

## ON THE COMPLEXITY OF GENERAL GRAPH FACTOR PROBLEMS\*

D. G. KIRKPATRICK<sup>†</sup> AND P. HELL<sup>‡</sup>

**Abstract.** For arbitrary graphs  $G$  and  $H$ , a  $G$ -factor of  $H$  is a spanning subgraph of  $H$  composed of disjoint copies of  $G$ .  $G$ -factors are natural generalizations of 1-factors (or perfect matchings), in which  $G$  replaces the complete graph on two vertices. Our results show that the perfect matching problem is essentially the only instance of the  $G$ -factor problem that is likely to admit a polynomial time bounded solution. Specifically, if  $G$  has any component with three or more vertices, then the existence question for  $G$ -factors is NP-complete. (In all other cases the question can be resolved in polynomial time.)

The notion of a  $G$ -factor suggests a natural generalization where  $G$  is replaced by an arbitrary family of graphs. This generalization gives rise not only to further NP-completeness results but also to new polynomial algorithms and duality theorems extending results of the traditional theory of matching. An indication of the nature and scope of these new results is presented.

**Key words.** algorithms, complexity, factor, graph, matching, NP-completeness

**1. Introduction.** Let  $H$  denote an arbitrary graph with vertex set  $V(H)$  and edge set  $E(H)$ . A *matching* in  $H$  is any subset  $M$  of  $E(H)$  such that no two elements of  $M$  have a vertex in common. A matching  $M$  is *perfect* (also called a 1-factor) if exactly one element of  $M$  is incident with each vertex in  $V(H)$ . If  $H$  is a weighted graph, then the weight of a matching  $M$  is just the sum  $\sum_{e \in M} \text{weight}(e)$ .

The notion of a matching in a graph has numerous applications in such diverse areas as transversal theory, assignment problems, network flows, multiprocessor scheduling, shortest path algorithms, and the Chinese postman and traveling salesman problems [3], [10], [12], [13], [16], [17], [22], [31], [33], [36]. The existence of polynomial time bounded algorithms for the construction of matchings of maximum cardinality (and hence determining the existence of perfect matchings) or maximum weight is well known [6], [7], [8], [9], although the exact complexity of the problems is not yet settled and work continues on this aspect of the problem [23], [26]. In addition there is a rich mathematical theory that has developed about the matching problem that includes characterizations of graphs that admit perfect matchings [39] and (more generally) duality theorems on maximum matchings [1], [8], [12], [17], [32], [33], [34].

A matching in  $H$  may be viewed as a collection of disjoint subgraphs of  $H$ , each isomorphic to  $K_2$ .<sup>1</sup> In a perfect matching the vertex set  $V(H)$  is completely partitioned by the vertex sets of the subgraphs. This suggests the following natural generalization: Let  $G$  be an arbitrary graph. A  $G$ -packing of a graph  $H$  is a set  $\{G_1, \dots, G_d\}$  of disjoint subgraphs of  $H$  such that each  $G_i$  is isomorphic to  $G$ . (Note that we do not require the  $G_i$ 's to be induced subgraphs; that variant of the problem is discussed later in the paper). A perfect  $G$ -packing or  $G$ -factor of a graph  $H$  is a  $G$ -packing such that the sets  $V(G_i)$  partition  $V(H)$ . Clearly, a  $K_2$ -packing is just a matching and a  $K_2$ -factor is a perfect matching.

\* Received by the editors August 17, 1981, and in final revised form September 8, 1982. A preliminary version of the main result was presented at the Tenth Annual ACM Symposium on Theory of Computing, [30].

<sup>†</sup> Department of Computer Science University of British Columbia, Vancouver, British Columbia, Canada V6T 1W5. The research of this author was supported by the Natural Sciences and Engineering Research Council grant A3583.

<sup>‡</sup> Departments of Computing Science and Mathematics, Simon Fraser University, Burnaby, British Columbia, Canada V5A 1S6. The research of this author was supported by the Natural Sciences and Engineering Research Council grant A5075.

<sup>1</sup>  $K_t$  denotes the complete graph on  $t$  vertices.

