$$\mathbf{H}=\mathbf{R}=\begin{bmatrix} 1 & \rho_{12} & 0 & & \cdots & & & 0 \\ \rho_{12} & 1 & \rho_{21} & 0 & \cdots & & & 0 \\ 0 & \rho_{21} & 1 & \rho_{12} & \ddots & & & \\ 0 & 0 & \rho_{12} & 1 & \ddots & 0 & & \\ & & & & \ddots & \rho_{\textcircled{21}} & 0 & 0 \\ & & \ddots & & & 1 & \rho_{\textcircled{12}} & 0 \\ & & & & 0 & \rho_{\textcircled{12}} & 1 & \rho_{\textcircled{21}} \\ 0 & & \cdots & & 0 & 0 & \rho_{\textcircled{21}} & 1 \end{bmatrix}. \qquad (4.28)$$

21 ↔ 12

The fact that the symmetric matrix **H** is zero except along the main diagonal and the first subdiagonal is crucial for decreasing the complexity of the receiver. It means that we can write the payoff function as

$$\Omega(\mathbf{b}) = \sum_{j=1-2M}^{2M+2} \lambda_j(b_{j-1}, b_j), \qquad (4.29)$$

where $b_{-2M} = 0$ and

$$\lambda_j(b_{j-1}, b_j) = \begin{cases} 2b_j y_j - 2\rho_{12} b_j b_{j\overset{-}{\swarrow}{+}1} - 1, & \text{if } j \text{ is even;} \\ 2b_j y_j - 2\rho_{21} b_j b_{j\underset{\leftarrow\,-}{+}1} - 1, & \text{if } j \text{ is odd.} \end{cases}$$

We see in (4.29) that the payoff function depends on each of its arguments sequentially, and only consecutive arguments are coupled. Consequently, to maximize (4.29), we can use the *dynamic programming* algorithm.

To explain the principle of operation of dynamic programming, we represent each possible sequence **b** as a path from the origin to the destination in the diagram in Figure 4.8. In this diagram we have labeled each arc with a "length." The payoff achieved by a given **b** is equal to the total length of its corresponding path. The dynamic programming algorithm is an efficient algorithm to find the longest (or the shortest) path from the origin to the destination in a layered graph. In a layered graph, nodes can