7-2012

Lagrangian Relaxation Techniques for Scalable Spatial Conservation Planning

Akshat KUMAR
Singapore Management University, akshatkumarR@smu.edu.sg

Xiaojian WU
University of Massachusetts Amherst

Shlomo ZILBERSTEIN
University of Massachusetts Amherst

Published inProceedings of the 26th AAAI Conference on Artificial Intelligence 2012, July 22-26, Toronto, Canada, pp. 309-315.

Lagrangian Relaxation Techniques for Scalable Spatial Conservation Planning

Akshat Kumar and **Xiaojian Wu** and **Shlomo Zilberstein**
Department of Computer Science
University of Massachusetts
Amherst, MA 01003, USA
{akshat,xiaojian,shlomo}@cs.umass.edu

Abstract

We address the problem of *spatial conservation planning* in which the goal is to maximize the expected spread of cascades of an endangered species by strategically purchasing land parcels within a given budget. This problem can be solved by standard integer programming methods using the *sample average approximation* (SAA) scheme. Our main contribution lies in exploiting the separable structure present in this problem and using Lagrangian relaxation techniques to gain scalability over the flat representation. We also generalize the approach to allow the application of the SAA scheme to a range of stochastic optimization problems. Our iterative approach is highly efficient in terms of space requirements and it provides an upper bound over the optimal solution at each iteration. We apply our approach to the Red-cockaded Woodpecker conservation problem. The results show that it can find the optimal solution significantly faster—sometimes by an order-of-magnitude—than using the flat representation for a range of budget sizes.

1 Introduction

We study the problem of allocating resources to maximize the population spread of an endangered species over the course of time. This important computational sustainability problem has recently received much attention in the AI literature (Sheldon et al. 2010; Ahmadizadeh et al. 2010; Golovin et al. 2011). In this problem, the spread of a species is modeled as a diffusion-based process or a cascade over a geographical network of land patches. That is, under certain conditions, the species spreads from an active region in which it is present to a neighboring region. Such diffusion-based stochastic processes have been widely studied in different contexts such as identifying influential members of a community for viral marketing (Kempe, Kleinberg, and Tardos 2003), early outbreak detection of water supply contamination (Krause et al. 2008), and selection of blogs to read when following a story (Leskovec et al. 2007). While the underlying process governing the spread of cascades in conservation planning is similar to propagation models in other contexts (Kempe, Kleinberg, and Tardos 2003; Leskovec et al. 2007), the key difference that makes conservation planning more challenging is the absence of submodularity (Sheldon et al. 2010)—an intuitive diminishing returns property and a discrete analogue of convexity that often facilitates the development of effective greedy solutions.

In the absence of submodularity, the computationally efficient greedy approach of Nemhauser, Wolsey, and Fisher (1978) cannot be used. As an alternative, Sheldon et al. formulated the conservation planning problem as a stochastic optimization problem and used the *sample average approximation* (SAA) scheme (Kleywegt, Shapiro, and Homem-de-Mello 2002) to transform the problem into a deterministic one. The resulting problem can then be formulated as a mixed-integer program (MIP) and solved using CPLEX. The drawback of this approach, however, is that to increase the accuracy of the SAA scheme, one must increase the number of samples, which in turn increases the size of the MIP leading to a significantly longer runtime. This adversely affects the applicability of the approach to large-scale conservation planning problems.

We address this scalability challenge by identifying and exploiting the separable structure present in the MIP for the conservation planning problem. We first modify the MIP by creating duplicate variables and constraints. Then we obtain the dual of this problem by relaxing certain consistency constraints using the well established technique of Lagrangian relaxation from the OR literature (Bertsekas 1999). The resulting dual has a particularly simple structure and is composed of several *independent* subproblems—precisely one per SAA sample. These subproblems can be solved efficiently independently and in parallel. This alleviates a major bottleneck of the global MIP that considers all the samples in one single program. Furthermore, the solution of these subproblems provides a subgradient of the dual problem, which is then optimized iteratively by using the projected subgradient method (Boyd, Xiao, and Mutapcic 2003). This approach, also called *dual decomposition*, provides a bound on the optimal solution at each iteration. To underscore the generality of this approach, we show how to apply it to the SAA scheme for generic stochastic optimization problems, which are useful in other sustainability applications such as energy system management (Lubin et al. 2011). We also discuss ways to vary the relaxation technique so as to make it more efficient in different contexts.

