TECHNION-Israel Institute of Technology
Computer Science Department

אלון עי

COVERING A GRAPH BY CIRCUITS

by

Alon Itai* and Michael Rodeh**

Technical Report #104

July 1977

* Technion-Israel Institute of Technology, Computer Science Dept., Haifa, Israel.
** IBM Israel Scientific Center, Haifa, Israel.

COVERING A GRAPH BY CIRCUITS

by

Alon Itai

Technion, Israel Institute of Technology, Haifa, Israel

Michael Rodeh

IBM Israel Scientific Center, Haifa, Israel

ABSTRACT

A circuit cover is a set of circuits which cover all the edges of a graph; its length is the sum of the lengths of the circuits. In analyzing irrigation systems and electrical circuits it is necessary to find a short circuit cover. It is shown that every bridge-free connected undirected graph with n vertices and e edges has a circuit cover the length of which is less than or equal to e+2nlogn. A probabilistic algorithm for finding such a cover is presented; its expected running time is $O(n^2)$, independent of the input graph. This constitutes one of the first examples of solving a graph-theoretical problem by a probabilistic algorithm - the class of algorithms introduced by Rabin.

If the graph contains two edge-disjoint spanning trees then there exists a circuit cover of length at most e+n-1.

The relationship of the circuit cover problem to the Chinese postman problem is also discussed.

1. INTRODUCTION

In analyzing irrigation systems by the Hardy Cross method [C] all the edges of a connected undirected graph must be covered by circuits. The set of resulting circuits is called a circuit cover. Once found, a certain function must be computed along these circuits. Since the computational effort depends on the sum of the lengths of the circuits (the length of the cover), it is worthwhile to first find a short circuit cover. A similar problem can arise in analyzing electrical circuits.

This problem is related to that of the Chinese postman [EJ] : to find the shortest tour such that each edge is traversed at least once. (That is, the postman must deliver mail along each edge of a graph and return to his starting point.) A circuit cover can be easily converted to a tour, which need not be the shortest. However, not every Chinese postman tour can be decomposed into a circuit cover, even if such a cover exists. For example, the tour illustrated in Figure 1 can not be converted to a circuit cover because a circuit may not contain an edge more than once.

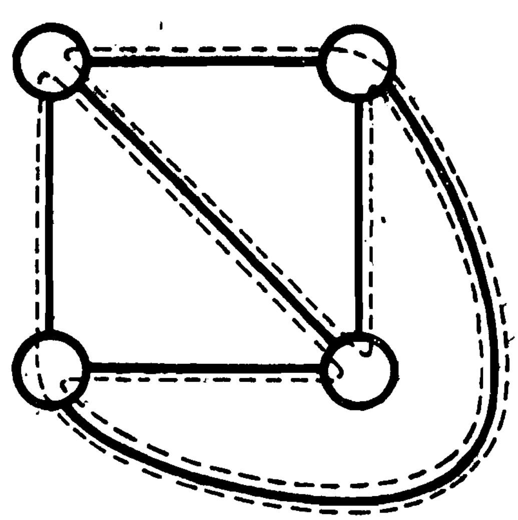


Fig. 1

A graph has a circuit cover if and only if it does not contain any bridge (an edge the deletion of which increases the number of connected components). Henceforth, we assume that all the input graphs are connected and bridge-free. (These properties can be checked in linear time [AHU].)

Note that an Euler circuit is a circuit cover in which each edge is covered exactly once. Therefore, a connected graph G=(V,E) with n vertices and e edges has a circuit cover of length e if and only if it is an Euler graph (equivalently, the degree of each vertex is even). Note also that the edges of each of the faces of a planar graph form a circuit cover which covers every edge exactly twice.

