Maximum Cardinality 1-restricted Simple 2-matchings*

David Hartvigsen
University of Notre Dame
Notre Dame, IN 46556-5646
Hartvigsen.1@nd.edu

Submitted: Mar 19, 2007; Accepted: Oct 14, 2007; Published: Nov 5, 2007
Mathematics Subject Classification: 05C70, 05C38, 90C27

Abstract

A *simple 2-matching* in a graph is a subgraph all of whose nodes have degree 1 or 2. A simple 2-matching is called *k-restricted* if every connected component has $> k$ edges. We consider the problem of finding a k-restricted simple 2-matching with a maximum number of edges, which is a relaxation of the problem of finding a Hamilton cycle in a graph. Our main result is a min-max theorem for the maximum number of edges in a 1-restricted simple 2-matching. We prove this result constructively by presenting a polynomial time algorithm for finding a 1-restricted simple 2-matching with a maximum number of edges.

Keywords: matchings, Hamilton cycle

1 Introduction

All graphs considered in this paper are undirected with no parallel edges or loops. A *simple 2-matching* in a graph is a subgraph all of whose nodes have degree 1 or 2. (For the sake of brevity, we henceforth drop the adjective *simple*.) Hence the connected components of a 2-matching are paths and cycles. A well-studied problem in the literature is to find a 2-matching in a graph with a maximum number of edges, a so-called *maximum 2-matching*. A polynomial time algorithm, Tutte-type theorem and Tutte-Berge-type theorem are known for this problem, among many other things (see [19] for an excellent survey). In this paper we present analogous results for a restricted version of this problem, which we describe next.

A *k-restricted 2-matching* is a 2-matching such that each connected component has $> k$ edges. The *k-restricted 2-matching problem* is to find a k-restricted 2-matching with

*An extended abstract (without proofs) of an earlier version of this paper appeared in [10]. The main theorem in the current paper, a min-max theorem for the maximum number of edges in a 1-restricted simple 2-matching, did not appear in [10].

a maximum number of edges, which we call a *maximum k-restricted 2-matching*. Hence the 0-restricted 2-matching problem is to find a maximum 2-matching and the 1-restricted 2-matching problem is to find a maximum 2-matching that contains no "isolated edges." Note that the k-restricted problem is a relaxation of the problem of finding a Hamilton cycle in a graph due to the following observation: A graph G with n nodes has a Hamilton cycle if and only if G has a k-restricted 2-matching with n edges, where $k > \lfloor n/2 \rfloor$.

Before describing our main results, let us introduce some notation. Let $\nu_k(G)$ denote the number of edges in a maximum k-restricted 2-matching for a graph G. Note that $n \geq \nu_0(G) \geq \nu_1(G) \geq \nu_2(G) \geq \cdots$.

Our main result is a min-max formula for $\nu_1(G)$. An analogous result for $\nu_0(G)$ appears in [19] and is proved using a construction of Tutte [22] and the Tutte-Berge theorem [2]; the Tutte-Berge theorem is an analogous result for classical matchings (hence we refer to such results as "Tutte-Berge-type"). In addition, we present a formula for $\nu_0(G) - \nu_1(G)$ and we characterize those graphs G for which $\nu_0(G) = \nu_1(G)$. (These two additional results were stated without proof in [10].) An analogous result, characterizing the graphs for which $n = \nu_0(G)$, was obtained by Belck [1] and Gallai [9], and, in a more general form, by Tutte [21]; an analogous result for classical matchings was also proved by Tutte [20]) (hence we refer to such theorems as "Tutte-type").

In this paper, we also present a polynomial time algorithm for the 1-restricted 2-matching problem. The algorithm's validity and our main result are proved simultaneously. The algorithm is a modified and simplified version of an algorithm that appeared in [13] for a related problem (see discussion below). An interesting feature of the algorithm is that it starts with a maximum 2-matching and modifies it to obtain a maximum 1-restricted 2-matching. As a corollary to the validity of the algorithm, we also show that in every graph there always exists a maximum 2-matching that can be transformed into a maximum 1-restricted 2-matching by removing its isolated edges. (This corollary also follows easily from an analogous result in [13].)