We note that recently, Lagrangian relaxations techniques have been employed successfully in several other domains

such as inference in graphical models (Sontag, Globerson, and Jaakkola 2010), natural language processing (Rush et al. 2010) and computer vision (Komodakis, Paragios, and Tziritas 2011). They provide a robust approach for increasing scalability, while also providing good quality bounded solutions in such diverse domains. Our experimental results underscore these advantages. Empirically, we test the dual decomposition approach on a publicly available conservation planning benchmark for the Red-cockaded Woodpecker (RCW) (Ahmadizadeh et al. 2010)—a federally listed endangered species. Our approach quickly provides the optimal primal solution for a range of problems, significantly faster than CPLEX. The dual solution quality provides a certificate of optimality, confirming that the solution is within 90% of the optimal for most problem instances.

2 The Conservation Planning Problem

We briefly describe the conservation planning problem that is more fully detailed in (Sheldon et al. 2010). In this problem, the goal is to maximize the spread of the target species through a network of *habitat patches* over some period of time. The species can survive only within land patches that are conserved. The management actions consist of a land acquisition strategy within some fixed budget. The spread of the species over time or the *cascade* is modeled as a stochastic process over a graph $G = (V, E)$ with an initial set of *active nodes*, representing the locations where the species is initially present. Each node represents a habitat patch and edges represent geographical connectivity among habitat patches. In each step, new nodes are activated based on local activation rules among neighbors that take into account the suitability score of the given habitat patch.

The habitat patches are partitioned into a set of non-overlapping *parcels* $\mathcal{P}_1, \ldots, \mathcal{P}_L$. Let $\mathcal{A} = \{1, \ldots, L\}$ be a set of management actions, where action l consists of buying the corresponding land parcel $\mathcal{P}_l$ at some given cost c_l. An action strategy is indicated by the vector $\mathbf{y} = \langle y_1, \ldots, y_L \rangle$. The binary variable y_l indicates whether action l is taken ($y_l = 1$) or not ($y_l = 0$).

Given a fixed budget and a set of target nodes, we need to select management actions such that the expected number of active target nodes is maximized over some fixed horizon.

Let the variables $\{X_v(\mathbf{y})\}$ capture the outcome of a cascade under a given strategy $\mathbf{y}$. Each variable $X_v(\mathbf{y})$ is binary with $X_v(\mathbf{y}) = 1$ indicating that node v is activated. Let $\mathcal{B}$ denote the given budget and $\mathcal{T}$ denote the set of target nodes. The optimization problem is:

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

This is an instance of the stochastic optimization problem studied by Kleywegt, Shapiro, and Homem-de-Mello (2002), of the following form:

$$\max_{x \in S} \big\{ g(x) := \mathbb{E}_P\big[G(x, W)\big] \big\} \tag{2}$$

where W is a random vector with distribution P, S is a finite set, $G(x, w)$ is a function of two variables x and w and $\mathbb{E}_P\big[G(x, W)\big]$ denotes the expected value under distribution P. Such stochastic optimization problems are particularly hard because even the objective function cannot be written in a closed form. Therefore, such problems are approximated by using a sample average approximation scheme in which multiple random samples of W are generated and the expected value function is approximated by the *deterministic* sample average function. The sample average approximation can then be solved using standard solvers such as CPLEX. This SAA scheme in the context of conservation planning amounts to generating a number of *training cascades* of the target species in the graph G and then solving the resulting deterministic optimization problem. For details about sampling cascades, we refer to (Sheldon et al. 2010). The resulting SAA optimization problem can be written as a mixed integer program (MIP).

Let $k \in [1..N]$ denote a training cascade. Let the activation of a node v under a cascade k be denoted by the binary variable x_v^k. The binary vector $\mathbf{y}$ denotes the action strategy. Let $\mathcal{A}(v)$ be the subset of actions that purchase the node v. Each training cascade k defines a subgraph $G_k = (V, E_k)$ of G which includes only active edges and nodes. The set V_0 denotes initially conserved nodes and $\mathcal{S}$ denotes initial locations of species. The following MIP denotes the optimization problem:

$$\begin{aligned}
\max_{\boldsymbol{x},\boldsymbol{y}} \quad & \frac{1}{N} \sum_{k=1}^{N} \sum_{v \in \mathcal{T}} x_v^k \\
\text{s.t.} \quad & \sum_{l=1}^{L} c_l y_l \leq \mathcal{B} \\
& x_v^k \leq \sum_{l \in \mathcal{A}(v)} y_l, \quad \forall v \notin V_0,\ \forall k \\
& x_v^k \leq \sum_{(u,v) \in E_k} x_u^k, \quad \forall v \notin \mathcal{S},\ \forall k \\
& x_v^k \in [0, 1],\ y_l \in \{0, 1\}
\end{aligned} \tag{3}$$

For a detailed explanation of the above constraints we refer to Sheldon et al. (2010). Intuitively, they represent a continuous path description for a cascade.

