populations. Although the earliest reserve site selection algorithms largely ignored spatial considerations, many newer models incorporate spatial objectives or constraints directly into the optimization problems. Williams, Revelle, and Levin (2005) argue that a primary reason for the importance of spatial attributes is the fact that they capture properties of the landscape that are favorable for the underlying population dynamics, and that an important, but computationally difficult, research direction is to directly optimize with respect to a model for the population dynamics instead of using spatial attributes as a proxy. This paper defines an important and useful subproblem of conservation planning *with respect to a specific widely adopted model of population dynamics*, and develops an efficient algorithm for that problem that can be used by practitioners.

Recently, Sheldon et al. (2010) studied a restricted, but still challenging, version of this problem, which we will refer to as *upfront conservation design*. In this problem, the planner is given a budget and a stochastic *metapopulation model* (Hanski and Ovaskainen 2000) that describes how the species will spread throughout a landscape of available habitat. The objective is to select a set of parcels to immediately purchase and conserve, subject to the budget constraint, that will maximize the spread of the population within a specified time horizon. A key simplification of this approach is that purchases are made upfront, which limits its utility in several ways. First, conservation budgets generally arrive in increments over time, so it is unrealistic to purchase a large set of parcels in advance. Moreover, it is often *unnecessary* to purchase parcels that are spatially remote from the current population until the species has spread enough to make them relevant. Second, this assumption requires planners to commit in advance to conservation strategies that may take many years to play out, which ignores the potential advantage of observing and responding to the stochastic outcomes of the population spreading process as it unfolds (e.g. by diverting money from failed subpopulations to purchase more parcels surrounding thriving ones).

An ideal approach would be *fully adaptive*: at regular decision epochs the planner would make purchase decisions based on the most recent population and budgetary information. Unfortunately, no currently-available adaptive planning tools can scale to realistic conservation scenarios. For example, while the problem can be encoded as a Markov Decision Process (MDP), the resulting state and action spaces would be far too big for state-of-the-art solvers. For example, recent advances in solving large spatio-temporal MDPs (Crowley and Poole 2011) require significant restrictions to the solution space, which are not acceptable in our application. Another approach would be to formulate the adaptive planning problem as a multi-stage stochastic integer program. However, the size of such a problem formulation scales exponentially with the number of stages, and the running time is already very costly for a single stage (Sheldon et al. 2010), or for a two-stage problem in a simpler setting that is not fully adaptive (Ahmadizadeh et al. 2010).

Recently, Golovin et al. (2011) proved that a simple greedy planning strategy provides near-optimal solutions in an adaptive conservation setting similar to ours. However, in order to provide approximation guarantees, the authors restrict the population dynamics so that no spread occurs between distinct land parcels. While this may be a reasonable assumption for slow-moving species, it ignores critical aspects of the population dynamics of highly-mobile animals such as birds, including the Red-cockaded Woodpecker on which our experiements are based.

In this paper, we consider the problem of *conservation design scheduling*, which is an important middle-ground between the upfront and fully adaptive approaches. Given a conservation design (i.e. a set of parcels to purchase) the problem is to schedule the purchase time of each parcel in a way that (1) achieves nearly the best possible population spread over the time horizon of any schedule (including the schedule that purchases everything upfront), and (2) maximizes flexibility by delaying purchases as long as possible. A solution to this problem yields a useful tool to conservation planners, who can first develop conservation designs that capture their own complex decision-making objectives, and then schedule the purchases to obtain the most efficient and cost-effective implementation of that design.

Importantly, our scheduling algorithm can also be used as a component of an adaptive planner. A common and successful approach for many adaptive planning problems is *re-planning*, where at each decision epoch a non-adaptive plan is computed from the current state and its first actions are executed. Our work enables a replanning approach that computes an upfront design using existing work (e.g. (Sheldon et al. 2010)), and then schedules the purchases so that one can execute only the first actions while guaranteeing no loss in population spread. Note that Ahmadizadeh et al. (2010) also explore re-planning using a two-stage non-adaptive problem formulation, and find that it can indeed offer advantages over upfront planning. Unlike that work, we separate the decision of which parcels to buy from the decision of when to buy them, so that we may develop efficient special-purpose algorithms for the latter problem that scale much more easily to bigger problems and more stages.

In addition to introducing and formalizing the problem of conservation design scheduling (Section 2), the main contribution of our paper is to develop a principled algorithm for solving it (Section 4). We formulate the problem in terms of network cascade optimization and show how it is equivalent to solving a novel variant of the Steiner tree problem with weights on *sets* of edges. We develop a primal-dual algorithm for the problem, which computes a feasible solution along with a bound on the quality of the optimal solution. Our experiments (Section 5), which use real data from the Red-cockaded Woodpecker problem, show that the approach produces near optimal solutions and is much more scalable than standard optimization tools.

