



May 2017

Two short proofs of the Perfect Forest Theorem

Yair Caro

University of Haifa-Oranim, yacaro@kvgeva.org.il

Josef Lauri

University of Malta, josef.lauri@um.edu.mt

Christina Zarb

University of Malta, christina.zarb@um.edu.mt

Follow this and additional works at: <http://digitalcommons.georgiasouthern.edu/tag>



Part of the [Discrete Mathematics and Combinatorics Commons](#)

Recommended Citation

Caro, Yair; Lauri, Josef; and Zarb, Christina (2017) "Two short proofs of the Perfect Forest Theorem," *Theory and Applications of Graphs*: Vol. 4 : Iss. 1 , Article 4.

Available at: <http://digitalcommons.georgiasouthern.edu/tag/vol4/iss1/4>

This article is brought to you for free and open access by the Journals at Digital Commons@Georgia Southern. It has been accepted for inclusion in Theory and Applications of Graphs by an authorized administrator of Digital Commons@Georgia Southern. For more information, please contact digitalcommons@georgiasouthern.edu.

Abstract

A perfect forest is a spanning forest of a connected graph G , all of whose components are induced subgraphs of G and such that all vertices have odd degree in the forest. A perfect forest can be thought of as a generalization of a perfect matching since, in a matching, all components are trees on one edge. Scott first proved the Perfect Forest Theorem, namely, that every connected graph of even order has a perfect forest. Gutin then gave another proof using linear algebra.

We give two very short proofs of the Perfect Forest Theorem which use only elementary notions from graph theory. Both of our proofs yield polynomial time algorithms for finding a perfect forest in a connected graph of even order.

1 Introduction

We consider a graph G on n vertices. A spanning subgraph F of a graph G is called a *perfect forest* if

- F is a forest
- the degree of every vertex in F is odd
- each subtree of F is an induced subgraph of G .

An important graph structure is a *matching*, which is a set of pairwise non-adjacent edges, that is without common vertices. A *maximum matching* is a matching that contains the largest possible number of edges, and such a matching is said to be *perfect* if it includes all the vertices in the graph [5]. It is well-known that a perfect/maximum matching in a graph can be found in polynomial time [2].

In the case of a connected graph G on n vertices where n is even and G is without a perfect matching, one would like to find other structures that are in a sense reminiscent of perfect matchings. Clearly in a perfect matching each edge is induced and both degrees are odd. This leads to the above definition of perfect forests. Establishing the existence of a perfect forest, and finding one in polynomial time are comparable to such questions with respect to perfect matchings.

Taking another point of view, the question of the maximum size of an induced subgraph of G with all degrees odd has also been treated in several papers, and is also related to perfect forests [1, 6, 7].

Scott [8] proves that every connected graph G with an even number of vertices contains a perfect forest. We call this the Perfect Forest Theorem. Scott's relatively long proof is graph theoretical and constructive. Gutin [3] provides a

shorter proof using simple linear algebra arguments, which in turn can give a polynomial time algorithm to find a perfect forest.

Gutin and Yeo consider perfect forests in directed graphs in [4]. They look at four generalizations to directed graphs of the concept of a perfect forest and extend Scott's theorem to digraphs in a non-trivial way.

An interesting algorithmic application of perfect forests can be found in [9], where Sharan and Wigderson use perfect forests, which they call pseudo-matchings, in an algorithm which yields a perfect matching in a bipartite cubic graph. This again justifies finding short proofs for the Perfect Forest Theorem, which yield a polynomial time algorithm.

Motivated by this interest, we provide two simple graph theoretic proofs of the Perfect Forest Theorem, one by induction on the number of vertices, and the other by induction on the number of edges. Both proofs yield a polynomial time algorithm for finding a perfect forest in a graph with an even number of vertices.

In our proofs, we define the *union* of two graphs G_1 and G_2 , denoted by $G_1 \cup G_2$, as the graph with vertex set $V(G_1) \cup V(G_2)$ and edge set $E(G_1) \cup E(G_2)$. Also, let v be a vertex of a tree T — the branches at v are defined to be the maximal subtrees of the forest $T - v$.

2 Two short proofs

We start with a simple lemma which will be used in the first proof.

Lemma 2.1. *Let G be a connected graph on n vertices where n is even, and G is not a tree. Then G has a spanning tree with at least two vertices of even degree.*

Proof. Let T be a spanning tree of G . Then clearly, the number of vertices of even degree must be even since the total number of vertices is even.

Let us assume that all vertices of T have odd degree. Since G is not a tree, there must be at least one edge $e \in E(G \setminus T)$. Let this edge join the vertices u and v . Consider the tree T^* obtained from T by adding the edge e and deleting the edge connecting u to w where w is the unique neighbour of u on the path from u to v in T . Then the degree of v and w are now even, and the spanning tree T^* is as required. \square

Theorem 2.2. *Let G be a connected graph on n vertices where n is even. Then G has a perfect forest.*

Proof. First proof.

We prove this by induction on n . For $n = 2$, the statement is clearly true. So suppose $n \geq 4$ is even.