We next discuss a closely related problem that has been studied in the literature. A *C_k-free 2-matching* is a 2-matching that contains no cycles of length $\leq k$. An algorithm to find a maximum C_3-free 2-matching appears in [11] and an algorithm to find a maximum C_4-free 2-matching in bipartite graphs appears in [12]. See also [18]. This problem is a relaxation of the Hamilton cycle problem, in the same way the k-restricted 2-matching problem is. The edge-weighted version of this problem on complete graphs can be shown to provide increasingly accurate approximations for the travelling salesman problem as k increases (see [6]). The C_5-free 2-matching problem is known to be NP-hard (see [3] for a proof due to Papadimitiou). Related work can be found in [3], [4], and [7].

Let us discuss one other related line of research. A *k-piece* is a connected graph with maximum degree equal to k. A *k-piece packing* in a graph is a subgraph whose connected components are k-pieces. The *node-max k-piece packing problem* is to find a k-piece packing in a graph that contains a maximum number of nodes; the *edge-max k-piece packing problem* is to find a k-piece packing in a graph that contains a maximum number of edges. Observe that the node-max and edge-max 1-piece packing problems are equivalent to the classical matching problem. However, for higher values of k, the node-max and edge-max problems are different from one another (that is, the sets of feasible solutions are not identical). Finally, note that the edge-max 2-piece packing problem is identical to the 1-restricted 2-matching problem.

For $k \geq 2$, only the node-max version of the k-piece packing problem has been considered in the literature. The node-max 2-piece packing problem was recently considered by Kaneko in [16], where he presented a Tutte-type theorem. This result was extended in [17] by Kano, Katona, and Király, where the authors presented a Tutte-Berge-type theorem. These results were further extended by Hartvigsen, Hell, and Szabó in [13], where the authors presented a polynomial time algorithm and Tutte-type and Tutte-Berge-type theorems for the node-max k-piece packing problem. Finally, a Gallai-Edmonds decomposition theorem for the general problem is presented in [15] by Janata, Loebl, and Szabó.

Let us consider two algorithmic approaches to proving the main results in this paper. The first is to use the algorithm in [13] for the node-max k-piece packing problem. For the case $k = 2$, this algorithm also solves the edge-max 2-piece packing problem, which is equivalent to our problem, the 1-restricted 2-matching problem. (For $k > 2$, the algorithm in [13] can be modified, in a straightforward manner, to polynomially solve the edge-max k-piece packing problem[1].) However, the algorithm in [13] is complex, even for $k = 2$. The key factor in this complexity is the need to identify so called "critical" subgraphs (called "blossoms" by Edmonds in [5] for $k = 1$, "suns" in [16] for $k = 2$, and "galaxies" in [13] in general). Proving the results in this paper, by making reference to results in [13], requires defining suns, proving some results about their structure, describing how suns appear in the algorithm in [13], and relying on the validity of the algorithm in [13]. It turns out that with only a little more work we can develop, from scratch, an algorithm for solving the 1-restricted 2-matching problem. There are several advantages of this second approach: (1) the resulting algorthm does not require the use of suns (or any "complex" critical graphs), hence it is considerably simpler (and shorter) than the algorithm in [13]; (2) we gain better insight into the problem and the results being proved; and (3) the proof is self-contained. Furthermore, optimal solutions produced by this simpler algorithm need not be optimal solutions to the node-max 2-piece packing problem, hence the simpler algorithm is clearly different from the algorithm in [13] (thus illustrating the above observation that the node-max and edge-max 2-piece packing problems are not identical). So we have decided to take the latter approach in this paper. Let us finally remark that Tutte- and Tutte-Berge-type theorems also appear in [13] for the node-max k-piece packing problem, but they have a different form from the corresponding results that appear in this paper (since we are dealing with edges here instead of nodes), and do not directly yield the results that appear here.

