We construct a bipartite graph G with bipartition (X, Y), where $X = \{x_1, x_2, \ldots, x_n\}$, $Y = \{y_1, y_2, \ldots, y_n\}$, and x_i is joined to y_j if and only if worker X_i is qualified for job Y_j. The problem becomes one of determining whether or not G has a perfect matching. According to Hall's theorem (5.2), either G has such a matching or there is a subset S of X such that $|N(S)| < |S|$. In the sequel, we shall present an algorithm to solve the personnel assignment problem. Given any bipartite graph G with bipartition (X, Y), the algorithm either finds a matching of G that saturates every vertex in X or, failing this, finds a subset S of X such that $|N(S)| < |S|$.

The basic idea behind the algorithm is very simple. We start with an arbitrary matching M. If M saturates every vertex in X, then it is a matching of the required type. If not, we choose an M-unsaturated vertex u in X and systematically search for an M-augmenting path with origin u. Our method of search, to be described in detail below, finds such a path P if one exists; in this case $\hat{M} = M \,\Delta E(P)$ is a larger matching than M, and hence saturates more vertices in X. We then repeat the procedure with $\hat{M}$ instead of M. If such a path does not exist, the set Z of all vertices which are connected to u by M-alternating paths is found. Then (as in the proof of theorem 5.2) $S = Z \cap X$ satisfies $|N(S)| < |S|$.

Let M be a matching in G, and let u be an M-unsaturated vertex in X. A tree $H \subseteq G$ is called an *M-alternating tree rooted at u* if (i) $u \in V(H)$, and (ii) for every vertex v of H, the unique (u, v)-path in H is an M-alternating path. An M-alternating tree in a graph is shown in figure 5.11.

