RM-73-4

# A GROUP CUT FOR THE TRAVELING SALESMAN PROBLEM

David E. Bell

October 1973

# A Group Cut for the Traveling Salesman Problem

David E. Bell*

## Abstract

Held and Karp have shown how the good minimum spanning tree algorithm may be used in solving the Traveling Salesman problem by means of a generalized linear program involving 1-trees. The size and structure of their dual problem are introduced. From this analysis a group cut is produced and methods by which this may resolve any duality gap are presented.

## 1. Introduction

In two papers Held and Karp [6,7] showed how the fast algorithm for calculating the minimum spanning tree of a graph (the greedy algorithm, see, for example, Edmonds [3]) could be incorporated in an approach for solving the Traveling Salesman problem. In the second of these papers, they exhibit some graphs for which this approach has a gap, that is, their lower bound is strictly less than the value of the optimal solution. Here it is intended to propose methods of resolving any gap which may occur, by considering equivalence properties of 1-trees. Standard graph theory terminology will be used throughout (for example, Harary [5]).

The next section is a brief statement of the problem and of the Held and Karp approach. Section 3 examines the size and structure of their dual problem, characterizing the small number of 1-trees which need ever be explicitly generated by their method.

---

*This research was part of a dissertation supervised by Professor J.G. Shapiro at the Operations Research Center, M.I.T., U.S.A.

Section 4 presents a way of resolving any duality gap by consideration of the group formulation of integer programs introduced by Gomory [4].

## 2. The Problem

Given a graph having nodes and edges, a subset of edges is connected if every pair of edges in the subset has a path of subset edges between them. The degree of a node with respect to the subset is the number of edges in the subset incident to it. If each edge is given a weight or cost, the traveling salesman problem is to find a minimum weight connected subset of edges having degree two at each node. Thus the two requirements for a feasible solution or tour, are

(i) connectedness

(ii) regularity of degree.

Whereas Dantzig et al. [2] chose to relax the connectedness constraint and solve the simpler problem which resulted, Held and Karp took the complementary approach by enforcing connectedness but relaxing the regularity of degree.

A tree is a connected subset of edges for which the path joining two edges is unique. A spanning tree is a tree which is a proper subset of no other, a minimum spanning tree being one with minimum weight. For example the removal of any edge from a tour gives a spanning tree. It is well known that for a connected graph having n nodes, all spanning trees have n - 1 edges. Consider a spanning tree with one extra edge. It could be a tour if the regularity condition is met. Consider a graph with different edge weights, edge (i,j) having its weight increased by an amount $\Pi_i + \Pi_j$ for some real vector $\Pi = (\Pi_1,\ldots,\Pi_n)$.

<u>Lemma 1</u>. The minimum tour for the new edge weights is the same as that for the old edge weights.

<u>Proof</u>. For any subset of edges of the graph, let $d = (d_1,\ldots,d_n)$ represent the degree at each node. Thus if the edge weight of the subset was c it is now $c + \sum_{i=1}^{n} d_i\Pi_i$. Consider two tours with weights $c_1,c_2$ where $c_1 < c_2$, say. If $c_1^*,c_2^*$ are their new weights

$$c_1^* = c_1 + \Sigma\, d_i\Pi_i = c_1 + 2\,\Sigma\,\Pi_i < c_2 + 2\,\Sigma\,\Pi_i = c_2^* \quad .$$

Thus the ordering of the tours with respect to the weights is unaltered and the minimum tour is invariant. ||

The purpose of this transformation of the edge weights is that although the minimum tour is invariant, the minimum spanning tree is not. Define a <u>1-tree</u> as a spanning tree on nodes $\{2,\ldots,n\}$ together with two edges incident to node 1. This is an example of a spanning tree on $\{1,2,\ldots,n\}$ plus an edge. Let the weight of a minimum 1-tree with node weights $\Pi$ be $w(\Pi) + 2\,\Sigma\,\Pi_i$. The definition of $w(\Pi)$ is made so that if ever the minimum 1-tree is a tour (and thus the minimum tour) $w(\Pi)$ will give its correct weight.