There is a fairly large literature addressing problems like the ones mentioned above: algorithms for finding (node or edge) maximum subgraphs in a graph whose connected components are in special classes, and theorems describing structural properties of these subgraphs. Much of this work has been recently surveyed in [14] by Hell (see also [19]).

[1] For $k > 2$, change the definition of *tip value of a galaxy* in [13] from the minimum number of nodes in a tip of the galaxy to the minimum number of *edges* in a tip of the galaxy. The proof of the modified algorithm's validity for $k = 2$ and $k > 2$ must also be changed.

The paper is organized as follows: Section 2 contains statements of our main theorems. Section 3 contains a description of a key subgraph in the algorithm, called an alternating structure, and how it interacts with the rest of the graph that contains it. Section 4 contains the statement of the main algorithm and an analysis of its complexity. Section 5 contains the proofs of our main theorems and a proof of the algorithm's validity.

2 The Theorems

In this brief section we state our main theorems. We also state some closely related results from the literature. The proofs of our main results make use of these related results and the structures in the algorithm, and appear in Section 5.

For a graph $G = (V, E)$ and $T \subset V$, we let $G - T$ denote the graph obtained from G by deleting the nodes in T (and all edges incident to these nodes). A connected component of G that contains exactly one edge is called an *isolated edge* of G. We let $Edges(G)$ denotes the number of isolated edges in G. For node-disjoint subgraphs H_1 and H_2 of G, we let $E[H_1, H_2]$ denote the number of edges of G with one endnode in H_1 and the other endnode in H_2. If M is a 2-matching, then the 2-matching obtained by removing the isolated edges from M is called the *1-restricted 2-matching associated with M*.

The following theorem characterizes the maximum size of a 2-matching. The statement is taken from the treatment in Schrijver [19]; its proof is based on a construction of Tutte [22] and the Tutte-Berge Theorem [2]. We present this result for comparison with our main results and also because we use in it proving our main results.

Theorem 1 *For any graph $G = (V, E)$, $\nu_0(G)$ is equal to the minimum value of*

$$|V| + |U| - |I| + \sum_K \left\lfloor \frac{1}{2} E\left[K, I\right] \right\rfloor$$

where U and I are disjoint subsets of V, I is stable, and where K ranges over the connected components of $G - U - I$.

The following is an easy corollary (the statement is also taken from [19]), which was first proved by Belck [1]

Corollary 2 *For any graph $G = (V, E)$, $n = \nu_0(G)$ iff*

$$|I| \leq |U| + \sum_K \left\lfloor \frac{1}{2} E\left[K, I\right] \right\rfloor$$

for all disjoint subsets U and I of V, where I is stable, and K ranges over the connected components of $G - U - I$.

Our main results follow. They characterize the maximum size of a 1-restricted 2-matching and the difference between the maximum sizes of a maximum 2-matching and a maximum 1-restricted 2-matching.

Theorem 3 *For any graph $G = (V, E)$, $\nu_1(G)$ is equal to the minimum value of*

$$|V| + |U| - |I| + \sum_K \left\lfloor \frac{1}{2} E\left[K, I\right] \right\rfloor - \{Edges(G - T) - 2 * |T|\}$$

where U and I are disjoint subsets of V, $T \subseteq U$, I is stable, and K ranges over the connected components of $G - U - I$.

Theorem 4 *For any graph $G = (V, E)$,*

$$\nu_0(G) - \nu_1(G) = \max_{T \subset V} \{Edges(G - T) - 2 * |T|\}.$$

Oberve that Theorem 3 is not a direct combination of Theorems 1 and 4, because it requires $T \subseteq U$. Similarly, Theorem 4 does not immediately follow from Theorems 1 and 3. Theorem 4 easily yields the following corollary.

