Nature Reserve Selection Problem: A Tight Approximation Algorithm

Magnus Bordewich and Charles Semple

Abstract—The Nature Reserve Selection Problem is a problem that arises in the context of studying biodiversity conservation. Subject to budgetary constraints, the problem is to select a set of regions to be conserved so that the phylogenetic diversity of the set of species contained within those regions is maximized. Recently, it has been shown in a paper by Moulton et al. that this problem is NP-hard. In this paper, we establish a tight polynomial-time approximation algorithm for the Nature Reserve Section Problem. Furthermore, we resolve a question on the computational complexity of a related problem left open by Moulton et al.

Index Terms—Combinatorial algorithms, phylogenetic diversity, biodiversity conservation.

1 Introduction

A central task in conservation biology is measuring, predicting, and preserving biological diversity as species face extinction. In this regard, individual species are often the focus of attention. However, as pointed out by Rodrigues et al. [13], this is not necessarily the best way of conserving diversity:

> Although conservation action is frequently targeted toward single species, the most effective way of preserving overall species diversity is by conserving viable populations in their natural habitats, often by designating networks of protected areas.

In this paper, we consider a natural computational problem in the context of conserving whole habitats instead of individual species.

Dating back to 1992 [1], phylogenetic diversity (PD) is a prominent quantitative tool for measuring the biodiversity of a collection of species. This measure is based on the evolutionary distance among the species in the collection. Loosely speaking, if $\mathcal{T}$ is a phylogenetic tree whose leaf set X represents a set of species and whose edges have real-valued lengths (weights), then the PD score of a subset S of X is the sum of the weights of the edges of the minimal subtree of $\mathcal{T}$ connecting the species in S. The standard PD optimization problem is to find a subset of X of a given size, which maximizes the PD score among all subsets of X of that size. Perhaps surprisingly, the so-called greedy algorithm solves this problem exactly [1], [10], [16].

A canonical extension of the standard problem allows for the consideration of conserving various regions such as nature reserves at some cost. In particular, aside from an edge-weighted phylogenetic tree $\mathcal{T}$ with leaf set X, we have a collection $\mathcal{A}$ of regions or areas containing species in X, with each region having an associated cost of preservation. Given a fixed budget B, the PD optimization problem for this extension is to find a subset of the regions in $\mathcal{A}$ to be preserved, which maximizes the PD score of the species contained within at least one preserved region while keeping within the budget. This problem is called the Budgeted Nature Reserve Selection (BNRS) and generalizes the analogous unit cost problems described in [9], [11], [12], and [13]. Allowing the cost of conserving each region to vary provides additional cost structure that is important in practice but, as commented in [2] and [5], is often omitted from such problems in conservation biology. For applications of BNRS with unit costs and using the maximum PD score across areas to make assessments in conservation planning, see, for example, [8], [12], and [15].

Moulton et al. [9] showed that a particular instance of BNRS (and, therefore, BNRS itself) is NP-hard; that is, there is no polynomial-time algorithm for solving it, unless $\mathrm{P} = \mathrm{NP}$. Despite this negative result, in this paper, we show that there is a polynomial-time $(1 - 1/e)$-approximation algorithm for this problem. That is, an efficient algorithm that generates a solution that has at least a $(1 - 1/e)$ fraction ($\approx$ 63 percent) of the PD of the optimal solution. Moreover, this approximation ratio is the best possible.

This paper is arranged as follows: Section 2 contains a formal definition of BNRS and a discussion of related work. Section 3 contains the description of the approximation algorithm and the statement of the main theorem, the proof of which is established in Section 4. In Section 5, we answer a computational complexity question on a related problem that was left open in [9]. Throughout most of this paper, we restrict ourselves to PD in the setting of unrooted trees. However, in Section 6, we extend our earlier results to the rooted analog of BNRS (RBNRS). The notation and terminology in this paper follows [14].

2 Budgeted Nature Reserve Selection

In order to define BNRS formally, we require the following definitions. A *phylogenetic X-tree* $\mathcal{T}$ is an (unrooted) tree

- *M. Bordewich is with the Department of Computer Science, University of Durham, South Road, Durham DH1 3LE, UK. E-mail: m.j.r.bordewich@durham.ac.uk.*
- *C. Semple is with the Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch, New Zealand. E-mail: c.semple@math.canterbury.ac.nz.*

Manuscript received 18 Feb. 2007; revised 2 July 2007; accepted 11 Sept. 2007; published online 13 Sept. 2007.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0019-0207.
Digital Object Identifier no. 10.1109/TCBB.2007.70252.

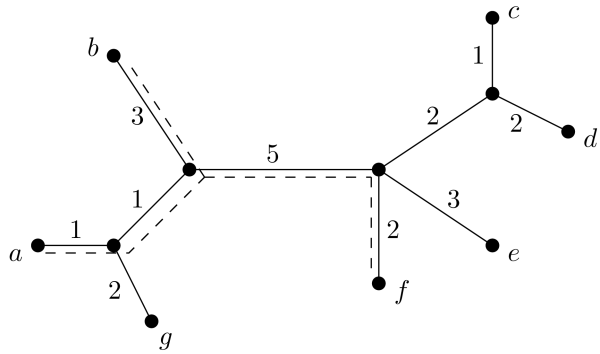

Fig. 1. A phylogenetic X-tree with edge lengths, where $X = \{a, b, c, d, e, f, g\}$.

with no degree-2 vertices and whose leaf set is X. Let $\mathcal{T}$ be a phylogenetic X-tree with edge set E and let $\lambda : E \rightarrow \mathbb{R}^{\geq 0}$ be an assignment of lengths (weights) to the edges of $\mathcal{T}$. Ignoring the dashed edges, Fig. 1 illustrates a phylogenetic X-tree with nonnegative real-valued edge weights, where $X = \{a, b, c, d, e, f, g\}$.

