Algorithm A.9 Branch and bound algorithm

Procedure Branch-and-Bound (
  score, // Score function
  bound, // Upper bound function
  σ_{best}, // Best full assignment so far
  $\text{score}_{\text{best}}$, // Best score so far
  i, // Variable to be assigned next
  $\boldsymbol{y}_{1\ldots i-1}$, // Current partial assignment
) // Recursive algorithm, called initially with the following arguments: some arbitrary full assignment σ_{best}, $\text{score}_{\text{best}} = \text{score}(\sigma_{\text{best}})$, $i = 1$, and the empty assignment.
  for each $x_i \in Val(X_i)$
    $\boldsymbol{y}_{1\ldots i} \leftarrow (\boldsymbol{y}_{1\ldots i-1}, x_i)$ // Extend the assignment
    if $i = n$ and $\text{score}(\boldsymbol{y}_{1\ldots n}) > \text{score}_{\text{best}}$ **then**
      $(\sigma_{\text{best}}, \text{score}_{\text{best}}) \leftarrow (\boldsymbol{y}_{1\ldots n}, \text{score}(\boldsymbol{y}_{1\ldots n}))$
        // Found a better full assignment
    else if $\text{bound}(\boldsymbol{y}_{1\ldots i}) > \text{score}_{\text{best}}$ **then**
      $(\sigma_{\text{best}}, \text{score}_{\text{best}}) \leftarrow$ Branch-and-Bound$(\text{score}, \text{bound}, \sigma_{\text{best}}, \text{score}_{\text{best}}, i+1, \boldsymbol{y}_{1\ldots i})$
        // If bound is better than current solution, try current partial assignment; otherwise, prune and move on
  return $(\sigma_{\text{best}}, \text{score}_{\text{best}})$

early. The better the current assignment σ_{best}, the better we can prune suboptimal trajectories. Other heuristics intelligently select, at each point in the search, which variable to assign next, allowing this choice to vary across different points in the search. When available, one can also use a lower bound as well as an upper bound, allowing pruning to take place based on partial (not just full) trajectories. Many other extensions exist, but are outside the scope of this book.

A.5 Continuous Optimization

In the preceding section, we discussed the problem of optimizing an objective over a discrete space. In this section we briefly review methods for solving optimization problems over a *continuous* space. See Avriel (2003); Bertsekas (1999) for more thorough discussion of nonlinear optimization, and see Boyd and Vandenberghe (2004) for an excellent overview of convex optimization methods.

A.5.1 Characterizing Optima of a Continuous Function

At several points in this book we deal with maximization (or minimization) problems. In these problems, we have a function $f_{\text{obj}}(\theta_1, \ldots, \theta_n)$ for several *parameters*, and we wish to find joint values of the parameters that maximizes the value of f_{obj}.

Formally, we face the following problem:

Find values $\theta_1, \ldots, \theta_n$ such that $f_{\text{obj}}(\theta_1, \ldots, \theta_n) = \max_{\theta'_1, \ldots, \theta'_n} f_{\text{obj}}(\theta'_1, \ldots, \theta'_n)$.