Let T be a spanning tree of a graph G=(V,E). Each edge (u,v) of G-T together with the unique path from u to v in T form a circuit. These e-(n-1) circuits are called the <u>fundamental circuits</u> of the graph with respect to T. Obviously, the fundamental circuits form a circuit cover. However, the length of such a cover may be as large as $O(n^3)$ (e.g. for the complete graph with n vertices).

We prove that any bridge-free graph has a circuit cover the length of which is at most e+2nlogn. A probabilistic algorithm (an algorithm which contains some random choices) for finding such a cover is presented. It takes $O(n^2)$ time on the average, independent of the input graph. This algorithm may not terminate (with probability zero) and as such it belongs to Rabin's third category of probabilistic algorithms [R]. In fact it constitutes one of the first examples of using probabilistic algorithms for graph-theoretical problems. A deterministic version of the algorithm requires at most $O(n^3)$ time. We also show that every graph which contains two edge-disjoint spanning trees has a circuit cover of length $e+n-1$.

Both the circuit cover and the Chinese postman problems can be defined for strongly connected directed graphs. However, a solution for one of the problems directly yields a solution for the other. A shortest tour may be found in $O(n^3 \log n)$ time by constructing a Hitchcock transportation problem [EJ] and solving it using the scaling method of Edmonds and Karp [EK].

2. A REDUCTION TO SPARSE GRAPHS

To construct a cover of length $e+O(n \log n)$, we first cover at least $e-n$ edges and then cover the remainding edges by circuits contained in an auxiliary subgraph. In fact, every undirected graph G contains subgraphs H and F such that:

(i) H is connected, bridge-free and has n vertices and at most $2n-3$ edges.

(ii) F is a forest contained in H.

(iii) Each connected component of G-F is an Euler graph.

Therefore, G-F is a set of edge-disjoint circuits. This set can be extended to a circuit cover of G by adding a set of circuits in H which covers every edge of F.

To find H and F, let T be a depth-first-search (DFS) spanning tree of G [AHU]. Now conduct a DFS on T. Each tree-edge is traversed twice, once in the forward direction and once backwards. When traversing the tree-edge (u,v) backwards from v to u, if the degree of v in G is odd, then delete the edge (u,v) from G. Let F be the set of the deleted edges. F is a forest since it is a subgraph of the tree T. Each connected component of the graph G-F is an Euler graph since the degree of each vertex is even. Let H be a subgraph of G which contains T and, in addition, at most one back edge from each vertex: for each vertex v from which back edges emanate we choose the back edge which leads to the lowest vertex (the vertex first visited by the DFS). H is connected, bridge-free, and has n vertices and at most 2n-3 edges, since H contains n-1 tree-edges and at most n-2 back edges, (back edges may emanate from all the vertices except the root and its sons). A cover for F in H and an Euler circuit for every connected component of G-F yields a circuit cover for G. Figure 2 illustrates a graph G, a DFS tree T, the corresponding forest F, and the subgraph H of G.

