NATURE RESERVE SELECTION PROBLEM: A TIGHT APPROXIMATION ALGORITHM

MAGNUS BORDEWICH[1] AND CHARLES SEMPLE[2]

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 conserve so that the phylogenetic diversity of the set of species contained within those regions is maximized. Recently, it was 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 in Moulton *et al.*

Date: 3 July 2007.

1991 *Mathematics Subject Classification.* 05C05; 92D15.

Key words and phrases. Combinatorial algorithms, phylogenetic diversity, biodiversity conservation.

The first author was supported by the EPSRC, while the second author was supported by the New Zealand Marsden Fund.

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 to conserve diversity:

> Although conservation action is frequently targeted towards 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 Faith (1992) [1], phylogenetic diversity is a prominent quantitative tool for measuring the biodiversity of a collection of species. This measure is based on the evolutionary distance amongst 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 phylogenetic diversity (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 that maximizes the PD score amongst 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, as well as 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 preserve that 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 problem (BNRS) and generalizes the analogous unit cost problems described in [9, 11, 12, 13]. Allowing the cost of conserving each region to vary provides additional cost structure that is important in practice but which, as commented in [2, 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, 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 P=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 which has at least a $(1 - 1/e)$ fraction ($\approx 63\%$) of the phylogenetic diversity of the optimal solution. Moreover, this approximation ratio is the best possible.

The 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 the paper, we restrict ourselves to phylogenetic diversity in the setting of unrooted trees. However, in the last section of the paper, we extend our earlier results to the rooted analogue of BNRS. The notation and terminology in the 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 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 non-negative real-valued edge weights, where $X = \{a, b, c, d, e, f, g\}$.

For a subset S of X, the *phylogenetic diversity* (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 non-negative (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.

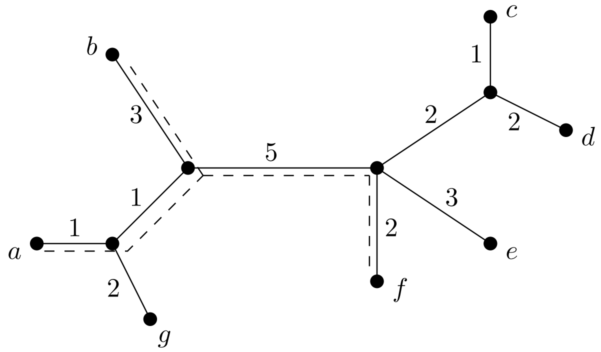

FIGURE 1. A phylogenetic X-tree with edge lengths, where $X = \{a, b, c, d, e, f, g\}$.

Question: Find a subset $\mathcal{A}'$ of $\mathcal{A}$ that 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 that 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 $\mathcal{T}$ be 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 be 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 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 $c(\{b\}) + c(\{f, c\}) + c(\{c, d\}) + c(\{e, g\}) = 4 + 8 + 6 + 5 = 23$ and

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

The problem BNRS 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. Lastly, BNRS is the "$0 \xrightarrow{c_r} 0/1$ Nature Reserve Problem" briefly discussed in the appendix in [11].

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 of size at most two that 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. The 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, amongst all available sets, the ratio of incremental diversity to cost is maximized and we keep within 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.

```
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 $\mathcal{G}$
```

FIGURE 2. The greedy algorithm Greedy.

```
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 $H_2$
```

FIGURE 3. The approximation algorithm ApproxBNRS.

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* P=NP.

In terms of the running time 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 in the special case that all regions have the same cost, this term can be removed from the running time. In this situation, the greedy algorithm starting from a subset $\mathcal{G}_0$ of $\mathcal{A}$ of size two that maximizes the PD score amongst all 2-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 $\mathcal{A}$ that 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 $\mathcal{S}_{\mathrm{opt}}$ such that S_1 and S_2 are chosen to maximize $PD(S_1 \cup S_2)$ amongst all subsets of $\mathcal{S}_{\mathrm{opt}}$ of size two, and S_3 maximizes $PD(S_1 \cup S_2 \cup S_3)$ amongst all sets in $\mathcal{S}_{\mathrm{opt}} \backslash \{S_1, S_2\}$. 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 $\mathcal{S}_{\mathrm{opt}} - \mathcal{G}_0$ 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.