We were motivated to study this generalization of matching by both practical and theoretical considerations. Graph partitioning problems arise in a number of applications [2], [4], [11], [15], [21], [24], [29]. Our original motivation [19], [30] stemmed from the study of examination scheduling. After an assignment of courses to examination periods, eliminating what could be called first-order conflicts (essentially a graph coloring problem), has been accomplished, the problem arises of assigning the examination periods to real time periods, under fairly standard constraints (normally some  $k$  examination periods are scheduled in sequence each day). The objective here is to minimize second-order conflicts (or inconveniences). Typically, this might include an occurrence of a student writing two examinations on the same day or perhaps two consecutive examinations on the same day. Suppose  $H$  is the graph whose vertex set is the set of examination periods  $\{p_i | 1 \leq i \leq t\}$  and whose edge  $(p_i, p_j)$  is weighted by the number of students common to courses examined in periods  $p_i$  and  $p_j$ . Then minimization of the types of second-order conflicts illustrated above corresponds to the construction of minimum weight  $K_3$ - and  $P_3^2$ -factors in  $H$ .

As we shall see, our investigation of generalized matching can also be viewed as furthering our understanding of the perceived threshold between NP-complete and polynomial time solvable problems [5], [14], [27], [28]. Specifically, it is of interest to know which members of this family of problems admit polynomial time bounded algorithms and which, like the general subgraph isomorphism problem of which they are all special instances, are NP-complete. We are able to provide a complete characterization (in the above sense) of the complexity of finding a  $G$ -factor. This characterization is similar, in spirit, to the results of Schaefer [38], Yannakakis [40] and Lewis [35] each of which establishes NP-completeness results over a broad family of interesting problems.

While our results here are essentially negative, it should be noted that an extension of the notion of  $G$ -packings and  $G$ -factors (replacing  $G$  by a family of graphs) has pointed the way to a very natural setting in which to extend the traditional theory of matching, giving rise to new polynomial algorithms and simple duality results [19], [20].

**2. Generalizations of matching.** Our notion of a  $G$ -packing (and  $G$ -factor) is by no means the only natural extension of the familiar concept of matching. Indeed, in § 5, we introduce and motivate the notion of a  $\mathcal{G}$ -packing (and  $\mathcal{G}$ -factor), where  $\mathcal{G}$  denotes a family of graphs. This extension subsumes, and should not be confused with, the notion of  $F$ -factor introduced by Muhlbacher [37].

If the concept of matching is extended in the natural way to hypergraphs the problem of determining the existence of a perfect matching is known to be NP-complete. Karp [27] describes what is probably the simplest version of this problem as three-dimensional matching:

INSTANCE: An integer  $p$  and a set  $U \subseteq \{1, 2, \dots, p\}^3$ .

QUESTION: Is there a subset  $W \subseteq U$  of cardinality  $p$  such that no two elements of  $W$  agree in any coordinate?

It should be clear that the  $k$ -dimensional matching problem (replace three by  $k$  above) is also NP-complete, when  $k \geq 3$ . The two-dimensional matching problem is equivalent to the matching problem for bipartite graphs.

Expressed as a language recognition problem, the existence problem for  $G$ -factors, which we denote  $\text{FACT}(G)$ , becomes:

INSTANCE: A graph  $H$ .

QUESTION: Does  $H$  admit a  $G$ -factor?

<sup>2</sup>  $P_t$  denotes the path on  $t$  vertices.

The problem  $\text{FACT}(K_1)$  is trivial since every graph admits a  $K_1$ -factor. Furthermore,  $\text{FACT}(K_2)$  is just the question of existence of a perfect matching, and hence,  $\text{FACT}(K_2) \in P$ . More generally, if  $G = \alpha \cdot K_1 \cup \beta \cdot K_2$ , that is the disjoint union of  $\alpha$  copies of  $K_1$  and  $\beta$  copies of  $K_2$  (or, equivalently, if each connected component of  $G$  has at most two vertices), then  $H$  admits a  $G$ -factor if and only if  $|V(H)|$  is divisible by  $|V(G)|$  and  $H$  admits a matching with at least  $\beta|V(H)|/|V(G)|$  edges. Thus the usual algorithms for finding a maximum matching (e.g., [7], [34]) may be used to answer  $\text{FACT}(G)$  in polynomial time. Our central result suggests that all other problems  $\text{FACT}(G)$  are unlikely to admit efficient solutions.

**THEOREM 4.2.** *If  $G$  is not of the form  $\alpha \cdot K_1 \cup \beta \cdot K_2$ , then  $\text{FACT}(G)$  is NP-complete.*

Two important instances of this result,  $G = K_3$  and  $G = P_3$ , were established earlier by T. Schaeffer [14], [28] and D. S. Johnson [25].

