Chapter 4

Finding the best tree by heuristic search

If we cannot find the best trees by examining all possible trees, we could imagine searching in the space of possible trees. In this chapter we will consider heuristic search techniques, which attempt to find the best trees without looking at all possible trees. They are, of their very nature, a bit \textit{ad hoc}. They also do not guarantee us to have found all, or even any, of the best trees.

The fundamental technique is to take an initial estimate of the tree and make small rearrangements of branches in it, to reach "neighboring" trees. If any of these neighbors are better, we consider them, and continue, attempting more rearrangements. Finally, we reach a tree that no small rearrangement can improve. Such a tree is at a local optimum in the tree space. However, there is no guarantee that it is a global optimum. Figure 4.1 shows the problem for the case of searching in two spatial coordinates. Trees are a rather different case, but tree space is difficult to depict in a diagram like this.

In the diagram, we are trying to maximize a quantity — trying to find the highest point on the surface. In the case of the parsimony criterion, we are actually trying to minimize the number of evolutionary changes of state. We can convert that into a maximization problem by simply placing a minus sign before the number of changes of state, so that 272 becomes $-272$. Or, alternatively, we could subtract it from a large number, so that 272 becomes $10,000 - 272 = 9,728$. Maximization of the resulting quantity will minimize the number of changes of state. It is easier to show the diagram as a maximization problem than as a minimization problem, as maxima are more visible than minima.

In this diagram, we imagine that we have started with a particular point on the surface and then looked at its four neighbors. One of them is higher, so we move to that point. Then we examine its neighbors. We continue this until we have climbed to the highest point on the "hill." However, as the diagram shows, this
strategy is incapable of finding another point, one that is in fact higher, but that is not located on the hill where we started. Strategies of this sort are often called the greedy algorithm because they seize the first improvement that they see.

In this chapter we will examine some of the different kinds of rearrangements that have been proposed. Many others are possible. The techniques are more the result of common sense than of using any mathematical techniques. Later in the chapter we will also discuss some sequential addition strategies used for locating the starting point of the search. In the next chapter we will discuss branch and bound methods, a search technique guaranteed to find all of the most parsimonious trees.

Although the discussion here will be cast in terms of parsimony, it is important to remember that exactly the same strategies and issues arise with the other criteria for inferring phylogenies, and heuristic search techniques are employed for them in much the same way.

**Nearest-neighbor interchanges**

*Nearest-neighbor interchanges (NNI)* in effect swap two adjacent branches on the tree. A more careful description is that they erase an interior branch on the tree, and the two branches connected to it at each end (so that a total of five branches are erased). This leaves four subtrees disconnected from each other. Four subtrees
A subtree is rearranged by dissolving the connections to an interior branch and reforming them in one of the two possible alternative ways:

Figure 4.2: The process of nearest-neighbor interchange. An interior branch is dissolved and the four subtrees connected to it are isolated. These then can be reconnected in two other ways.

can be hooked together into a tree in three possible ways. Figure 4.2 shows the process. One of the three trees is, of course, the original one, so that each nearest-neighbor interchange examines two alternative trees. In an unrooted bifurcating tree with \( n \) tips, there will be \( n - 3 \) interior branches, at each of which we can examine two neighboring trees. Thus in all, \( 2(n - 3) \) neighbors can be examined for each tree. Thus a tree with 20 tips has 34 neighbors under nearest-neighbor interchange.

There is some ambiguity about how greedy we ought to be. If we accept the first neighboring tree that is an improvement, that will not be as good a search...
method as looking at all $2(n - 3)$ neighbors and picking the best one, but it will be quicker. We could also imagine trying multiple trees tied for best and evaluating rearrangements on each of them. The most sophisticated heuristic rearrangement strategies retain a list of all trees tied for best, and rearrange all of them.

Figure 4.3 shows what the space of all 15 possible unrooted trees looks like for 5 species, where trees that are adjacent by nearest-neighbor interchange are connected. Figure 4.4 shows the numbers of changes of state that are required for the data in Table 1.1 for each of these trees. Each tree has 4 neighbors. It will
be a useful exercise for the reader to pick a random starting point on this graph, and try various variations on nearest-neighbor interchange, using the lines on the graph as a guide. Does the process always find the most parsimonious tree, which requires 8 changes of state?

