To compute optimal policies in Markov Decision Processes (MDP), a good approach is to use Dynamic Programming. Dynamic Programming algorithms uses Bellman's equations to improve the value functions, thus increasing accuracy of the approximated values.

First, the policies will be evaluated, this process is also called prediction. This means computing the value functions of each policy using Bellman's Equation as update rule. Then the policies of each state will improve according to the value functions computed previously. By iterating the newly updated policies, improvements will still appear until it is already the optimal policy. If policy evaluation terminates abnormally, an alternative method is to do value iteration instead of evaluating the policy itself.

However, Dynamic Programming works on every state of MDP as a set. A worst case scenario is if a set is very large, like a chessboard for example, the computational cost is significantly large and takes a very long time. Asynchronous Dynamic Programming is introduced to solve such problems. It works with the states not as a set, but instead it updates the value of the states in-place ignoring the orders. Also, in the process of computation, the unimportant states will be ignored for the future, thus effectively reduces costs to compute the optimal policy.