1 Problem 4.12

Solution: According to the construction in the book, we ‘convert’ the asynchronous CDMA model to equivalent synchronous CDMA model, and we will compute $\Omega(b)$, which is the payoff function:

$$b^T H b = \sum_{j=1-MK}^{MK+K} \sum_{l=1-MK}^{MK+K} b_j b_l h_{j,l}$$

$$= \sum_{j=1-MK}^{MK+K} b_j \left[ A_{\kappa(j)}^2 b_j + 2 \sum_{l=j+1}^{j+K-1} b_l h_{i,j} \right]$$

$$= \sum_{j=1-MK}^{MK+K} b_j \left[ A_{\kappa(j)}^2 b_j + 2 \sum_{n=1}^{K-1} b_{j+n} h_{j+n,j} \right]$$

$$= \sum_{j=1-MK}^{MK+K} b_j \left[ A_{\kappa(j)}^2 b_j + 2 \sum_{n=1}^{K-1} b_{j+n} A_{\kappa(j)} A_{\kappa(j+n)} \rho_{\kappa(j),\kappa(j+n)} \right] \quad (1)$$
This comes from the property of symmetric matrix: \( \forall Q \in \mathbb{R}^{N \times N} \) and \( x \in \mathbb{R}^{N \times 1} \), we have:

\[
x^T Q x = \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij} x_i x_j \\
= \sum_{i=1}^{N} [Q_{ii} x_i^2 + \sum_{j=1,j \neq i}^{N} Q_{ij} x_i x_j] \\
= \sum_{i=1}^{N} [Q_{ii} x_i^2 + \sum_{j=1}^{i-1} Q_{ij} x_i x_j + \sum_{j=i+1}^{N} Q_{ij} x_i x_j] \\
\tag{2}
\]

We further note that:

\[
\sum_{i=1}^{N} \sum_{j=1}^{i-1} Q_{ij} x_i x_j = \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} Q_{ij} x_i x_j \\
= \sum_{j=1}^{N-1} \sum_{i=j+1}^{N} Q_{ij} x_i x_j \\
= \sum_{i=1}^{N} \sum_{j=i+1}^{N} Q_{ij} x_i x_j \text{ by interchanging } i \text{ and } j \\
= \sum_{i=1}^{N} \sum_{j=i+1}^{N} Q_{ij} x_i x_j \text{ by the symmetry of } Q \tag{3}
\]

Plug (3) into (2) we have:

\[
x^T Q x = \sum_{i=1}^{N} [Q_{ii} x_i^2 + 2 \sum_{j=i+1}^{N} Q_{ij} x_i x_j] \\
\tag{4}
\]

From equation (4) we can get equation (1)

Then we can compute \( \Omega(b) \):

\[
\Omega(b) = 2b^T A_M y - b^T H b \\
= 2 \sum_{j=1-MK}^{MK+K} A_{\kappa(j)} b_j y_j - \sum_{j=1-MK}^{MK+K} b_j [A_{\kappa(j)}^2 b_j + 2 \sum_{n=1}^{K-1} b_{j+n} A_{\kappa(j)} A_{\kappa(j+n)} \rho_{\kappa(j),\kappa(j+n)}] \\
= \sum_{j=1-MK}^{MK+K} A_{\kappa(j)} b_j [2y_j - A_{\kappa(j)} b_j - 2 \sum_{n=1}^{K-1} b_{j+n} A_{\kappa(j+n)} \rho_{\kappa(j),\kappa(j+n)}] \\
= \sum_{j=1-MK}^{MK+K} \mu_j (b_j, x_j) \tag{5}
\]

2
where

\[ \mu_j(b_j, x_j) = A_\kappa(j)b_j[2y_j - A_\kappa(j)b_j - 2\sum_{n=1}^{K-1} x_j(n)A_\kappa(j+n)\rho_{\kappa(j),\kappa(j+n)}] \] (6)

\[ x_j = (x_j(1), x_j(2), ..., x_j(K-1))^T \]
\[ = (b_{j+1}, b_{j+2}, ..., b_{j+K-1})^T \] (7)

\[ x_{j-1} = (b_j, b_{j+1}, ..., b_{j+K-2})^T \]
\[ = f(b_j, x_j) \] (8)

\[ x_{MK+K} = 0 \] (9)

To maximize \( \Omega(b) \) is equivalent to maximize \( \hat{\Omega}(b) = \sum_{j=1-MK}^{MK+K} \hat{\mu}_j(b_j, x_j) \), where:

\[ \hat{\mu}_j(b_j, x_j) = A_\kappa(j)b_j[y_j - \sum_{n=1}^{K-1} x_j(n)A_\kappa(j+n)\rho_{\kappa(j),\kappa(j+n)}] \] (10)

Then we can write down backward dynamic programming algorithm as follows:

\[ J_{MK+K}(x_{MK+K}) = 0 \] (11)

\[ J_{MK+K-1}(x_{MK+K-1}) = \max\{ \hat{\mu}_{MK+K}(1, x_{MK+K}), \hat{\mu}_{MK+K}(-1, x_{MK+K}) \} \] (12)

\[ J_{j-1}(x_{j-1}) = \max \{ J_j(x_{j-1}(2), x_{j-1}(3), ..., x_{j-1}(K-1), 1) + \hat{\mu}_j(x_{j-1}(1), x_{j-1}(2), ..., x_{j-1}(K-1), 1), \]
\[ J_j(x_{j-1}(2), x_{j-1}(3), ..., x_{j-1}(K-1), -1) + \hat{\mu}_j(x_{j-1}(1), x_{j-1}(2), ..., x_{j-1}(K-1), -1) \} \] (13)

As the same in the forward dynamic programming algorithm, since each level we need to calculate the \( J_j \)’s for \( 2^{K-1} \) states, so the time complexity per bit is \( O(2^K) \)

2 Problem 4.20

Solution: We first condition on \( b_2 \), so this becomes a simple binary hypothesis testing problem with the following hypotheses:

\[ H1 : y(t) = (1 + A_2b_2)^2s^2(t) + \sigma n(t) \] (14)

\[ H2 : y(t) = A_2^2b_2^2s^2(t) + \sigma n(t) \] (15)