Corollary 5 *For any graph $G = (V, E)$, $\nu_0(G) = \nu_1(G)$ if and only if*

$$2 * |T| \geq Edges(G - T), \text{ for every } T \subset V.$$

Observe that we obtain an if and only if characterization of the graphs G for which $n = \nu_1(G)$ by simply replacing $n = \nu_0(G)$ with $n = \nu_1(G)$ in Corollary 2 (since $n = \nu_0(G)$ iff $n = \nu_1(G)$).

The following result follows immediately from the validity of the algorithm.

Theorem 6 *In every graph there exists a maximum 2-matching whose associated 1-restricted 2-matching is maximum.*

3 The Alternating Structure

The main objective of this section is to define a special graph called an alternating structure. It plays a central role in the main algorithm. After defining it we explore some of its properties.

The *length* of a simple path is its number of edges.

Suppose M is a maximum 2-matching in a graph G. Let S be a subgraph of G and let S_M denote the subgraph of S induced by the edges of M (with no isolated nodes). We say that S is an *alternating structure* in G with respect to M if it has the following properties (see Example 7 and Figure 1 below):

1. S is a forest.

2. The connected components of S_M are simple paths of length 4. The endnodes of these paths are called *extreme nodes* and have degree 1 in M. The nodes adjacent to extreme nodes in these paths are called *near-extreme nodes*. The remaining nodes in these paths are called *center nodes*.

3. Each connected component in S has exactly one node that does not occur in S_M. We call such nodes *roots*.

4. Each root is contained in an isolated edge of M. This edge is not in S.

5. Each center node c is incident with exactly one edge of S that is not in M. The other endnode of this edge is not a center node of S and is not contained in the same connected component of S_M as c.

Let v be a node in an alternating structure S. Then the unique path in S from v to a root is called the *v-root path* in S.

Example 7 *Figure 1 contains an example of an alternating structure S. Bold edges denote edges in M. The solid nodes denote center nodes; the remaining nodes consist of four roots, a, b, c and d, and the extreme and near-extreme nodes. All the edges shown are in S, except ab and cd. The unique path from v to a is the v-root path.*

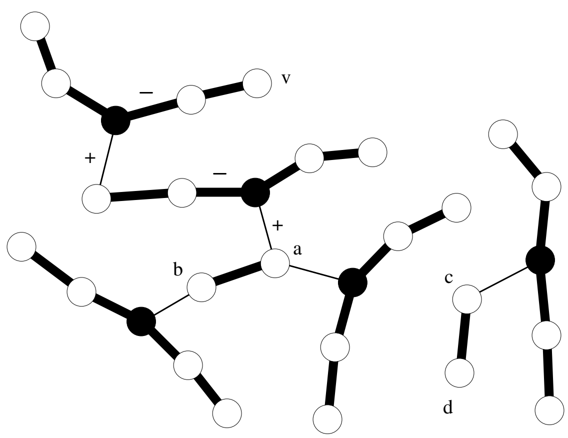

Figure 1: An alternating structure S (where edges ab and cd are not in S)

The following procedure is useful in analyzing alternating structures. It also plays a key role in several other procedures, introduced below, that are used in the main algorithm for finding maximum 1-restricted 2-matchings.

Path exchange
Input: A graph G with a maximum 2-matching M, an alternating structure S with respect to M, and a non-center node v in S.
Action: Let P be the v-root path in S. For every center node x of P do the following: Let xy be the unique edge in P and M; and let xz be the unique edge in P and not in M. Remove xy from M and add xz to M.

Example 8 *In the alternating structure S in Figure 1, the path exchange from v is performed along the v-root path by removing from M the edges with a minus sign and adding to M the edges with a plus sign.*

Remark 9 *The subgraph M of G obtained after performing a path exchange is a maximum 2-matching.*

We next list the possible ways that an alternating structure can interact with the graph that contains it. In Proposition 11, we show that certain of these possibilities cannot occur. For the remaining possibilities, we show, in three procedures, how S can be modified.

Let G be a graph with a maximum 2-matching M and an alternating structure S with respect to M. Let vw be an edge of G, but not of S or M, where v is a non-center node in S.