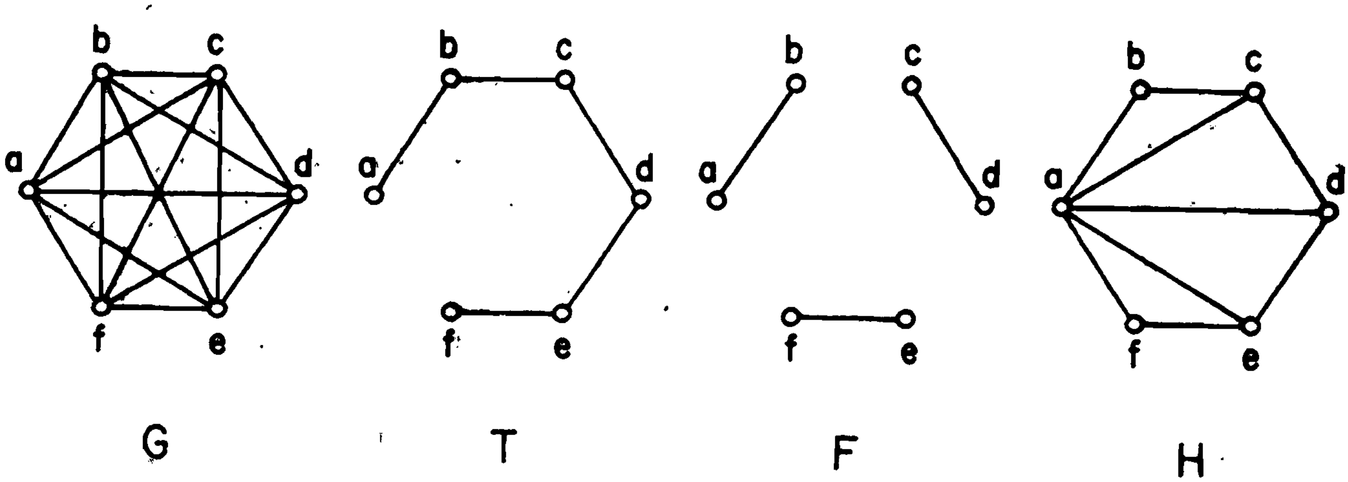


Fig. 2

3. COVERING THE FOREST F

A <u>generalized circuit</u> (g-circuit) is a union of edge-disjoint simple circuits. A set B of g-circuits is an <u>F-cover</u> if every edge of F belongs to at least one g-circuit of B. B is a <u>g-minimum F-cover</u> if it is an F-cover with the smallest number of g-circuits. A subset S of an F-cover B is <u>exact</u> (with respect to F) if there exists at least one edge in F which belongs to all the g-circuits of S and to no g-circuit of B-S. An F-cover is <u>irreducible</u> if all its nonempty subsets are exact.

We cover F by a set of g-circuits. The initial F-cover is the set of fundamental circuits obtained from the spanning tree T. We then try to reduce the number of g-circuits. The basic tool for reducing the number of g-circuits in a given F-cover B is the procedure REDUCE. REDUCE(B,S) accepts an F-cover B and a subset S of B. If S is not exact then B is updated and its cardinality is decreased by one.

```
   procedure REDUCE(B,S);
   begin comment let S={c_1,...,c_s};
     if s=1 then
1.   begin if every edge of c_1 is covered by B-S then B:=B-S end
2.   else begin S':={c_i ⊕ c_{i+1} |1≤i<s};
3.          if (B-S)∪S' is an F-cover then B:=(B-S)∪S'
        end
   end
```

To implement REDUCE we prepare a list $NCTE_B$ which for every edge $f \in F$ contains the number of g-circuits in the current F-cover B which pass through f. Initially the list can be computed in $O(n^2)$ time, and then updated without affecting the asymptotic running time.

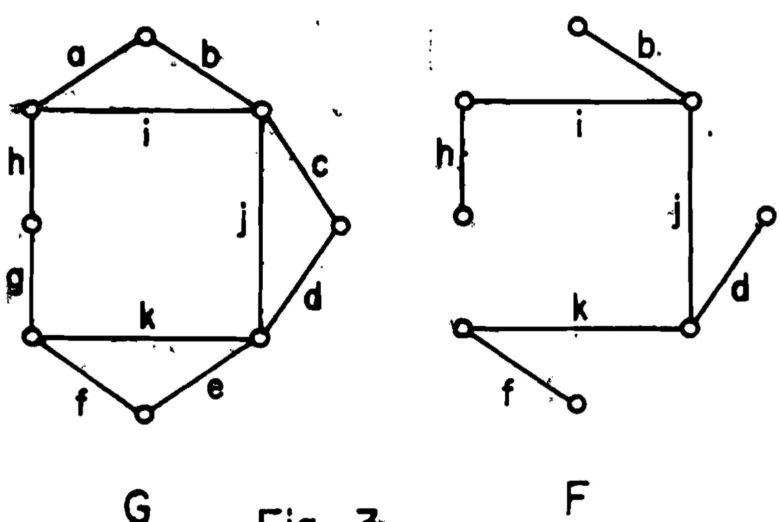
<u>Lemma 1</u>: The execution time of REDUCE(B,S) is $O(n|S|)$.

