AN ALGORITHM FOR THE ENUMERATION OF SPANNING TREES

PAWEL WINTER*

DIKU, Institute of Datalogy, University of Copenhagen, Sigtunaengade 41,
DK-2200 Copenhagen N, Denmark

Abstract.

Enumeration of spanning trees of an undirected graph is one of the graph problems that has received much attention in the literature. In this paper a new enumeration algorithm based on the idea of contractions of the graph is presented. The worst-case time complexity of the algorithm is $O(n + m + nt)$ where $n$ is the number of vertices, $m$ the number of edges, and $t$ the number of spanning trees in the graph. The worst-case space complexity of the algorithm is $O(n^2)$. Computational analysis indicates that the algorithm requires less computation time than any other of the previously best-known algorithms.

Keywords: Graph Theory, Spanning Tree, Enumeration Algorithm.

1. Introduction.

The problem of enumerating all spanning trees of a graph arises for example when a spanning tree satisfying certain criteria has to be determined but these criteria are so complicated or cannot be quantified that the direct approach is either impractical or impossible.

Enumeration of spanning trees in undirected graphs has been studied widely in recent years and several algorithms of varying efficiency have been developed (e.g. [2, 3, 4, 5, 8, 9, 10]). An excellent survey of algorithms that appeared prior to 1970 has been given by Chase [3].

The best algorithm in terms of worst-case asymptotic bounds is that of Gabow and Myers [4]. It requires $O(m + n + nt)$ time and $O(n + m)$ space where $n$, $m$ and $t$ are the number of vertices, edges and spanning trees respectively. The time bound $O(n + m + nt)$ is optimal to within a constant factor in the sense that listing the edges of all spanning trees requires $O(nt)$ time. If, however, a more structured form of output is acceptable, it may be possible to lower the bound $O(n + m + nt)$.

In 1968 Char [2] presented a conceptually simple algorithm requiring in the worst case $O(n + m + n(t + t_0))$ time where $t_0$ is the number of non-tree subgraphs.
with \( n - 1 \) edges. This algorithm has been analyzed by Jayakumar and Thulasiraman [7]. According to their computational analysis based on a number of randomly generated graphs, Char’s algorithm is superior to the Gabow and Myers algorithm. Furthermore, it becomes more and more efficient as the number of spanning trees increases. This is due to the fact that Gabow and Myers’ algorithm requires rather extensive bookkeeping while Char’s algorithm merely examines all possible sequences of \( n - 1 \) edges and discards those that do not form a spanning tree.

In 1970 Chase [3] developed an algorithm based on the idea of factoring, i.e., spanning trees are generated by adding to the currently connected subtree \( T_i \), a vertex \( v_j \) not yet in \( T_i \); all edges connecting \( v_j \) with the vertices of \( T_i \) are processed together in a single iteration. Although Chase did not derive the worst-case time complexity bound for his algorithm, he compared its performance with a large number of previously known algorithms (including Char’s algorithm and various versions of factoring algorithms). The derived computational results indicate the superiority of his algorithm.

In this paper we present a new algorithm for the enumeration of all spanning trees in an undirected graph. This algorithm requires \( O(n + m + nt) \) time and \( O(n^2) \) space. If a more structured form of output than the explicit list of all spanning trees is acceptable, the worst-case time complexity reduces to \( O(n + m + \min\{nt, n!(n-1)/2\}) \). The algorithm is based on the idea of consecutive contractions of the graph. To some extent the algorithm is related to that of Gabow and Myers. First it constructs all spanning trees containing some selected edge \( e_1 \), then all spanning trees containing another edge \( e_2 \) but not \( e_1 \), etc. However, by the initial, very simple, relabeling of the vertices, our algorithm does not require any bridge-tests as used in Gabow and Myers’ algorithm. The relabeling combined with an appropriately chosen sequence of contractions reduces the bookkeeping to very few pointer-adjustments which to some extent are similar to those employed in Char’s algorithm. However, while Char’s algorithm tests all sequences of \( n - 1 \) edges (also those not forming a spanning tree), our algorithm examines only combinations for which at least one spanning tree exists. Furthermore, while Char’s algorithm examines one sequence (or part of a sequence) at a time, our algorithm examines several different sequences simultaneously. This stems from the fact that when a graph is contracted, some of the edges become parallel. Hence our algorithm resembles the factoring algorithms with the exception that it expands forests rather than subtrees and, as mentioned previously, does not require any bridge-tests which, similar to the Gabow and Myers algorithm, are necessary in the factoring algorithms.

This paper is organized as follows. Most important concepts used throughout the paper are introduced in §2. In §3 it is shown how spanning trees can be enumerated by means of contractions of the underlying graph. The algorithm is described in detail in §4. Its time and space complexity is discussed in §5. Further modifications of the algorithm which improve its efficiency considerably are