First, suppose w is not in S; we have the following possibilities:

- A. w is not contained in a connected component of M.
- B. w is contained in a connected component, say C, of M:
  - i. C is a path and w is an endnode of C.
  - ii. C is a path and w is an interior node of C:
    - a. C is a path of length 4 and w is the centernode of C.
    - b. Otherwise.
  - iii. C is a cycle

Second, suppose w is a non-center node in S. Let P_v and P_w denote the v-root and w-root paths, respectively. We have the following possibilities:

- C. v and w are in different trees of S.
- D. v and w are in the same tree of S:
  - i. v is on P_w (or w is on P_v, which we do not consider due to symmetry):
    - a. w is extreme.
    - b. w is near-extreme.
  - ii. v is not on P_w and w is not on P_v:
    - a. v or w is extreme.
    - b. v and w are near-extreme.

Remark 10 *In possibility (Di), w cannot be a root because it is "above" v in the tree. In possibility (Dii), neither v nor w can be a root; if one were, then it would be on the root path of the other and we would have condition (Di).*

Proposition 11 *Possibilities (A), (Bi), (C), (Dia), and (Diia) cannot occur.*

Proof. Suppose possibility (A), (Bi), (Dia), or (Diia) holds. (If (Diia) holds, then assume, without loss of generality, that w is extreme.) Perform a path exchange on v and add vw to M. Then the new subgraph M is a 2-matching with more edges than the original 2-matching M, which is a contradiction of M having maximum cardinality.

Suppose possibility (C) holds. Perform path exchanges on v and w and add vw to M. Then the new subgraph M is a 2-matching with more edges than the original 2-matching M, which is again a contradiction. ■

We next consider the possibilities not addressed in Proposition 11. One of two things can be done. Either S can be enlarged (or "grown") or M can be modified (or "augmented") so that the new associated 1-restricted 2-matching has one more edge. The details are contained in the following three procedures.

Modify M and S by Growing
Input: G, M, S, vw, and C that satisfy possibility (Biia).
Action: Add to S the subgraph C and the edge vw.

Remark 12 *The subgraph S that is output by "Modify M and S by Growing" is an alternating structure.*

Modify M and S by a Type-1 Augmentation
Input: G, M, S, vw, and C that satisfy possibilities (Biib) or (Biii).
Action: (Note: If C is a path, then w is an interior node of C.)
Case: C is a path and there exists an edge wx in C where x has degree 1 in C. Perform a path exchange from v, remove wx from M, and add vw to M.
Case: C is a path and there does not exists an edge wx in C where x has degree 1 in C. Then (since we are not in possibility (Biia) or the previous case) there must be a degree 1 node in C, say y, that is distance 3 or more from w in C. Let x be the node of C adjacent to w and between w and y on C. Perform a path exchange from v, remove wx from M, and add vw to M.
Case: C is a cycle. Let wx be an edge of C. Perform a path exchange from v, remove wx from M, and add vw to M.

Modify M and S by a Type-2 Augmentation
Input: G, M, S, and vw that satisfy possibility (Dib) or (Diib).
Action: Let x be the extreme node of S adjacent to w in S_M. Perform a path exchange from v, remove wx from M, and add vw to M.

Proposition 13 *The Type-1 and Type-2 Augmentations result in a new maximum 2-matching M, whose associated 1-restricted 2-matching has one more edge than the 1-restricted 2-matching associated with the inputed 2-matching M.*

Proof. We leave the proof to the reader. ■

4 The Algorithm

In this section we present the main algorithm and analyze its complexity. We make use of the following simple procedure several times in the main algorithm.

Initialize the Alternating Structure S
Input: A graph G and a 2-matching M.
Action: Let S consist of the endnodes of all isolated edges of M.