Lemma 2. If the minimum tour has weight $c^*$ then $w(\Pi) \leq c^*$ for all $\Pi \varepsilon R^n$.

Proof. The optimal tour with weight $c^* + 2 \Sigma \Pi_i$ is an example of a 1-tree. ||

Hence $w(\Pi)$ represents a lower bound for $c^*$. This leads to a dual problem of finding a best lower bound $w^*$ for $c^*$:

$$w^* = \max_{\Pi} w(\Pi) \quad . \tag{1}$$

It would be desirable to find that $w^* = c^*$ for all graphs, however Held and Karp [7] exhibit some graphs which do have a duality gap and $w^* < c^*$. They use branch and bound to close the gap.

Problem (1) may be rewritten as

$$\max w$$

$$\text{s.t.} \quad w \leq c_k + \sum_{i=1}^{n} d_{ik}\Pi_i - 2 \Sigma \Pi_i \quad , \tag{2}$$

where the index k runs over the set of all 1-trees and $d_{ik}$ is the degree of node i in 1-tree k. Note that since $d_{1k}$ is constant with value 2 there is nothing to be gained by varying $\Pi_1$. Letting $v_{ik} = d_{ik} - 2$ and $\Pi_1 = 0$, (2) becomes

$$\max w$$

$$w \leq c_k + \sum_{i=2}^{n} \Pi_i v_{ik} \quad , \qquad k = 1,\ldots,T.$$

The dual of this L.P. is

$$\min \sum_{k=1}^{T} c_k\lambda_k$$

$$\text{s.t.} \sum_{k=1}^{T} v_{ik}\lambda_k = 0 \ , \qquad i = 2,\ldots,n-1$$

$$\sum_{k=1}^{T} \lambda_k = 1 \ , \qquad (3)$$

$$\lambda_k \geq 0 \ , \qquad \kappa = 1,\ldots,T.$$

Note that the equation for i = n has been omitted. Since $\sum_{i=2}^{n} v_{ik} = 0$ for all k, one of the constraints was redundant. If the condition "$\lambda_k$ integral" were added to (3) it would be an exact formulation of the traveling salesman problem. It is the removal of this constraint that is the relaxation used by Held and Karp.

Formulation (3) may be solved by column generation although dual ascent procedures are required to speed the computation. For any L.P. basis B a shadow price vector $\Pi = c_B B^{-1}$ is obtained which is used to find an improved 1-tree until optimality is reached. If at any time a generated 1-tree is a tour the procedure may be stopped.

## 3. The Size and Structure of the Dual Problem

In the following it will be assumed that the graph under consideration has n nodes and all $\frac{1}{2}n(n-1)$ possible edges, that is, every node is adjacent to every other node.

The following result, although well known, is included because the relation outlined in the proof will be important later.

<u>Lemma 3</u>. The number of distinct labelled spanning trees of the graph is $n^{n-2}$.

<u>Proof</u> The proof will set up a one to one relation between the spanning trees of the graph and the set of n - 2 vectors having elements from the set {1,2,...,n} of which there are clearly $n^{n-2}$. See Prüfer [9].

Every tree must have at least one end node (having degree one). The end node with smallest index is thus well defined and unique. For example, in Figure 1 it is node 1.

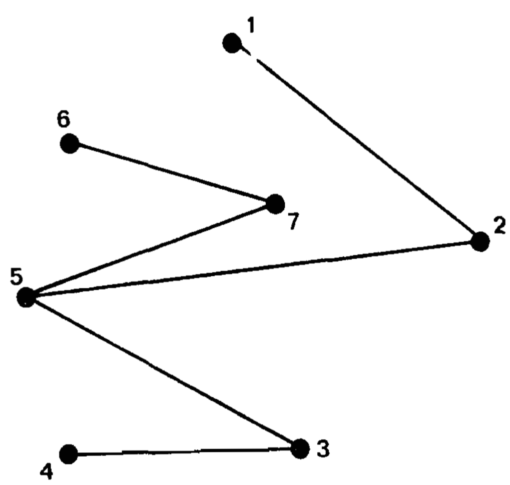


Figure 1

Let the first element of the n - 2 vector (the tree vector) be the index of the unique node to which this node is adjacent. For the example, the tree vector $(n_1,\ldots,n_5)$ has $n_1 = 2$. After removing the edge between these two nodes a new tree results. The above step may be repeated until only one edge remains, giving $n_2 = 5$, $n_3 = 3$, $n_4 = 5$, $n_5 = 7$ or $n = (2,5,3,5,7)$ as the tree vector for the example.

