COMPLEMENTARY SPANNING TREES

BY

GEORGE B. DANTZIG

TECHNICAL REPORT NO. CS 126
MARCH 1969

COMPUTER SCIENCE DEPARTMENT
School of Humanities and Sciences
STANFORD UNIVERSITY

COMPLEMENTARY SPANNING TREES

BY

GEORGE B. DANTZIG

TECHNICAL REPORT NO. 126

MARCH 1969

COMPUTER SCIENCE DEPARTMENT
STANFORD UNIVERSITY
STANFORD,CALIFORNIA

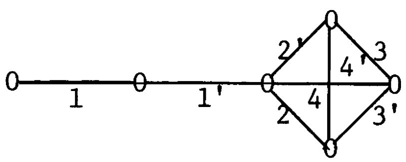
Research of George B. Dantzig and reproduction of this report was partially supported by National Science Foundation Grant GP-6431; Atomic Energy Commission, Contract AT[04-3] 326 PA #18; Office of Naval Research, Contracts N00014-67-A-0112-0011 and N00014-67-A-0112-0016; U.S. Army Research Office, Contract DAHC-04-67-C0028; and National Institutes of Health Grant GM 14789,

COMPLEMENTARY SPANNING TREES

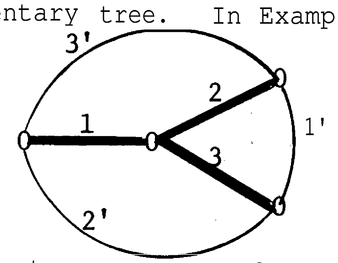
BY

GEORGE B. DANTZIG

Given a network G whose arcs partition into non-overlapping "clubs" (sets) R_i. D. Ray Fulkerson has considered the problem of constructing a spanning tree such that no two of its arcs belong to (represent) the same club and has stated necessary and sufficient conditions for such trees to exist [1].

(1)  [1]

In Example (1) no such "representative" tree exists. When each club R_i consists of exactly two arcs, we shall refer to each of the arc pair as the "complement" of the other, and the representative tree as a complementary tree. In Example (2) the heavy arcs {1,2,3}

(2) 

form such a tree. The complements of {1,2,3}, namely {1',2',3'} form a cycle. However, {1',2',3} form another complementary tree. Our objective is to prove

Main Theorem: <u>If there exists one complementary tree, there exists at least two.</u>

The general idea is to pass from one complementary tree to the other by a sequence of "adjacent" (or "neighboring") trees which are "almost" complementary, An almost-complementary tree is defined to be one where each set R_i furnishes exactly one arc with the exception of one "special" set which furnishes two and one other set which furnishes none. In Example (2), the almost complementary trees with respect to the special set {1,1'} are {1,1',2}, {1,1',2'}, {1,1',3} and {1,1',3'}. A sequence leading from {1,2,3} to {1',2',3} along a path of adjacent almost-complementary trees is {1,2,3}, {1,1',3}, {2',1',3}.

Two trees are said to be adjacent or neighbors if they differ by one arc. The general procedure for generating a sequence of adjacent almost-complementary trees is as follows: Start with a complementary tree, Add to it any out-of-tree arc, say A', forming a cycle.

Step I: If either A or A' is another arc of the cycle, delete it and terminate if the new tree thus formed is complementary. If not,

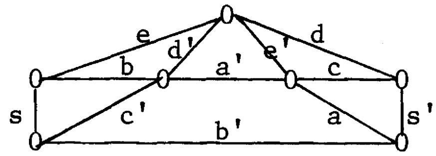
Step II: Arbitrarily*drop some other arc of the cycle forming an adjacent almost-complementary tree with respect to AA'.

Step III: Introduce as out-of-tree arc the complement of the arc dropped in Step II. Return to Step I.

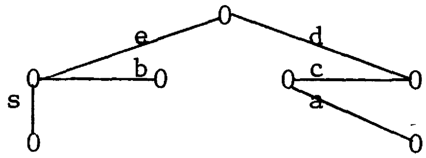
Note especially that the sequence of almost-complementary trees thus generated all contain A, A' as the special pair of arcs. In all discussion that follows the "almost" is defined with respect to a fixed pair of special arcs.