The main algorithm follows:

Maximum 1-restricted 2-matching Algorithm
Input: A graph G.
Action:
Obtain a maximum 2-matching M;
Initialize the Alternating Structure S;
While M contains isolated edges and there exists an edge vw, which is not in S or M, where v and w are not center nodes and v is in S, do
  Case: w is not in S;
    If possibility (Biia) holds, then Modify S by Growing;
    Else Modify M and S by a Type-1-Augmentation;
    Initialize the Alternating Structure S;
  Case: w is in S;
    Modify M and S by a Type-2-Augmentation;
    Initialize the Alternating Structure S;
Output the 1-restricted 2-matching associated with M.

We next analyze the complexity of the main algorithm. Let n denote the number of nodes and e the number of edges in a graph G.

Proposition 14 *The Maximum 1-restricted 2-matching Algorithm has worst case time complexity of* $O(en^2)$

Proof. Let us first consider the complexity of obtaining a maximum 2-matching M in G. A polynomial time reduction of Tutte [22] allows the problem of finding a maximum 2-matching in a graph to be reduced to the problem of finding a maximum cardinality matching, which, of course, can be done in polynomial time using Edmonds' algorithm [5]. However, this problem can be solved more efficiently by using a reduction algorithm due to Gabow [8], with a complexity of $O(n^{\frac{1}{2}}e)$.

The first execution of Initialize the Alternating Structure S can be performed in $O(n)$ time. Subsequent executions only require constant time since both Augmentation procedures simply eliminate one isolated edge of M.

The algorithm alternates between growing S (i.e., running the Modify S by Growing procedure) and performing augmentations (i.e., running one of the two Augmentation procedures each of which enlarges the 1-restricted 2-matching associated with M). During a growing phase, if we are careful, we need to look at each edge in G only once (in the

While statement). Since each growth of S requires constant time and since S can grow to size $O(n)$, it takes at most $O(e)$ time to grow S. The algorithm augments at most $O(n)$ times and each such augmentation requires $O(n)$ time. Hence the overall complexity of the algorithm is $O(en^2)$ time. ■

5 Proofs

In this section we present proofs of the results in Section 2 as well as a proof of the validity of the algorithm for finding a maximum 1-restricted 2-matching. We preface this with a couple of definitions and by proving a lower bound on $\nu_0(G) - \nu_1(G)$ and an upper bound on $\nu_1(G)$.

Let $G = (V, E)$ be a graph. We let $|G|$ denote the number of edges in G. Furthermore, if M is a 2-matching in G, then the *deficiency* of M is defined to be $\sum_{v \in V} (2 - \deg_M(v))$, where $\deg_M(v)$ denotes the degree of node v in M.

Lemma 15 *For any graph $G = (V, E)$,*

$$\nu_0(G) - \nu_1(G) \geq Edges(G - T) - 2 * |T|, \text{ for any } T \subset V.$$

Proof. Let M be a maximum 1-restricted 2-matching and let M' be a 2-matching that contains M as a subgraph and, subject to this, has a maximum number of edges. Let $T \subset V$. Observe that there are at most $2 * |T|$ edges of M' with one endnode in T and the other endnode contained in $Edges(G-T)$. Hence there are are least $Edges(G-T)-2*|T|$ isolated edges in $G - T$ that are not adjacent to an edge of M' in G and therefore must be in M' (due to its maximality). Thus we have $|M'| - |M| \geq Edges(G - T) - 2 * |T|$. The result now follows since $|M'| \leq \nu_0(G)$ and $|M| = \nu_1(G)$. ■

Lemma 16 *For any graph $G = (V, E)$,*

$$\nu_1(G) \leq |V| + |U| - |I| + \sum_K \left\lfloor \frac{1}{2} E[K, I] \right\rfloor - \{Edges(G - T) - 2 * |T|\}$$

where U and I are disjoint subsets of V; $T \subseteq U$; I is stable; and where K ranges over the connected components of $G - T - I$.

Proof. The result follows immediately from Theorem 1 and Lemma 15. Furthermore, the result is true even if T is not a subset of U. ■

We next prove Theorems 3 and 4 together, since their proofs share some structures.