The proof of Theorem 4.2 is deferred to § 4. The following lemma allows us to restrict our attention to problems  $\text{FACT}(G)$ , where  $G$  is a connected graph.

**LEMMA 2.1.** *Let  $G$  be a graph and  $G'$  any component of  $G$  with the maximum number of edges. Then,  $\text{FACT}(G') \leq_p \text{FACT}(G)$ .*

*Proof.* Suppose  $G'$  has  $p$  vertices and suppose  $G$  has  $r$  distinct components isomorphic to  $G'$ . If  $H$  is any graph with  $dp$  vertices, then let  $T(H)$  denote the graph  $H \cup d(G - G')$ . Obviously, if  $H$  admits a  $G'$ -factor, then  $T(H)$  admits a  $G$ -factor. Suppose  $T(H)$  admits a  $G$ -factor  $F$ .  $F$  must contain exactly  $dr$  components isomorphic to  $G'$ . But, by the maximality of  $G'$ , the restriction of  $F$  to  $d(G - G')$  contains at most  $d(r - 1)$  components isomorphic to  $G'$ . Hence the restriction of  $F$  to  $H$  must be a  $G'$ -factor of  $H$ . Thus  $H$  admits a  $G'$ -factor if and only if  $T(H)$  admits a  $G$ -factor.  $\square$

**3. Basic modules and their properties.** Our objective in this and the next section is to demonstrate how, for an arbitrary connected graph  $G$  on  $k$  vertices, the  $k$ -dimensional matching problem can be polynomially reduced to the problem  $\text{FACT}(G)$ . Our construction is component based (cf. [14]) in nature; in this section we describe the components (which we call modules) and their properties that we exploit in the general construction.

**3.1. Modules and coherences.** A *module* is a graph  $M$  with nonempty subset  $C \subseteq V(M)$  of distinguished vertices. We call the elements of  $C$  (respectively  $V(M) - C$ ) *connector vertices* (respectively *interior vertices*) of  $M$ . A  $G$ -*module* is any module that admits a  $G$ -packing covering all of its interior vertices (plus some, possibly empty, subset of its connector vertices).

A *modular extension* of the module  $M$  is any graph  $H$ , containing  $M$  as an induced subgraph, in which no interior vertex of  $M$  is adjacent to a vertex of  $H - M$  (that is,  $M$  is connected to the rest of  $H$  only through its connector vertices). Let  $\pi = \{G_1, \dots, G_d\}$  be any  $G$ -packing of some modular extension  $H$  of  $M$ . A vertex  $v$  of  $M$  is said to be *bound* to  $M$  by  $\pi$ , if  $v \in V(G_i)$  implies  $V(G_i) \subseteq V(M)$ . A  $G$ -module  $M$  is *internally  $G$ -coherent* if every  $G$ -factor of every modular extension of  $M$  binds to  $M$  all of its interior vertices (that is, it respects the modularity of  $M$ ).

The simplest example of internally  $G$ -coherent  $G$ -module is the (connected) graph  $G$  itself with any one vertex  $v \in V(G)$  designated as a connector vertex. We depict this schematically in Fig. 1.

**3.2. Diamond modules.** If  $G$  is any connected graph and  $v \in V(G)$ , then the graph, formed from  $G$  by splitting  $v$  into two nonadjacent vertices  $v_a$  and  $v_b$ , each of which is adjacent to all of the neighbors of  $v$  in  $G$ , is called a  $G$ -*diamond* and is denoted  $D[G; v]$ . We depict  $D[G; v]$  schematically in Fig. 2.

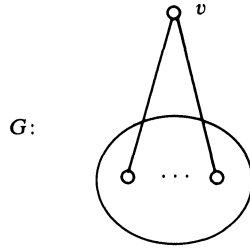


FIG. 1

If  $v_a$  and  $v_b$  are taken as connector vertices, then  $D[G; v]$  is a  $G$ -module. Its coherence, it turns out, depends on the choice of vertex  $v$ , but a choice ensuring  $G$ -coherence always exists. Specifically, let  $v^*$  be any vertex of  $G$  that is not a cutpoint and belongs to a biconnected component of  $G$  containing at most one cutpoint. Every graph  $G$  is guaranteed to contain at least one such vertex (cf. [18, p. 36]).