For a subset S of X, the *PD* of S on $\mathcal{T}$ is the sum of the edge lengths of the minimal subtree of $\mathcal{T}$ that connects S. This sum is denoted as $PD_{(\mathcal{T},\lambda)}(S)$; however, if there is no ambiguity, we usually shorten it to $PD(S)$. Referring to Fig. 1, if $S = \{a, b, f\}$, then $PD(S)$ is equal to the sum of the weights of the minimal subtree (dashed edges) that connects a, b, and f; in particular, $PD(S) = 12$.

BNRS is formally defined as follows:

Problem: BNRS
Instance: A phylogenetic X-tree $\mathcal{T}$, a nonnegative (real valued) weighting λ on the edges of $\mathcal{T}$, a collection $\mathcal{A}$ of subsets of X, a cost function c on the sets in $\mathcal{A}$, and a budget B.
Question: Find a subset $\mathcal{A}'$ of $\mathcal{A}$, which maximizes the PD score of $\bigcup_{A\in\mathcal{A}'} A$ on $\mathcal{T}$ such that $\sum_{A\in\mathcal{A}'} c(A) \leq B$.

Referring to the informal discussions in the Introduction, in the statement of BNRS, $\mathcal{A}$ is the collection of regions, and $\mathcal{A}'$ is an optimal subset of regions that we wish to conserve, which maximizes the PD score of the species contained in at least one of the preserved regions. Of course, the total cost of the preserving the regions in $\mathcal{A}'$ is at most B.

Example 2.1. As an example of an instance of BNRS, take $\mathcal{T}$ as the edge-weighted phylogenetic X-tree shown in Fig. 1, choose $\mathcal{A}$ to be

$$\big\{\{b\}, \{f, c\}, \{c, d\}, \{a, b\}, \{a, g\}, \{e\}, \{g, e\}\big\},$$

and set c as the cost function on $\mathcal{A}$ defined by $c(\{b\}) = 4$, $c(\{f, c\}) = 8$, $c(\{c, d\}) = 6$, $c(\{a, b\}) = 10$, $c(\{a, g\}) = 4$, $c(\{e\}) = 4$, and $c(\{g, e\}) = 5$. By setting $B = 24$, we now have an instance of BNRS.

A feasible solution of this instance is $\big\{\{f, c\}, \{a, b\}\big\}$ as $c(\{f, c\}) + c(\{a, b\}) = 8 + 10 = 18$, which is within the budget. Note that the PD score on $\mathcal{T}$ associated with this feasible solution is

$$PD(\{f, c\} \cup \{a, b\}) = 15.$$

An optimal solution is $\big\{\{b\}, \{f, c\}, \{c, d\}, \{e, g\}\big\}$. In this case,

$$\begin{aligned} &c(\{b\}) + c(\{f, c\}) + c(\{c, d\}) \\ &\quad + c(\{e, g\}) = 4 + 8 + 6 + 5 = 23, \end{aligned}$$

and

$$PD(\{b\} \cup \{f, c\} \cup \{c, d\} \cup \{e, g\}) = 21.$$

The BNRS problem extends the problem OPTIMIZING DIVERSITY VIA REGIONS described in [9]. The extension from the latter to the former is that, instead of each region having a unit cost, the cost of conserving each region is allowed to vary. Moulton et al. [9] showed that OPTIMIZING DIVERSITY VIA REGIONS is NP-hard and, so, consequently, BNRS is also NP-hard. BNRS also extends the problem BUDGETED MAXIMUM COVERAGE, in which each element of X has a weight, and the objective is to maximize the total weight of $\bigcup_{A\in\mathcal{A}'} A$ without the additional structure imposed by a tree [7]. An instance of the latter problem may be realized as a BNRS instance by taking $\mathcal{T}$ to be a star tree with leaf set X and assigning the weight of each element in X to be the length of the incident edge in $\mathcal{T}$. (Note that a star tree is a phylogenetic tree with a single interior vertex.) The approximation algorithm and its proof presented here closely follow those in [7] for the restricted "star tree problem" but must be extended to cover the more complicated interactions of PD score rather than a simple sum of weights. Last, BNRS is the "$0 \xrightarrow{c_r} 0/1$ Nature Reserve Problem" briefly discussed in [11, Appendix].

3 The Approximation Algorithm

In this section, we describe a tight polynomial-time approximation algorithm for BNRS called ApproxBNRS. The fact that it is such an algorithm is established in the next section. For a subset $\mathcal{G}$ of $\mathcal{A}$, the notations $c(\mathcal{G})$ and $PD(\mathcal{G})$ denote $\sum_{A\in\mathcal{G}} c(A)$ and $PD(\cup_{A\in\mathcal{G}} A)$, respectively.