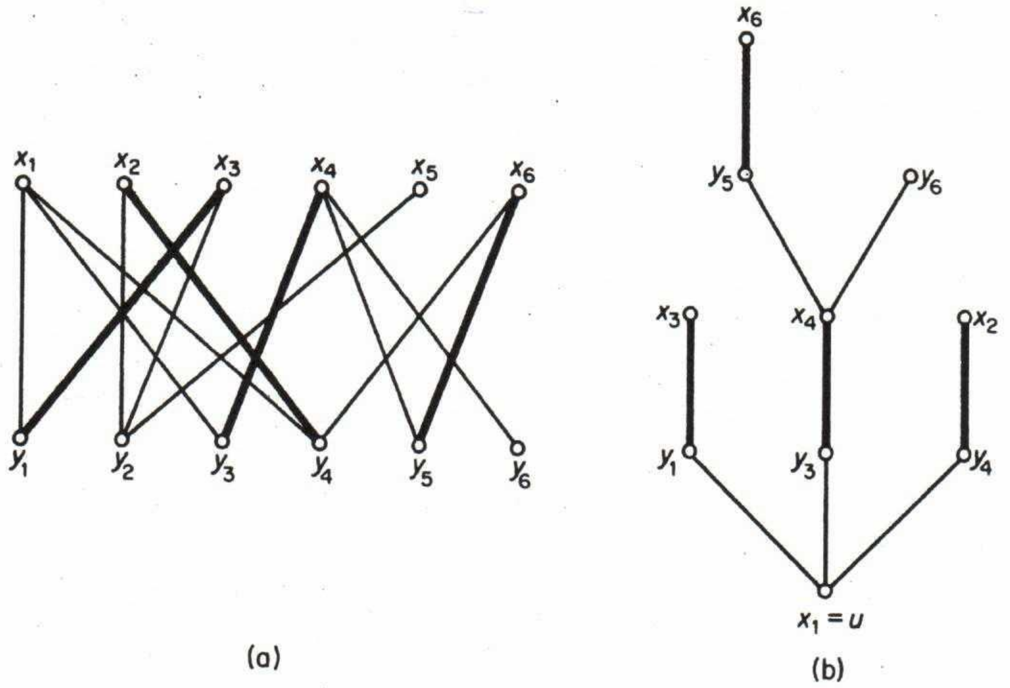

Figure 5.11. (*a*) A matching M in G; (*b*) an M-alternating tree in G

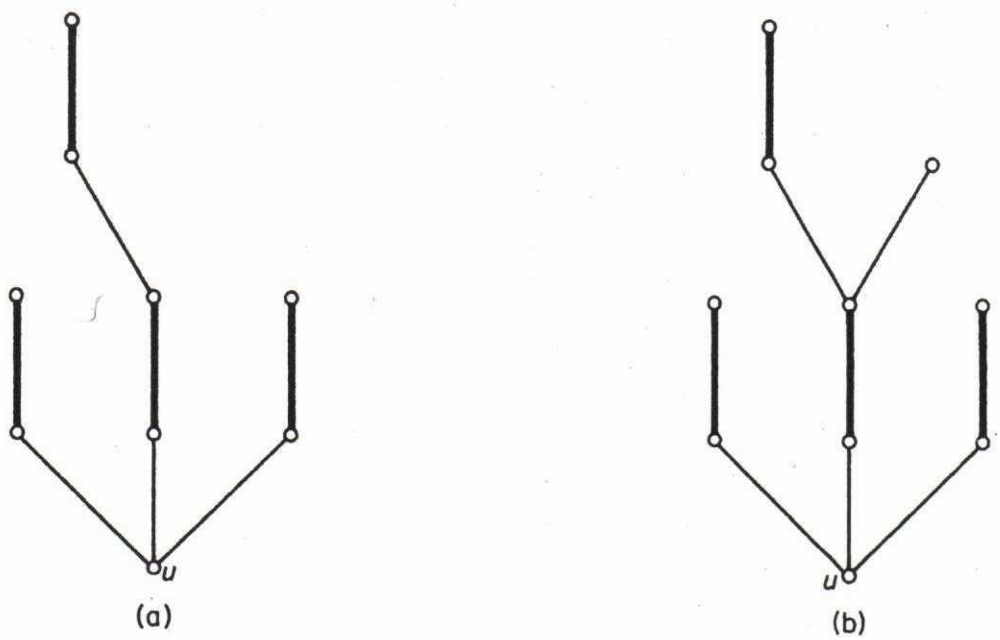

Figure 5.12. (*a*) Case (i); (*b*) case (ii)

The search for an M-augmenting path with origin u involves 'growing' an M-alternating tree H rooted at u. This procedure was first suggested by Edmonds (1965). Initially, H consists of just the single vertex u. It is then grown in such a way that, at any stage, either

(i) all vertices of H except u are M-saturated and matched under M (as in figure 5.12*a*), or
(ii) H contains an M-unsaturated vertex different from u (as in figure 5.12*b*).

If (i) is the case (as it is initially) then, setting $S = V(H) \cap X$ and $T = V(H) \cap Y$, we have $N(S) \supseteq T$; thus either $N(S) = T$ or $N(S) \supset T$.

(a) If $N(S) = T$ then, since the vertices in $S \backslash \{u\}$ are matched with the vertices in T, $|N(S)| = |S| - 1$, indicating that G has no matching saturating all vertices in X.
(b) If $N(S) \supset T$, there is a vertex y in $Y \backslash T$ adjacent to a vertex x in S. Since all vertices of H except u are matched under M, either $x = u$ or else x is matched with a vertex of H. Therefore $xy \notin M$. If y is M-saturated, with $yz \in M$, we grow H by adding the vertices y and z and the edges xy and yz. We are then back in case (i). If y is M-unsaturated, we grow H by adding the vertex y and the edge xy, resulting in case (ii). The (u, y)-path of H is then an M-augmenting path with origin u, as required.

Figure 5.13 illustrates the above tree-growing procedure.
The algorithm described above is known as the *Hungarian method*, and