<u>Proof</u>: Line 1 can be done simply by checking whether $NCTE_B(f) > 1$ for every $f \in c_1$. Line 2 requires $O(n|S|)$ time (remember that the number of edges in H is at most 2n-2). Line 3 can be executed by checking whether every entry in the updated $NCTE_B$ is non-zero. Q.E.D.

<u>Lemma 2</u>: Every g-minimum F-cover is irreducible.

<u>Proof</u>: Assume to the contrary that S is a non-exact subset of a g-minimum F-cover B. Apply REDUCE(B,S). The resultant F-cover has fewer g-circuits than the original one, contradicting the g-minimality of B. Q.E.D.

Note that not every irreducible F-cover is g-minimum, e.g. the F-cover B_1 for F and G as in Figure 3 is an irreducible F-cover but not g-minimum since B_2 contains fewer g-circuits,



$B_1 = \{c_1 = \{a,b,c,d,k,g,h\},$
$c_2 = \{a,b,j,e,f,g,h\},$
$c_3 = \{i,c,d,e,f,g,h\}\}$

$B_2 = \{c_4 = \{a,b,c,d,e,f,g,h\},$
$c_5 = \{i,j,k,g,h\}\}$

Fig. 3

Lemma 3: Every F-cover contains at most n-1 exact subsets.

Proof: For every exact subset S of an F-cover B, there exists at least one edge $f_S \in F$ which is covered by all g-circuits of S and by no other g-circuit of B. These edges are all distinct. Since F contains at most n-1 edges, B may contain at most n-1 such subsets. Q.E.D.

Lemma 4: Every irreducible F-cover contains at most $\lfloor \log n \rfloor$ g-circuits.

Proof: Let B be an irreducible F-cover. By Lemma 3, B contains at most n-1 exact subsets. By definition every nonempty subset of B is exact. Therefore, $2^{|B|}-1 \leq n-1$ or $|B| \leq \lfloor \log n \rfloor$. Q.E.D.

Theorem 1: Let G be a bridge-free graph with n vertices and e edges. Then G has a circuit cover the length of which is no more than $e+(2n-3)\lfloor \log n \rfloor$.

Proof: In the previous section we obtained a sparse subgraph H and a forest $F \subseteq H$. All the edges of G-F may be covered by exactly one g-circuit the length of which is at most e. Since H is bridge-free it has a g-minimum F-cover B. By Lemmas 2 and 4, B is irreducible and contains at most $\lfloor \log n \rfloor$ g-circuits. Since each of these g-circuits is contained in H and H has at most 2n-3 edges, the length of each g-circuit is at most 2n-3. Therefore, the length of the F-cover is at most $(2n-3)\lfloor \log n \rfloor$ which completes the proof. Q.E.D.

A graph G with n vertices is dense if it contains at least $n \log n$ edges.

Corollary 1: Every dense graph has a circuit cover the length of which is linear in the number of edges.

4. FINDING IRREDUCIBLE F-COVER

To produce an irreducible F-cover we proceed in two steps: First we use the procedure LG to construct an F-cover consisting of at most $\lceil \log n \rceil$ g-circuits. Then we apply the procedure IR to obtain an irreducible F-cover.

```
    procedure LG;
    begin B:=the set of fundamental circuits of H w.r.t. T;
1.    while (|B| choose 2)≥2n do
2.    begin S:=a random subset of B of cardinality 2;
3.       call REDUCE(B,S) end;
4.    while |B|>⌈logn⌉ do
5.       begin S:=a random non-empty subset of B of cardinality at
               most ⌈logn⌉;
6.       call REDUCE(B,S) end
    end
```

The procedure LG is probabilistic. Theoretically, it may loop forever without finding a non-exact subset S of B. Lemma 5 implies that in practice this does not happen. Later the lemma is used to estimate the behavior of the algorithm.