We begin with an informal overview of ApproxBNRS and its subroutine Greedy (see Figs. 2 and 3). By considering all possibilities, ApproxBNRS initially finds a feasible solution whose size is at most two and which maximizes the PD score on $\mathcal{T}$. The resulting solution is called H_1. Next, the algorithm, in turn, considers every subset of $\mathcal{A}$ of size three and applies the subroutine Greedy to each of these subsets. Algorithm Greedy is a greedy-like algorithm that takes a subset $\mathcal{G}_0$ of size three of $\mathcal{A}$ and sequentially adds sets from $\mathcal{A} - \mathcal{G}_0$. The only criteria for which set is selected is that, among all available sets, the ratio of incremental diversity to cost is maximized, and we keep within the budget. The resulting feasible solution that maximizes the PD score is called H_2. Finally, ApproxBNRS compares the two feasible solutions H_1 and H_2 and returns the one with the biggest PD score.

The main result of this paper is the following theorem, whose proof is given in the next section.

Theorem 3.1. *ApproxBNRS is a polynomial-time* $(1 - 1/e)$-approximation *algorithm for* BNRS. *Moreover, for any* $\epsilon > 0$, BNRS *cannot be approximated with an approximation ratio of* $(1 - 1/e + \epsilon)$, *unless* $\mathrm{P} = \mathrm{NP}$.

```
Greedy$(\mathcal{G}_0, U)$:
  $\mathcal{G} \leftarrow \mathcal{G}_0$
  Repeat
    select $A \in U$ that maximizes $\frac{PD(\mathcal{G} \cup A) - PD(\mathcal{G})}{c(A)}$
    if $c(\mathcal{G}) + c(A) \leq B$ then
      $\mathcal{G} \leftarrow \mathcal{G} \cup \{A\}$
    $U \leftarrow U \backslash A$
  Until $U = \emptyset$
  Return $\mathcal{G}$
```

Fig. 2. The greedy algorithm Greedy.

In terms of the runtime of ApproxBNRS, running the greedy subroutine is very efficient; however, repeating this for all subsets of $\mathcal{A}$ of size three incurs a multiplicative overhead of $O(|\mathcal{A}|^3)$. Typically, the number of regions or nature reserves under consideration will be small, and hence, this overhead is minor. Nevertheless, it is worth noting that in the special case that all regions have the same cost, this term can be removed from the runtime. In this situation, the greedy algorithm, starting from a subset $\mathcal{G}_0$ of $\mathcal{A}$ of size two, which maximizes the PD score among all two-element subsets of $\mathcal{A}$, achieves the approximation ratio $(1 - 1/e)$. The proof of this fact is a routine extension of [6], using the same insights regarding the difference between PD and the ordinary weight function as we have used in the proof of Theorem 3.1 given in the next section.

4 Proof of Theorem 3.1

This section consists of the proof of Theorem 3.1. Let $\mathcal{S}_{\text{opt}}$ denote a subset of $\mathcal{A}$, which is an optimal solution to BNRS. If $|\mathcal{S}_{\text{opt}}| \leq 2$, then ApproxBNRS finds a feasible solution whose PD score is equal to the PD score of $\mathcal{S}_{\text{opt}}$. Therefore, we may assume that $|\mathcal{S}_{\text{opt}}| \geq 3$, in which case it suffices to show that there is a subset $\mathcal{G}_0$ of $\mathcal{A}$, with $|\mathcal{G}_0| = 3$, whose input to Greedy (together with $\mathcal{A} - \mathcal{G}_0$) results in a subset of $\mathcal{A}$, whose PD score is within the approximation ratio stated in the theorem.