2 Problem Statement

Basic Setup. The study area is divided into *land parcels* that are the smallest units available for purchase. They contain distinct *habitat patches* which are the atomic units in the population dynamics model and can either be occupied or unoccupied by the species of interest. Each parcel p has a cost $c(p)$, which denotes the cost of purchasing the land and restoring or conserving all of its habitat patches so they are suitable for the species to occupy. A *conservation design* is a set of parcels that are intended to be purchased and conserved. Given a conservation design D, a *purchase schedule* π for D is a mapping from parcels in D to purchase times in $\{0, 1, \ldots, H\}$, where H is the time horizon of interest. Although the species dynamics have a yearly time step, the allowed purchase times (i.e. decision epochs) may be less frequent depending on the specific problem. An *upfront schedule* is one that assigns all parcels to purchase time $t = 0$.

Population Model. We use the same stochastic dynamics model as (Sheldon et al. 2010), which is an instance of a *metapopulation model* from the ecology literature (Hanski and Ovaskainen 2000). A patch i has two possible states in each time step, either unoccupied or occupied, and only conserved patches may be occupied. The population dynamics consists of two types of stochastic events. Colonization events occur when a population from patch i colonizes an unoccupied patch j, which happens with probability p_{ij}. Extinction events occur when a patch that is occupied at time t becomes unoccupied at time $t+1$, which happens with probability $1 - p_{ii}$. All events are independent. The single-step colonization probability p_{ij} typically decays with distance and encodes spatio-temporal dynamics in which populations slowly spread from a source population when new habitat is made available. Thus, in long-term planning for population spread, it is often unnecessary to purchase parcels that are distant from a source population at time $t = 0$. By delaying such purchases until they become relevant to the design, a conservation organization can use limited funds much more flexibly.

Stochastic Problem. Our problem statement will rely on two important concepts: the *reward* of a schedule and the *flexibility* of a schedule. The reward $R(\pi)$ is a random variable that encodes the amount of population spread as the number of occupied patches at time H. It is easy to show in our model that the upfront schedule always achieves at least as much reward as any other schedule and thus maximizes expected reward, so let $R^* = E[R(\pi_{\text{upfront}})]$. Our goal is to find the most flexible schedule (in a sense to be defined later) among the set of schedules $\Pi(\epsilon) = \{\pi : E[R(\pi)] \geq R^* - \epsilon\}$ with near-optimal expected reward.

Scenario Graphs. As in prior work (Sheldon et al. 2010), our algorithm does not work directly with the stochastic problem definition; instead, it uses the Sample Average Approximation (SAA) technique to optimize over a finite set of samples. The main idea is to simulate a set of *cascade scenarios* from the probabilistic spread model, each of which describes one outcome of the population spread process, and to approximate expected reward as the average over the scenarios. The scenarios are combined into a single *scenario graph*, which is illustrated in Figure 1 and explained in detail in the remainder of this section.

More concretely, a cascade scenario is a layered graph with a vertex $v_{i,t}$ for each patch i and each time step t. For each pair of patches (i, j) and time step t, a coin is flipped with probability p_{ij} to determine if the directed edge $(v_{i,t}, v_{j,t+1})$ is present. If this edge is present and patch i is occupied at time t (through previous colonizations or non-extinctions), then patch j will be colonized and become occupied at time $t+1$, as long as it is conserved. No other edges are present. A cascade scenario graph encodes occupancy as reachability: assuming (for now) that all patches are conserved, then patch j is occupied at time t exactly when $v_{j,t}$ is reachable from a vertex $v_{i,0}$ corresponding to an initially occupied patch i.

To approximate the probabilistic spread model, we sample a set of N i.i.d. cascade scenarios $\{C_1, \ldots, C_N\}$, where we will denote the vertices in C_k by $\{v^k_{i,t}\}$. These scenarios are combined into a single *scenario graph*, which has an additional root vertex r with directed edges $(r, v^k_{i,0})$ to each vertex representing an initially occupied patch i.

A schedule π is said to purchase node $v^k_{i,t}$ and all of its incoming edges if patch i is purchased no later than time t (i.e. $\pi(p) \le t$ where i belongs to parcel p). Vertex $v^k_{i,t}$ becomes occupied under π if there is a path through *purchased* edges from r to $v^k_{i,t}$. We let the variable $X^k_\pi(i,t)$ equal 1 if $v^k_{i,t}$ is occupied under π and 0 otherwise.