3 Our Approach

First we rewrite the above optimization problem compactly as follows. We also change the maximization to minimization by negating the objective.

$$\begin{aligned}
\min_{\{\boldsymbol{x^k}\},\boldsymbol{y}} \quad & -\frac{1}{N} \sum_{k=1}^{N} f_k(\boldsymbol{x^k}, \boldsymbol{y}) \\
\text{s.t.} \quad & \sum_{l=1}^{L} c_l y_l \leq \mathcal{B} \\
& \boldsymbol{x^k} \in \boldsymbol{X^k},\ \forall k \\
& y_l \in \{0, 1\},\ \forall l
\end{aligned} \tag{4}$$

where each function f_k and the corresponding feasible set $\boldsymbol{X^k}$ are defined as follows:

$$f_k(\boldsymbol{x^k}, \boldsymbol{y}) = \sum_{v \in \mathcal{T}} x_v^k \tag{5}$$

$$x_v^k \leq \sum_{l \in \mathcal{A}(v)} y_l, \ \forall v \notin V_0 \tag{6}$$

$$x_v^k \leq \sum_{(u,v) \in E_k} x_u^k, \ \forall v \notin \mathcal{S} \tag{7}$$

$$x_v^k \in [0,1] \tag{8}$$

So far we have only reformulated the global MIP (3). However, this reformulation highlights the *separable structure* present in this MIP and lends itself to the application of the Lagrangian relaxation method.

3.1 Lagrangian Relaxation of the Global MIP

The main challenge in solving the global MIP (3) is that as the number of samples, N, increases, the number of variables and constraints increase as does the complexity of solving the MIP. Our main goal is to provide a general approach to address this computational bottleneck while also providing quality bounded solutions. Next, we show how Lagrangian relaxation technique can achieve this objective by exploiting the separable structure present in the MIP (4). To expose this structure, we present an equivalent form of MIP (4) by creating one copy of variables $\boldsymbol{y}$ for each cascade k and a vector of global variables $\boldsymbol{d}$ as follows:

$$\begin{aligned} \min_{\{\boldsymbol{x^k}, \boldsymbol{y^k}\}, \{d_l\}} \quad & -\frac{1}{N} \sum_{k=1}^{N} f_k(\boldsymbol{x^k}, \boldsymbol{y^k}) \\ \text{s.t.} \quad & \sum_{l=1}^{L} c_l y_l^k \leq \mathcal{B}, \ \forall k \\ & \boldsymbol{x^k} \in \boldsymbol{X^k}, \ \forall k \\ & y_l^k = d_l, \ \forall l, \ \forall k \\ & y_l^k \in \{0,1\}, \ \forall l, \ \forall k \end{aligned} \tag{9}$$

Notice the constraint $y_l^k = d_l$ that enforces different copies of the variable y_l to be globally consistent and equal to d_l. Next, we write the Lagrangian of the above MIP by relaxing this constraint:

$$L(\{\boldsymbol{x^k}, \boldsymbol{y^k}\}, \boldsymbol{d}, \boldsymbol{\lambda}) = -\frac{1}{N} \sum_{k=1}^{N} f_k(\boldsymbol{x^k}, \boldsymbol{y^k}) + \sum_{l,k} \lambda_l^k (y_l^k - d_l)$$
$$\boldsymbol{x^k} \in \boldsymbol{X^k} \ \forall k \ , \ \boldsymbol{y^k} \in \boldsymbol{Y^k} \ \forall k \tag{10}$$

where each $\boldsymbol{Y^k}$ is defined as follows:

$$\boldsymbol{Y^k} = \big\{ \langle \ldots, y_l^k, \ldots \rangle | y_l^k \in \{0,1\} \ \forall l \wedge \sum_{l=1}^{L} c_l y_l^k \leq \mathcal{B} \big\} \tag{11}$$

Therefore, the dual, $\min_{\{\boldsymbol{x^k}, \boldsymbol{y^k}\}, \boldsymbol{d}} L(\{\boldsymbol{x^k}, \boldsymbol{y^k}\}, \boldsymbol{d}, \boldsymbol{\lambda})$, is:

$$\begin{aligned} q(\boldsymbol{\lambda}) &= \min_{\{\boldsymbol{x^k}, \boldsymbol{y^k}\}, \boldsymbol{d}} \left\{ -\frac{1}{N} \sum_{k=1}^{N} f_k(\boldsymbol{x^k}, \boldsymbol{y^k}) + \sum_{l,k} \lambda_l^k (y_l^k - d_l) \right\} \\ &= \min_{\{\boldsymbol{x^k}, \boldsymbol{y^k}\}, \boldsymbol{d}} \sum_{k=1}^{N} \left\{ -\frac{f_k(\boldsymbol{x^k}, \boldsymbol{y^k})}{N} + \sum_{l} \lambda_l^k y_l^k \right\} - \sum_{l,k} \lambda_l^k d_l \end{aligned}$$