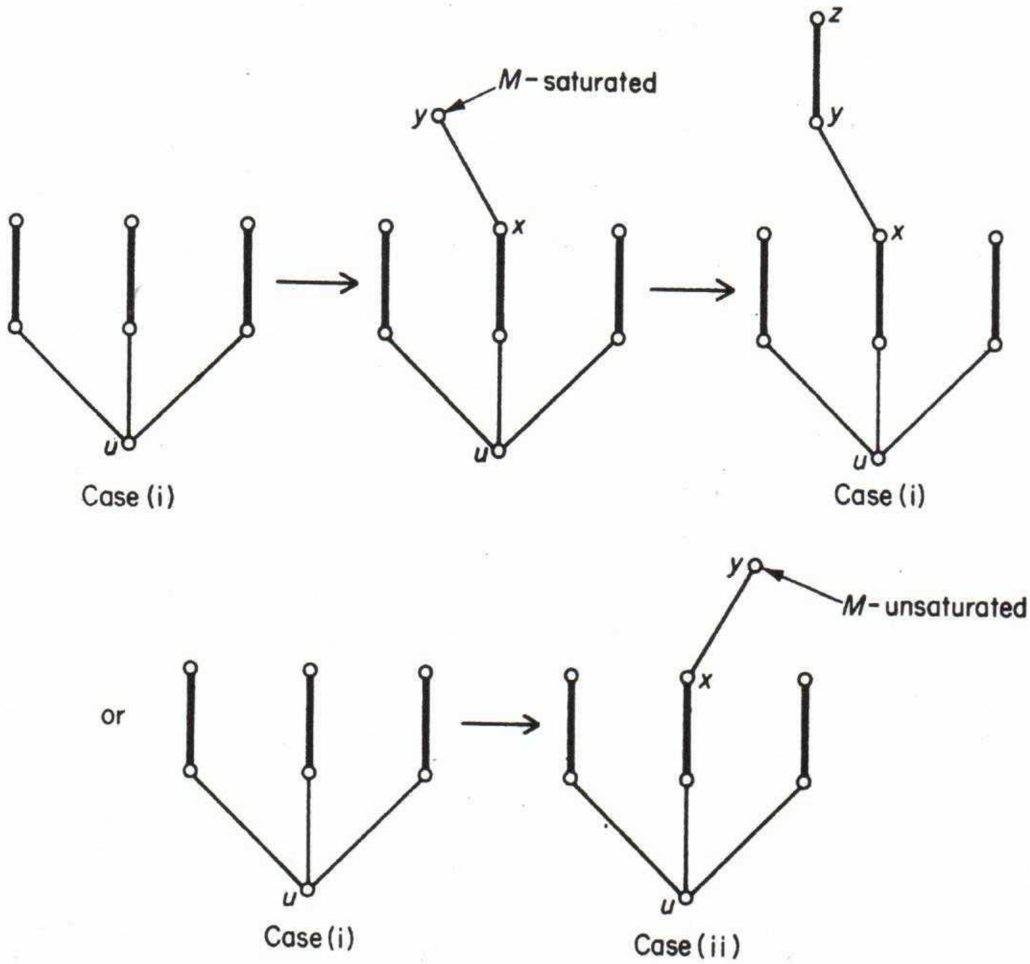

Figure 5.13. The tree-growing procedure

can be summarised as follows:

Start with an arbitrary matching M.

1. If M saturates every vertex in X, stop. Otherwise, let u be an M-unsaturated vertex in X. Set $S=\{u\}$ and $T=\emptyset$.
2. If $N(S)=T$ then $|N(S)|<|S|$, since $|T|=|S|-1$. Stop, since by Hall's theorem there is no matching that saturates every vertex in X. Otherwise, let $y \in N(S)\backslash T$.
3. If y is M-saturated, let $yz \in M$. Replace S by $S \cup \{z\}$ and T by $T \cup \{y\}$ and go to step 2. (Observe that $|T|=|S|-1$ is maintained after this replacement.) Otherwise, let P be an M-augmenting (u, y)-path. Replace M by $\hat{M}=M\,\Delta E(P)$ and go to step 1.