LEMMA 3.1. *The module  $D[G; v^*]$  with  $v_a^*$  and  $v_b^*$  as connectors is internally  $G$ -coherent.*

*Proof.* Let  $H$  be any modular extension of  $D[G; v^*]$  and let  $\varphi$  be any  $G$ -factor of  $H$ .  $\varphi$  induces a partition  $\pi$  of the interior vertices of  $D[G; v^*]$ . Since  $D[G; v^*]$  has exactly two connector vertices and each graph in  $\varphi$  is connected,  $\pi$  has at most two cells. All of the vertices of  $D[G; v^*]$  that do not belong to the same biconnected component as  $v_a^*$  must belong to the same partition of  $\pi$  (otherwise there must be two vertex-disjoint paths from this set to  $v^*$  in  $G$ , contradicting the choice of  $v^*$ ). Hence, if  $\pi$  has two cells then some element of  $\varphi$  must contain all of the vertices of  $D[G; v^*]$  that do not belong to the same biconnected component as  $v_a^*$ , either  $v_a^*$  or  $v_b^*$ , and at least one vertex of  $H - D[G; v^*]$ . But this component has at least one more cutpoint (namely  $v_a^*$  or  $v_b^*$ ) than  $G$ , a contradiction. Thus,  $\pi$  has exactly one cell and hence  $D[G; v^*]$  is internally  $G$ -coherent.  $\square$

We can summarize the relevant properties of diamond modules as follows:

Property 3.2. (a) Every  $G$ -factor of every modular extension of  $D[G; v^*]$  binds to  $D[G; v^*]$  its interior vertices plus exactly one of its connector vertices.

(b) The graph  $D[G; v^*]$  minus either one of its connector vertices admits a  $G$ -factor.

Thus, diamond modules, wherever they appear in a larger graph, force a “choice” of one or the other of their connector vertices.

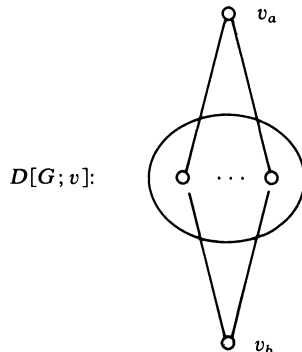


FIG. 2

**3.3. Star modules.** A  $G$ -star, denoted  $S[G; v]$ , is the graph formed from  $G$  by identifying, with each vertex  $w \in V(G)$ , the  $v_a$ -connector of a distinct copy of  $D[G; v]$ . If the  $v_b$ -connectors of the  $|V(G)|$   $G$ -diamonds used in the construction, relabelled as  $x_1, \dots, x_{|V(G)|}$ , are taken as connector vertices, then  $S[G; v]$  is a  $G$ -module. We depict  $S[G; v]$  schematically in Fig. 3.

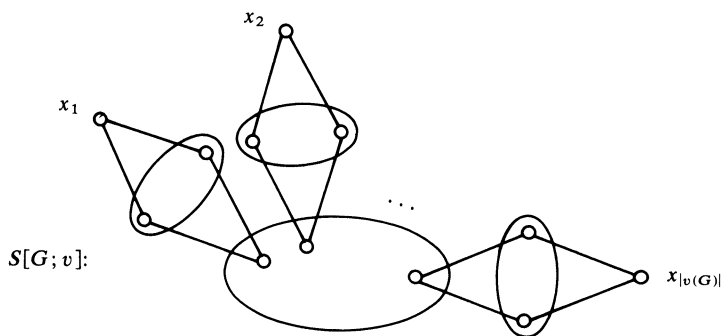


FIG. 3

$S[G; v]$  can be seen as a modular extension of  $|V(G)|$  disjoint copies of  $D[G; v]$ . As would be expected the coherence of  $S[G; v]$  depends on the coherence of  $D[G; v]$ . Specifically,

**LEMMA 3.3.** *If  $D[G; v]$  is internally  $G$ -coherent, then so is  $S[G; v]$ .*

*Proof.* Let  $H$  be any modular extension of  $S[G; v]$  and let  $\varphi$  be any  $G$ -factor of  $H$ .  $H$  must also be a modular extension of each of the  $|V(G)|$  copies of  $D[G; v]$  used in the construction of  $S[G; v]$ . Since  $D[G; v]$  is internally  $G$ -coherent, it follows that all of the interior vertices of  $S[G; v]$  that are internal to one of the copies of  $D[G; v]$  must be bound to  $S[G; v]$  by  $\varphi$ . Since none of the remaining interior vertices of  $S[G; v]$  are adjacent to any of the connectors of  $S[G; v]$ , it follows that  $\varphi$  binds to  $S[G; v]$  all of its interior vertices.  $\square$