We note that the variable d_l is unconstrained. Therefore, to prevent the dual from being unbounded from below, the Lagrange multipliers must satisfy the following constraints:

$$\{\lambda_l^k\} \in \Lambda_l \ \ \forall l \tag{12}$$

$$\Lambda_l = \big\{ \{\lambda_l^k\} | \sum_{k=1}^{N} \lambda_l^k = 0 \big\} \tag{13}$$

The above condition further simplifies the dual $q(\boldsymbol{\lambda})$ as the last term in the dual vanishes. We finally have the dual:

$$q(\boldsymbol{\lambda}) = \sum_{k=1}^{N} \min_{\{\boldsymbol{x^k}, \boldsymbol{y^k}\}} \left\{ -\frac{f_k(\boldsymbol{x^k}, \boldsymbol{y^k})}{N} + \sum_{l} \lambda_l^k y_l^k \right\} \tag{14}$$

Scalability: Interestingly, the above expression of the dual highlights the key reason for scalability. The dual can be evaluated by solving the inner minimizations in the above problem *independently* of each other. Crucially, each inner minimization problem corresponds to a MIP for a *single* SAA sample augmented by dual variables. Hence, the size of the inner minimization problem is defined by the size of the MIP for a *single* sample regardless of the total number of SAA samples. This significantly improves the scalability of solving the dual problem instead of a single global MIP.

3.2 Maximizing the Dual

It is well known that the dual solution provides a lower bound over the primal solution. Therefore, the dual optimization problem we will solve is the following:

$$\begin{aligned} \max_{\boldsymbol{\lambda}} \quad & q(\boldsymbol{\lambda}) = \sum_{k=1}^{N} g^k(\boldsymbol{\lambda^k}) \\ \text{s.t.} \quad & \{\lambda_l^k\} \in \Lambda_l, \ \forall l \end{aligned} \tag{15}$$

where $g^k(\boldsymbol{\lambda^k})$ denotes the inner minimization problem of Eq. (14) for the SAA sample k. A fundamental result in the optimization literature is that the dual problem is always concave regardless of the primal problem (Boyd and Vandenberghe 2004). Therefore, there are no local optima in the dual problem (15). We can maximize it by using iteratively the projected subgradient technique (Bertsekas 1999). According to this technique, the dual variables $\boldsymbol{\lambda^k}$ are updated as follows:

$$\boldsymbol{\lambda^{k,i+1}} \leftarrow \big[\boldsymbol{\lambda^{k,i}} + \alpha_{i+1} \nabla g^k(\boldsymbol{\lambda^{k,i}}) \big]_{\Lambda} \tag{16}$$

where i denotes the iteration number, $[\cdot]_{\Lambda}$ denotes the projection onto the constraint set Λ, and α is the step size. It is well known that for a function of the type

$$h(\boldsymbol{\lambda}) = \min_{x} \big\{ a(x) + \boldsymbol{\lambda} \cdot b(x) \big\},$$

the gradient $\nabla h(\boldsymbol{\lambda}) = b(\bar{x})$, where $\bar{x}$ is the optimal solution to the problem $\min_x \left\{a(x)+\boldsymbol{\lambda}\cdot b(x)\right\}$. Each function $g^k(\boldsymbol{\lambda^k})$ in our case is of this type. Therefore the gradient of each dual variable $\lambda_l^{k,i}$ is given by

$$\nabla g^k(\lambda_l^{k,i}) = \bar{y}_l^{k,i} \tag{17}$$

where $\bar{y}_l^{k,i}$ denotes the solution of the inner minimization problem of Eq. (14) for sample k for iteration i. Intuitively, it denotes the optimal decision as to buy the parcel l or not for iteration i. This subgradient is available essentially for free without requiring any extra computation other than solving the individual minimization problem for each sample. The projection onto the constraint set of Eq. (13) essentially amounts to the averaging operation and the final update equation for the dual variables is as follows:

$$\lambda_l^{k,i+1} \leftarrow \lambda_l^{k,i} + \alpha_{i+1}\left[\bar{y}_l^{k,i} - \frac{\sum_{k'=1}^{N} \bar{y}_l^{k',i}}{N}\right] \tag{18}$$

Theorem 1 (Weak duality)**.** *If $p^\star$ is the optimal solution of the global MIP (4) and $q^\star$ is the optimal solution of Eq. (15), then $q^\star \leq p^\star$ (Bertsekas 1999).*

Therefore, the dual solution always provides a quality guarantee over the primal solution at each iteration. This can be used effectively to provide a certificate of optimality or near-optimality when the gap between the primal and dual is diminished.