This produces a map from the set of spanning trees to the set of n - 2 vectors. It remains to show that from any n - 2 vector a spanning tree can be constructed whose tree vector is the given vector. The idea, given an arbitrary n - 2 vector with elements in $\{1,2,\ldots,n\}$ is to reverse the algorithm given previously. Consider the end node with smallest index. Since it has degree zero when the edge incident to it is removed, it cannot appear as a coefficient in the tree vector. Conversely any index not appearing in the tree vector must have been an end node. Find the smallest index k, not appearing in the n - 2 vector. Such an index must exist. This node must have been adjacent to node $n_1$ so this edge $(k,n_1)$ may be entered in the spanning tree to be constructed. Now consider the vector $(n_2,\ldots,n_{n-2})$ as a tree vector for the graph on nodes $\{1,2,\ldots,n\} - \{k\}$. The process may be repeated until n - 2 edges nave been included in the spanning tree. The missing edge is that between the unique two nodes which at no time in the construction were end nodes. This can be seen to reverse the process used to construct the tree vector, hence the spanning tree created from the n - 2 vector has the n - 2 vector as its tree vector.

The n - 2 vector (2,5,3,5,7) would yield:

1) Smallest missing element from {1,2,3,4,5,6,7} is 1. Enter edge (1,2).
2) Smallest missing element from {2,3,4,5,6,7} in (5,3,5,7) is 2. Enter edge (2,5).
3) Smallest missing element from {3,4,5,6,7} in (3,5,7) is 4. Enter edge (4,3).
4) Smallest missing element from {3,5,6,7} in (5,7) is 3. Enter edge (3,5).
5) Smallest missing element from {5,6,7} in (7) is 5. Enter (5,7).
6) The remaining nodes are {6,7}. Enter edge (6,7).

This reconstructs Figure 1. ||

Corollary. The number of distinct 1-trees is $\frac{1}{2}(n - 1)^{n-2}(n - 2)$.

Proof. The number of spanning trees on {2,...,n} is $(n - 1)^{n-3}$ together with $\frac{1}{2}(n - 1)(n - 2)$ ways to select two edges to be incident to node 1. ||

This corollary reveals that even for a graph with only 11 nodes there are $4.5 \times 10^9$ possible 1-trees. It will be shown however that only a small proportion of these could ever be generated by the minimum spanning tree algorithm, so that by use of dual ascent methods even fewer will actually be considered. The number of active 1-trees is reduced to 92,368 in the 11 city case and approximately $7.74 \times 10^{23}$ in the 42 city case, of which exactly one is a tour.

Lemma 4. Let $d_i$ be the number of times index i appears in a given tree vector, then node i has degree $d_i + 1$ in the corresponding spanning tree.

Proof. A given node cannot be an end node until it has degree one at which time it must have had all but one of its edges removed and thus appeared that many times in the tree vector. The process does not stop until every node has reached degree one or less. ||

Definition. Call two 1-trees equivalent if they have the same degree at each node.

The following 1-trees are equivalent.

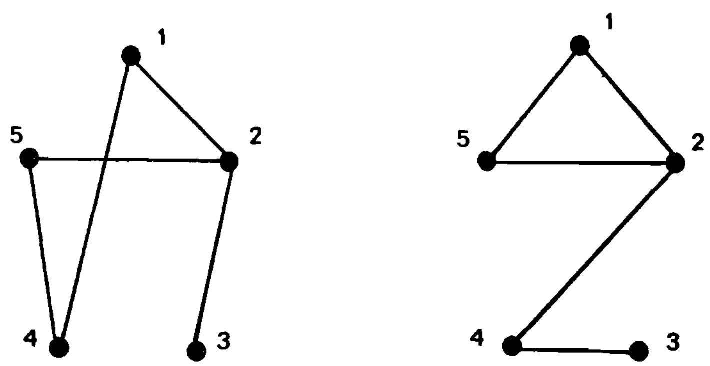


**Figure 2**

This definition divides the set of 1-trees into equivalence classes. Notice that all tours are in the same equivalence class since they all have the same degree vector.