Lemma 5: The probability is less than or equal to 1/2 that the subset S of B chosen at random in Line 2 or Line 5 of LG is exact.

Proof: At Line 2, $\binom{|B|}{2} \geq 2n$. Since there are at most n-1 exact subsets (Lemma 3) the probability of choosing one is at most $(n-1)/2n<1/2$.

At Line 5, $|B| \geq 1+\lceil \log n \rceil$. Therefore, the number of non-empty subsets of B with cardinality at most $\lceil \log n \rceil$ is $\sum_{i=1}^{\lceil \log n \rceil} \binom{|B|}{i} \geq \sum_{i=1}^{\lceil \log n \rceil} \binom{1+\lceil \log n \rceil}{i} = 2^{\lceil \log n \rceil + 1} - 2 \geq 2n-2$. Therefore, the probability of choosing an exact subset in Line 5 is at most $(n-1)/(2n-2)=1/2$. Q.E.D.

Lemma 6: The average execution time of LG is $O(n^2)$.

Proof: Since the probability of choosing an exact subset either in Line 2 or in Line 5 is less than or equal to 1/2 (Lemma 5) we conclude that REDUCE will not be invoked more than twice on the average without reducing the cardinality of B. Therefore, Line 3 is executed no more than twice on the average until $|B|$ decreases. Since $|B|<n-1$, Lines 2-3 are repeated no more than $2(n-1)=O(n)$ times on the average. By Lemma 1, each invocation of REDUCE in Line 3 requires $O(n|S|)=O(2n)=O(n)$ time. Therefore, the average execution time of Lines 1-3 is bounded by $O(n^2)$.

At Line 4 $\binom{|B|}{2}<2n$ and thus $|B|(|B|-1)<4n$ which implies $|B|<1+2\sqrt{n}$. Since REDUCE is called no more than twice on the average until $|B|$ decreases, Lines 5-6 are repeated $2(1+2\sqrt{n})=O(\sqrt{n})$ times on the average. By Lemma 1 each invocation of Line 5 requires $O(n \log n)$ time, hence the loop of Lines 5-6 requires at most $O(n^{1.5} \log n) \leq O(n^2)$ time on the average. Q.E.D.

LG is a probabilistic algorithm which produces an F-cover B with cardinality at most $\lceil \log n \rceil$. The final step is to obtain an irreducible F-cover using the deterministic algorithm IR(B) below.

<u>procedure</u> IR(B);

1. <u>begin</u> <u>while</u> there exists a nonempty subset S of B <u>do</u>
2.   <u>call</u> REDUCE(B,S)

<u>end</u>

Let $B=\{c_1, c_2, \ldots, c_b\}$ and $p=2^b-1$. Let $S_1, S_2, \ldots, S_p$ be the list of nonempty subsets of B sorted in lexicographic order. Let $NCTE_S$ be a list which records the number of g-circuits of S covering every edge f of F. S is exact if and only if there exists an edge f of F for which $NCTE_B(f)=NCTE_S(f)=|S|$. To produce $NCTE_{S_i}$ we use $NCTE_{S_{i-1}}$. The time needed for such a computation is $O(n|S_{i-1} \oplus S_i|)$. Therefore, the total execution time of Line 1 for a given B is at most $O(n \sum_{i=2}^{p} |S_{i-1} \oplus S_i|)$. For the lexicographic order, this expression is $O(np)$.

When a non-exact subset is found, the value of p is halved. Therefore, the total execution time of Line 1 is bounded by $O(np \sum_{i=0}^{\infty} 2^{-i})=O(np)$.