Proof of Theorems 3 and 4. We prove Theorem 4 first. Run the algorithm on G; let M be the resulting maximum 2-matching; let R be the associated 1-restricted 2-matching; and let $S = (V^S, E^S)$ be the final alternating structure. If M has no isolated edges, then let T be the empty set; the result follows from Lemma 15. So let us assume this is not the case. Let T be the set of center nodes of S. We show that for this choice of T,

$$\nu_0(G) - \nu_1(G) = Edges(G - T) - 2 * |T|,$$

which proves Theorem 4.

Consider the set F of isolated edges in $G-T$. By the structure of S at the end of the algorithm, F contains (at least) the following two types of edges:

F_1 = the set of edges xy in M where x is an extreme node and y is a near-extreme node;

F_2 = the set of edges xy in M where x and y are roots of S.

By the properties of S, we have the following two facts:

- Every edge in F_1 is adjacent to exactly one edge in M and the other endnode of this edge is in T.
- Every node in T is incident with exactly two edges of M whose other endnodes are in an edge of F_1.

Thus no edge of $F - F_1$ can be adjacent to an edge in M. Hence all such edges must be isolated edges in M, since M is a maximum cardinality 2-matching. Observe that the isolated edges of M are precisely the edges in F_2 (due to the initialization of S), hence $F = F_1 \cup F_2$. Thus, the number of isolated edges in M is equal to $Edges(G-T) - 2 * |T|$. Hence, we have that $|M| - |R| = \nu_0(G) - |R| = Edges(G-T) - 2 * |T|$. Thus, by Lemma 15, R is a maximum 1-restricted 2-matching, which concludes the proof of Theorem 4. (This also shows that the algorithm works.)

To prove Theorem 3, we show that there exist sets U and I in G such that

$$\nu_0(G) = |V| + |U| - |I| + \sum_K \left\lfloor \frac{1}{2} E\left[K, I\right] \right\rfloor,$$

where T, the node set defined above, is a subset of U. With this, Theorem 3 follows from Theorem 4.

To begin, let $V^* = V - V^S$, and define $S^* = (V^*, E^*)$ to be the subgraph of G induced by V^*. Apply Theorem 1 to S^*. Let U^* and I^* be optimal sets that yield the minimum in the theorem. Let I^F be a set of nodes consisting of one node from each edge in F. Let $U = U^* \cup T$ and let $I = I^* \cup I^F$. Observe that, by the structure of S, I is stable and there are no edges of M from S to S^*. Thus $\nu_0(S^*) + \nu_0(S) = \nu_0(G)$ and, by our choice of U^* and I^* and Theorem 1, we have

$$\nu_0(S^*) = |V^*| + |U^*| - |I^*| + \sum_K \left\lfloor \frac{1}{2} E\left[K, I^*\right] \right\rfloor.$$

So it remains to show that

$$\nu_0(S) = |V^S| + |T| - |I^F| + \sum_K \left\lfloor \frac{1}{2} E\left[K, I^F\right] \right\rfloor.$$

To see that this is sufficient, observe that there are no edges from $S-T$ to S^* (otherwise we would have continued growing S in the algorithm). Hence the components K considered in the above two summations (and the edges from them to I) are the same as the components K considered in the summation in Theorem 3.

First, observe that, by our choice of I^F and the structrue of S, $E\left[K, I^F\right] = 1$, for all connected components K of $V^S - T - I^F$. Thus the above summation term for $\nu_0(S)$ equals zero. Second, by the structure of S, $S - T$ is a set of isolated edges (i.e., F), hence $2\,|F| = 2|I^F|$ is the minimum number of deficiencies in $S - T$. Hence, $2|I^F| - 2\,|T|$ is the minimum number of deficiencies in S. Therefore $\nu_0(S) = \left|V^S\right| - \frac{1}{2}\left[2\left|I^F\right| - 2|T|\right]$, and the result follows. ■

