Mathematical Programming 1 (1971) 6–25. North-Holland Publishing Company

THE TRAVELING-SALESMAN PROBLEM AND MINIMUM SPANNING TREES: PART II *

Michael HELD
IBM Systems Research Institute New York, New York, U.S.A.

and

Richard M. KARP **
University of California, Berkeley, California, U.S.A.

Received 19 October 1970

The relationship between the symmetric traveling-salesman problem and the minimum spanning tree problem yields a sharp lower bound on the cost of an optimum tour. An efficient iterative method for approximating this bound closely from below is presented. A branch-and-bound procedure based upon these considerations has easily produced proven optimum solutions to all traveling-salesman problems presented to it, ranging in size up to sixty-four cities. The bounds used are so sharp that the search trees are minuscule compared to those normally encountered in combinatorial problems of this type.

0. Introduction

In a previous paper [7], the authors explored the relationship between the symmetric traveling-salesman problem and the minimum spanning tree problem. By means of this relationship a lower bound on the cost of an optimum tour was derived which is quite sharp, and hence of value in connection with branch-and-bound procedures. The methods proposed in [7] for computing this bound proved inadequate. The present paper gives an efficient method for approximating the bound closely from below, and reports on the use of this method in a highly successful algorithm for the *exact* solution of symmetric traveling-salesman problems.

* This paper was presented at the 7th Mathematical Programming Symposium 1970, The Hague, The Netherlands.

** This research has been partially supported by the National Science Foundation under Grant GP-25081 with the University of California. Reproduction in whole or in part is permitted for any purpose of the United States Government.

The bound derived in [7] is expressed as the maximum of a function describable as the minimum of a finite number of linear functions. Two methods were proposed for maximizing this function: a linear programming method using column generation, and an ascent method which at each step increases the value of the function. A branch-and-bound procedure using bounds derived from the ascent method to aid in the search for an optimum tour was also suggested.

The method presented here for approximating the bound is of a different nature. It is an iterative method related to the relaxation method for the solution of systems of linear inequalities [1], [12]. The iteration does not necessarily improve the function value at each step, but instead reduces the Euclidean distance from a maximum point of the function.

An improved branch-and-bound procedure incorporating the new ascent method has easily produced proven optimum solutions to all traveling-salesman problems presented to it, ranging in size up to 64 cities. The bounds used are so sharp that the resulting search trees are minuscule compared to those normally encountered in combinatorial problems of this type.

1. A lower bound on the cost of an optimum tour

In this section we briefly review the approach taken in [7]. Let the $n \times n$ symmetric matrix (c_{ij}) specify the weights assigned to the edges of the complete undirected graph K_n with vertex set $\{1, 2, ..., n\}$. Each subgraph of K_n is assigned a weight equal to the sum of the weights of its edges. A *tour* is a cycle passing through each vertex exactly once. The *traveling-salesman problem* seeks a tour of minimum weight.

A *tree* is a connected graph without cycles. A 1-*tree* is a tree having vertex set $\{2, 3, ..., n\}$, together with two distinct edges at vertex 1. Our approach exploits the fact that a minimum-weight 1-tree is easy to compute, together with the following relationships between tours and 1-trees:

(i) a tour is simply a 1-tree in which each vertex has degree 2;
(ii) if a minimum-weight 1-tree is a tour, then it is a tour of minimum weight;

(iii) for any real n-vector $\pi = (\pi_1, \pi_2, ..., \pi_n)$, the transformation on edge weights

$$c_{ij} \to c_{ij} + \pi_i + \pi_j$$

leaves the traveling-salesman problem invariant, but changes the minimum 1-tree.

Clearly, with respect to the weights $c_{ij} + \pi_i + \pi_j$ the weight of a minimum tour is greater than or equal to the weight of a minimum-weight 1-tree. Thus

$$C^* + 2 \sum_{i=1}^{n} \pi_i \geq \min_k \left[c_k + \sum_{i=1}^{n} \pi_i d_{ik} \right] \tag{1}$$

where C^* is the weight of a minimum tour with respect to the weights c_{ij}, k indexes the 1-trees, c_k is the weight of the k-th 1-tree with respect to the weights c_{ij}, and d_{ik} is the degree of vertex i in the k-th 1-tree. Let

$$w(\pi) = \min_k \, [c_k + \pi \cdot v_k],$$

where v_k is the n-vector having d_{ik}-2 as its i-th component, and $\cdot$ denotes inner product. Then (1) is equivalent to

$$C^* \geq w(\pi). \tag{2}$$

Thus we obtain an infinite family of lower bounds on the weight of an optimum tour, and the best such bound is $\max_\pi w(\pi)$.