1. If G is a tree with all degrees odd, then G is the required perfect forest.

2. If G is a tree with at least one vertex w of even degree, let the branches of w be B_1, \dots, B_t , where t is even. Then at least one of these branches, say B_1 , contains an even number of vertices, otherwise the number of vertices in G is odd. Let G_1 be the induced subgraph of G on B_1 , and G_2 the subgraphs induced by w and B_2, \dots, B_t . Clearly G_1 and G_2 are both connected and have even order, and hence by induction, there exists perfect forests F_1 and F_2 in G_1 and G_2 respectively. Hence, $F_1 \cup F_2$ is a perfect forest of G .
3. If G is not a tree, then by Lemma 2.1, G has a spanning tree T with at least two vertices of even degree, and hence by the same argument presented in (2) above, T enables us to split G into two vertex-disjoint connected components each of even order, G_1 and G_2 , and applying induction on both components gives a perfect forest.

Remark: This proof tells us that either G is a tree with all vertices having odd degree or $V(G)$ can be split as $V = A \cup B$, $A \cap B = \emptyset$ and both A and B induce connected components each of even order (hence induction applies).

Now we give the second proof.

Second Proof.

We now use induction on the number of edges m . The case $m = 1$ is trivial. So assume $m \geq 2$ and n is even. Consider $G - e$, for some $e \in E(G)$.

1. Consider the case where $G - e$ is disconnected. If the two components G_1 and G_2 each have an even number of vertices, then by the induction hypothesis, they each have a perfect forest F_1 and F_2 respectively, and $F_1 \cup F_2$ is a perfect forest in $G - e$ and hence in G .

If G_1 and G_2 each have an odd number of vertices, then consider $G'_1 = G_1 + e$ and $G'_2 = G_2 + e$. Each has an even number of vertices because we are adding a new vertex to each, and hence by the induction hypothesis, they each have a perfect forest F'_1 and F'_2 respectively. But then $F'_1 \cup F'_2$ is a perfect forest for G .

2. If $G - e$ is still connected, by the induction hypothesis there exists a perfect forest F in $G - e$. If the edge e joins different components of F , then F is still a perfect forest in G . If, on the other hand, e joins two vertices, say u and v , in the same component H of F , let K be the union of the other components of F (we are not excluding the possibility that K is empty, that is, $F = H$, that is, F is a tree).

Now consider the subgraph induced in G by the vertices of H . The key fact here is that this induced subgraph of G contains no edges other than those in $E(H)$ except e , since H is a connected induced subgraph of $G - e$.

Therefore $H + e$ is an induced subgraph of G . This induced subgraph $H + e$ now has exactly one cycle C which contains the edge e . Consider $H - E(C)$, that is, the subgraph of H obtained by removing all the edges of this cycle. Now $H - E(C)$ contains no cycles so it is forest all of whose components are induced sub-trees of G because every edge in $E(C)$ joins two vertices in different components of $H - E(C)$. Now, the degrees of all vertices in $H - E(C)$ except u and v are reduced by 2 when the edges in $E(C)$ are removed. Therefore they are still odd in $H - E(C)$ since they were odd in H . The degrees of u and v in $H - E(C)$ are, however, one less than in H since they are reduced by 1 when removing the edges of $E(C)$. Of course the degrees of the other vertices of F , those in K , are unchanged.

So we put back the edge e into $H - E(C)$, giving the graph H' , in which all vertices, including u and v , have odd order. Also, H' is induced in G by the key fact pointed out above. Therefore F' , the forest containing the union of H' and K rather than H and K , is a perfect forest in G .

□

Remark: Clearly, both proofs are constructive and can be implemented in polynomial time.

3 Conclusion

Perfect forests have attracted the attention of various graph theorists as one natural extension of the idea of perfect matchings. In the literature two proofs of the Perfect Forest Theorem were known, one using techniques from linear algebra. We here give two very short proofs which use only elementary notions from graph theory, both of which yield polynomial time algorithms for finding a perfect forest in a graph of even order.

References

- [1] Y. Caro. On induced subgraphs with odd degrees. *Discrete Mathematics*, 132(1):23–28, 1994.
- [2] Z Galil. Efficient algorithms for finding maximum matching in graphs. *ACM Comput. Surv.*, 18(1):23–38, March 1986.
- [3] G. Gutin. Note on perfect forests. *Journal of Graph Theory*, 82:229–231, 2016.
- [4] G. Gutin and A. Yeo. Note on perfect forests in digraphs. *Journal of Graph Theory*, 2016. to appear.

- [5] L. Lovász and M.D. Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.
- [6] A.J. Radcliffe and A.D. Scott. Every tree contains a large induced subgraph with all degrees odd. *Discrete mathematics*, 140(1):275–279, 1995.
- [7] A.D. Scott. Large induced subgraphs with all degrees odd. *Combinatorics, Probability and Computing*, 1(04):335–349, 1992.
- [8] A.D. Scott. On induced subgraphs with all degrees odd. *Graphs and Combinatorics*, 17(3):539–553, 2001.
- [9] R. Wigderson and Sharan A. A new afc algorithm for perfect matching in bipartite cubic graphs. In *Proceedings of the Fourth Israel Symposium on Theory of Computing and Systems: June 10-12, 1996, Jerusalem, Israel*, volume 96, page 202. IEEE Computer Society, 1996.