Let $\mathcal{G}_0$ be the subset $\{S_1, S_2, S_3\}$ of $\mathcal{S}_{\text{opt}}$ such that S_1 and S_2 are chosen to maximize $PD(S_1 \cup S_2)$ among all subsets of $\mathcal{S}_{\text{opt}}$ of size two and S_3 maximizes $PD(S_1 \cup S_2 \cup S_3)$ among all sets in $\mathcal{S}_{\text{opt}} \setminus \{S_1, S_2\}$. Now, consider Greedy applied to $(\mathcal{G}_0, \mathcal{A} - \mathcal{G}_0)$. Let p denote the first iteration, in which a member A_{l+1}, say, of $\mathcal{S}_{\text{opt}} - \mathcal{G}_0$ is considered but, because of budgetary reasons, is not added to the current greedy solution. Up to iteration p, let, in order, $A_1, A_2, \ldots, A_l$ denote the members of $\mathcal{A} - \mathcal{G}_0$ that are added to $\mathcal{G}_0$ and, for $i = 1, \ldots, l$, let $\mathcal{G}_i = \mathcal{G}_0 \cup \{A_1, A_2, \ldots, A_i\}$. Observe that $\mathcal{G}_l$ is a feasible solution and a subset of the final output $\mathcal{G}^*$ of the greedy subroutine, and hence, $PD(\mathcal{G}^*) \geq PD(\mathcal{G}_l)$. For convenience, we also let $\mathcal{G}_{l+1} = \mathcal{G}_l \cup \{A_{l+1}\}$, but note that $\mathcal{G}_{l+1}$ is not a feasible solution, as $c(\mathcal{G}_{l+1}) > B$. Furthermore, for all i, let c_i denote $c(A_i)$. For a subset $\mathcal{S}$ of $\mathcal{A}$, denote the minimal subtree of $\mathcal{T}$ that connects the elements of X that are contained in at least one member of $\mathcal{S}$ by $\mathcal{T}(\mathcal{S})$. Also, let $E(\mathcal{T}(\mathcal{S}))$ denote the edge set of $\mathcal{T}(\mathcal{S})$. We begin the proof with two lemmas.

```
ApproxBNRS$(\mathcal{T}, \lambda, \mathcal{A}, c, B)$:
  Find $\mathcal{G}'$ in $\{\mathcal{G} : \mathcal{G} \subseteq \mathcal{A}, c(\mathcal{G}) \leq B, |\mathcal{G}| \leq 2\}$ that maximizes PD
  $H_1 \leftarrow \mathcal{G}'$
  $H_2 \leftarrow \emptyset$
  For all $\mathcal{G}_0 \subseteq \mathcal{A}$, such that $|\mathcal{G}_0| = 3$ and $c(\mathcal{G}_0) \leq B$ do
    $U \leftarrow \mathcal{A} \backslash \mathcal{G}_0$
    $\mathcal{G} \leftarrow$Greedy$(\mathcal{G}_0, U)$
    if $PD(\mathcal{G}) > PD(H_2)$ then $H_2 \leftarrow \mathcal{G}$
  If $PD(H_1) > PD(H_2)$ then Return $H_1$, otherwise Return $H_2$
```

Fig. 3. The approximation algorithm ApproxBNRS.

Lemma 4.1. *For all* $i \in \{1, 2, \ldots, l+1\}$

$$PD(\mathcal{G}_i) - PD(\mathcal{G}_{i-1}) \geq \frac{c_i}{B - c(\mathcal{G}_0)} (PD(\mathcal{S}_{\text{opt}}) - PD(\mathcal{G}_{i-1})).$$

Proof. One crucial point to be observed for the approach in [7] to be applicable in our setting is that the incremental diversity from adding the entire optimal solution to the current partial greedy solution is bounded by the sum of the increments that would be obtained from adding each set in the optimal solution individually. We formalize this as follows: Let i be any element in $\{1, 2, \ldots, l+1\}$. Let F denote the set of edges in $E(\mathcal{T}(\mathcal{S}_{\text{opt}} \cup \mathcal{G}_{i-1})) - E(\mathcal{T}(\mathcal{G}_{i-1}))$. Observe that $PD(\mathcal{S}_{\text{opt}} \cup \mathcal{G}_{i-1}) - PD(\mathcal{G}_{i-1})$ is equal to $\sum_{e \in F} \lambda(e)$. Since $\mathcal{G}_{i-1}$ is nonempty, there is, for each $e \in F$, an element in $\bigcup_{A \in (\mathcal{S}_{\text{opt}} - \mathcal{G}_{i-1})} A$ such that e is on the path from that element to a vertex in $\mathcal{T}(\mathcal{G}_{i-1})$. In particular, there is a set A_e in $\mathcal{S}_{\text{opt}} - \mathcal{G}_{i-1}$ such that $\mathcal{T}(\mathcal{G}_{i-1} \cup A_e)$ contains e. Since A_i is chosen so that $\frac{PD(\mathcal{G}_i) - PD(\mathcal{G}_{i-1})}{c_i}$ is maximized, we have, for all $A \in \mathcal{S}_{\text{opt}} - \mathcal{G}_{i-1}$

$$\frac{PD(\mathcal{G}_{i-1} \cup A) - PD(\mathcal{G}_{i-1})}{c(A)} \leq \frac{PD(\mathcal{G}_i) - PD(\mathcal{G}_{i-1})}{c_i}.$$

Therefore, as the total cost of the elements in $\mathcal{S}_{\text{opt}} - \mathcal{G}_{i-1}$ is at most $B - c(\mathcal{G}_0)$

$$
\begin{aligned}
PD(\mathcal{S}_{\mathrm{opt}}) - PD(\mathcal{G}_{i-1}) &\leq PD(\mathcal{S}_{\mathrm{opt}} \cup \mathcal{G}_{i-1}) - PD(\mathcal{G}_{i-1}) \\
&= \sum_{e \in F} \lambda(e) \\
&\leq \sum_{A \in (\mathcal{S}_{\mathrm{opt}} - \mathcal{G}_{i-1})} \left[\sum_{\{e \in F:\, e \in \mathcal{T}(\mathcal{G}_{i-1} \cup A)\}} \lambda(e) \right] \\
&= \sum_{A \in (\mathcal{S}_{\mathrm{opt}} - \mathcal{G}_{i-1})} \\
&\quad \frac{PD(\mathcal{G}_{i-1} \cup A) - PD(\mathcal{G}_{i-1})}{c(A)} c(A) \\
&\leq \sum_{A \in (\mathcal{S}_{\mathrm{opt}} - \mathcal{G}_{i-1})} \frac{PD(\mathcal{G}_i) - PD(\mathcal{G}_{i-1})}{c_i} c(A) \\
&\leq \frac{PD(\mathcal{G}_i) - PD(\mathcal{G}_{i-1})}{c_i} (B - c(\mathcal{G}_0)).
\end{aligned}
$$

Rearrangement now gives the inequality in the statement of the lemma, and the result follows. □

Lemma 4.2. *For all* $i \in \{1, 2, \ldots, l+1\}$

$$
\begin{aligned}
&PD(\mathcal{G}_i) - PD(\mathcal{G}_0) \\
&\geq \left[1 - \prod_{k=1}^{i} \left(1 - \frac{c_k}{B - c(\mathcal{G}_0)} \right) \right] (PD(\mathcal{S}_{\mathrm{opt}}) - PD(\mathcal{G}_0)).
\end{aligned}
$$

Proof. The proof is by induction on i. The result for $i = 1$ immediately follows from Lemma 4.1.