Consider, for example, the graph G in figure 5.14a, with initial matching $M=\{x_2y_2, x_3y_3, x_5y_5\}$. In figure 5.14$b$ an M-alternating tree is grown, starting with x_1, and the M-augmenting path $x_1y_2x_2y_1$ found. This results in a new matching $\hat{M}=\{x_1y_2, x_2y_1, x_3y_3, x_5y_5\}$, and an $\hat{M}$-alternating tree is now grown from x_4 (figures 5.14c and 5.14d) Since there is no $\hat{M}$-augmenting

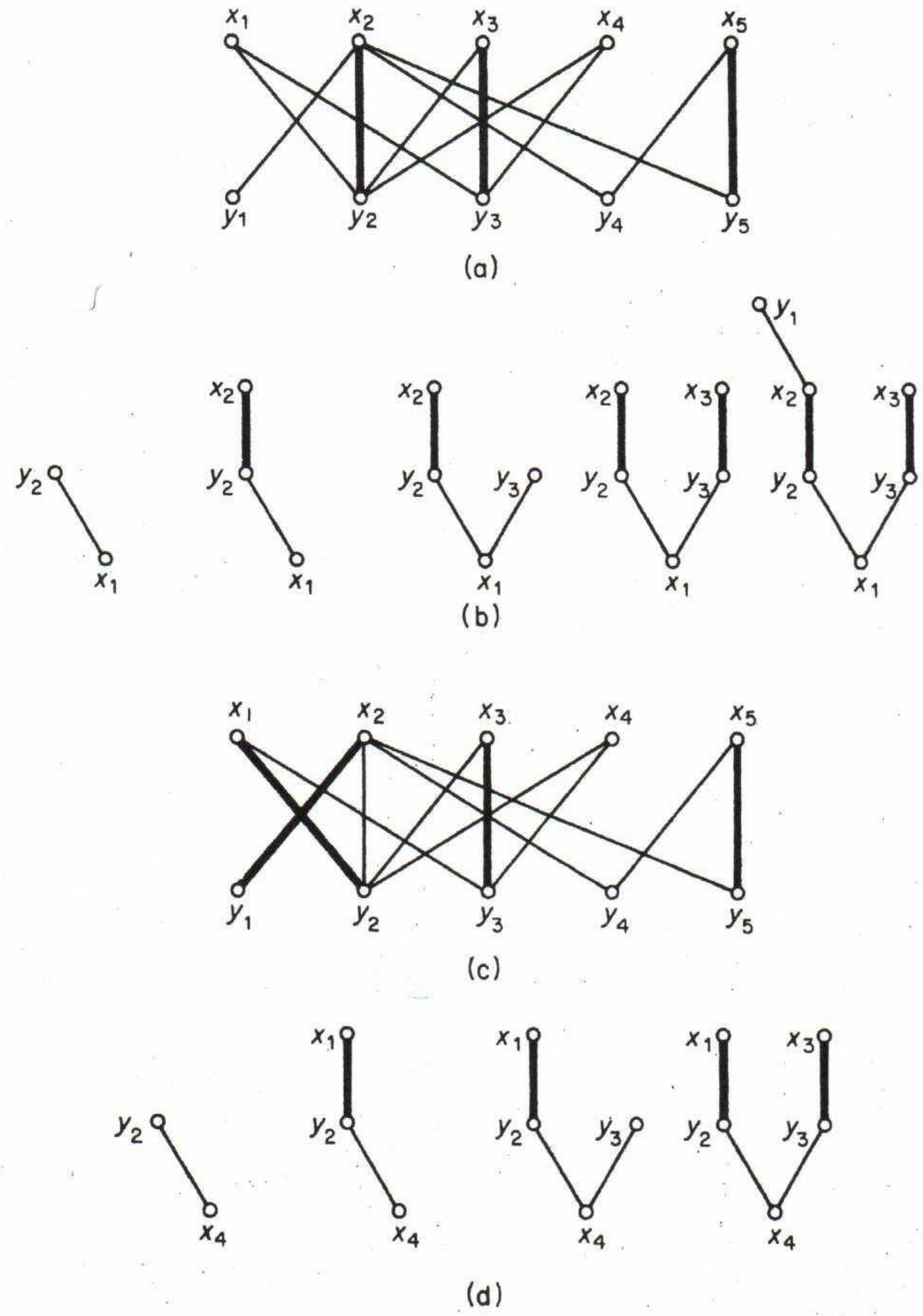

Figure 5.14. (*a*) Matching M; (*b*) an M-alternating tree; (*c*) matching $\hat{M}$; (*d*) an $\hat{M}$-alternating tree

path with origin x_4, the algorithm terminates. The set $S=\{x_1, x_3, x_4\}$, with neighbour set $N(S)=\{y_2, y_3\}$, shows that G has no perfect matching.