**COROLLARY 3.4.**  *$S[G; v^*]$  is internally  $G$ -coherent.*

In fact,  $S[G; v^*]$  satisfies the following somewhat stronger property. A  $G$ -module  $M$  is  $G$ -coherent if it is internally  $G$ -coherent, if every  $G$ -factor of every modular extension of  $M$  binds to  $M$  either all or none of its connector vertices, and if in addition, both  $M$  and  $M - C$  admit  $G$ -factors.  $G$ -coherence places a strong restriction on the  $G$ -modularity of  $M$ . It is clear from the definitions that  $G$ -coherent modules  $M$ , wherever they appear in a larger graph, can be viewed as forcing a “choice” of either all or none of their connector vertices (both of which are possible). Our central construction rests on the following:

**LEMMA 3.5.**  *$S[G, v^*]$  is  $G$ -coherent.*

*Proof.* Let  $\varphi$  be any  $G$ -factor of any modular extension of  $S[G, v^*]$ . Note that  $S[G, v^*]$  contains  $|V(G)|^2$  vertices of which  $|V(G)|$  are connectors. Since  $S[G, v^*]$  is internally  $G$ -coherent and the total number of vertices bound to  $S[G, v^*]$  by  $\varphi$  must be a multiple of  $|V(G)|$ , it follows that either all or none of the connector vertices must be bound to  $S[G, v^*]$  by  $\varphi$ .  $\square$

**4. The general construction.** We are now prepared to state and prove our central lemma.

**LEMMA 4.1.** *If  $G$  is a connected graph, then  $|V(G)|$ -dimensional matching  $\leq_p$  FACT( $G$ ).*

*Proof.* Let  $p$  be any positive integer and  $U \subseteq \{1, 2, \dots, p\}^k$ , where  $k = |V(G)|$ . It suffices to show how to construct (in polynomial time) a graph  $R(U)$  with the property that  $R(U)$  admits a  $G$ -factor if and only if  $U$  admits a  $k$ -dimensional matching.

$R(U)$  contains, among others, an independent set of  $kp$  vertices labelled by the pairs  $(i, j)$ , where  $1 \leq i \leq k$  and  $1 \leq j \leq p$ . For each  $k$ -tuple  $(t_1, \dots, t_k) \in U$ ,  $R(U)$  contains a distinct copy of  $S[G; v^*]$  whose  $k$  connector vertices are arbitrarily identified with the  $k$  vertices labelled  $(1, t_1), \dots, (k, t_k)$ .

Suppose that  $U$  admits a  $k$ -dimensional matching  $W$ . We construct a  $G$ -factor  $\varphi$  of  $R(U)$  as follows. To those copies of  $S[G; v^*]$  associated with  $k$ -tuples in  $W$ ,  $\varphi$  binds all of their vertices (in particular, their connector vertices). To all other copies of  $S[G; v^*]$ ,  $\varphi$  binds only their interior vertices. Thus,  $\varphi$  binds the vertex  $(i, j)$  to the star module associated with that unique  $k$ -tuple in  $W$  containing  $j$  in position  $i$ . It follows that  $\varphi$  is a  $G$ -factor of  $R(U)$ .

Conversely, suppose that  $R(U)$  admits a  $G$ -factor  $\varphi$ . We construct a  $k$ -dimensional matching  $W$  of  $U$  as follows. Call a copy of  $S[G; v^*]$  in  $R(u)$  “chosen” if  $\varphi$  binds to that copy all of its connector vertices. By Lemma 3.5  $\varphi$  chooses exactly  $p$  of the star-modules in  $R(u)$ . Let  $W$  be the set of  $k$ -tuples associated with chosen star modules. Since each vertex  $(i, j)$  is bound to exactly one chosen star module, it follows that exactly one element of  $W$  contains  $j$  in its  $i$ th component. Hence  $W$  is a  $k$ -dimensional matching of  $U$ .  $\square$

We now restate and give a direct proof of our central result.