Now, assume that $i \geq 2$ and that the result holds for all j, where $j < i$. Then, by Lemma 4.1 (for the first inequality) and by induction (for the second inequality), we have

$$
\begin{aligned}
PD(\mathcal{G}_i) - PD(\mathcal{G}_0) &= PD(\mathcal{G}_{i-1}) - PD(\mathcal{G}_0) + PD(\mathcal{G}_i) \\
&\quad - PD(\mathcal{G}_{i-1}) \\
&\geq PD(\mathcal{G}_{i-1}) - PD(\mathcal{G}_0) \\
&\quad + \frac{c_i}{B - c(\mathcal{G}_0)} (PD(\mathcal{S}_{\mathrm{opt}}) - PD(\mathcal{G}_{i-1})) \\
&= PD(\mathcal{G}_{i-1}) - PD(\mathcal{G}_0) \\
&\quad + \frac{c_i}{B - c(\mathcal{G}_0)} (PD(\mathcal{S}_{\mathrm{opt}}) - PD(\mathcal{G}_0) \\
&\quad - (PD(\mathcal{G}_{i-1}) - PD(\mathcal{G}_0))) \\
&= \left(1 - \frac{c_i}{B - c(\mathcal{G}_0)} \right) (PD(\mathcal{G}_{i-1}) - PD(\mathcal{G}_0)) \\
&\quad + \frac{c_i}{B - c(\mathcal{G}_0)} (PD(\mathcal{S}_{\mathrm{opt}}) - PD(\mathcal{G}_0)) \\
&\geq \left(1 - \frac{c_i}{B - c(\mathcal{G}_0)} \right) \\
&\quad \left[1 - \prod_{k=1}^{i-1} \left(1 - \frac{c_k}{B - c(\mathcal{G}_0)} \right) \right] \\
&\quad (PD(\mathcal{S}_{\mathrm{opt}}) - PD(\mathcal{G}_0)) \\
&\quad + \frac{c_i}{B - c(\mathcal{G}_0)} (PD(\mathcal{S}_{\mathrm{opt}}) - PD(\mathcal{G}_0)) \\
&= \left[1 - \prod_{k=1}^{i} \left(1 - \frac{c_k}{B - c(\mathcal{G}_0)} \right) \right] \\
&\quad (PD(\mathcal{S}_{\mathrm{opt}}) - PD(\mathcal{G}_0)).
\end{aligned}
$$

This completes the proof of the lemma. □

Proof of Theorem 3.1. Since $c(\mathcal{G}_{l+1}) > B$, we have $\sum_{k=1}^{l+1} c_k = c(\mathcal{G}_{l+1}) - c(\mathcal{G}_0) > B - c(\mathcal{G}_0)$. Furthermore, the function

$$
\prod_{k=1}^{l+1} \left(1 - \frac{c_k}{\sum_k c_k} \right)
$$

has a maximum at $c_k = \frac{\sum_k c_k}{(l+1)}$ for all k. Therefore,

$$
\begin{aligned}
1 - \prod_{k=1}^{l+1} \left(1 - \frac{c_k}{B - c(\mathcal{G}_0)} \right) &\geq 1 - \prod_{k=1}^{l+1} \left(1 - \frac{c_k}{\sum_k c_k} \right) \\
&\geq 1 - \left(1 - \frac{1}{l+1} \right)^{l+1} \\
&\geq 1 - 1/e.
\end{aligned}
$$

Hence, by Lemma 4.2, we have

$$
PD(\mathcal{G}_{l+1}) - PD(\mathcal{G}_0) \geq (1 - 1/e)(PD(\mathcal{S}_{\mathrm{opt}}) - PD(\mathcal{G}_0)). \quad (1)
$$

Recalling that $\mathcal{G}_0 = \{S_1, S_2, S_3\}$, we now show that

$$
PD(S_1 \cup S_2 \cup S_3) - PD(S_1 \cup S_2) \leq PD(\mathcal{G}_0)/3. \quad (2)
$$

Let $A_j = E(\mathcal{T}(S_1 \cup S_2 \cup S_3)) - E(\mathcal{T}((S_1 \cup S_2 \cup S_3) - S_j))$ for $j = 1, 2, 3$. Since

$$
\begin{aligned}
PD(S_1 \cup S_2 \cup S_3) &= PD(S_1 \cup S_2) + \sum_{e \in A_3} \lambda(e) \\
&= PD(S_1 \cup S_3) + \sum_{e \in A_2} \lambda(e) \\
&= PD(S_2 \cup S_3) + \sum_{e \in A_1} \lambda(e),
\end{aligned}
$$

and since S_1 and S_2 were chosen to maximize $PD(S_1 \cup S_2)$, it follows that

$$
\sum_{e \in A_3} \lambda(e) \leq \sum_{e \in A_j} \lambda(e), \qquad j = 1, 2.
$$

