Section 16.4. Perfect Matchings and Factors

Note. In Section 16.2. Matchings in Bipartite Graphs we saw necessary and sufficient conditions for the existence of a perfect matching of bipartite graphs (in Hall's Theorem [Theorem 16.4] and Corollary 16.5). In this section we give a necessary and sufficient condition for the existence of a perfect matching that applies to general graphs (in Tutte's Theorem, Theorem 16.13). The proofs require some of the results of Section 16.3. Matchings in Arbitrary Graphs (in particular, the idea of a barrier and the Tutte-Berge Theorem, Theorem 16.11), so this is necessary background for the current section. We define a factor and give an algorithm that reduces the problem of finding a general factor into finding a special type of factor (a "1-factor").

Theorem 16.13. Tutte's Theorem.
A graph G has a perfect matching if and only if $o(G - S) \leq |S|$ for all $S \subseteq V$.

Note. Tutte's Theorem (Theorem 16.13) does not gie a clear class of graph which have perfect matchings, but instead gives a necessary and sufficient condition for the existence of a perfect matching. Julius Petersen was concerned with a problem about factoring polynomials into irreducible factors (he corresponded with David Hlbert about this problem). He associated perfect matchings with factors of degree one. For this reason, perfect matchings are also called 1-factors. Pertersen's work appears in "Die Theorie der Regulären Graphs," *Acta Mathematica*, **15**, 193–220 (1891). An English translation of the paper appears in Chapter 10, "The Factorization of Graphs," of N. L. Biggs, E. K. Lloyd, and R. J. Wilson's *Graph Theory: 1736-1936*, Oxford University Press (1976) (this source was mentioned in in Section 1.7. Further Reading). Petersen focused on degree three polynomials, which correspond to 3-regular graphs, resulting in the next theorem. As a quick historical note, Bondy and Murty state that Petersen's interest in k–degree polynomials and there association with k-regular graphs is how the term "degree" migrated from polynomials to its use in graph theory as the "degree of a vertex."

Theorem 16.14. Petersen's Theorem.
Every 3-regular graph without cut edges has a perfect matching.

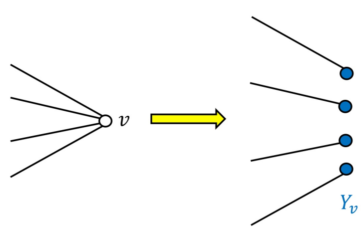
Note. The condition that G has no cut edges in Petersen's Theorem (Theorem 16.14), as shown by the fact that the 3-regular Sylvester graph has no perfect matching (as argued in Note 16.3.A).

Definition. Let G be a graph and let f be a nonnegative integer-valued function on V. An *f-factor* of G is a spanning subgraph F of G such that $d_F(v) = f(v)$ for all $v \in V$. A *k-factor* of G is an f-factor with $d_F(v) = f(v) = k$ for all $v \in V$.

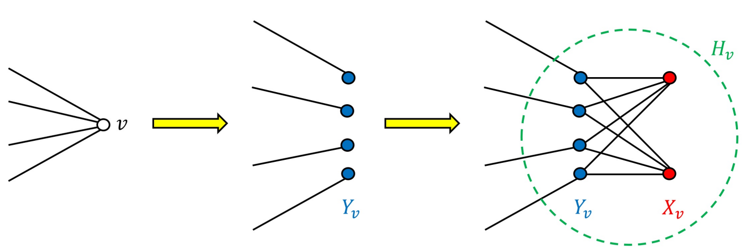
Note. A 1-factor of G is a spanning subgraph of G in which each vertex is degree 1. Hence a 1-factor is a perfect matching. A 2-factor is a spanning subgraph of G in which each vertex is degree 2. Hence a 2-factor is a collection of disjoint cycles which span the graph.

Note. Bondy and Murty state (on page 437): "Many interesting graph-theoretic problems can be solved in polynomial time by reducing them to problems about 1-factors." As an example, William Tutte showed that the question of deciding whether a given graph has an f-factor can be reduced to deciding whether a related graph G' has a 1-factor. His result appears in "A Short Proof of the Factor Theorem for Finite Graphs," *Canadian Journal of Mathematics*, **6**, 347–352 (1954). A copy is online on the University of Michigan website (accessed 7/1/2022). His reduction process is as follows.

Note 16.4.A. For G to have an f-factor, we must certainly have $d(v) \geq f(v)$ for all $v \in V$. We assume G is loopless "for simplicity." For each v of G, we replace v by set Y_v of $d(v)$ vertices, each of degree one so that each edge incident to v is then replaced by an edge incident to one of the Y_v as follows:



Next we add a set X_v of $d(v) - f(v)$ vertices and form a complete bipartite graph H_v by joining each vertex of X_v to each vertex of Y_v. For example, if we are looking for a 2-factor so that $f(v) = 2$, then when $d(v) = 4$ we have:



The resulting graph H is obtained from graph G, more simply, by replacing each vertex v of G by a complete bipartite graph $H_v[X_v, Y_v]$ and then joining each edge incident to v to a different vertex of Y_v. Figure 16.9 illustrates the construction in the case of 2-factors where $f(v) = 2$ for all $v \in V$; each dotted circle contains a copy of H_v. Notice that the corner vertices are of degree 3 so that $H_v = K_{3,1}$ in this case. When the degree is 4, as in the center vertex, $H_v = K_{4,2}$ (also as illustrated above).