**THEOREM 4.2.** *If  $G$  is not of the form  $\alpha \cdot K_1 \cup \beta \cdot K_2$ , then  $\text{FACT}(G)$  is NP-complete.*

*Proof.* It is clear that all problems  $\text{FACT}(G)$  are in NP. By Lemma 2.1, it suffices to show that if  $G$  is a connected graph with at least three vertices, then  $\text{FACT}(G)$  is NP-complete. But this is immediate from Lemma 4.1 and the NP-completeness of  $k$ -dimensional matching, for  $k \geq 3$ .  $\square$

Thus virtually all uniform factoring problems (with the exception of matching) are NP-complete. A similar characterization holds for what we call “strict”  $G$ -factors.

A  $G$ -packing (or  $G$ -factor) of  $H$  is *strict* if each  $G_i$  belonging to the packing is an induced subgraph of  $H$ . Corresponding to  $\text{FACT}(G)$  we have the question  $S\text{-FACT}(G)$  expressed as:

INSTANCE: A graph  $H$ .

QUESTION: Does  $H$  admit a strict  $G$ -factor?

Note that  $H$  admits a strict  $G$ -factor if and only if its complement  $\bar{H}$  admits a strict  $\bar{G}$ -factor. Then  $S\text{-FACT}(G)$  and  $S\text{-FACT}(\bar{G})$  are polynomially equivalent, and it is sufficient to consider problems  $S\text{-FACT}(G)$  for connected graphs  $G$  only. Clearly, if  $G$  has fewer than three vertices a polynomial algorithm for  $S\text{-FACT}(G)$  follows from algorithms for maximum matching (e.g., [7]). As with  $\text{FACT}(G)$  all other cases appear to be intractable.

**THEOREM 4.3.** *If  $G$  has at least three vertices, then  $S\text{-FACT}(G)$  is NP-complete.*

*Proof.* Observe that, in the proof of Lemma 4.1., the graph  $R(U)$  admits a  $G$ -factor if and only if it admits a strict  $G$ -factor. (This follows from the construction of star modules.) Hence, Lemma 4.1 also proves that  $|V(G)|$ -dimensional matching  $\leq_p S\text{-FACT}(G)$ . Thus, the result follows from the obvious fact that  $S\text{-FACT}(G) \in \text{NP}$  and the NP-completeness of  $k$ -dimensional matching for  $k \geq 3$ .  $\square$

**5. Family packings and factors.** We introduced  $G$ -packings and  $G$ -factors as a generalization of conventional matchings and have reached the unfortunate conclusion

that this is an unlikely direction in which to generalize the rich theory—most notably the existence of polynomial time bounded algorithms—that is associated with the matching problem. However, a straightforward extension of the notion of  $G$ -packing suggests itself as another natural generalization of matching. While negative results still abound, this extension does give rise to a number of positive results which hint at a new generalized theory of matching including both polynomial algorithms and elegant duality results.

We extend the notion of  $G$ -packing by replacing  $G$  by a family  $\mathcal{G}$  of “packing” graphs. A  $\mathcal{G}$ -packing of a graph  $H$  is a set  $\{G_1, \dots, G_d\}$  of disjoint subgraphs of  $H$  such that each  $G_i$  is isomorphic to some element of  $\mathcal{G}$ . A  $\mathcal{G}$ -factor is defined similarly. The existence problem for  $\mathcal{G}$ -factors, denoted  $\text{FACT}(\mathcal{G})$ , becomes:

INSTANCE: A graph  $H$ .

QUESTION: Does  $H$  admit a  $\mathcal{G}$ -factor?

As an example, if  $C_t$  denotes the cycle on  $t$  vertices and  $\mathcal{G} = \{K_2, C_3, C_4, C_5, \dots\}$ , then  $\text{FACT}(\mathcal{G})$  can be solved as an assignment problem, [33].

The NP-completeness of many problems  $\text{FACT}(\mathcal{G})$  stems directly from our earlier constructions. As a simple example, consider:

*Example 5.1.*  $\text{FACT}(\{K_t | t \geq 3\})$  is NP-complete.

*Proof.* It suffices to observe that the graph used in our reduction of three-dimensional matching to  $\text{FACT}(K_3)$  contains no complete subgraphs of order four. Thus any  $\{K_t | t \geq 3\}$ -factor must also be a  $K_3$ -factor.  $\square$

It is interesting to note that  $H$  has a  $\{K_t | t \geq 3\}$ -factor if and only if its complement has a coloring in which each color class contains at least three vertices. This connection with coloring is explored in more detail in [19]. Example 5.1 is subsumed by the following theorem whose proof will appear elsewhere.