It is easily seen that each edge in $E(\mathcal{T}(S_1 \cup S_2 \cup S_3))$ occurs in at most one A_j. Hence,

$$
\begin{aligned}
PD(S_1 \cup S_2 \cup S_3) &\geq \sum_{j=1}^{3} \sum_{e \in A_j} \lambda(e) \\
&\geq 3 \sum_{e \in A_3} \lambda(e),
\end{aligned}
$$

and so

$$
PD(S_1 \cup S_2 \cup S_3) - PD(S_1 \cup S_2) = \sum_{e \in A_3} \lambda(e) \leq PD(\mathcal{G}_0)/3,
$$

giving (2).

Next,

$$
PD(\mathcal{G}_{l+1}) - PD(\mathcal{G}_l) \leq PD(S_1 \cup S_2 \cup A_{l+1}) - PD(S_1 \cup S_2),
$$

and so

$$
\begin{aligned}
PD(\mathcal{G}_{l+1}) - PD(\mathcal{G}_l) &\leq PD(S_1 \cup S_2 \cup S_3) \\
&\quad - PD(S_1 \cup S_2) \leq PD(\mathcal{G}_0)/3.
\end{aligned} \quad (3)
$$

Otherwise, A_{l+1} would have been chosen, instead of S_3, to be in $\mathcal{G}_0$. Putting together (1) and (3), we get

$$\begin{aligned} PD(\mathcal{G}_l) &\geq PD(\mathcal{G}_{l+1}) - PD(\mathcal{G}_0)/3 \\ &\geq (1-1/e)(PD(\mathcal{S}_{\text{opt}}) - PD(\mathcal{G}_0)) + \left(1-\frac{1}{3}\right)PD(\mathcal{G}_0) \\ &> (1-1/e)PD(\mathcal{S}_{\text{opt}}). \end{aligned}$$

This proves the first part of the theorem.

For the proof of the second part, we begin by defining the problem MAXIMUM k-COVERAGE:

Problem: MAXIMUM k-COVERAGE
Instance: A collection $\mathcal{A}$ of subsets of X and an integer k.
Question: Find a subset $\mathcal{A}' = \{A_1, A_2, \dots, A_k\}$ of $\mathcal{A}$ of size k, which maximizes the size of the set $A_1 \cup A_2 \cup \cdots \cup A_k$.

Feige [3] showed that no polynomial-time approximation algorithm for MAXIMUM k-COVERAGE can have an approximation ratio better than $(1-1/e)$, unless $\text{P} = \text{NP}$. Observing that BNRS is a generalization of MAXIMUM k-COVERAGE (see the following), it follows that no approximation algorithm can exist for BNRS with a ratio better than $(1-1/e)$, unless $\text{P} = \text{NP}$.

Given an instance of MAXIMUM k-COVERAGE, take $\mathcal{T}$ to be the star tree on leaf set X, in which each edge has weight 1. Assign a cost of 1 to each element of $\mathcal{A}$ and take the budget $B = k$. Under this setup, it is clear that MAXIMUM k-COVERAGE can be interpreted as a special case of BNRS. Hence, a polynomial-time approximation algorithm for BNRS with approximation ratio α would yield an approximation algorithm for MAXIMUM k-COVERAGE with approximation ratio α. According to Feige [3], no such algorithm can exist for $\alpha = (1 - 1/e + \epsilon)$, unless $\text{P} = \text{NP}$. □

5 Optimizing Diversity with Coverage

The problem OPTIMIZING DIVERSITY WITH COVERAGE was defined in [9], where a very restricted version was shown to have a polynomial-time algorithm. While, superficially, this problem is similar to BNRS, the problem behaves very differently. Loosely speaking, we are given an edge-weighted phylogenetic X-tree $\mathcal{T}$ and a collection $\mathcal{A}$ of subsets of X. Here, the members of $\mathcal{A}$ represent some attributes that the species possess. For example, $\mathcal{A} = \{A_1, A_2, \dots, A_s\}$ may be a collection of taxonomic groups, and each A_i contains the species in X that belong to the group. Given a fixed positive integer k and positive integers $n_1, n_2, \dots, n_s$, the PD optimization problem is to find a subset X' of X of size k, which contains, for all i, at least n_i species with attribute A_i and maximizes the PD score among all such subsets of X of size k. Formally, we have the following problem.

Problem: OPTIMIZING DIVERSITY WITH COVERAGE
Instance: A phylogenetic X-tree $\mathcal{T}$, a nonnegative real-valued weighting λ on the edges of $\mathcal{T}$, a collection $\mathcal{A}$ of subsets of X, a threshold n_A for each $A \in \mathcal{A}$, and a positive integer k.
Question: Find a subset X' of X, which maximizes the PD score of X' on $\mathcal{T}$ such that $|X'| \leq k$ and, for each $A \in \mathcal{A}$, at least n_A species from A are included in X'.