**Subtree pruning and regrafting**

A second, and more elaborate, rearrangement strategy is *subtree pruning and regrafting* (SPR). This is shown in Figure 4.5. It consists of removing a branch from the tree (either an interior or an exterior branch) with a subtree attached to it. The subtree is then reinserted into the remaining tree in all possible places, each of which inserts a node into a branch of the remaining tree. In Figure 4.5 the 11-
Figure 4.5: Subtree pruning and regrafting (SPR) rearrangement. The places where the subtree could be reinserted are shown by arrows. The result of one of these reinsertions (at the branch that separates B and C from the other species) is shown.

species tree has a 5-species subtree removed, and it is inserted into the remaining tree of 6 species, in one of the 9 possible places. One of these is of course the original tree. In general, if a tree of $n_1 + n_2$ species has a subtree of $n_2$ species removed from it, there will be $2n_1 - 3$ possible places to reinsert it. One of these is the original location. In fact, considering both subtrees (the one having $n_1$ species and the
Finding the best tree by heuristic search

One having \( n_2 \) species, there are \((2n_1 - 3 - 1) + (2n_2 - 3 - 1) = 2n - 8\) neighbors generated at each interior branch. It is also not hard to show that when an exterior branch is broken, there are \(2n - 6\) neighbors that can be examined. Thus, as there are \( n \) exterior branches on an unrooted bifurcating tree and \( n - 3 \) interior branches, the total number of neighbors examined by SPR will be \( n(2n - 6) + \)

---

**Figure 4.6: Tree bisection and reconnection (TBR).** A branch is broken and the two tree fragments are reconnected by putting in branches between all possible branches in one and all possible branches in the other. One of these reconnections and its result are shown here.
\[(n-3)(2n-8) = 4(n-3)(n-2).\] However, some of these may be the same neighbor (to see that, consider \(n = 4\)). For the tree shown in Figure 4.5, which has \(n = 11\), there are thus up to 304 neighbors under SPR. Of course, \(2(n-3) = 16\) of them are the same neighbors that NNI examines. But it is clear simply from the numbers that SPR carries out a much wider search and is thus more likely to find a better peak in the space of all trees.

The issues of how greedy to be, whether to delay accepting a new tree until all SPR rearrangements have been examined, and how many tied trees to retain as the basis for further rearrangement, arise for SPR just as they do for NNI.

**Tree bisection and reconnection**

*Tree bisection and reconnection (TBR)* is more elaborate yet. An interior branch is broken, and the two resulting fragments of the tree are considered as separate trees. All possible connections are made between a branch of one and a branch of the other. One such rearrangement is shown in Figure 4.6. If there are \(n_1\) and \(n_2\) species in the subtrees, there will then be \((2n_1 - 3)(2n_2 - 3)\) possible ways to reconnect the two trees. One of these will, of course, be the original tree. In this case there is no general formula for the number of neighbors that will be examined. It depends on the exact shape of the tree. For the 11-species tree in Figure 4.6 (which is the same one shown in Figure 4.5), for the interior branches there can be up to 296 neighbors that will be examined. As in the other types of rearrangement, there are issues of greediness and of how many tied trees to base rearrangement on. Allen and Steel (2001) calculate how many neighbors there will be under TBR and SPR rearrangement, and calculate bounds on the maximum number of these operations needed to reach any tree from any other.

**Other tree rearrangement methods**

*Tree-fusing*

The NNI, SPR, and TBR methods hardly exhaust the possible tree rearrangement methods. The repertoire of rearrangement methods continues to expand. Goloboff (1999) has added two additional rearrangement methods. One is *tree-fusing*. This requires two trees that have been found to be optimal or nearly so, and alters them by exchanging subgroups between the two trees. This requires that both trees have a subtree on them that contains the same list of species. Thus if one tree has on it the subtree \(((D,F),(G,H))\) and another the subtree \(((D,G),(F,H))\) one could swap the subtrees. Each tree would thus propose to the other a particular resolution of that four-species group. The proposals would be expected to be better than random resolutions of that group, as they were found by heuristic search on that tree. They thus become candidates of particular interest for resolving the same group on other trees.