Monte Carlo is one learning method suitable under the condition of no complete knowledge of the environment. The agent learns from experiences, the sample interactions with the environment. Much like a bandit problem, Monte Carlo sample the returns from each state-action and average the rewards, while Monte Carlo deals with multiple states (multiple bandit problems). Since the action selections require learning, they are considered nonstationary problems, and thus policy & value iterations are performed to compute value functions, just like Dynamic Programming.

There are few advantages of Monte Carlo over Dynamic Programming: The ability to learn from actual experiences, simulated experiences, and the simulation of sample episodes only from the states of interest, greatly reduces computational costs. However, by only evaluating states of interest, some state-action will not be observed, which should be avoided since the purpose of reinforcement learning is to determine an optimal action among all actions from each state. To do so, each action should be set as a starting point by a probability of a non-zero value. By applying exploring starts to policy/value iteration and policy improvement, the optimal policy will still be computed while without obtaining any environmental knowledge.

The above method is an example of on-policy method. On-policy methods estimate the value of a policy while using it as control. On-policy methods are usually simpler and therefore considered first. For off-policy method, a simple approach is to use two policies: one for optimal (target policy) and one for exploration (behavior policy). The separation of the two policies allows the target policy to be used as decision, while using the behavior policy to sample different actions that might be missed. One problem of using behavior policy is if non-greedy actions are selected commonly, the process of learning is very slow. Even though off-policy methods generate data from another policy, thus it has a large variance and slower to converge to a specific value. But off-policy methods generally have broader applications than on-policy methods.