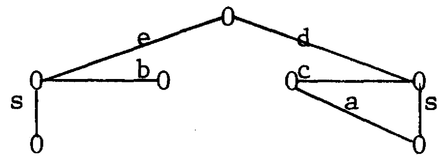
* This will be changed later.

Let us see what happens if we apply these steps to Example (3).

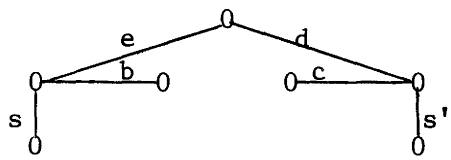
(3) 

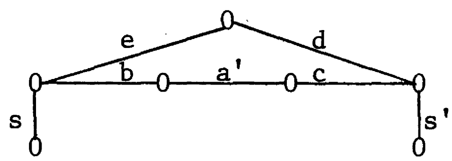
The given starting complementary tree is {a,b,c,d,e,s} , see (4).

(4) 

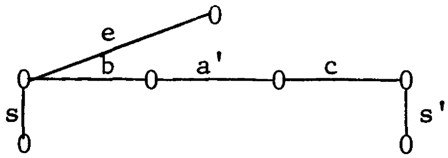
(4a) 

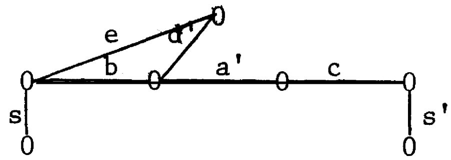
In (4a) we have chosen s' as the starting out-of-tree arc so that the sequence (path) of adjacent almost-complementary trees generated by the rules will be with respect to the special set **s,s'** . According to Step II we can elect to break the cycle by arbitrarily dropping arc a to obtain (5). Since a is dropped, Step III requires that a' , its complement, must be the next out-of-tree arc see (5a).

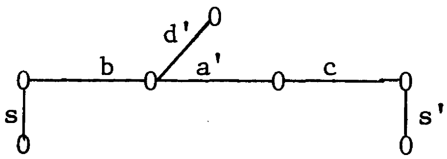
(5) 

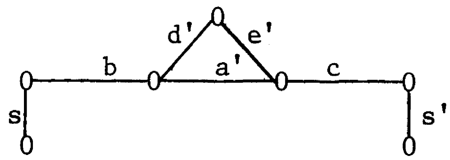
(5a) 

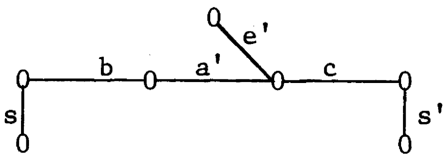
We arbitrarily break the cycle by dropping d, see (6), then in (6a) introduce its complement d'. Next we drop e and introduce e', see (7) and (7a). Next we drop d' and introduce back d, see (8) and (8a). Next we drop e' and introduce back e, see (9) and (9a).

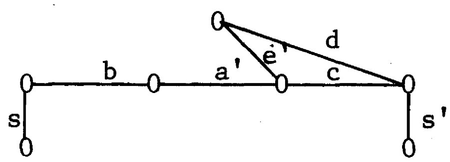
(6) 

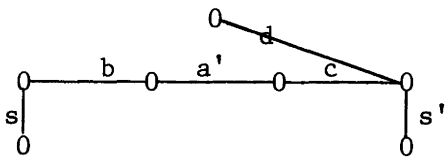
(6a) 

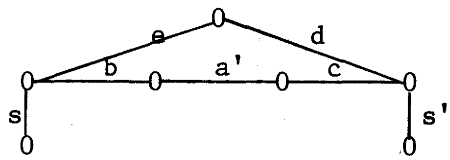
(7) 

(7a) 

(8) 

(8a) 

(9) 

(9a) 

Note that **(9a)** is identical to (5a) and our rules allow us to drop d so that we return to **(6),** Le., the path circles back on itself.