Proof of Corollary 5. Clearly, if $2 * |T| \geq Edges(G - T)$, for every $T \subset V$, then max in Theorem 4 is 0 and it is achieved by $T = \emptyset$. Hence $\nu_0(G) = \nu_1(G)$. Conversely, if $2 * |T| < Edges(G - T)$, for some $T \subset V$, then $\nu_0(G) > \nu_1(G)$ by Theorem 15. ■

Theorem 17 *The 1-restricted 2-matching Algorithm finds a maximum 1-restricted 2-matching.*

Proof. The theorem follows immediately from the proof of Theorem 4. ■

Proof of Theorem 6. The theorem follows immediately from Theorem 17, since the matching M in the algorithm is always a maximum 2-matching and the algorithm outputs the 1-restricted 2-matching associated with the final M. ■

References

[1] H.-B. Belck, Reguläre Faktoren von Graphen, Journal für die reine und angewandte Mathematik 188 (1950) 228-252.

[2] C. Berge, Sur le couplate maximum d'un graphe, Comptes Rendus Hebdomadaires des Séances de l'Académie Sciences [Paris] 247 (1958) 258-259.

[3] G. Cornuéjols and W.R. Pulleyblank, A matching problem with side conditions, Discrete Math. 29 (1980) 135-159.

[4] W.H. Cunningham and Y. Wang, Restricted 2-factor polytopes, Mathematical Programming 87 (2000) 87-111.

[5] J. Edmonds, Paths, trees and flowers, Canad. J. of Math. 17 (1965) 449-467.

[6] M.L. Fisher, G.L. Nemhauser and L.A. Wolsey, An analysis of approximations for finding a maximum weight Hamiltonian circuit, Operations Res. 27, 4 (1979) 799-809.

[7] A. Frank, Restricted t-matchings in bipartite graphs, Discrete Appl. Math. 131, 2, (Sept. 2003) 337-346.

[8] H.N. Gabow, An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems, in: Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing (Boston, 1983), The Association for Computing Machinery, New York, 1983, 448-456.

[9] T. Gallai, On factorisation of graphs, Acta Mathematica Academiae Scientiarum Hungaricae 1 (1950) 133-153.

[10] D. Hartvigsen, 1-restricted simple 2-matchings (extended abstract), in Electronic Notes in Discrete Mathematics 18, Elsevier (2004) 145-149.

[11] D. Hartvigsen, Extensions of Matching Theory, Ph.D. Thesis, Department of Mathematics, Carnegie-Mellon U. (1984); under the supervision of Gérard Cornuéjols.

[12] D. Hartvigsen, Finding maximum square-free 2-matchings in bipartite graphs, J. Combin. Theory, Series B 96 (2006) 693-705.

[13] D. Hartvigsen, P. Hell, J. Szabó, The k-piece packing problem, J. Graph Theory 52:4 (August 2006) 267-293.

[14] P. Hell, Packing in graphs, Electronic Notes in Discrete Mathematics, 5 (2000).

[15] M. Janata, M. Loebl, and J. Szabó, A Gallai-Edmonds type theorem for the k-piece packing problem, Elec. J. Combin. 12 (2005).

[16] A. Kaneko, A necessary and sufficient condition for the existence of a path factor every component of which is a path of length at least two, J. Combin. Theory, Series B 88 (2003) 195-218.

[17] M. Kano, G.Y. Katona, and Z. Király, Packing paths of length at least two, Discrete Math., 283 (2005) 129-135.

[18] Y. Nam, Matching Theory: Subgraphs with Degree Constraints and other Properties, Ph. D. Thesis, Department of Mathematics, University of British Columbia (1994); under the supervision of Richard Anstee.

[19] A. Schrijver. Combinatorial Optimization, Polyhedra and Efficiency. Springer 2003.

[20] W.T. Tutte, The factorization of linear graphs, J. London Math. Soc. 22 (1947) 107-111.

[21] W.T. Tutte, The factors of graphs, Canad. J. Math. 4 (1952) 314-328.

[22] W.T. Tutte, A short proof of the factor theorem for finite graphs, Canad. J. Math. 6 (1954) 347-352.