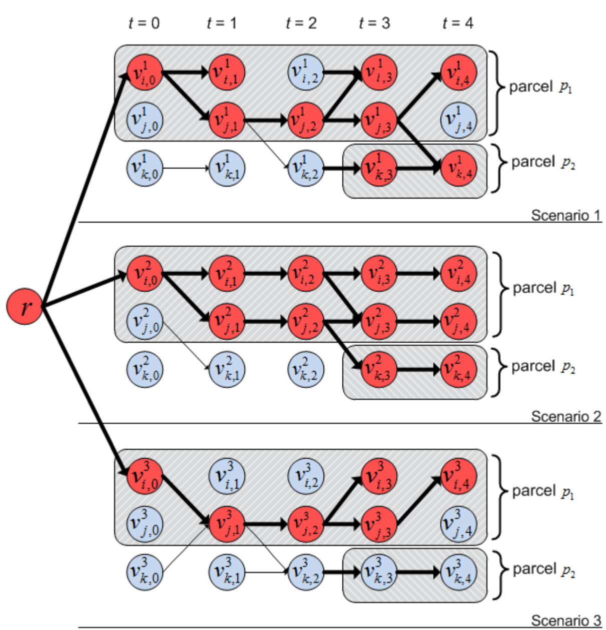

Figure 1: Example scenario graph ($N = 3$) for problem with parcels $p_1 = \{i, j\}$, $p_2 = \{k\}$. The schedule $(\pi(p_1) = 0, \pi(p_2) = 3)$ is also illustrated, using shaded boxes to indicate purchased nodes, and heavy line weights to indicate purchased edges. Vertices representing occupied patches under this schedule are colored red.

The average reward of a schedule π relative to the scenarios $\{C_1, \ldots, C_N\}$ is denoted by

$$\hat{R}(\pi) = \frac{1}{N} \sum_{k=1}^{N} \sum_{i} X^k_\pi(i, H)$$

which is just the average across scenarios of the number of occupied patches at time H. Importantly, as $N \to \infty$ we have that $\hat{R}(\pi)$ converges to $E[R(\pi)]$ for any fixed π, and the set $\{\pi : \hat{R}(\pi) \ge \hat{R}(\pi_{\text{upfront}}) - \epsilon\}$ converges to $\Pi(\epsilon)$. In our experiments, we show that optimizing over the set of schedules π such that $\hat{R}(\pi)$ is *exactly* equal to the upfront reward $\hat{R}^* = \hat{R}(\pi_{\text{upfront}})$ on $\{C_1, \ldots, C_N\}$ achieves very good results, and henceforth we restrict to that case.

Optimization Problem. Our goal is now to find a schedule that achieves reward $\hat{R}^*$ and has maximum "flexibility". We know that the upfront schedule achieves $\hat{R}^*$, however, it requires commitment to all expenditures at the first time step and is thus the least flexible. Indeed, we formalize flexibility in terms of expenditures over time.

Given any schedule π we can define its corresponding *cost curve* C_π to be a function from purchase times to accumulated cost, so that $C_\pi(t)$ is equal to the total cost of parcels purchased under π from time 0 up to and including time t. This curve is non-decreasing and provides a view of a schedule's spending profile over the time horizon. We define a strict partial order relation $<_c$ on schedules such that $\pi_1 <_c \pi_2$ if and only if (1) $C_{\pi_1}(t) \le C_{\pi_2}(t)$ for all t and (2) there is some t for which $C_{\pi_1}(t) < C_{\pi_2}(t)$. That is, the total expenditures of π_1 never exceed those of π_2 and there is at least one time where they are strictly less. In this case we can say that π_1 offers more flexibility in terms of budget management compared to π_2 and should be preferred if all else is equal. Note that some cost curves may cross and hence they are incomparable under $<_c$. If all parcels have positive costs, then the upfront schedule is the unique maximum element under $<_c$, and the schedule that defers all purchases until time H is the unique minimum. However, when we restrict to schedules with reward $\hat{R}^*$, the latter schedule will be excluded and there may no longer be a unique minimum.

We can now specify the problem of *conservation design scheduling*, which is to find a schedule π^* from the set of all possible schedules Π such that:

$$\pi^* = \arg\min_c \{\pi \in \Pi \mid \hat{R}(\pi) = \hat{R}^*\} \qquad (1)$$

where $\min_c$ returns a minimal element under the ordering $<_c$. That is, out of all schedules that achieve reward $\hat{R}^*$ we want to return one that is minimal in terms of its cost curve (i.e. it has maximal flexibility). Note that the solution may not be unique, since $<_c$ is not a total order.

3 Set Weighted Steiner Graph Formulation

One challenge in finding a solution to Equation 1 is that we must minimize over a rather non-standard ordering. To deal with this issue, below we show that minimizing a more traditional surrogate cost function yields solutions that are minimal in the cost curve partial order. We then show how to formulate the resulting problem as a novel variant of the Steiner tree problem.

Surrogate Cost Function. Let f be a function from times in $\{0, \ldots, H\}$ to real numbers. The surrogate cost function is

$$\text{cost}_f(\pi) = \sum_p c(p) \cdot f(\pi(p))$$

where the sum is over all parcels p in the conservation design. This cost function is simply a weighted sum of the parcel costs, where the weight is determined by f based on the parcel's purchase time. Intuitively, if f were a strictly decreasing function of purchase times, then minimizing with respect to cost_f would favor schedules that delay purchasing. It turns out that for any decreasing f there is a strong relationship to the $<_c$ ordering based on cost curves.