2. An ascent method for the computation of $\max_\pi w(\pi)$

In this section we present an iterative method for computing or approximating $\max_\pi w(\pi)$. The iteration computes a sequence of n-vectors $\{\pi^m\}$ according to the formula

$$\pi^{m+1} = \pi^m + t_m v_{k(\pi^m)} \tag{3}$$

where:

for any π, $k(\pi)$ is the index of a minimum-weight 1-tree at the point π,

and

$\{t_m\}$ is a sequence of scalars.

The following lemma establishes the rationale for this iteration.

Lemma 1:

Let $\bar{\pi}$ and π be such that $w(\bar{\pi}) \geq w(\pi)$. Then $(\bar{\pi} - \pi) \cdot v_{k(\pi)} \geq w(\bar{\pi}) - w(\pi) \geq 0$.

Proof:

(i) $c_{k(\pi)} + \pi \cdot v_{k(\pi)} = w(\pi)$

(ii) $c_{k(\pi)} + \bar{\pi} \cdot v_{k(\pi)} \geq \min\limits_{k} [c_k + \bar{\pi} \cdot v_k] = w(\bar{\pi})$.

Substracting (i) from (ii)

$$\bar{\pi} \cdot v_{k(\pi)} - \pi \cdot v_{k(\pi)} \geq w(\bar{\pi}) - w(\pi) \geq 0.$$

Thus the hyperplane through π having $v_{k(\pi)}$ as its normal determines a closed half-space containing all points $\bar{\pi}$ such that $w(\bar{\pi}) - w(\pi) \geq 0$, and the iteration step moves into the half-space along the normal $v_{k(\pi)}$. In particular, this half-space includes any point where $w(\cdot)$ assumes its maximum value, and a sufficiently small step produces a point closer than π to any such maximum point.

The next lemma indicates the limits on appropriate step sizes.

Lemma 2:

If

$$0 < t < \frac{2(w(\bar{\pi}) - w(\pi))}{\|v_{k(\pi)}\|^2}$$

then

$$\|\bar{\pi} - (\pi + t v_{k(\pi)})\| < \|\bar{\pi} - \pi\|. \ *$$

Proof:

$$\|\bar{\pi} - (\pi + t v_{k(\pi)})\|^2 = \|\bar{\pi} - \pi\|^2 - 2t(\bar{\pi} - \pi) \cdot v_{k(\pi)} + t^2 \|v_{k(\pi)}\|^2 =$$

$$\|\bar{\pi} - \pi\|^2 + t\,[\,t\,\|v_{k(\pi)}\|^2 - 2(\bar{\pi} - \pi) \cdot v_{k(\pi)}] \leq$$

$$\|\bar{\pi} - \pi\|^2 + t\,[\,t\,\|v_{k(\pi)}\|^2 - 2(w(\bar{\pi}) - w(\pi))] < \|\bar{\pi} - \pi\|^2.$$

* $\|\ \|$ denotes Euclidean norm.

Thus, it t is in the indicated range, then the point $\pi + tv_{k(\pi)}$ is closer to $\bar{\pi}$ than π is.

Iterations of the type (3) have been investigated previously in connection with the relaxation method for the solution of linear inequalities [1] and [12]. Given a system of inequalities

$$\sum_{j=1}^{g} a_{ij} x_j \geq b_i, \qquad i = 1, 2, ..., h$$

the relaxation method constructs a sequence $\{x^s\}$, where x^{s+1} is obtained from x^s by selecting an inequality violated by x^s, and then taking a step toward the corresponding hyperplane along the normal to it through x^s. The relation of the relaxation method to the present problem is seen by noting that $\max_{\pi} w(\pi)$ is the optimum value of the linear program

$$\begin{aligned}&\text{Maximize } w\\ &\text{Subject to } w \leq c_k + \pi \cdot v_k \text{ for all } k.\end{aligned} \tag{4}$$

Moreover, given a "target value" $\bar{w}$, finding a point $\bar{\pi}$ such that $w(\bar{\pi}) \geq \bar{w}$ is equivalent to solving the system of inequalities

$$\bar{w} \leq c_k + \pi \cdot v_k \text{ for all } k. \tag{5}$$

Indeed a version of the relaxation method in which that violated inequality at π is selected which minimizes $c_k + \pi \cdot v_k$, and hence maximizes the "residual" $\bar{w} - (c_k + \pi \cdot v_k)$, yields an iteration scheme of precisely the form (3). A geometric viewpoint will perhaps clarify this.