The restricted case solved in [9] is when each element of X appears in exactly one set $A \in \mathcal{A}$ and the subtrees in $\{\mathcal{T}(A) : A \in \mathcal{A}\}$ are vertex disjoint. While this restricted version is shown to be solvable in polynomial time, the question of the computational complexity of the problem under less stringent or no restrictions is left open. We end this section by observing that determining if there is even a feasible solution to the general problem OPTIMIZING DIVERSITY WITH COVERAGE is NP-hard, let alone finding an optimal solution. This is because determining if there is a feasible solution is equivalent to the classic NP-complete decision problem HITTING SET [4].

Problem: HITTING SET
Instance: A collection $\mathcal{A}$ of subsets of X and an integer k.
Question: Does there exist a subset X' of X of size at most k such that $A \cap X' \neq \emptyset$ for all $A \in \mathcal{A}$?

For an instance of HITTING SET as aforementioned, consider the instance of OPTIMIZING DIVERSITY WITH COVERAGE by taking the same sets X and $\mathcal{A}$ and integer k. Now, take $n_A = 1$ for all $A \in \mathcal{A}$ and let $\mathcal{T}$ be an arbitrary phylogenetic X-tree. Then, a subset of X is a feasible solution to the latter problem if and only if it is a feasible solution to the former problem. Conversely, for an instance of OPTIMIZING DIVERSITY WITH COVERAGE, consider the instance of HITTING SET by taking the ground set to be X, the bound to be k, and choosing the collection of subsets of X to be

$$\{B : \exists A \in \mathcal{A}, B \subseteq A, |B| = |A| - n_A + 1\}.$$

In short, this collection consists of, for each $A \in \mathcal{A}$, all subsets of A of size $|B| = |A| - n_A + 1$. It is now easily seen that a subset of X is a feasible solution to this instance of HITTING SET if and only if it is a feasible solution to the original instance of OPTIMIZING DIVERSITY WITH COVERAGE.

The above-mentioned equivalence suggests that the restrictions required to make OPTIMIZING DIVERSITY WITH COVERAGE solvable or even approximable must be fairly severe. Certainly, they must at least make the associated restricted version of HITTING SET tractable. One example could be to restrict k to be at least $\sum_{A \in \mathcal{A}} n_A$. In this case, HITTING SET is trivial, and hence, a feasible solution to OPTIMIZING DIVERSITY WITH COVERAGE can be found easily. However, it is still not clear whether the optimal solution can be found efficiently.

6 Rooted Phylogenetic Trees

In practice, one frequently wants to work with the rooted analog of PD. In this short section, we briefly describe how ApproxBNRS can be applied to RBNRS and the consequences of Theorem 3.1 for this problem.

A *rooted phylogenetic* X-tree $\mathcal{T}$ is a rooted tree with no degree-2 vertices, except for, perhaps, the root and whose leaf set is X. Let E denote the edge set of $\mathcal{T}$ and let $\lambda : E \rightarrow \mathbb{R}^{\geq 0}$ be an assignment of lengths (weights) to the edges of $\mathcal{T}$. For a subset S of X, the *rooted PD* (rPD) of S on $\mathcal{T}$ is the sum of the edge lengths of the minimal subtree of $\mathcal{T}$ that connects S and the root of $\mathcal{T}$. RBNRS is the same as that in the unrooted setting but with the rooted phylogenetic tree replacing the unrooted phylogenetic tree and using rPD instead of PD. In particular, it is formally defined as follows:

Problem: RBNRS
Instance: A rooted phylogenetic X-tree $\mathcal{T}$, a nonnegative (real-valued) weighting λ on the edges of $\mathcal{T}$, a collection $\mathcal{A}$ of subsets of X, a cost function c on the sets in $\mathcal{A}$, and a budget B.
Question: Find a subset $\mathcal{A}'$ of $\mathcal{A}$, which maximizes the rPD score of $\bigcup_{A \in \mathcal{A}'} A$ on $\mathcal{T}$ such that $\sum_{A \in \mathcal{A}'} c(A) \leq B$.

We can interpret an instance of RBNRS as an instance of BNRS as follows: Given an instance of RBNRS, let $\mathcal{T}_\rho$ denote the unrooted phylogenetic tree obtained from $\mathcal{T}$ by adjoining a new leaf ρ via a new edge to the root of $\mathcal{T}$ and then viewing the resulting tree as an unrooted phylogenetic tree with leaf set $X \cup \rho$. Let $\mathcal{A}_\rho$ denote the set $\{\{A \cup \rho\} : A \in \mathcal{A}\}$ and let c_ρ denote the cost function on $\mathcal{A}_\rho$ by setting $c_\rho(A \cup \rho) = c(A)$ for all $A \in \mathcal{A}$. Furthermore, let λ_ρ be the weighting on the edges of $\mathcal{T}_\rho$ by setting the weight of the edge incident with ρ to be 0 and let $\lambda_\rho(e) = \lambda(e)$ for all $e \in E(\mathcal{T})$.