A flow diagram of the Hungarian method is given in figure 5.15. Since the algorithm can cycle through the tree-growing procedure, I, at most $|X|$ times before finding either an $S\subseteq X$ such that $|N(S)|<|S|$ or an M-augmenting path, and since the initial matching can be augmented at most $|X|$ times

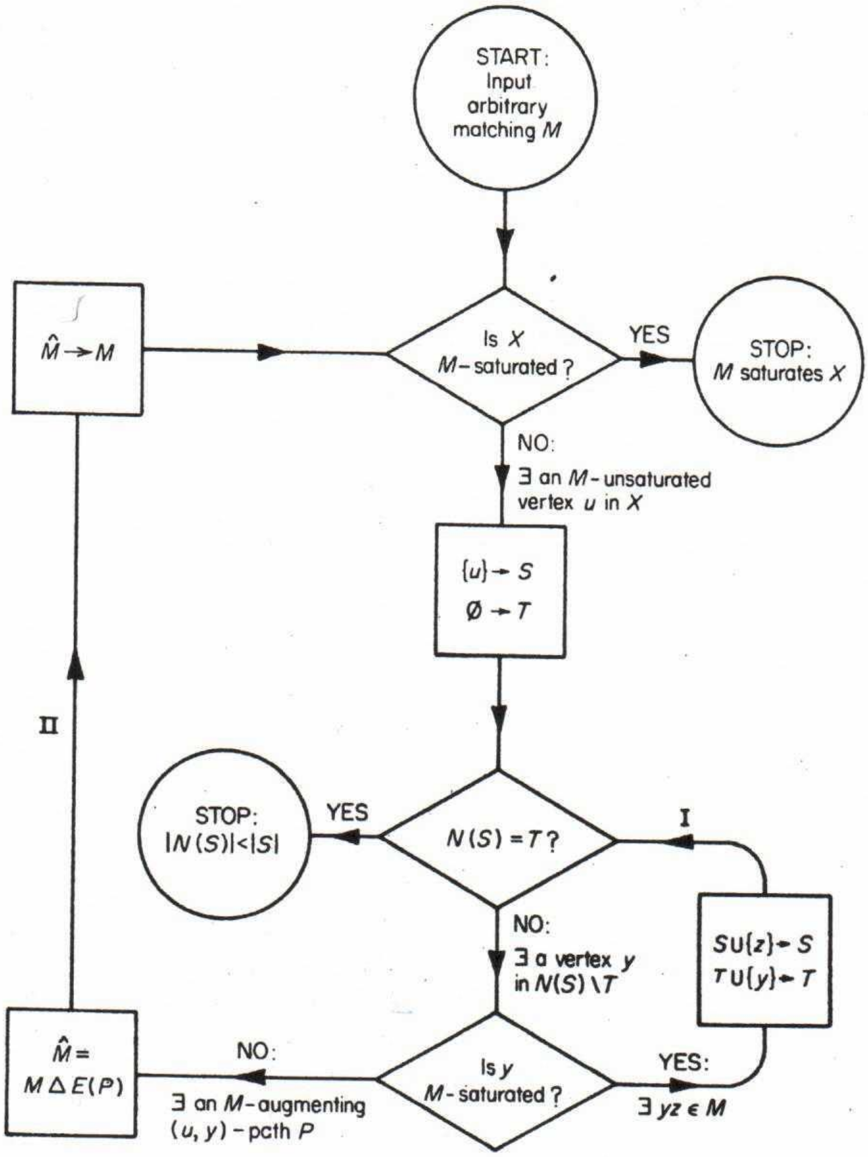

Figure 5.15. The Hungarian method

before a matching of the required type is found, it is clear that the Hungarian method is a good algorithm.

One can find a maximum matching in a bipartite graph by slightly modifying the above procedure (exercise 5.4.1). A good algorithm that determines such a matching in any graph has been given by Edmonds (1965).

Exercise

5.4.1 Describe how the Hungarian method can be used to find a maximum matching in a bipartite graph.