Lemma 1 and $|B| \leq \lceil \log n \rceil$ imply that the total execution time of Line 2 is bounded by $O(n \log^2 n)$.

Using $p \leq 2^{\lceil \log n \rceil} \leq 2n$ we conclude:

<u>Theorem 2</u>: An irreducible F-cover may be produced in $O(n^2)$ time on the average.

Similar considerations, but using deterministic instead of probabilistic methods, yield the following result:

Theorem 3: An irreducible F-cover may be produced in at most $O(n^3)$ time.

5. COVERING A GRAPH WHICH CONTAINS TWO EDGE-DISJOINT SPANNING TREES

Let G be a graph which contains two edge-disjoint spanning trees, T_1 and T_2. In Section 2 a method was presented for deleting some of the edges of a spanning tree to yield an Euler subgraph. We use the same method to cover G. First, we delete m edges of T_1 and obtain a connected Euler subgraph of G, which may be covered by e-m edges. Only m edges of T_1 have not yet been covered. To cover them we apply the same method to the graph H consisting of T_2 and the uncovered edges of T_1. Since T_2 is a spanning tree of H deleting some edges of T_2 yields a subgraph containing all the m uncovered edges. Moreover, each connected component of this subgraph is an Euler graph. The length of this cover is at most m+n-1. Combining these two covers yields a result which covers every edge at most twice and whose length is at most e+n-1. The running time of the method is bounded by O(ne) - the time required to find two disjoint spanning trees or show that no such two trees exist [K].

6. RELATIONSHIP TO THE CHINESE POSTMAN AND OPEN PROBLEMS

Every Euler circuit induces a Chinese postman tour. On the other hand, every postman tour P of G defines an Euler multigraph G': G' contains G and some additional copies of the edges of G, so that the multiplicity of every edge is equal to the number of times it is traversed in the tour. Let D be the set of edges which appear in G' but not in G.

Lemma 7: D does not contain circuits.

Proof: Assume that D contains a circuit C. Deleting C from G' yields a multigraph G'' which also contains G. Since G' is an Euler graph so is G''. Therefore, an Euler circuit for G'' constitutes a postman tour which is shorter than P, contradicting the minimality of P.

Corollary 2: The Chinese postman tour consists of at most e+n-1 edges.

Note that this corollary holds even if the edges have positive weights (and the problem is to find a tour of minimum weight).

Every circuit cover induces a postman tour. It is unknown to us whether a reduction in the opposite direction is possible, i.e., given a postman tour, does there exist a circuit cover in which each edge is covered the same number of times as in the tour.

Weaker versions of this problem are:

(i) Does every graph have a circuit cover of length is less than or equal to $e+n-1$?

(ii) Does every graph have a circuit cover in which each edge is covered at most twice?

Possible other extensions are to find a cover of minimum length and to investigate the problem of finding a circuit cover of small weight in a graph with weights on the edges.

REFERENCES

|AHU| A. V. Aho, J. E. Hopcroft and J. D. Ullman, The Design and Analysis of Computer Algorithms, Addison-Wesley (1974).

|C| H. Cross, "Analysis of Flow in Networks of Conduits of Conductors" Bull. No. 286, Univ. of Illinois Engineering Experimental Station, Urbana, Ill. (1936).

|EJ| J. Edmonds and E. L. Johnson, "Matching, Euler Tours and the Chinese Postman", Math. Programming 5(1973) 88-124.

|EK| J. Edmonds and R. M. Karp, "Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems", J. ACM 19(1972) 248-264.

|K| T. Kameda, "On Maximally Distant Spanning Trees of a Graph", Computing 17(1976) 115-119.

|L| C. L. Liu, Introduction to Combinatorial Mathematics, McGraw-Hill (1968).

|R| M. O. Rabin, "Probabalistic Algorithms", Proc. Sym. on New Directions and Recent Results in Algorithms and Complexity, Carnegie-Mellon University, Academic Press (April 1976).