3.3 Extracting the Primal Solution

So far, we have only discussed the update of the dual variables $\boldsymbol{\lambda}$. While solving the dual, the dual variables act as penalty parameters for violating the consistency constraint among different copies of the action variables $\boldsymbol{y}$. If all the copies of an action variable are consistent, extracting the primal solution is trivial. However, inconsistencies may arise among different copies of variables $\boldsymbol{y}$, resulting in $y_l^k \neq y_l^{k'}$ for two samples k and k'. In that case, extracting a consistent primal solution is non-trivial.

Our strategy to extract the primal is based upon the insight that thanks to the dual variables, most of the copies of action variables will be consistent, with a few exceptions. Thus, we create L binary random variables, one for each action l, and initialize them as follows:

$$P(r_l = 1) = \sum_{k=1}^{N} \frac{\mathbb{1}_{[X=1]}(\bar{y}_l^k)}{N},\ \forall l \tag{19}$$

All the variables for which $P(r_l = 1) = 1$ or $P(r_l = 0) = 1$ are consistent and their corresponding primal variable y_l is set accordingly. For the rest of the variables, it might happen that *most* of its copies agree on a single assignment, except a few. For example, consider $N = 30$ samples. After solving the dual, 28 copies of a variable y_l may agree on the value 0 with only 2 suggesting it to be 1. In this case, we set the primal variable $y_l = 0$ based on the majority vote. While setting variables in this manner, we also take into account the budget constraint—a variable is never set to 1 if it violates the available budget. More precisely, we use a threshold parameter $\mathcal{P}$ (which can be 0.8 or 0.9) with the following rules to extract the primal:

$$y_l = 1 \quad \text{iff } P(r_l) \geq \mathcal{P} \tag{20}$$

$$y_l = 0 \quad \text{iff } P(r_l) \leq 1 - \mathcal{P} \tag{21}$$

It turns out that most of the action variables can be set according to this rule depending on the threshold $\mathcal{P}$. However, there can still be a few action variables which have too much uncertainty associated with them. Instead of setting them randomly, we formulate a *reduced* version of the global MIP (3). In this MIP, the only integer variables are those action variables y_l which cannot be extracted using the threshold rules. In our experimental setting, we found that only a small fraction of the total action variables fall into this category (about 20%). Thus, this reduced MIP is much easier to solve. Furthermore, the threshold parameter $\mathcal{P}$ can be adjusted to further reduce the size of this reduced MIP.

We now discuss how to set the step size α for an iteration i, which is based on the quality of the extracted primal solution. Let APX_i denote the primal solution quality and $DUAL_i$ be the dual solution quality. We set α as follows:

$$\alpha_i = \frac{APX_i - DUAL_i}{\sum_{l,k}\left(\bar{y}_l^{k,i}\right)^2} \tag{22}$$

The motivation for this rule is that when the gap between the primal and dual is large, we take large steps in the gradient direction. As the gap narrows, the step size also decreases. Bertsekas (1999) provides a theoretical justification of this approach. Moreover, based on the properties of the step size, it can be shown that the subgradient approach converges to the optimal solution of problem (15) (Bertsekas 1999).

3.4 Extension to Stochastic Optimization

The idea of Lagrangian relaxation is quite general and not limited to the conservation planning problem. We can easily apply this approach of creating multiple copies of variables to the sample average approximation scheme to any generic stochastic optimization problem of the form shown in Eq. (2). The SAA optimization problem is:

$$\max_{\boldsymbol{x}\in S} \hat{g}_N(\boldsymbol{x}) = \frac{1}{N}\sum_{k=1}^{N} G(x, W^k) \tag{23}$$

Even though the above problem is deterministic, its complexity increases, often exponentially, with the number of samples N (Kleywegt, Shapiro, and Homem-de-Mello 2002). We can decompose the above optimization problem by creating multiple copies of the variable $\boldsymbol{x}$ and relaxing the consistency constraints in the same manner we have treated the conservation planning problem. Thus, the proposed approach using Lagrangian relaxation appears promising for a range of stochastic optimization problems.