Proposition 1. *For any schedules π_1 and π_2, and any strictly decreasing function f, if $\pi_1 <_c \pi_2$, then $cost_f(\pi_1) < cost_f(\pi_2)$.*

The proof is in the extended version of the paper. This result implies that for any decreasing f if we minimize cost_f over a set of schedules, then the resulting schedule will also be minimal under $<_c$ as called for by Equation 1. Note that by varying the choice of f it may be possible to generate different solutions to Equation 1 corresponding to different minima in $<_c$. In our experiments, we use a simple discounted f given by $f(t) = \beta^t$ for a discount factor $\beta \in (0, 1)$.

Set Weighted Directed Steiner Graph. Combining the above ideas for a set of N cascade scenarios and a decreasing function f we arrive at our final optimization problem:

$$\pi^* = \arg\min_{\pi \in \Pi} \text{cost}_f(\pi) \text{ s.t. } \hat{R}(\pi) = \hat{R}^*. \tag{2}$$

We can view this problem as a type of Steiner graph problem on the scenario graph. In particular, we say that any vertex at time $t = H$ is a *terminal vertex* if it is reachable from the root r, which is the set of nodes for which $X^k_{\pi_{\text{upfront}}}(i, H) = 1$ and hence contribute to the upfront reward $\hat{R}^*$. The only way for π to satisfy the constraint $\hat{R}(\pi) = \hat{R}^*$ is to purchase a set of edges in the scenario graph that connect all of those target nodes to r. Thus, the constraint in Equation 2 corresponds to purchasing edges such that r has a path to each terminal, as in the Steiner tree problem.

However, in the standard Steiner tree problem, each edge is associated with a distinct weight and can be purchased individually, with the goal of connecting all terminals using a set of edges of minimum total weight (which always comprise a tree). Our situation is more complicated because we purchase parcels, which correspond to subsets of edges in the scenario graph. In particular, purchasing a parcel p at time t, which incurs cost $c(p)f(t)$, corresponds to purchasing an edge set $E_{p,t}$ which contains all the edges arriving at any vertex $v^k_{i,t'}$ with $i \in p$ and $t' \geq t$.

From the above we see that our problem is an instance of a problem that we will call the *Set Weighted Directed Steiner Graph (SW-DSG)* problem, a novel variant of the Steiner tree problem. For the remainder of the paper we will discuss this problem in its general form to simplify notation. The input for SW-DSG is a directed graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$ with a single root vertex r, a set of terminal vertices $T \subseteq \mathbb{V}$, a set of M edge sets $\mathcal{E} = \{E_1, \ldots, E_M\}$ where each $E_s \subseteq \mathbb{E}$, and a non-negative cost c_s for each E_s. (The conservation problem has edge sets $\mathcal{E} = \{E_{p,t}\}$ with $E_{p,t} = \{(u, v^k_{i,t'}) : i \in p, t' \geq t\}$ and $c_{p,t} = c(p)f(t)$). A subset of $\mathcal{E}$ forms a Steiner graph if the union of the edges connect r to all vertices in T. The desired output is a minimum cost subset of $\mathcal{E}$ that forms a Steiner graph. Note that the optimal Steiner graph need not be a tree in SW-DSG.

4 Primal-Dual Algorithm

The SW-DSG problem is a generalization of the traditional directed Steiner tree (DST) problem, which is known to be NP-complete. Further, under standard complexity assumptions, DST is hard to approximate by a factor better than $\log(|T|)$ (Charikar et al. 1998). Note that these results hold for even acyclic directed graphs, which is the case for the scenario graphs from our conservation problem. There are a number of effective heuristic algorithms for DST (Drummond and Santos 2009), with many of the most successful relying on shortest path computations as a subroutine. While shortest paths can be computed in edge weighted graphs efficiently, this turns out to not be the case for our set weighted problem. In particular, note that the shortest path problem is a special case of DST (or SW-DSG) where there is a single terminal vertex. This problem turns out to be NP-Hard for SW-DSG, even when restricted to acyclic graphs.

Theorem 1. *The SW-DSG problem is NP-hard even when restricted to a single terminal vertex and acyclic graphs.*

The proof is in the full paper and is by reduction from the set cover problem. Thus, it is difficult to extend prior shortest-path-based heuristics to our problem.

Another approach is to encode an SW-DSG problem as a mixed integer program (MIP), which is straightforward, and to then use a MIP optimizer. However, as our experiments show, this approach does not scale well. One could also consider a rounding procedure for the MIP's LP-relaxation. However, our experiments show that the scalability of LP solvers is also poor. Instead, we exploit the MIP encoding in another way, by following the primal-dual schema (Vazirani 2001) to derive a scalable algorithm that performs near optimally in our experiments. Our work can be considered as a non-trivial generalization of previous work (Wong 1984), where the primal-dual schema was applied to DST.