With the above setup, let $\mathcal{G}$ be a feasible solution to RBNRS and let $\mathcal{G}_\rho = \{A \cup \rho : A \in \mathcal{G}\}$. Then, $\mathcal{G}_\rho$ is a feasible solution to the above instance of BNRS, and $rPD(\mathcal{G}) = PD(\mathcal{G}_\rho)$. Similarly, if $\mathcal{G}'_\rho$ is a feasible solution of the above instance of BNRS, then $\mathcal{G}' = \{A : A \cup \rho \in \mathcal{G}'_\rho\}$ is a feasible solution of RBNRS, and $PD(\mathcal{G}'_\rho) = rPD(\mathcal{G}')$. It is now easily seen from this equivalence that ApproxBNRS provides a polynomial-time $(1 - 1/e)$-approximation algorithm for RBNRS. Moreover, the argument at the end of the proof of Theorem 3.1, showing that MAXIMUM k-COVERAGE can be interpreted as a special case of BNRS, still works for RBNRS but uses a rooted star tree instead of an unrooted star tree. Thus, no approximation algorithm for RBNRS exists with a ratio better than $(1 - 1/e)$, unless $\mathrm{P} = \mathrm{NP}$.

Acknowledgments

Magnus Bordewich was supported by the EPSRC, and Charles Semple was supported by the New Zealand Marsden Fund.

References

[1] D.P. Faith, "Conservation Evaluation and Phylogenetic Diversity," *Biological Conservation,* vol. 61, pp. 1-10, 1992.
[2] D.P. Faith and A.M. Baker, "Phylogenetic Diversity (PD) and Biodiversity Conservation: Some Bioinformatics Challenges," *Evolutionary Bioinformatics Online,* pp. 70-77, 2006.
[3] U. Feige, "A Threshold of $\ln n$ for Approximating Set Cover," *J. ACM,* vol. 45, pp. 634-652, 1998.
[4] M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman, 1979.
[5] K. Hartmann and M. Steel, "Maximizing Phylogenetic Diversity in Biodiversity Conservation: Greedy Solutions to the Noah's Ark Problem," *Systematic Biology,* vol. 55, pp. 644-651, 2006.
[6] D. Hochbaum, "Approximating Covering and Packing Problems: Set Cover, Vertex Cover, Independent Set, and Related Problems," *Approximation Algorithms for NP-Hard Problems.* PWS, 1997.
[7] S. Khuller, A. Moss, and J. Naor, "The Budgeted Maximum Coverage Problem," *Information Processing Letters,* vol. 70, pp. 39-45, 1999.
[8] C. Moritz and D.P. Faith, "Comparative Phylogeography and the Identification of Genetically Divergent Areas for Conservation," *Molecular Ecology,* vol. 7, pp. 419-429, 1998.
[9] V. Moulton, C. Semple, and M. Steel, "Optimizing Phylogenetic Diversity under Constraints," *J. Theoretical Biology,* vol. 246, pp. 186-194, 2007.
[10] F. Pardi and N. Goldmann, "Species Choice for Comparative Genomics: Being Greedy Works," *PLoS Genetics 1,* p. e71, 2005.
[11] F. Pardi and N. Goldman, "Resource-Aware Taxon Selection for Maximising Phylogenetic Diversity," *Systematic Biology,* vol. 56, no. 3, pp. 431-444, June 2007.
[12] A.S.L. Rodrigues and K.J. Gaston, "Maximising Phylogenetic Diversity in the Selection of Networks of Conservation Areas," *Biological Conservation,* vol. 105, pp. 103-111, 2002.
[13] A.S.L. Rodrigues, T.M. Brooks, and K.J. Gaston, "Integrating Phylogenetic Diversity in the Selection of Priority Areas for Conservation: Does It Make a Difference," *Phylogeny and Conservation,* A. Purvis, J.L. Gittleman, and T. Brooks, eds., Cambridge Univ. Press, 2005.
[14] C. Semple and M. Steel, *Phylogenetics.* Oxford Univ. Press, 2003.
[15] T.B. Smith, K. Holder, D. Girman, K. O'Keefe, B. Larison, and Y. Chan, "Comparative Avian Phylogeography of Cameroon and Equatorial Guinea Mountains: Implications for Conservation," *Molecular Ecology,* vol. 9, pp. 1505-1516, 2000.
[16] M. Steel, "Phylogenetic Diversity and the Greedy Algorithm," *Systematic Biology,* vol. 54, pp. 527-529, 2005.

Magnus Bordewich received the MMath degree and the DPhil degree in mathematics from Oxford University in 1998 and 2003, respectively. He was a postdoctoral research fellow in the Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand, and, then, in the School of Computer Science, Leeds University, Leeds, United Kingdom. In 2006, he joined Durham University, where he is currently a lecturer in the Department of Computer Science and has recently been awarded an EPSRC postdoctoral fellowship in theoretical computer science, researching on randomized algorithms and approximation in phylogenetics.

Charles Semple received the BSc degree (Hons) in mathematics from Massey University and the MSc and PhD degrees in mathematics from Victoria University of Wellington. Initially a postdoctoral fellow, since 2001, he has been a permanent staff member at the University of Canterbury, where he is currently an associate professor in the Department of Mathematics and Statistics. He was a visiting research fellow in Merton College, University of Oxford, in 2003, a visiting professor at the Université de Montpellier II in 2005, and a visiting fellow in the Isaac Newton Institute for Mathematical Sciences, University of Cambridge, in 2007. His research interests include combinatorics, computational complexity, and computational biology.