3.5 Variants of Lagrangian Relaxation

The Lagrangian relaxation approach we developed in this work offers a general paradigm that can be easily adapted

CPLEX				DD-Plan				
# Samples	Quality	First Achieved	Total Time	Quality	U.B.	First Achieved	Total Time	Time per Iteration
10	55.30	980	1029	55.30	59.75	63	3583	32.28
15	55.33	1126	2851	55.33	60.28	148	5131	46.23
20	52.65	13636	29847	52.65	59.34	1217	6688	60.25
25	51.12	22006	84188	51.12	58.10	8428	9272	83.53
30*	48.46	4496	36000	48.46	57.38	196	11510	103.69
35*	48.40	27598	36000	48.71	57.73	145	11837	106.64
20% Budget								
10	99.90	78	139	99.90	102.95	13	1286	10.63
15	99.27	577	590	99.27	103.21	14	1936	16.00
20	98.20	254	647	98.20	101.77	51	2638	21.80
25	97.36	5960	7927	97.36	101.57	95	4054	33.50
30	95.50	9221	16178	95.50	100.48	749	7688	63.54
35	95.54	16217	22259	95.54	100.44	852	7839	64.79
40	95.23	17654	29855	95.23	100.12	770	12693	104.90

Table 1: Solution quality and time (in seconds) comparison between CPLEX and the Lagrangian relaxation method

to the characteristics of different optimization contexts. We discuss several useful variants below.

In the above relaxation, each dual subproblem corresponds to a single SAA sample. We can strengthen this relaxation in a flexible way by clustering multiple SAA samples together into a single dual subproblem. This will decrease the duality gap at the expense of increasing the computational complexity of solving the dual. One way to cluster different samples is based on the notion of *similarity*. If two samples (or cascades in the conservation planning problem) are not similar as measured by some distance function such as the Hamming distance, then we can include them in the *same* cluster as otherwise they are most likely to increase the inconsistency among different copies of variables.

Another variant, which mainly applies to situations where each dual subproblem is a MIP, is based on the number of integer variables. In the current formulation, each dual subproblem has the same number of integer variables as the global MIP. This approach worked well in our case as there are relatively few integer variables (144 to be precise). The complexity is dominated by the number of continuous variables and constraints, which can range up to 820K variables and 1.6M constraints for 40 samples and horizon 50. This makes the branch-and-bound approach of CPLEX space and time consuming for the global MIP. We can easily adapt the Lagrangian relaxation approach to handle cases where the complexity is dominated by the number of *integer variables*. Consider a scenario with N samples and M integer variables in the global MIP. We construct N dual subproblems as before by creating multiple copies of integer variables. However, instead of each dual subproblem having M integer variables, we use the following approach. We first partition the set of integer variables into N non-overlapping sets S_i (assuming $M > N$). For the dual subproblem i, only the variables in the set S_i are made of integer type, and the rest are continuous. This formulation effectively decreases the number of integer variables while also providing the bounds on solution quality. If there is no duality gap, then the optimal dual solution will be equal to the optimal primal solution as consistency constraints among all the copies of the variables will ensure that all the variables are integer.
These variations illustrate that the Lagrangian relaxation technique provides a general and flexible framework to handle stochastic optimization problems and can prove useful in other sustainability tasks.

4 Experimental Evaluation

We used a publicly available conservation planning benchmark which represents a geographical region on the coast of North Carolina (Ahmadizadeh et al. 2010). The conservation target is the Red-cockaded Woodpecker. The graph consists of 411 territories or habitat patches grouped into 146 parcels. Each parcel can be purchased at a certain cost, establishing 146 action variables. The spread of species or the cascade is modeled using a habitat suitability score. We used a plan horizon of 50 years to make the problem realistic and challenging. We evaluated the performance of the Lagrangian relaxation technique on several different metrics: a) solution quality, b) runtime, and c) the quality of bounds provided by the dual solution with the increasing number of SAA samples. In the experiments, we treat the problem as a maximization problem, with the dual solution providing an upper bound on the optimal primal solution. All the experiments were done on a 12-core Mac Pro with 18GB RAM. The global MIP was solved using CPLEX 12.2, which was allowed to utilize all the cores. Our algorithm was implemented in C++ and used CPLEX as a subroutine to solve each dual subproblem. It also utilized the 12 cores to solve the dual subproblems in parallel.

Solution quality: Table 1 shows the solution quality and runtime comparisons between CPLEX and the Lagrangian relaxation approach, which we refer to as the dual-decomposition planning (DD-Plan) method, with an increasing number of SAA samples. The top half of the table shows results when the available budget is 10% of the total amount required to purchase all the parcels and the bottom half