<u>Definition</u> A <u>1-tree vector</u> is $\underline{n} = (n_1,\ldots,n_{n-1})$ where $n_i \varepsilon \{2,\ldots,n\}$, $(n_1,\ldots,n_{n-3})$ is the tree vector for the spanning tree on nodes $(2,\ldots,n)$, $n_{n-2} < n_{n-1}$ are the nodes to which node 1 is adjacent.

For example, the 1-tree vectors for the two graphs of Figure 2 are (2524) and (4225).

<u>Theorem 5</u>. Two 1-trees are equivalent if and only if their 1-tree vectors are permutations of each other.

<u>Proof</u>. The definition of a 1-tree vector inherits the tree vector property given by Lemma 4. The proof follows immediately. ||

<u>Lemma 6</u>. The ordering by weight of each equivalence class is unaffected by the edge weight transformation $c_{ij} \to c_{ij} + \Pi_i + \Pi_j$.

<u>Proof</u>. As in Lemma 1,

$$c_1^* = c_1 + \Sigma\, d_i \Pi_i < c_2 + \Sigma\, d_i \Pi_i = c_2^* \quad . \qquad ||$$

Hence only the minimum 1-tree in each equivalence class will ever be generated by the dual problem. An upper bound for for the number of 1-trees which could be generated by varying the node weight vector $\Pi$ is thus the number of equivalence classes.

<u>Theorem 7</u>. The number of equivalence classes is

$$\binom{2n-3}{n-1} - (n-1) \quad .$$

<u>Proof</u>. The set of 1-trees has a one to one correspondence with the set of all n - 1 vectors having elements from $\{2,\ldots,n\}$ with the last two coefficients satisfying $n_{n-2} < n_{n-1}$. What is required is the maximum number of such vectors none of which is a permutation of any other. Ignoring the condition on the last two coefficients for the moment, this is the number of distinct non-negative integer solutions to

$$\sum_{i=2}^{n} d_i = n - 1 \quad , \tag{4}$$

where $d_i$ is the number of times i appears in the vector. This is the same as the combinatorial problem of assigning n - 1 unlabelled objects (the R.H.S. of (4)) to n - 1 labelled boxes (the $d_i$), which is well solved with solution

$$\binom{2n-3}{n-1} \quad .$$

In general, A unlabelled objects to B labelled boxes has $\binom{A+B-1}{A}$ distinct solutions, see for example Riordan [10].

Now a vector which cannot be permuted to a vector which satisfies $n_{n-2} < n_{n-1}$ is not a valid solution. But this only occurs if every element of the vector is the same, of which there are n - 1 cases.

Hence the number of classes is

$$\binom{2n-3}{n-1} - (n-1) \quad . \qquad ||$$

For the case n = 4 there are $\binom{5}{3} - 3 = 7$ cases:

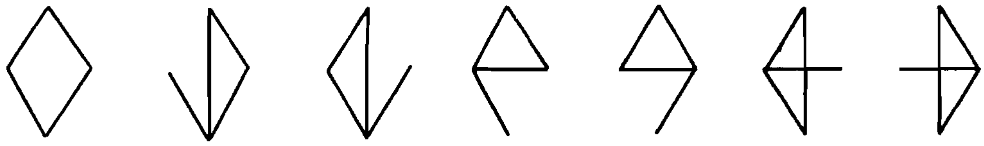

**Figure 3**

The ratio of 1-trees to equivalence classes is approximately $20\left(\frac{n-2}{4}\right)^n$ using Stirling's formula.

