Topic 5.3 Improving the Performance of the MINIMAX search Strategy

A. Common Factors:

1) Database of experience like ‘best opening move’, ‘great winning [/loosing] combination’, …

2) Maintaining transposition table that marks repeated states

3) Forward pruning, that excludes equivalent states

4) Singular expansion, where only ‘clearly better’ moves are considered

B. Using 'cutoff test' and heuristic evaluation function

i) To have a reduced search space 'cutoff tests' are applied in place of 'terminal tests':

- Simple: Set a fixed depth; Not so good.
- Sophisticated: quiescence search; [A quiescent state does not exhibit wild swing in near future.]

ii) To apply on states at cutoff depths, utility functions are replaced with heuristic functions, which return expected utility of states. [In Chess, material values like pawn = 1, knight/bishop = 3, queen = 9, etc. are used.]

C. Using alpha-beta pruning:

- Cutting off in compliance with already calculated minimax values, that is, values of
  - best choice for the maximizing player, α,
  - best choice for the minimizing player, β.

- Example:

Say, in absence of true minimax values, we are given the following sequence of numbers from which we are supposed to assign a value to each newly generated 'terminal' state, that is, a state at the cutoff depth:

4, 3, -1, 4, 5, 2, 1, -2, -5, 3, 2, 1. [-5, +5]

Further:
Branching factor = 2;
Cutoff depth = 2 moves or 4 plies;
MAX makes the opening move;
Left to right expansion of the tree.

✓Try to prune a branch in every occasion and go up;
✓ To go up, one must know at least the range of all the siblings, and it is not enough.

……

4, 3, -1, 4, 5, 2, 1, -2, -5, 3, 2, 1.