Consider an expert to be each method of making a decision in chess. Let us call it $m_i(S), i=1, ..., n$, where you have $n$ different methods. How to combine experts to build a better expert? How to learn what weights to place on each of the experts? Our proposed approach is a multiscale-recursive method that group pairs of evaluation functions at each iteration.

First, let us write down different evaluation functions. If you want, you can add new ones or not use the ones proposed below (make sure to consider at least 6 different evaluation functions). There will be bonus for the top 3 checker programs on the checker competition (gold, silver, bronze).

Suggested evaluation functions. Please describe which ones you use in your homework. Feel free to choose others, but say which ones you use.

/* 1: Single Pieces Count. Sum total number of your pieces (subtract from adversary) */
/* 2: King Pieces Count. Sum total number of your king pieces (subtract from opponent king pieces) */
/* 3: Defense. How well defended are the pieces? For red defense, sum total number of red pieces at previous neighbors (n3, n4) to a red piece. For white defense sum total number of white pieces at previous neighbors (n1, n2) to a white piece. You can take the difference of both. */
/* 4: Defense against kings. For red defense, sum total number of red pieces post neighbors (n1,n2) to a red piece. Multiply by the number of white kings. For white, sum total number of white pieces post neighbors (n1, n2) to a white piece and multiply by the number of red kings. */
/* 5: Dynamic defense on sides. Sum the total value of piece coordinate “b”, a formula that only values pieces if b=1 or b=8 (b=0 or b=7), by only counting these states. */
/* 6: Dynamic position. Count number of moves by pieces (to get a fast computation, do not include jumps, i.e., just check for each piece if a neighbor spot is empty or not. Compare with opponents. */

/* 7: Offense Single Pieces: count how advanced are your single pieces (how close to become a king). For example, the sum of “a” component for all your single pieces and subtract from adversary. Well if you are moving forward in “a” you sum “a”, otherwise, if you are moving backwards use sum (8-“a”). */

1. Let us see which methods are good methods.

**Graph1.** First check that methods used are good methods. Run checkers games between each method $m_i$ against a method that simply chooses randomly the successor. For each method $m_i$ run four games: two games the random method plays with white and two games the random method plays red. For the methods that do win against the random method, report on the average number of moves method $m_i$ takes to win (average over four games). Make sure you have at least four methods that do win against a random player. Plot a graph where the x-axis represent each method $i=1, ..., n$, that is a “good” (winning method) and the y-axis represent the number of moves in average method $m_i$ takes to win against the random method. Well, to make a graph where larger numbers means better, plot the difference: $y_i = 100 - \text{“average number of moves by method $y_i$”}$. In this way the larger is this value the fewer moves in average it took a method to win. Finally, to look like a probability, divide each value $y_i$ by the sum of all $\sum y_i$. In this way you can interpret the normalized $y_i$ value as a probability of a method to be the best method. Comment: Decisions made by scaling a method $m_i(S)$ by a factor “s”, i.e., by choosing a new method “$M_i= s m_i(S)$”, will not affect the outcome of plays during the game. Decisions are made by simply ordering the list of candidate moves according to the evaluation criteria. If the evaluation criteria is scaled by “s” the order of the moves is still the same and thus the decisions are not affect by it.

2. Let us choose two different evaluation functions that are close in “concept”. Say evaluations 1 & 2 and we refer to them as $m_1$ and $m_2$.

Run two checkers games between these two experts, $m_1$ and $m_2$. One game $m_1$ plays with white and one with red. If there is a tie, choose the winner, $m_w$, as being the one that won in less moves, the other call it looser, $m_L$. Now combine the two methods as follows

$$M_{wL}(S) = m_w(S) + \alpha m_L(S).$$

Note that one parameter is sufficient to combine two methods, since a general scale parameter will not affect any decision as discussed above. How to choose a good $\alpha$?
Play two games between $m_w + \alpha_1 m_L$ and $m_w + \alpha_2 m_L$ (each game alternates color), where you arbitrary choose the values $\alpha_1 < \alpha_2$. If there is a tie, break it based on number of moves. So keep track of the number of moves of each game. If $\alpha_1$ is the winner, choose a new value $\alpha_2 < \alpha_1$ and play again. For example $\alpha_2 = \alpha_1 / 2$. If $\alpha_2$ is the winner choose a new value $\alpha_1 > \alpha_2$ and play again. For example, $\alpha_1 = 2 \alpha_2$. Do this procedure recursively a few times to select a parameter $\alpha$. Which values $\alpha_1 < \alpha_2$ to choose the start? Suggestion to start with $\alpha_1 = 0$ (start with the winner method, $m_w$) and $\alpha_2 = 1$ (or some scale related to the values used by the methods, $m_1$ and $m_2$). Verify that the final value of $\alpha$ is different than zero (improvement was achieved). In this way you grouped two methods that have the same concept.

3. Same as in the previous one, b, but now for the concepts 3 & 4. Also do for 5 & 6. Refer to this new methods as $M_{12}$ and $M_{34}$ and $M_{56}$.

4. Combine $M_{34}$ and $M_{56}$ and refer to it as $M_{3456}$. Report on the new parameter $\alpha$.

5. Combine $M_{12}$ and $M_{3456}$ and refer to it as $M_{123456}$. Report on the new parameter $\alpha$. This is your expert checker player.