4. <u>The Dual Problem Considered as an Integer Program</u>

It has already been noted that if the solution to (3) is integral then it represents the optimal solution to the traveling salesman problem, so regarding it as an integer program and applying the group reformulation of Gomory [4] with respect to an optimal L.P. basis B, with $\Pi = -c_B B^{-1}$ the set of multipliers, then (3) is equivalent to

$$\min \sum_{k \varepsilon n} (c_k + \Pi v_k)\,\lambda_k$$

$$B^{-1} N \lambda_N \leq B^{-1} e_{n-1}$$

$$B^{-1} N \lambda_N \equiv B^{-1} e_{n-1} \qquad (\text{mod } 1)$$

$$\lambda_N \geq 0 \text{ and integer} \quad ,$$

where $e_{n-1}$ is the n - 1 vector $\begin{matrix} 0 \\ \vdots \\ 0 \\ 1 \end{matrix}$ and N is the matrix of non-basic 1-trees. Note that if two 1-trees have degrees satisfying

$$B^{-1}\binom{v^1}{1} \equiv B^{-1}\binom{v^2}{1} \qquad (\text{mod } 1) \quad ,$$

then their contribution to the group equation in (5) is equal. Thus the set of 1-trees may be decomposed according to the value of $B^{-1}\binom{v}{1}$ (mod 1). To avoid confusion with the earlier definition of equivalence amongst 1-trees, this decomposition will be called <u>group equivalence</u>.

<u>Lemma 8</u>. If two 1-trees are equivalent then they are group equivalent.

<u>Proof</u>. The definition of group equivalence relies only on the value of v which is equal for elements of the same equivalence class. ||

This lemma serves to confirm that the solution is not affected by the exclusion of all but the minimum 1-tree in each equivalence class.

<u>Theorem 9</u>. All the 1-trees in the basis are in the same group equivalence class. Moreover, this class contains no tours.

<u>Proof</u>. Since all the basic 1-trees are columns of B they must all satisfy

$$B^{-1}\binom{v}{1} \equiv 0 \qquad (\text{mod } 1)$$

since each is a unit vector. Each tour trivially satisfies

$$B^{-1}\binom{v}{1} = B^{-1}\binom{0}{1}$$

so that it only need be shown that

$$B^{-1}\binom{0}{1} \not\equiv 0 \qquad (\text{mod } 1) \quad .$$

But $B^{-1}\binom{0}{1}$ (mod 1) is the right hand side of the group equation in (5) and thus is zero if and only if the L.P. solution is integral. This has been assumed not to be the case. ||

This theorem applies equally well to any problem with constraints of the form

$$A\lambda = b$$
$$1\lambda = 1$$
$$\lambda \geq 0 \quad ,$$

that is, problems having a convexity row and no slacks.

Theorem 9 provides a group restriction which may be applied to future selections of 1-trees, since they should now be chosen to satisfy either

$$B^{-1}\begin{pmatrix} v \\ 0 \end{pmatrix} \equiv 0 \qquad (\text{mod } 1)$$

or

$$B^{-1}\begin{pmatrix} v \\ 1 \end{pmatrix} \not\equiv 0 \qquad (\text{mod } 1) \quad ,$$

each of which makes all the existing basic variables infeasible without affecting the feasibility of any tour. It is possible to convert it to an ordinary cutting plane in the following manner.

Let $\beta_i$ be the $i^{th}$ row of $B^{-1}$ and let $\varepsilon_i \equiv -\beta_i e_{n-1}$ (mod 1) with $0 \leq \varepsilon_i$, $i = 1,\ldots,n - 1$. Since $\varepsilon \not\equiv 0$, $\varepsilon_j > 0$ for some particular j so that the constraint

$$\beta_j \begin{pmatrix} v \\ 0 \end{pmatrix} \equiv 0 \qquad (\text{mod } 1)$$

also excludes all the basic 1-trees.