To apply the primal-dual schema we start by giving a primal MIP for the SW-DSG problem along with the dual of its LP-relaxation in Figure 2. The primal MIP uses a standard network-flow encoding of the Steiner graph constraint. The only integer variables are the $y(E_s)$ variables, one for each edge set in $\mathcal{E}$, which is a binary indicator of whether E_s was purchased or not. Thus, the objective corresponds to the total cost of purchased edge sets, as desired. The flow variable $x^k_{i,j}$ encodes the flow on edge (i, j) destined for terminal k. The flow balance constraints (2) guarantee that one unit of flow is carried on a path from the root node r to k. The LP-relaxation of the primal simply replaces the integer constraints on the $y(E_s)$ with a positivity constraint. The dual of the relaxed primal includes dual variables u^k_i and $w^k_{i,j}$ corresponding to the primal flow constraints, one of which is redundant so that we can set $u^k_r = 0$ for all $k \in T$, simplifying the dual objective.

At a high level, our primal-dual algorithm is iterative where each iteration increases the value of the dual objective and purchases a single edge set E_s, which corresponds to setting the primal variable $y(E_s) = 1$. The iteration stops

$$\textbf{(Primal)} \quad \min \sum_{s=1}^{M} y(E_s) \times c_s, \quad \text{subject to:} \tag{1}$$

$$\sum_{h \in \mathbb{V}} x_{i,h}^k - \sum_{j \in \mathbb{V}} x_{j,i}^k = \begin{cases} 1, & \text{if } i = r \\ -1, & \text{if } i = k \\ 0, & \text{if } i \neq r, k \end{cases}, k \in T, i \in \mathbb{V} \tag{2}$$

$$x_{i,j}^k \leq \sum_{s:(i,j) \in E_s} y(E_s), \; k \in T, (i,j) \in \mathbb{E} \tag{3}$$

$$x_{i,j}^k \geq 0, \; (i,j) \in \mathbb{E}, k \in T \tag{4}$$

$$y(E_s) \in \{0, 1\} \tag{5}$$

$$\textbf{(Dual)} \quad \max \sum_{k \in T} u_k^k - u_r^k, \quad \text{subject to:} \tag{6}$$

$$\sum_{k,(i,j) \in E_s} w_{i,j}^k \leq c_s, s \in \{1, \dots, M\} \tag{7}$$

$$u_j^k - u_i^k - w_{i,j}^k \leq 0, k \in T, (i,j) \in \mathbb{E} \tag{8}$$

$$w_{i,j}^k \geq 0 \tag{9}$$

Figure 2: MIP for the SW-DSG problem and the corresponding dual LP of the MIP's LP-relaxation.

when the purchased edges form a Steiner graph (i.e. the primal becomes feasible). The value of the dual objective at the end of the iteration serves as a lower-bound on the optimal primal objective, which provides a worst-case indication of how far from optimal the returned solution is.

Algorithm 1 gives pseudo-code for the algorithm. The main data structure is an auxiliary graph $\mathbb{G}' = (\mathbb{V}, \mathbb{A})$ with the same vertices as the input graph $\mathbb{G}$. The auxiliary graph edge set $\mathbb{A}$ is initially empty and then each iteration adds the newly purchased edges $E_s \in \mathcal{E}$. The algorithm terminates when the edges in $\mathbb{A}$ form a Steiner graph. Given a current auxiliary graph we denote all vertices having paths to terminal node k via edges in $\mathbb{A}$ by $C(k)$, which is considered to include k. Also, we define the *cut set* of a terminal node k, denoted by Cut(k) to be the set of all edges (i, j) such that $j \in C(k)$ and $i \notin C(k)$. Intuitively, if k is not already reachable from the root, we know that at least one edge in Cut(k) must be added to $\mathbb{A}$ to arrive at a Steiner graph.

After initializing all dual variables to zeros, each iteration randomly selects a terminal vertex k that is not connected to r in the auxiliary graph. At an intuitive level, the algorithm will then select an edge set E_s that contains a cutset edge of k according to a heuristic $\Delta(s, k)$ that is derived by applying the primal-dual schema. More concretely, the aim of each iteration is to raise the dual objective value by increasing the value of u_k^k while maintaining feasibility. Increasing u_k^k by itself will violate constraints of type (8) in the dual and lines 5 through 8 maintain feasibility by selecting an edge set E_{s^*} among those that intersect the cut set of k and then raising all variables corresponding to vertices in $C(k)$ and edges in Cut(k) by a value $\Delta(s^*, k)$ (including u_k^k). This is done in a way that causes the dual constraint of type (7) corresponding to edge set E_{s^*} to become tight. Since this constraint corresponds to primal variable $y(E_{s^*})$, the algorithm effectively sets $y(E_{s^*}) = 1$, indicating a purchase, by adding the

Algorithm 1 Primal-Dual Algorithm for SW-DSG.

