$k = 1, \ldots, n$, the vertices u_k and w_k both occur exactly once in the 1-factor F, and hence, the edge set

$$F' = \{e' \mid e \in F\}$$

contains exactly two edges of G that are incident on the vertex v_k. It follows that F' is a 2-factor of G. The graph obtained from G by deleting the edges of this 2-factor is a $(2r-2)$-regular graph, and the result follows by induction. ♢

Maximum Matchings and Minimum Vertex Covers

The theme of *max-min* pairs of optimization problems, seen earlier in this chapter and in Chapter 5, appears once again in the context of *vertex covers*.

DEFINITION: Let G be a graph, and let C be a subset of the vertices of G. Then set C is a ***vertex cover*** of graph G if every edge of G is incident on at least one vertex in C.

DEFINITION: A ***minimum vertex cover*** is a vertex cover with the least number of vertices.

Example 13.4.7: For the bipartite graph shown in Figure 13.4.8, a maximum matching (the bold edges) and minimum vertex cover (the solid vertices) both have cardinality 5.

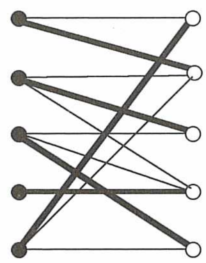

Figure 13.4.8 **A maximum matching and minimum vertex cover.**

Proposition 13.4.7 [Weak Duality for Matchings]. *Let M be a matching in a graph G, and let C be a vertex cover of G. Then $|M| \leq |C|$.* ♢ *(Exercises)*

Corollary 13.4.8 [Certificate of Optimality for Matchings]. *Let M be a matching in a graph G, and let C be a vertex cover of G such that $|M| = |C|$. Then M is a maximum matching and C is a minimum vertex cover.* ♢ *(Exercises)*

Remark: The converse of Corollary 13.4.8 does not hold in general (see Exercises); however, it does hold for bipartite graphs.

NOTATION: The neighbor set of a given subset W of vertices in a graph G is denoted $N_G(W)$ (instead of the usual $N(W)$) when there is more than one graph involved.

Theorem 13.4.9 [König, 1931]. *Let G be a bipartite graph. Then the number of edges in a maximum matching in G is equal to the number of vertices in a minimum vertex cover of G.*

Proof: Let $\{X, Y\}$ be the vertex bipartition of bipartite graph G, and let C^* be a minimum vertex cover of G. Then C^* is the disjoint union of its set of X-vertices, $C^*_X = C^* \cap X$, and its set of Y-vertices, $C^*_Y = C^* \cap Y$, as illustrated in Figure 13.4.9.

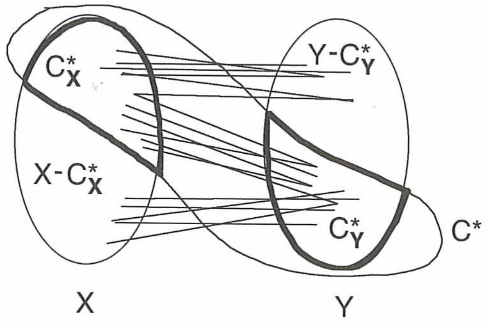

Figure 13.4.9 **Minimum vertex cover** $C^* = C^*_X \cup C^*_Y$.

Consider the bipartite subgraph G_1 of graph G induced on the vertex bipartition $\{C^*_X, Y - C^*_Y\}$. Let W be any subset of C^*_X. If $|W| > |N_{G_1}(W)|$, then there would exist $w \in W$ such that $N_{G_1}(W - \{w\}) = N_{G_1}(W)$. But this would imply that $(C^*_X - \{w\}) \cup C^*_Y = C^*$ is a vertex cover of graph G, contradicting the minimality of C^*. Thus, the bipartite graph G_1 satisfies Hall's Condition, and by Hall's Theorem (Theorem 13.4.3), G_1 has a C^*_X-saturating matching M^*_1, with $|M^*_1| = |C^*_X|$.

Next, let G_2 be the bipartite subgraph induced on the vertex bipartition $\{X - C^*_X, C^*_Y\}$. Then a similar argument applied to graph G_2 shows that it has a C^*_Y-saturating matching M^*_2, with $|M^*_2| = |C^*_Y|$. The edge set $M = M^*_1 \cup M^*_2$ is clearly a matching in graph G, and $|M| = |C^*_X| + |C^*_Y| = |C^*|$. Thus, by Corollary 12.4.6, M is a maximum matching. ◇

0-1 Matrices and the König-Egerváry Theorem

An interesting interpretation of this last theorem involves *0-1 matrices*, which are matrices each of whose entries is 0 or 1.

Theorem 13.4.10 [König-Egerváry, 1931]. *Let A be a 0-1 matrix. Then the maximum number of 1's in matrix A, no two of which lie in the same row or column, is equal to the minimum number of rows and columns that together contain all the 1's in A.*

Proof: Let G be a bipartite graph with vertex bipartition $\{X, Y\}$, such that A is an adjacency matrix of graph G, where X is the set of vertices corresponding to the rows of matrix A, and Y is the vertex set corresponding to the columns. The result follows by applying Theorem 13.4.9 (see Exercises). ◇

Application 13.4.4 *The Bottleneck Problem*: Suppose that a manufacturing process consists of five operations that are performed simultaneously on five machines. The time in minutes that each operation takes when executed on each machine is given in the table below. Determine whether it is possible to assign the operations so that the process is completed within 4 minutes.

	M1	M2	M3	M4	M5
Op1	4	5	3	6	4
Op2	5	6	2	3	5
Op3	3	4	5	2	4
Op4	4	8	3	2	7
Op5	2	6	6	4	5