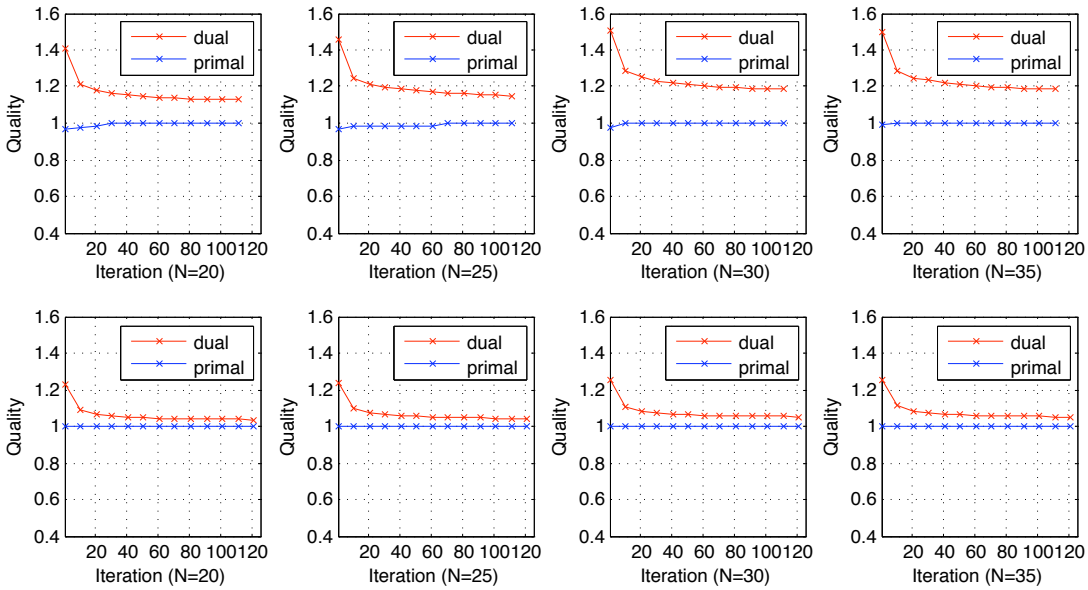

Figure 1: Solution quality vs. iterations for different sample sizes (N). Top row corresponds to 10% budget, bottom row to 20% budget.

shows the results for 20% budget. The 10% budget setting was particularly challenging for CPLEX, as also noted in (Ahmadizadeh et al. 2010). CPLEX was run until optimality was proved except for starred ($\star$) entries, for which we used a time cutoff of 36K seconds as the optimality gap provided by CPLEX was relatively high. The 'Quality' column under 'CPLEX' denotes the optimal quality achieved by CPLEX (except for starred entries). An encouraging observation is that the DD-Plan approach achieves optimal quality for every sample size and budget setting. As the 10% budget setting was particularly challenging, we performed additional experiments for each sample size from 10 to 25 (not shown in the table for brevity). The results showed that the DD-Plan algorithm achieves optimal solutions for each of them. This confirms that DD-Plan can consistently extract good primal solutions using the technique of Sec. 3.3.

Run time: Table 1 also shows runtime comparison between CPLEX and DD-Plan. One striking observation is that DD-Plan finds the optimal primal solution significantly faster than CPLEX. The column 'First Achieved' shows when the best primal solution quality was first achieved for both CPLEX and DD-Plan. For both budget settings, DD-Plan is significantly faster than CPLEX, by more than an order of magnitude. For example, for sample size 35, CPLEX achieves its best solution quality at 27598 seconds, whereas the DD-Plan approach achieves the same in just 145 seconds. For the 20% budget setting, our primal extraction scheme often provided the optimal solution within the first 10 iterations. These results are reflected by the timing results under the 'First Achieved' column for DD-Plan.

Another observation is that as the number of samples increases, the runtime of CPLEX needed to prove optimality increases significantly. For example, for sample size 10 and 10% budget, CPLEX takes only 1029 seconds, whereas for sample size 20, it takes about 29847 seconds. This happens because the resulting global MIP can become very large as the sample size increases, with the number of continuous variables around 820K and the number of constraints around 1.6M for sample size 40. This adversely affects the space and time requirement of the branch-and-bound procedure of CPLEX, where each node corresponds to this MIP.

Unlike CPLEX, the runtime of DD-Plan increases gracefully with a larger number of samples. This is theoretically justified as the runtime of each iteration of DD-Plan is *linear* in the number of samples. This is because each dual subproblem always solves a MIP of a fixed size regardless of the total number of samples.

The last column of Table 1, 'Time per Iteration', shows the average time required by the DD-Plan approach per iteration, which includes solving the dual and extracting the primal solution using the reduced MIP. The results show roughly a linear increase in the runtime per iteration with the increasing number of samples. Moreover, because each dual subproblem can be solved in parallel, the runtime can be further decreased by using more CPU cores. These results also confirm that solving the reduced MIP is computationally fast, because most of the primal variables could be set using the threshold rule. We used a particularly strict threshold value of 0.9. This shows that the dual variables were effective in forcing *most* of the copies of action variables to agree on a single value. These observations substantiate our claim that DD-Plan provides a scalable technique to solve large-scale conservation planning problems and produce good solution quality.