1: {**Inputs:** Graph $\mathbb{G} = (\mathbb{V}, \mathbb{E})$, edge sets $\mathcal{E} = \{E_1, \dots, E_M\}$, costs $\{c_1, \cdots, c_M\}$, terminals $T \subseteq \mathbb{V}$}
2: Initialize:
$u_i^k = 0$, for each $k \in T, i \in \mathbb{V}$; $w_{i,j}^k = 0$, for each $(i,j) \in \mathbb{E}, k \in T$
$\mathbb{G}' = (\mathbb{V}, \mathbb{A})$ with $\mathbb{A} = \emptyset$
lowerBound $= 0$, solution $= \emptyset$
3: **while** $\mathbb{G}'$ is not a Steiner graph **do**
4: Let k be random vertex in T not connected to r in $\mathbb{G}'$
5: $S = \{s \mid E_s \cap \text{Cut}(k) \neq \emptyset, s \notin \text{solution}\}$
6: $s^* = \arg\min_{s \in S} \Delta(s, k)$
where $\Delta(s, k) = \left(c_s - \sum_{k' \in T, (m,n) \in E_s} w_{m,n}^{k'}\right) / |E_s \cap \text{Cut}(k)|$
7: $u_j^k = u_j^k + \Delta(s^*, k)$, for each $j \in C(k)$
8: $w_{i,j}^k = w_{i,j}^k + \Delta(s^*, k)$, for each $(i,j) \in \text{Cut}(k)$
9: $\mathbb{A} = \mathbb{A} \cup E_{s^*}$
10: lowerBound $=$ lowerBound $+ \Delta(s^*, k)$
11: solution $=$ solution $\cup \{s^*\}$
12: **end while**
13: Pruning: solution $=$ solution $- \{s \mid \exists s' \in solution, E_s \subset E_{s'}\}$

edges in E_{s^*} to $\mathbb{A}$. The dual objective value at termination is the sum across iterations of $\Delta(s^*, k)$ and is returned as the lower-bound. The key property is that each iteration maintains feasibility of the dual, which guarantees that the dual corresponds to a true lower bound on the optimal value.

Theorem 2. *Each iteration of the primal-dual algorithm produces a feasible dual solution with increased objective.*

Proof. (sketch) As the base case, the initialization assigns all dual variables to be zero, which is a feasible solution. Now suppose that iteration $q - 1$ starts with a feasible solution $\{u_i^l, w_{i,j}^l\}$, which satisfies the dual constraints of type (7) and (8). Now if the algorithm terminates, we get a feasible solution. Otherwise let k be the terminal vertex selected. For all variables $\{u_i^l, w_{i,j}^l\}$ with $l \neq k$ the values are not changed, so (8) is satisfied. For the remaining variables with $l = k$, there are three cases. *Case 1:* For $j \notin C(k)$, the variables u_j^k and $w_{i,j}^k$ are unchanged, so they cannot contribute to a violation of (7) or (8). *Case 2:* For any edge (i, j) with both $j, i \in C(k)$, we increase both u_j^k and u_i^k by Δ^* and continue to satisfy the corresponding constraint of (8). *Case 3:* For any cut set edge $(i, j) \in \text{Cut}(k)$, we increase u_j^k and $w_{i,j}^k$ by Δ^* so that (8) remain satisfied. Since the $w_{i,j}^k$ for edges in the cut set are increased, we must ensure that constraints of type (7) do not become violated. The choice of Δ^* made by the algorithm can be verified to never violate any of those constraints and makes at least one of them tight. □

After the main portion of the algorithm terminates, a pruning step is conducted to remove any edge set that is a subset of some other edge set in the solution, which decreases the total cost while maintaining feasibility. In particular, it ensures that each parcel is purchased no more than once in the final solution.

5 Experiments

Our empirical evaluation uses the same dataset as in prior work by Sheldon et al. (2010) on computing upfront conservation designs. The data is derived from a conservation