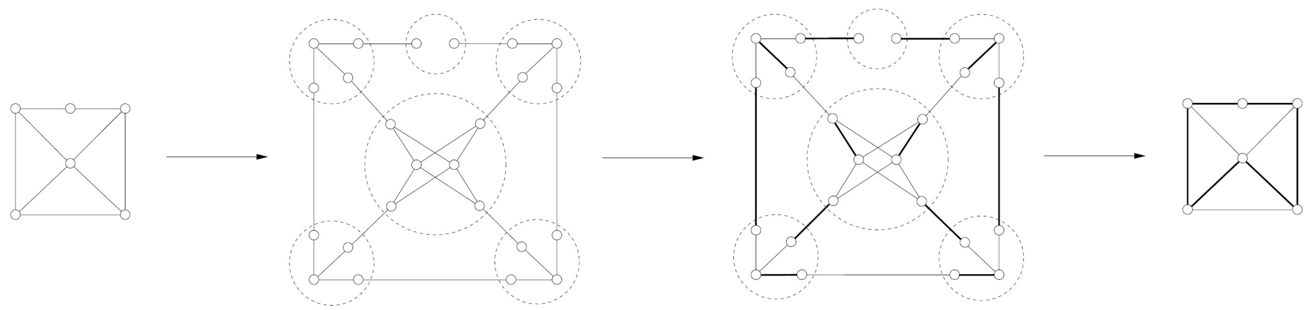


Fig. 16.9. Polynomial reduction of the 2-factor problem to the 1-factor problem

Notice that the dotted circles show how to recover G from H (by shrinking the encircled bipartite graphs H_v to a single vertex v). In H, the vertices of X_v are joined only to the vertices of Y_v. This <u>if</u> F is a 1-factor of H, then all $d(v) - g(v)$ vertices of X_v are matched by F with $d(v) - f(v)$ of the $d(v)$ vertices of Y_v. The remaining $f(v)$ vertices of Y_v must be matched by F with $f(x)$ vertices in $V(H) \setminus V(H_v)$ (these edges of F are "between" vertices of H_v and vertices of H_w where w is a neighbor of v in G). Next, shrinking H to G by collapsing the circles bipartite graphs of H back down to vertices of H yields an f-factor of G (conversely, any f-factor of G can be converted into a 1-factor of H, though this is not the direction we are interested in). Figure 16.9 gives the steps (1) the creation from graph G of graph H, (2) the introduction of a 1-factor (by observation), and (3) the shrinking of H down to G and the resulting f-factor (a 2-factor in this example).

Note. In Exercise 16.4.2 it is to be shown that the reduction of the f-factor problem to the 1-factor problem is a polynomial time procedure. In the next section we'll see a polynomial time algorithm that finds a 1-factor of a graph (Edmond's Algorithm). Together, these give a polynomial time algorithm for finding a general f-factor of a graph.

Note. In the pursuit of spanning subgraphs with degree parities, we introduce the next definition.

Definition. Let G be a graph and let T be an even subset of V. A spanning subgraph H of G is a *T-join* if $d_H(v)$ is odd for all $v \in T$ and even for all $v \in V \setminus T$.

Note. If H is a 1-factor of a graph then it is a V-join since each vertex of the 1-factor with vector set T is degree 1 and $V \setminus T = \varnothing$ so vacuously all $v \in V \setminus T$ are of even degree. If $H = P$ is a spanning xy-path in G then for $T = \{x, y\}$ we have that P is a T-join (or $\{x, y\}$-join) of G since each vertex of the path other than x and y is of degree two and each vertex of $T = \{x, y\}$ is of degree one in the path.

Note. In the setting of weighted graphs, we now state two problems. One is related directly to a T-join and the other is related to perfect matchings.

Problem 16.15. The Weighted T-Join Problem.

GIVEN: A weighted graph $G = (G, w)$ and a subset T of V.

FIND: A minimum-weight T-join of G (if one exists).

Note. The Shortest Path Problem (Problem 6.11) can be viewed as a special case of the Weighted T-Join Problem. We would start with graph G and vertices x and y. Then we would consider a minimum-weight $\{x, y\}$-join of G with the restriction that we replace G with the graph spanned by the edges of the path (so that the path is a spanning subgraph, as needed). Another special case is the Postman Problem of Exercise 16.4.22: "In [their] job, a postman picks up mail at the post office, delivers it, and then returns to the post office. [They] must, of course, cover each street at least once. Subject to this condition, [they wish] to choose a route entailing as little walking as [is] possible." This relates to finding a minimum-weight Eulerian spanning digraph. Notice that priority here is on the edges (i.e., the "streets") whereas in the Traveling Salesman Problem (Problem 2.6) the priority is on the vertices (i.e., the "cities").

Note. It is to be shown in Exercise 16.4.21 that the Weighted T-Join Problem can be reduced in polynomial-time to the following problem.

Problem 16.16. The Minimum-Weight Matching Problem.

GIVEN: A weighted complete graph $G = (G, w)$ of even order.

FIND: A minimum-weight perfect matching in G.

Note. The Minimum-Weight Matching Problem includes the Maximum Matching Problem (Problem 16.1). The input graph for which we want a maximum matching can be embedded in a complete graph of even order, and then a weight of 0 assigned to each vertex of the original graph and a weight of 1 to all other edges. Since weight is being minimized, the edges of the original graph (of weight 0) are included in the solution whenever possible. Of course, the edges of weight 1 in the minimum weight matching of the complete graph are then ignored to give the maximum matching of the original graph. Jack Edmonds gave a polynomial time algorithm in "Maximum Matching and a Polyhedron with 0,1 Vertices," *Journal of Research of the National Bureau of Standards—B*, **69B**(1–2), 125–130 (1965); a copy is online on National Institute of Standards and Technology website (accessed 7/2/2022). Some of his ideas are addressed in Section 17.4, "Coverings by Perfect Matchings"; see Exercises 17.4.5 to 17.4.7.

Revised: 7/2/2022