**THEOREM 5.2.** *Let  $\mathcal{G}$  be any subset of  $\{K_t | t \geq 1\}$ . If  $K_1 \in \mathcal{G}$  or  $K_2 \in \mathcal{G}$ , then  $\text{FACT}(\mathcal{G})$  is in P, otherwise  $\text{FACT}(\mathcal{G})$  is NP-complete.*

One further example should help to substantiate our claim that the study of family factorizations is a fertile setting in which to generalize the traditional theory of matching.

*Example 5.3.*  $\text{FACT}(\{K_{1,t} | t \geq 1\})$  is in P.

*Proof.* It is straightforward to confirm that a graph  $H$  admits a  $\{K_{1,t} | t \geq 1\}$ -factor if and only if it contains no isolated vertices.

The facility location (or domination number) problem [4], [11] can be viewed as trying to find a minimal  $\{K_{1,t} | t \geq 1\}$ -factor. Our framework makes it natural to express the related problem of determining the existence of factors using only a restricted subset of facilities (star graphs). Example 5.3 is just one special case of the following:

**THEOREM 5.4.** *Let  $\mathcal{G}$  be any subset of  $\{K_{1,t} | t \geq 1\}$ . If for some  $t \geq 1$ ,  $K_{1,t} \notin \mathcal{G}$  and  $K_{1,t+1} \in \mathcal{G}$ , then  $\text{FACT}(\mathcal{G})$  is NP-complete. Otherwise  $\text{FACT}(\mathcal{G})$  is in P.*

The proof of Theorem 5.4 also includes a duality result analogous to the theorems of Tutte [39] and Berge [1, p. 159] for star matchings; it will appear elsewhere. We have similar results for any set of complete bipartite graphs.

**6. Conclusions.** Our results completely classify the complexity of factorization problems. The vast majority of them are NP-complete. The few remaining cases can be reduced to the maximum matching problem and hence admit polynomial time algorithms. The results of § 5 suggest that the study of the more general family factorization problems is a fruitful direction in which to pursue a generalized theory of matching, including new polynomial algorithms.