Define $0 \leq \alpha_k < 1$ by

$$\alpha_k \equiv \beta_j \begin{pmatrix} v_k \\ 0 \end{pmatrix} \qquad (\text{mod } 1)$$

for each 1-tree; then the constraint

$$\sum_{k=1}^{T} \lambda_k \alpha_k \leq \theta \qquad (6)$$

is a valid cut for any $0 \leq \theta < \varepsilon_j$ since the existing optimal L.P. solution $\lambda^*$ satisfies

$$\lambda^* \alpha = \varepsilon_j \quad .$$

The cut (6) is related to the usual Gomory cut. Let $\lambda_k \equiv \alpha_k - \varepsilon_j$, then the Gomory cut is

$$\Sigma\ \gamma_k \lambda_k \geq 1 - \varepsilon_j > 0 \quad , \tag{7}$$

which is valid since $\lambda^* \gamma = 0$. The cut (6) with $\theta = 0$ is much stronger than (7) since it removes all the 1-trees not in the correct class, instead of merely making $\lambda^*$ infeasible.

It would be preferable if the group constraint

$$\beta_j \binom{v}{0} \equiv 0 \qquad (\text{mod } 1) \tag{8}$$

could be incorporated into the subproblem which generates the 1-trees, that is, the minimum spanning tree algorithm.

A group weight $g_i \equiv \beta_j e_i$ (mod 1) is assigned to node i of the graph for i = 2,...,n - 1 with group weight zero for nodes 1,n. Now each edge (i,j) may be considered to have a group weight $g_i + g_j$ (mod 1). Figure 4 shows a graph in which each edge has a real weight as well as a group weight from {0,1}, the modulo 2 group.

The minimum spanning tree has group weight zero, but it may be that (8) requires the minimum spanning tree with group weight 1 to be found. This would be a simple matter if there existed an efficient dynamic programming formulation of the minimum spanning tree problem since the state space could be extended to include the group (see section 2.2 in [1]). Without such a formulation the only direct procedure for finding a minimum spanning tree of a given group equivalence class is by a $k^{th}$ best solution procedure (see Lawler [8])

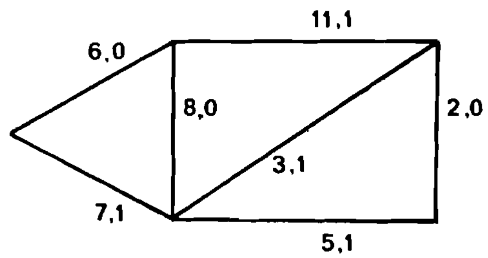


Figure 4

by which successive best solutions are generated until one satisfies the side constraint.

In the only problem for which any computation was tried a six node problem had an optimal L.P. basis with determinant 4, optimal value 473. The group constraint (6) with $\theta = 0$ gave a bound of 508 which was indeed the cost of the optimal tour.

## References


[1] Bell, D.E. "The Resolution of Duality Gaps in Discrete Optimization." Technical Report No. 81, Operations Research Center, Massachusetts Institute of Technology, August 1973.

[2] Dantzig, G.B., Fulkerson, D.R., and Johnson, S.M. "Solution of a Large Scale Traveling Salesman Problem," Operations Research, 2 (1954), 393-410.

[3] Edmonds, J. "Matroids and the Greedy Algorithm," Mathematical Programming, 1 (1971), 127-136.

[4] Gomory, R.E. "Some Polyhedra Related to Combinatorial Problems," Linear Algebra, 2 (1969), 451-558.

[5] Harary, F. Graph Theory. Reading, Mass., Addison-Wesley, 1969.

[6] Held, M. and Karp, R.M. "The Traveling Salesman Problem and Minimum Spanning Trees," Operations Research, 18 (1970), 1138-1162.

[7] Held, M. and Karp, R.M. "The Traveling Salesman Problem and Minimum Spanning Trees: Part II," Mathematical Programming, 1 (1971), 6-25.

[8] Lawler, E.L. "A Procedure for Computing the $k^{th}$ Best Solution to Discrete Optimization Problems and Its Application to the Shortest Path Problem," Management Science, 18 (March 1972), 401-405.

[9] Prüfer, H. "Neuer Beweis Eines Satzes Über Permütation," Archiv der Mathematischen Physik, 27 (1918), 742-744.

[10] Riordan, J. An Introduction to Combinatorial Analysis. New York, Wiley, 1958.