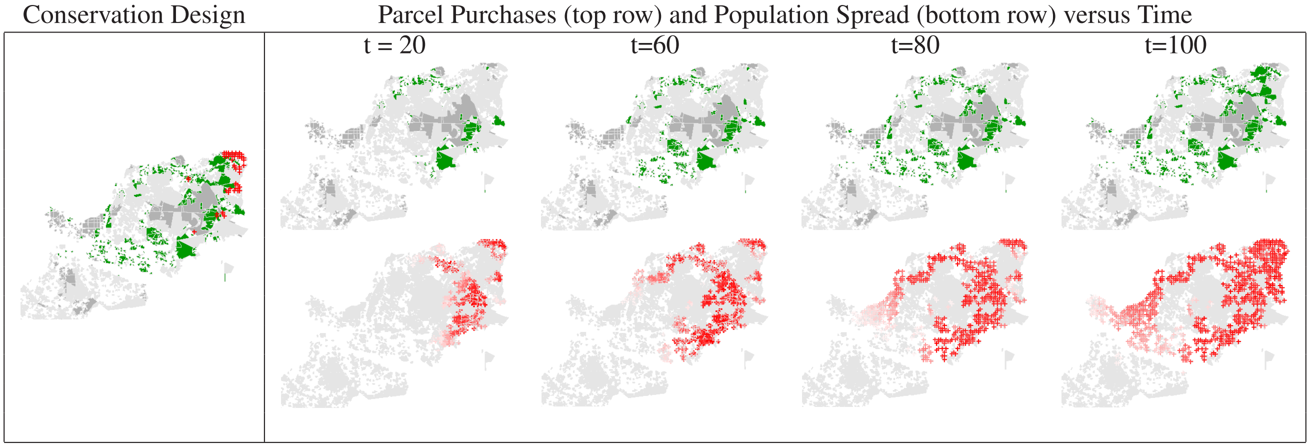

Figure 3: (Left) Original conservation design used for scheduling shown as green shaded parcels. Free parcels are also shaded in dark grey and red '+' indicates initially occupied patches. (Right) The top row shows the parcels purchased (shaded green) by our schedule over a horizon of 100 years. The bottom row shows the population spread over the same horizon for the schedule, where lighter red shading of a patch indicates a smaller probability of being occupied (as measured by 20 simulations).

problem involving the Red-cockaded Woodpecker (RCW) in a large land region of the southeastern United States that was of interest to The Conservation Fund. The region was divided into 443 non-overlapping parcels (each with area at least 125 acres) and 2500 patches serving as potential habitat sites. Parcel costs were based on land prices and some land parcels were already conserved and thus had cost zero. We use the same population spread model as Sheldon et al., which was based on individual-based models of the RCW. Since our approach requires a conservation design as input, we use the design computed by Sheldon et al. using a total budget constraint of \$320M. The map of the area is shown in the left cell of Figure 3, with parcels making up the design shaded green and free parcels shaded grey; red '+' marks indicate initially occupied patches. Our method also requires specifying a strictly decreasing function for defining the surrogate cost function, for which we use $f(t) = \beta^t$ for $\beta = 0.96$ (the results are not very sensitive to β).

Comparing to Optimal Solutions. Here we compare the solutions of our primal-dual algorithm to optimal solutions found using the CPLEX solver applied to a MIP encoding of the SW-DSG problem. Since the optimal solver can't scale to large versions of the problem, we consider problems involving cascade networks with 2 scenarios and horizons ranging from 15 to 40 years. We also use CPLEX to compute solutions to the LP-relaxation of the MIP. The objective value returned for the LP provides an alternative approach to computing a lower-bound on the optimal solution and thus is interesting to compare to our lower bound in terms of tightness and runtime. Since our primal-dual algorithm is stochastic due to the random selection of terminal nodes, we report averages over 20 runs, noting that the standard errors are negligible.

The first two data columns of Table 1 show the (surrogate) cost of the solutions found by MIP and our algorithm (PD) for increasing horizons, where larger horizons correspond to larger problems. When a method fails to return a solution due to memory constraints no value is shown in the table. We see that for horizons where MIP is able to yield solutions, our algorithm produces solutions that have very similar costs (here lower cost is better).

The next two columns of Table 1 provide results for the lower-bound computed by the LP and by the PD algorithm. We see that the lower-bound produced by the LP is significantly tighter than the bound produced by our algorithm. However, the LP is unable to be solved for the largest problem, while our approach still yields a lower bound. Overall, while our lower bound is not as good as the LP (when it can be computed) it is still quite useful as it is generally within a factor of two of the optimal solution.

The final three columns of Table 1 presents the time used by the approaches for each problem, where blank cells indicate that the method ran out of memory. Our algorithm is significantly faster than the MIP approach, which fails for the two largest problems, and is comparable with the LP approach, which only provides a lower bound and fails for the largest problem. This later result indicates that a solution based on LP-rounding would face difficulty, since even solving the LP for these large problems (40 time steps with 2500 patches each) is computationally demanding. An advantage of the primal-dual algorithm is that it avoids encoding the LP and rather works directly with a graph.

Quality of Conservation Schedules. We now evaluate our algorithm on problems of more realistic sizes. Here, we consider problems based on 10 cascades (following Sheldon et al. (2010)) and horizons ranging from 20 to 100 years, which are well beyond the range approachable for the MIP and LP. The solution times for our algorithm ranged from 15 seconds for $H = 20$ to 29 minutes for $H = 100$.

First, we evaluate the average accumulated reward of the schedule returned by our method for each horizon. This is done by running 20 simulations of the stochastic population spread model. Each simulation provides a reward value (number of occupied patches at the horizon) and we averaged the results. This was done for the schedules produced by our primal-dual algorithm and for the upfront schedule.