## REFERENCES

- [1] C. BERGE, *Graphs and Hypergraphs*, North-Holland, Amsterdam, 1973.
- [2] F. T. BOESCH AND J. F. GIMPEL, *Covering the points of a digraph with point-disjoint paths and its application to code optimization*, J. Assoc. Comput. Mach., 24 (1977), pp. 192–198.
- [3] N. CHRISTOFIDES, *Worst-case analysis of a new heuristic for the travelling salesman problem*, GSIA, Carnegie-Mellon University, 1976.
- [4] E. COCKAYNE, S. GOODMAN AND S. HEDETNIEMI, *A linear algorithm for the domination number of a tree*, Inform. Processing Letters, 4 (1975), pp. 41–44.
- [5] S. A. COOK, *The complexity of theorem-proving procedures*, Proc. Third ACM Symposium on Theory of Computing, pp. 151–158.
- [6] W. H. CUNNINGHAM AND A. B. MARSH, III, *A primal algorithm for optimum matching*, Math. Programming Study, 8 (1978), pp. 50–72.
- [7] J. EDMONDS, *Paths, trees, and flowers*, Canad. J. Math., 17 (1965), pp. 449–467.
- [8] ———, *Maximum matching and a polyhedron with  $(0, 1)$  vertices*, J. Res. Nat. Bureau of Standards, 69B (1965), pp. 125–130.
- [9] J. EDMONDS AND E. L. JOHNSON, *Matching: a well-solved class of integer linear programs*, in Combinatorial Structures and their Applications, R. K. Guy et al., eds., Gordon and Breach, New York, 1970, pp. 89–92.
- [10] ———, *Matching, Euler tours, and the Chinese postman*, Math. Programming, 5 (1973), pp. 88–124.
- [11] M. FARBER, *Domination and duality in weighted trees*, Congressus Numerantium, 33 (1981), pp. 3–13.
- [12] L. R. FORD, JR. AND D. R. FULKERSON, *Flows on Networks*, Princeton University Press, Princeton, NJ, 1962.
- [13] M. FUJII, T. KASAMI AND K. NINAMIYA, *Optimal sequencing of two equivalent processors*, SIAM J. Appl. Math., 17 (1969), pp. 784–789; Erratum, *ibid.*, 20 (1971), p. 141.
- [14] M. R. GAREY AND D. S. JOHNSON, *Computers and Intractability*, W. H. Freeman, San Francisco, 1979.
- [15] L. L. GARNISHTEYN, *The partitioning of graphs*, Engineering Cybernetics, 1 (1969), pp. 76–82.
- [16] P. C. GILMORE AND R. E. GOMORY, *Sequencing a one-stage variable machine: a solvable case of the travelling salesman problem*, Oper. Res., 12 (1964), pp. 655–679.
- [17] M. HALL, *Distinct representations of subsets*, Bull. Amer. Math. Soc., 54 (1948), pp. 922–926.
- [18] F. HARARY, *Graph Theory*, Addison-Wesley, Reading, MA, 1968.
- [19] P. HELL AND D. G. KIRKPATRICK, *Scheduling, matching and coloring*, in Algebraic Methods in Graph Theory, G. R. Szasz et al., eds., Colloq. Math. Soc. Janos Bolyai, Hungary, 1981.
- [20] ———, *On generalized matching problems*, Inform. Processing Letters, 12 (1981), pp. 33–35.
- [21] L. J. HERBERT, *Some applications of graph theory to clustering*, Psychometrika, 39 (1974), pp. 283–309.
- [22] A. J. HOFFMAN AND H. M. MARKOWITZ, *A note on shortest path, assignment, and transportation problems*, Naval Res. Logist. Quart., 10 (1963), pp. 375–380.
- [23] J. E. HOPCROFT AND R. M. KARP, *An  $n^{5/2}$  algorithm for maximum matchings in bipartite graphs*, SIAM J. Comput., 2 (1973), pp. 225–231.
- [24] A. K. HOPE, *Component placing through graph partitioning in computer-aided printed-wiring-board design*, Electronic Letters, 8 (1972), pp. 87–88.
- [25] D. S. JOHNSON, private communication, August 1977.
- [26] O. KARIV, *An  $O(n^{2.5})$  algorithm for finding maximum matching in a general graph*, Ph.D. Thesis, Weizmann Institute, Rehovot, Israel, 1976.
- [27] R. M. KARP, *Reducibility among combinatorial problems*, in complexity of Computer Computations, R. E. Miller and J. W. Thatcher, eds., Plenum Press, New York, 1972, pp. 85–103.
- [28] ———, *On the complexity of combinatorial problems*, Networks, 5 (1975), pp. 45–68.
- [29] B. W. KERNIGHAN AND S. LIN, *An efficient heuristic procedure for partitioning graphs*, Bell System Tech. J., 49 (1970), pp. 291–307.
- [30] D. G. KIRKPATRICK AND P. HELL, *On the completeness of a generalized matching problem*, in Proc. Tenth Annual ACM Symposium on Theory and Computing, 1978, pp. 240–245.
- [31] J. M. KLEIN AND H. TAKAMORI, *Parallel line assignment problems*, Management Sci. 19 (1972), pp. 1–10.
- [32] D. KÖNIG, *Graphs and matrices*, Mat. Fiz. Lapok, 38 (1931), pp. 116–119.
- [33] H. W. KUHN, *The Hungarian method for the assignment problem*, Naval Res. Logist. Quart., 2 (1955), pp. 83–97.
- [34] E. L. LAWLER, *Combinatorial Optimization*, Holt, Rinehart and Winston, New York, 1976.
- [35] J. M. LEWIS, *On the complexity of the maximum subgraph problem*, Proc. Tenth Annual ACM Symposium on Theory of Computing, 1978, pp. 265–274.
- [36] L. MIRSKY, *Transversal Theory*, Academic Press, New York, 1971.



- [37] J. MÜHLBACHER, *F-factors of graphs: a generalized matching problem*, Information Processing Lett., 8 (1979), pp. 207–214.
- [38] T. J. SHAEFER, *The complexity of satisfiability problems*, Proc. Tenth Annual ACM Symposium on Theory of Computing, 1978, pp. 216–226.
- [39] W. T. TUTTE, *The factorisation of linear graphs*, J. London Math. Soc., 22 (1947), pp. 107–111.
- [40] M. YANNAKAKIS, *Node- and edge-deletion NP-complete problems*, Proc. Tenth Annual ACM Symposium on Theory of Computing, 1978, pp. 253–264.