Thus we see in Example (3) that the idea of moving from one **almost-**complementary tree to the next by arbitrarily dropping an arc of a cycle fails to terminate with another complementary tree. Instead it generates a. cycle of almost-complementary trees that repeat ad infinitum. Note that Tree (6) is adjacent to Tree (5) as well as Tree (7) and Tree (9).

What we need is a modified rule for dropping an arc of a cycle so that each almost-complementary tree so generated is adjacent to exactly two others, one or both of which could be completely complementary. If this could be arranged it is easy to see that the method would never repeat an almost-complementary tree nor could it return to the original complementary tree because we have arranged it so that there is only one path out of it. We need a dropping rule which would give rise to a set S of trees which satisfy the following abstract properties:

(i) Given a finite set S and a relation "neighbor".

(ii) If i is a neighbor of j then j is a neighbor of i.

(iii) No element has more than two neighbors.

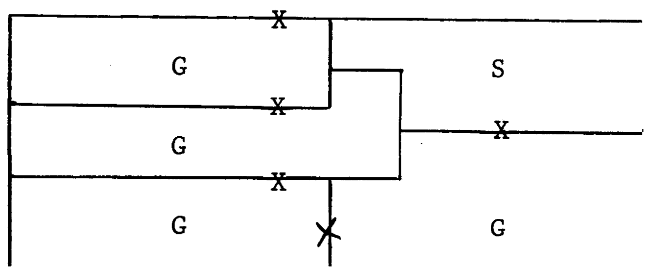
(iv) At least one element has exactly one neighbor.

Theorem: S contains at least two elements with exactly one neighbor.

This type of theorem is used by Euler to resolve the Koenigsberg Bridge problem. Lemke and Howson were the first to turn the underlying idea into a constructive procedure for proving theorems by rigging the network relations to have the abstract properties. Lemke showed that the complementary pivot algorithms used to solve linear and positive-definite quadratic programming problems could be modified to find complementary solutions to bi-matrix games and certain other non-convex problems [See References 2-9.].

Curtis Eaves tells the following Ghost Story to illustrate Lemke's principle, Once upon a time, there was a haunted house. A brave lad entered the front door, (Doors are marked by an x in (10).) Suddenly, he saw a Ghost. He turned to flee but a gust of wind slammed shut the front door and it would not open. He ran from the room through a second door only to discover himself in another room with a Ghost. He fled from room to room with a Ghost hoping to find sanctuary by exiting through a door which led to the outside or led to a room without a Ghost. The house had property that if a room contained a Ghost it had exactly two doors. Query, did the brave young man find Sanctuary?

(10)



Lemke was able to apply his principle because his elements ("rooms") were a selected subset of the extreme points of a convex set. Two elements were adjacent if they had an edge (door) in common. We shall establish the main theorem by setting up a correspondence between certain trees of graph G and certain extreme points of a linear program, namely the following network flow problem:

Arbitrarily order the nodes in G . Next orient each arc (i,j) as a directed arc from i to j, if $i < j$ and from j to i if $j < i$. Assign to the arcs of the given complementary tree arbitrary values $a_{ij} > 0$ and $a_{ji} = -a_{ij}$ if (i,j) is a directed arc of the tree, for all other (i,j) let $a_{ij} = 0$. Let node values $b_i = \sum_j a_{ij}$.

The network flow problem is then to find $x_{ij} \geq 0$ such that

$$\sum_{i \epsilon U_j} x_{ij} - \sum_{k \epsilon V_j} x_{jk} = b_j$$

where $U_j = \{i | (i,j)$ is a directed arc of $G\}$

$V_j = \{k | (j,k)$ is a directed arc of $G\}$

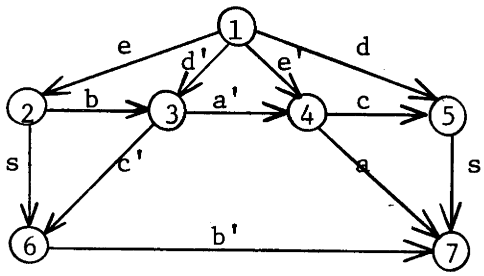
It is well known that the arcs (i,j) corresponding to basic variables (feasible or not) form a tree, If feasible basic solutions are non-degenerate and the feasible set is bounded, then a new basic feasible solution can be obtained by increasing sufficiently the flow x_{ij} on a directed out-of-tree arc (i,j) while adjusting the flows on basic arcs. The arc dropping out of the cycle will then correspond to the unique basic variable whose value decreased to zero.

