The General Branch and Bound Algorithm

- **Each solution is assumed to be expressible as an array X[1:n] (as was seen in Backtracking).**
- **A predictor, called an approximate cost function CC, is assumed to have been defined.**
- **Definitions:**
  - **A live node is a node that has not been expanded**
  - **A dead node is a node that has been expanded**
  - **The expanded node (or E-node for short) is the live node with the best CC value.**
- **The general B&B algorithm follows:**

```
Procedure B&B()
begin
    E: nodepointer;
    E := new(node);    -- this is the root node which
                       -- is the dummy start node
    H: heap;           -- A heap for all the live nodes
    -- H is a min-heap for minimization problems,
    -- and a max-heap for maximization problems.
   while (true) do
      if (E is a final leaf) then
         -- E is an optimal solution
         print out the path from E to the root;
         return;
      endif
      Expand(E);
      if (H is empty) then
        report that there is no solution;
        return;
      endif
      E := delete-top(H);
   endwhile
end
```

```
Procedure Expand(E)
begin
  - Generate all the children of E;
  - Compute the approximate cost value CC of each child;
  - Insert each child into the heap H;
end
```