Quality bounds: The DD-Plan approach, though not guaranteed to find the optimal solution, provides an upper bound on the optimal primal solution at each iteration. Using this dual solution, we can bound the gap between the optimal solution and the current primal solution. The column 'U.B.' in Table 1 shows the best bounds DD-Plan provided

upon termination. For the 10% budget setting, we get a guarantee of the primal solution within 85 to 90% of the optimal (calculated as $Primal * 100/Dual$) for most sample sizes. Since the problem with this budget setting is particularly difficult, the DD-Plan approach could not prove optimality (although it did produce the optimal primal solution), but it provided good upper bound. For the 20% budget setting, which is relatively easier, we can see that DD-Plan provides excellent bounds with the solution guaranteed to be within 95 to 97% of the optimal value. Fig. 1 shows how the primal solution and dual solution evolve with the number of iterations. We show the normalized quality on the y-axis with 1 denoting the optimal solution. A key observation, also highlighted earlier, is that DD-Plan is able to provide a good solution, often near-optimal, very early. The upper bound also decrease nearly monotonically with the increasing number of iterations. For the 20% budget setting, the gap between the upper bound and the primal solution becomes quite narrow. Furthermore, these bounds can be tightened using the clustering techniques discussed in Sec. 3.5.

5 Conclusion

We address an important problem of spatial conservation planning, in which the goal is to maximize the expected spread of an endangered species using limited resources. This stochastic optimization problem has been previously addressed using the technique of sample average approximation and solved using off-the-shelf mixed integer programming (MIP) solvers. We identify a key bottleneck of this approach—the rapid increase in runtime with an increasing number of samples—and tackle it by exploiting the separable structure present in the global MIP. Using the Lagrangian relaxation approach, we create a dual problem composed of computationally independent subproblems, thereby achieving a very significant reduction in complexity. The dual problem itself is optimized using the projected subgradient method. We also discuss several variants of the Lagrangian relaxation approach that are beneficial in different optimization contexts. Empirically, the resulting algorithm DD-Plan performs well. It produces optimal solutions for a range of problems significantly faster than CPLEX. These results demonstrate that the Lagrangian relaxation technique offers a general method to handle stochastic optimization problems that arise in computational sustainability, and they pave the way to tackling much larger benchmark problems.

Acknowledgments

We thank anonymous reviewers for their helpful comments. This work was funded in part by the National Science Foundation grants number IIS-1116917 and IIS-1160424.

References

Ahmadizadeh, K.; Dilkina, B.; Gomes, C. P.; and Sabharwal, A. 2010. An empirical study of optimization for maximizing diffusion in networks. In *Proceedings of the 16th International Conference on Principles and Practice of Constraint Programming*, 514–521.

Bertsekas, D. P. 1999. *Nonlinear Programming*. Athena Scientific, 2nd edition.

Boyd, S., and Vandenberghe, L. 2004. *Convex Optimization*. New York, NY, USA: Cambridge University Press.

Boyd, S.; Xiao, L.; and Mutapcic, A. 2003. Subgradient methods. Lecture notes.

Golovin, D.; Krause, A.; Gardner, B.; Converse, S. J.; and Morey, S. 2011. Dynamic resource allocation in conservation planning. In *Proceedings of the 25th Conference on Artificial Intelligence*, 1331–1336.

Kempe, D.; Kleinberg, J.; and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD 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. *SIAM Journal on Optimization* 12:479–502.

Komodakis, N.; Paragios, N.; and Tziritas, G. 2011. MRF energy minimization and beyond via dual decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33(3):531 –552.

Krause, A.; Leskovec, J.; Guestrin, C.; VanBriesen, J.; and Faloutsos, C. 2008. Efficient sensor placement optimization for securing large water distribution networks. *Journal of Water Resources Planning and Management* 134(6):516–526.

Leskovec, J.; Krause, A.; Guestrin, C.; Faloutsos, C.; VanBriesen, J.; and Glance, N. 2007. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 420–429.

Lubin, M.; Petra, C. G.; Anitescu, M.; and Zavala, V. 2011. Scalable stochastic optimization of complex energy systems. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 64:1–10.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions – I. *Mathematical Programming* 14:265–294.

Rush, A. M.; Sontag, D.; Collins, M.; and Jaakkola, T. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of the International Conference on Empirical Methods in Natural Language Processing*, 1–11.

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 *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, 517–526.

Sontag, D.; Globerson, A.; and Jaakkola, T. 2010. Introduction to dual decomposition for inference. In Sra, S.; Nowozin, S.; and Wright, S., eds., *Optimization for Machine Learning*. MIT Press. 219–254.