Uniqueness is a consequence of non-degeneracy, One way to avoid degeneracy is to assign as the n-l arc flows of the starting complementary tree n-l different powers of $\varepsilon > 0$. Arc flows in subsequent almost-complementary trees will then be polynomials expressions in ε which will be strictly positive for some range $0 < \varepsilon < \varepsilon_o$.

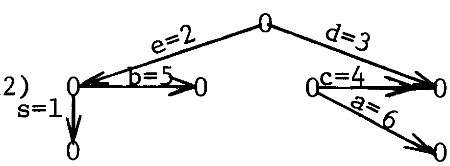
Boundedness is a consequence of first ordering the nodes and then orienting the arcs consistent with this node ordering. If this is done there can be no directed cycles -in G. In general, the feasible set is bounded if and only if there is no cycle in which all arcs are oriented in the same direction around the cycle.

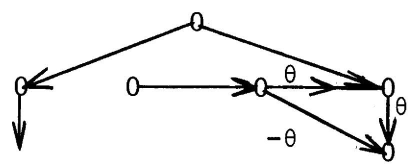
The almost-complementary trees correspond to the sequence of basic feasible solutions can now be easily shown to satisfy the conditions of Theorem 2 and the main Theorem follows as a consequence.

We illustrate the procedure on Example (3). The letters

(11) 

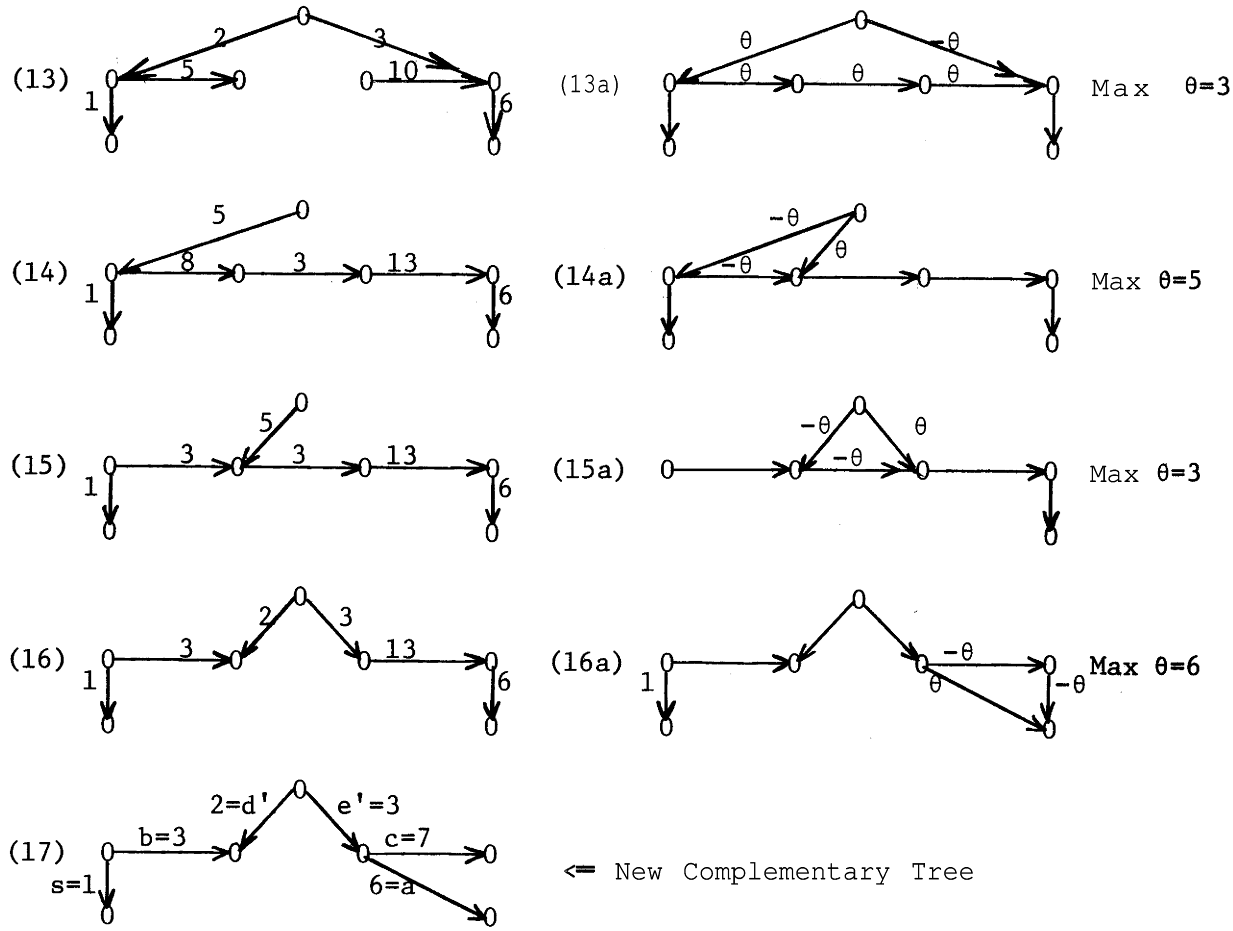
will now represent not only the name of the arc but also the directed flow on the arc. The node ordering was chosen arbitrarily. For starting flow in the complementary starting tree we assume

