例子：假设程序f的输入域划分为4个不相交的分区c_1, c_2, c_3, c_4，每个分区中有k_1, k_2, k_3, k_4个测试用例。初始化 MAPT 和 RAPT 的两个参数$\gamma = \epsilon = 0.005$, $\tau = \delta = 0.001$。根据以往的测试经验或者测试历史，不妨设置每个分区的惩罚上限$Pun_i = 10 \ \ (i = 1,2,3,4)$。

根据 MAPT 算法步骤，测试程序 f 的过程如下：

测试任务开始前，设置初始测试剖面$\mathrm{tf} = \{< c_1, 0.25 >, < c_2, 0.25 >, < c_3, 0.25 >, < c_4, 0.25 >\}$，初始状态转移矩阵如下：

$$\mathrm{P} = \begin{pmatrix} 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$

第一次测试时，根据tf选择分区，不妨假设c_1被选中，然后从c_1中随机选择测试用例t并执行。

- 如果 t 揭示了软件故障，根据公式 6 和 7，调整状态转移矩阵的第一行，结果如下：

$$\mathrm{P} = \begin{pmatrix} 0.25123 & 0.24959 & 0.24959 & 0.24959 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$

- 如果 t 没有揭示软件故障，根据公式 8 和 9，调整状态转移矩阵的第一行，结果如下：

$$\mathrm{P} = \begin{pmatrix} 0.24975 & 0.25007 & 0.25007 & 0.25007 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$$

然后根据更新后的转移概率$p_{1j} (j = 1,2,3,4)$选取下一个测试用例所在的分区c_j，依据测试用例的执行结果更新状态转移矩阵的第j行。

根据 RAPT 算法步骤，测试程序 f 的过程如下：

测试任务开始前，设置初始测试剖面$\mathrm{tf} = \{< c_1, 0.25 >, < c_2, 0.25 >, < c_3, 0.25 >, < c_4, 0.25 >\}$，每一个分区的惩罚因子$pun_i = 0$，奖励因子$rew_i = 0$，其中$\mathrm{i} = 1,2,3,4$。

在测试过程中，根据tf选择分区，不妨假设c_1被选中，然后从c_1中随机选择测试用例t并执行。

- 如果 t 揭示了软件故障，令$rew_1 = rew_1 + 1 = 1$，$pun_1 = 0$。接下来在c_1中选择的第二个测试用例揭示了故障，则$rew_1 = rew_1 + 1 = 2$，第 3 个测试用例没

揭示故障，此时$rew_1 = 2$, $pun_1 = pun_1 + 1 = 1$。根据公式 10 和 11 调整测试剖面 $\mathrm{tf} = \{< c_1, 0.25846 >, < c_2, 0.24718 >, < c_3, 0.24718 >, < c_4, 0.24718 >\}$，然后令$rew_1 = 0$。

- 如果 t 没有揭示故障并且$pun_1 < 10$，根据公式 12 和 13 调整测试剖面$\mathrm{tf} = \{< c_1, 0.24900 >, < c_2, 0.25033 >, < c_3, 0.25033 >, < c_4, 0.25033 >\}$，然后令$pun_1 = pun_1 + 1$。
- 如果 t 没有揭示故障并且$pun_1 = 10$，认为c_1的失效率很低，根据公式 12 和 13 调整测试剖面 $\mathrm{tf} = \{< c_1, 0 >, < c_2, 0.33333 >, < c_3, 0.33333 >, < c_4, 0.33333 >\}$，然后令$pun_1 = 0$。（$c_1$选取概率不会一直是 0，原因是下一个选取分区中的测试用例没有揭示故障时，c_1选取概率增加。）

注意：在测试过程中，计算机的计算精度导致可能出现分区选取概率的和不等于 1 或者转移矩阵某一行的状态转移概率的和不等于 1 的情况。