Table 1: Comparison of PrimalDual (PD) with MIP and LP.

	Cost (M$)		Lower Bound		Run Time (s)		
	MIP	PD	LP	PD	MIP	LP	PD
H = 15	126.8	126.2[1]	122.2	84.9	5.5	6.1	0.9
H = 20	123.6	125.7	117.7	71.9	8.2	7.6	2.5
H = 25	117.6	121.4	104.7	61.5	28	10	9.0
H = 30	130.4	134.0	117.3	56.9	5126	15	11
H = 35		131.3	109.9	64.1		18	25
H = 40		127.5		59.7			45

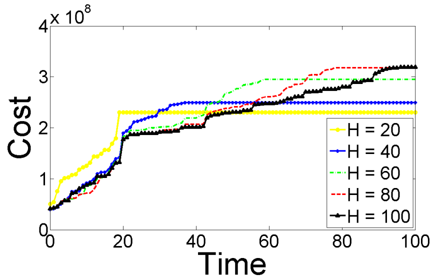

Figure 4: Cost curves for horizons 20 to 100.

Recall that the intention is to nearly match the reward of the upfront schedule. *The average reward of the upfront schedule ranged from* 332 *for* $H = 20$ *to* 615 *at* $H = 100$ *and for all time horizons the primal-dual solution attained average reward at least* 95.3% *of optimal, with negligible error bars about the averages.* The small gap indicates that for 10 scenarios the SAA approximation is quite good—the gap could be further reduced by increasing the number of scenarios.

Of course, we must also consider the cost curves corresponding to the schedules, since that is what affords the flexibility criterion of our problem. Figure 4 presents the cost curves for our schedules. Note that the cost curve for a schedule produced for horizon H will only increase until time H and then remain flat, reflecting that no purchases are made after that time. We see that for all horizons the cost curves show a fairly gradual increase in cost expenditures over time, indicating that the schedules are indeed providing a significant amount of flexibility regarding purchase times, particularly compared to an upfront schedule. In experiments not shown, we found that the cost curves vary by a small amount for different values of β, but the same general trend is present. Interestingly, in all curves there is a sudden jump in cost at around 20 years. To understand this in Figure 3 we show both the parcel purchases made by our schedule and the population spread on the map over the 100 year horizon. We see that at t=20 the sharp increase in cost is due to the purchase of some relatively expensive and vast parcels in the southern part of the design. Looking at the population spread dynamics it is apparent that those parcels are a critical gateway for ensuring reliable spread to the southwestern part of the design in later years. Delaying the purchase any longer significantly increases the probability that such spread does not occur, which our approach discovers.

[1]Here PD cost is less than the "optimal" MIP cost. After investigating, it appears to be due to finite precision issues of CPLEX.

6 Future Work

We plan to consider a variant of our current problem where we do not require that all terminals be connected, but rather charge a penalty for unconnected terminals. This might allow for even more flexibility in a schedule with little loss in reward. To further improve our algorithm, we will consider an approach based on "simultaneous raising" of dual variables where we consider all terminals at each iteration, which has shown improved results in other contexts. Finally, we plan to consider fully adaptive design approaches, including replanning strategies that leverage our scheduling algorithm.

References

Ahmadizadeh, K.; Dilkina, C.; Gomes, C. P.; and Sabharwal, A. 2010. An empirical study of optimization for maximizing diffusion in network. In *16th International Conference on Principles and Practice of Constraint Programming*.

Charikar, M.; Chekuri, C.; Cheung, T.; Dai, A.; Guha, S.; and Li, M. 1998. Approximation algorithm for dircted steiner tree problems. In *The Ninth Annual ACM-SIAM Symposium on Discrete Algorithm*.

Crowley, M., and Poole, D. 2011. Policy gradient planning for environmental decision making with existing simulators. In *AAAI '11: Proceedings of the Twenty-Fifth Conference on Artificial Intelligence, Special Track on Computational Sustainability and AI.*

Drummond, L. M., and Santos, M. 2009. A distributed dual ascent algorithm for steiner problems in multicast routing. *Networks* 53:170–183.

Golovin, D.; Krause, A.; Gardner, B.; Converse, S. J.; and Morey, S. 2011. Dynamic resource allocation in conservation planning. In *AAAI '11: Proceedings of the Twenty-Fifth Conference on Artificial Intelligence*.

Hanski, I., and Ovaskainen, O. 2000. The metapopulation capacity of a fragmented landscape. *Nature* 404(6779):755–758.

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

Vazirani, V. V. 2001. *Approximation Algorithms*. Berlin: Springer.

Williams, J.; ReVelle, C.; and Levin, S. 2005. Spatial attributes and reserve design models: a review. *Environmental Modeling and Assessment* 10(3):163–181.

Wong, R. T. 1984. A dual ascent approach for steiner tree problems on a directed graph. *Mathematical Programming* 28:271–287.