(12) 

(12a) 

Max $\theta=6$

In (12a) we arbitrarily introduce the out-of-tree arc s' with flow $s' = \theta$, this causes a change of flows about the cycle in order that the net-flow around each node remains the same. Thus the net flow at node (5) in (12) is $c+d = 7$; if s is increased from $s = 0$ to $s = \theta$ then c changesfrom $c = 4$ to $c = 4 + \theta$; similarly, $a = 6$ changesto $a = 6 - \theta$. The maximum change in θ that preserves feasibility is $\theta = 6$ at which value $a = 0$ and arc a drops out, see (13). Therefore $a' = \theta$ is introduced in (13a).



<u>Modified Algorithm</u>: After node ordering, arc orientation, assigning basic feasible flows, and **chosing** a special basic arc, increase flow on its complement.

Step I': Drop arc of the cycle as in simplex algorithm. If arc dropped is a special arc, terminate. If not,

Step II': Introduce as incoming arc the complement of the arc dropped. Return to Step I'.

REFERENCES

[1] FULKERSON, D.R., Verbal communication, This theorem is a corollary of either Theorem lc or Theorem 2c of Edmonds and Fulkerson, "Transversals and matroid partition", N.B.S. Journal, Vol. 69B, 147-153.

[2] D.I.A. Cohen, On the Sperner Lemma, J. Combinatorial Theory 2 [1967] 585-587.

[3] R.W. Cottle and G.B. Dantzig, Complementary pivot theory of mathematical programming, J. Linear Alg. Appl. 1 [1968] 103-125.

[4] K. Fan, Simplicial maps from an orientable n-pseudo manifold into S^m with the octahedral triangulation, J. Combinatorial Theory 2 [1967] 588-602.

[5] D. Gale and H. Nikaidô, The Jacobian matrix and the global univalence of mappings, Math. Ann. 159 [1965] 81-93.

[6] C.E. Lemke, Bimatrix equilibrium points and mathematical programming, Management Sci. 11 [1965] 681-689.

[7] C.E. Lemke and J.T. Howson, Jr,, Equilibrium points of bimatrix games, J. Soc. Indust. Appl. Math. 12 [1964] 413-423.

[8] O. Ore, Theory of graphs, American Mathematical Society, Providence, Rhode Island 1962.

[9] H. Scarf, The approximation of fixed points of a continuous mapping, SIAM J. Appl. Math. 15 [1967], 1328-1343.