在强化学习任务中，如果 reward function 和状态转移模型是确定的，那么我们要找的最优 policy 也是确定的。但由于 reward 往往是稀疏的，同时状态空间往往很大，导致我们需要很长的时间才能探索到最优 policy。为了加速这个探索的过程，我们想要人为设计一些附加 rewards（在环境本身的 reward 之外），这些 rewards 一般是在直觉上能够引导 agent 到达更好的状态、离目标更近。那么我们想要知道的是，我们有怎样的自由修改 reward function，使得最优 policy 保持不变。

为了探究这个问题，我们先看一个具体的例子，对于一个让自行车自动驶向目的地的任务，如果我们奖励自行车靠近目的地的动作:如果 s'比 s 离终点更近，定义 reward F(s,a,s')>0，否则 F=0。那么自行车会陷入一个状态序列的循环 s1→s2→s3→s1…，因为 F(s1,a,s2)+F(s2,a,s3)+F(s3,a,s4)>0。也就是说自行车最后会选择在一个区域转圈，因为它会不断得到正的得分，而不是直接到达目的地。这是一个错误设计附加 reward 的样例，因为它得到了错误的最优 policy。

简单起见，我们先对任务先做一些限定，首先 MDP 的状态是有限的，然后我们假设系统的 reward 是不具有随机性的，也就是说 R 是一个有限（bounded）的函数 R(s,a)，为了和附加 reward 的形式对齐，我们可以将它写成更一般化的形式 R(s,a,s')，这样 reward 可以和 next state 有关，也可以只取决于 action；折扣参数 γ≤1，当 γ=1 时，任务是一个 undiscounted MDP，我们假设存在一个 terminal state s_0，并且对于任何的 policy 都会以 1 的概率转移到 s_0（all policy are proper），这其实是对状态转移模型 T 的限制，本文假设 MDP 是满足该限制的（T is proper）。对于 discounted MDP，没有 terminal state。

在原始 MDP M 中 reward function 为 R(s,a,s')，加入 reward F(s,a,s')后，新的 MDP M'的 reward function 为 R'=R+F。M 和 M'有相同的状态、动作空间，有相同的转移概率。那么，F 是什么形式可以保证在 M'得到的最优 policy 也是 M 中的最优 policy 呢？

受启发于上面自行车的例子，我们知道需要给自行车靠近目的地的 action 奖励的同时，要惩罚远离目的地的 action，使得 F(s1,a,s2)+F(s2,a,s3)+F(s3,a,s4)=0。于是我们设计了一种 potential-based shaping function F：Φ 是一个实值函数：S→R，可以看做每个状态定义了一个势能，则 F 为势能差：

$$F(s,a,s') = \gamma\Phi(s') - \Phi(s)$$

这种形式的 shaping 是 M'与 M 最优 policy 一致的充分必要条件，我们主要看充分性：

$$Q^*_M(s,a) = \mathrm{E}_{s'\sim P_{sa}(\cdot)}\left[R(s,a,s') + \gamma \max_{a'\in A} Q^*_M(s',a')\right]$$

$$Q^*_M(s,a) - \Phi(s) = \mathrm{E}_{s'}\Big[R(s,a,s') + \gamma\Phi(s') - \Phi(s) + \gamma \max_{a'\in A}\left(Q^*_M(s',a') - \Phi(s')\right)\Big]$$

定义 $\hat{Q}_{M'}(s,a) \triangleq Q^*_M(s,a) - \Phi(s)$，并且利用 $F(s,a,s') = \gamma\Phi(s') - \Phi(s)$：

$$\begin{aligned}&\hat{Q}_{M'}(s,a)\\&\quad=\mathrm{E}_{s'}\left[R(s,a,s')+F(s,a,s')+\gamma\max_{a'\in A}\hat{Q}_{M'}(s',a')\right]\\&\quad=\mathrm{E}_{s'}\left[R'(s,a,s')+\gamma\max_{a'\in A}\hat{Q}_{M'}(s',a')\right]\end{aligned}$$

由于上式满足 M'下的贝尔曼最优方程，$Q^*_{M'}(s,a) = Q_{M'}(s,a)$。因此：

$$\begin{aligned}\pi^*_{M'}(s) &\in \arg\max_{a\in A} Q^*_{M'}(s,a)\\&= \arg\max_{a\in A} Q^*_{M}(s,a)-\Phi(s)\\&= \arg\max_{a\in A} Q^*_{M}(s,a)\end{aligned}$$

以上证明了使用 potential-based reward function 可以保证最优 policy 一致，我们现在进一步研究这种 reward function 的性质。首先在上面的证明中我们知道了：

$$Q^*_{M'}(s,a) = Q^*_{M}(s,a)-\Phi(s)$$

用类似的证明方法可知，对于最优 policy 下的 value function，也存在：

$$V^*_{M'}(s) = V^*_{M}(s)-\Phi(s)$$

实际上，对于任意的 policy（不一定是最优 policy），都存在 $V^{\pi}_{M'}(s)=V^{\pi}_{M}(s)-\Phi(s)$。这意味这在整个学习的过程中，M'下 policy 的好坏，都同样反映在 M 中。M'下近似最优的 policy，在 M 中也是近似最优的。

为什么我们要选择一个和 action 无关的函数 F 呢，这是因为如果在原始 MDP 下我们得到了某个 policy 的 Q value，Q(s,a)>Q(s,b)，此时如果为 action a、b 定义了不同的附加 reward，那么可能会造成 Q'(s,a)<Q'(s,b)，这样两个 MDP 会产生不同的 policy。而如果只是为 Q(s,a)、Q(s,b)同时加上一个函数 Φ(s)，则不会影响它们的相对大小。

事实上，如果定义 F 为一个 MDP 的 reward function，那么任何 policy 都是最优的，也就是说 F 对任何 policy 都没有倾向，此时 Q(s,a)= -Φ(s)，是从 s 到 s_0 的势能差（$\Phi(s_0)=0$），在任何 policy 下都相同。F 作为附加 policy 自然也就不会产生错误的最优 policy。

最后考虑如何确定 Φ(s)，Φ(s)的一个好的选择是 $V^*_M(s)$，此时 $V^*_{M'}(s)$是 0，$Q^*_{M'}(s,a) = A^*_M(s,a)$（代入前两个公式），此时可以很快学到 Q 和最优 policy。在实际操作的时候，我们没有 $V^*_M(s)$，所以可以用已有的知识去估计各 state 的好坏，作为 $V^*_M(s)$的近似，这同样对训练速度有提升，而且有正确性的保证。

注意以上证明都是针对有限状态的情况，无限状态时，我们要求 Φ 是有限的（bounded），从而 reward 是有限的。

当我们应用 potential-based shaping 进行 Q-learning 时，有文章指出，这其实相当于用 Φ(s)对 Q-value 进行初始化，然后再进行正常的训练。也就是说，如果定义两个 agent，第一个初始化为$Q(s,a)=Q_0(s,a)$，然后用 potential-based shaping reward F 辅助训练；第二个初始化为$Q'_0(s,a)=Q_0(s,a)+\Phi(s)$，然后只用系统本身的 reward 进行训练，两个 agent 使用 advantage-based policy 选择 action，效果是

完全一样的。这侧面反映了对 Q value 初始化的重要性。直观上来看，agent 的任务是要找到一个 policy，使得它可以尽快的达到 state 空间的目标区域，如果 Q value 都用最优 state value 初始化，（相当于已知 state 的好坏，只要去学习不同 action 的相对好坏），那么 agent 可以很快的达到目标区域。