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

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

Proof. One crucial point to observe in order for the approach of [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 non-empty, 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 $B - c(\mathcal{G}_0)$,

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

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

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

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

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 induction (for the second inequality), we have

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

This completes the proof of the lemma. □

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

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

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

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

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 Eqn (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

(3)

$$PD(\mathcal{G}_{l+1}) - PD(\mathcal{G}_l) \leq PD(S_1 \cup S_2 \cup S_3) - PD(S_1 \cup S_2) \leq PD(\mathcal{G}_0)/3;$$

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

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

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 $\mathcal{A}' = \{A_1, A_2, \ldots, A_k\}$ of $\mathcal{A}$ of size k that 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 P=NP. Observing that BNRS is a generalization of Maximum k-Coverage (see below), it follows that no approximation algorithm can exist for BNRS with ratio better than $(1-1/e)$ unless P=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 set-up, 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. By [3], no such algorithm can exist for $\alpha = (1 - 1/e + \epsilon)$ unless P=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, $\mathcal{A} = \{A_1, A_2, \ldots, A_s\}$ 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 $n_1, n_2, \ldots, n_s$, the PD optimization problem is to find a subset X' of X of size k that contains, for all i, at least n_i species with attribute A_i

and maximizes the PD score amongst 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 non-negative 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 that 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 above, 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 words, 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 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 analogue of phylogenetic diversity. In this short section, we briefly describe how ApproxBNRS can be applied to the rooted analogue of BNRS 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 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 phylogenetic diversity* (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 $\mathcal{T}$. The rooted analogue of the Budgeted Nature Reserve Selection problem, denoted 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 non-negative (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 $\mathcal{A}$ that 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 in the following way. 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 $\lambda_\rho(e) = \lambda(e)$ for all $e \in E(\mathcal{T})$.

With the above set-up, 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 using 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 P=NP.

References

[1] Faith, D.P., 1992. Conservation evaluation and phylogenetic diversity. Biol. Conserv. 61, 1-10.

[2] Faith, D.P., Baker, A.M., 2006. Phylogenetic diversity (PD) and biodiversity conservation: some bioinformatics challenges. Evol. Bioinf. Online 2, 70-77.

[3] Feige, U., 1998. A threshold of $\ln n$ for approximating set cover. J. ACM 45, 634-652.

[4] Garey, M.R., Johnson, D.S., 1979. Computers and intractability: A guide to the theory of NP-completeness, Freeman, San Francisco, CA.

[5] Hartmann, K., Steel, M., 2006. Maximizing phylogenetic diversity in biodiversity conservation: greedy solutions to the Noah's Ark Problem. Syst. Bio. 55, 644-651.

[6] Hochbaum, D., 1997. Approximating covering and packing problems: set cover, vertex cover, independent set, and related problems. In: Hochbaum, D. (Ed.), Approximation Algorithms for NP-Hard Problems. PWS, Boston.

[7] Khuller, S., Moss, A., Naor, J., 1999. The budgeted maximum coverage problem. Inform. Process. Lett. 70, 39-45.

[8] Moritz, C., Faith D.P., 1998. Comparative phylogeography and the identification of genetically divergent areas for conservation. Mol. Ecol. 7, 419-429

[9] Moulton, V., Semple, C., Steel, M., 2007. Optimizing phylogenetic diversity under constraints. J. Theoret. Biol. 246, 186-194.

[10] Pardi, F., Goldmann, N., 2005. Species choice for comparative genomics: being greedy works. PLoS Genetics 1, e71.

[11] Pardi, F., Goldman, N., 2007. Resource aware taxon selection for maximising phylogenetic diversity. Syst. Biol., in press.

[12] Rodrigues, A.S.L., Gaston, K.J., 2002. Maximising phylogenetic diversity in the selection of networks of conservation areas. Biological Conservation 105, 103-111.

[13] Rodrigues, A.S.L., Brooks, T.M., Gaston, K.J., 2005. Integrating phylogenetic diversity in the selection of priority areas for conservation: does it make a difference In Purvis, A., Gittleman, J.L., Brooks, T. (Eds.), Phylogeny and Conservation. Cambridge University Press, Cambridge.

[14] Semple, C., Steel, M., 2003. Phylogenetics. Oxford University Press, Oxford.

[15] Smith T.B., Holder, K., Girman, D., O'Keefe, K., Larison, B., Chan, Y., 2000. Comparative avian phylogeography of Cameroon and Equatorial Guinea mountains: implications for conservation. Mol. Ecol. 9, 1505-1516.

[16] Steel, M., 2005. Phylogenetic diversity and the greedy algorithm. Syst. Biol. 54, 527-529.

[1]Department of Computer Science, Durham University, Durham DH1 3LE, United Kingdom

E-mail address: m.j.r.bordewich@durham.ac.uk

[2]Biomathematics Research Centre, Department of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand

E-mail address: c.semple@math.canterbury.ac.nz