Let $P_{\bar{w}}$ denote the polyhedron of feasible solutions to (5). We note that

(i) $w_1 < w_2 \Rightarrow P_{w_1} \supset P_{w_2}$;

(ii) if $\bar{w} > w(\pi)$ then $P_{\bar{w}}$ is contained in the half-space determined by the hyperplane through π with normal vector $v_{k(\pi)}$;

(iii) the ray $\{\pi + tv_{k(\pi)} \mid t > 0\}$ is perpendicular to the face of $P_{\bar{w}}$ whose equation is $\bar{w} = c_{k(\pi)} + \bar{\pi} \cdot v_{k(\pi)}$; the ray intersects the face at the point corresponding to $t = \dfrac{\bar{w} - w(\pi)}{\|v_{k(\pi)}\|^2}$.

The situation is portrayed in fig. 1.

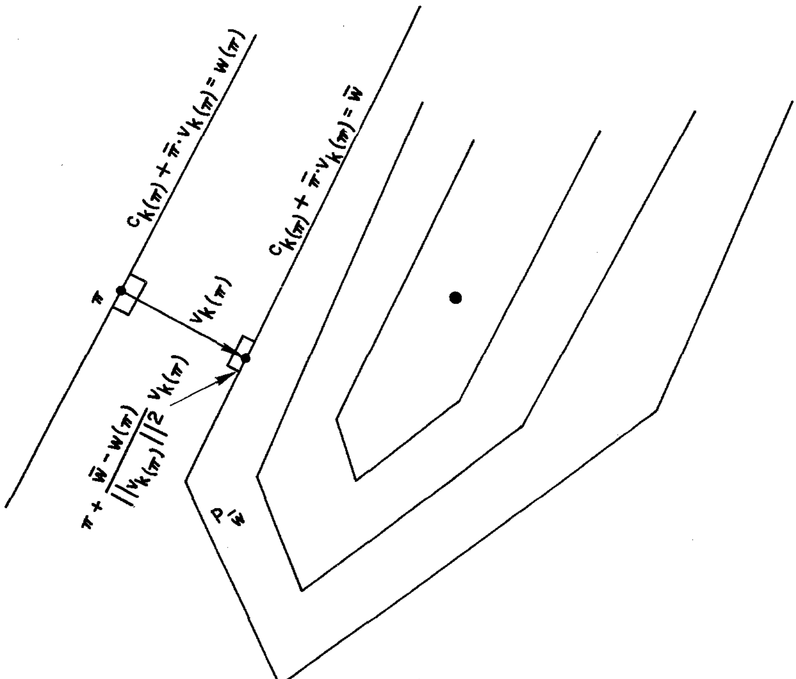


Fig. 1

Certain results about the convergence properties of various forms of the relaxation method [1], [12] can be adapted to give insight into the proper choice of the parameters $\{t_m\}$ in (3). The following lemma is similar to theorem 1 in [12].

Lemma 3:

We assume that $\bar{w} < \max_{\pi} w(\pi)$. Let $\{\pi^m\}$ be a sequence of points obtained by the process

$$\pi^{m+1} = \pi^m + \lambda_m \left(\frac{\bar{w} - w(\pi^m)}{\|v_{k(\pi^m)}\|^2} \right) v_{k(\pi^m)} \tag{6}$$

There are two cases:

Case 1:

If, for some ϵ, $0 < \epsilon < \lambda_m \leq 2$ for all m, then $\{\pi^m\}$ either includes a point $\pi^l \in P_{\bar{w}}$ or converges to a point on the boundary of $P_{\bar{w}}$.

Case 2.

If $\lambda_m = 2$ for all m, then the sequences $\{\pi^m\}$ always includes a point $\pi^l \in P_{\bar{w}}$.

This theorem differs from theorem 1 of [12] in two respects. First it does not require the relaxation parameter λ_m to be constant, and second, it uses a "maximum residual," rather than "maximum distance" rule for selecting the violated inequality.

Proof:

Case 2 is merely a specialization of Case 2 in theorem 1 of [12]. To treat case 1 we need a concept from [12]. Let A be a point set and $\{y^m\}$ a sequence of points. Then $\{y^m\}$ is *Féjer-monotone* relative to A if, for every $x \in A$, the sequence $\{\|y^m - x\|\}$ is monotone nonincreasing. It is shown in [12] that, if A is of full dimension (i.e., contains a neighborhood) then any sequence which is Féjer-monotone relative to A converges. In the present case, assume that the sequence $\{\pi^m\}$ does not contain a $\pi^l \in P_{\bar{w}}$. Then, by lemma 2, the sequence $\{\pi^m\}$ is Féjer-monotone relative to the full-dimensional set $P_{\bar{w}}$, and thus converges. Hence, $\|\pi^{m+1} - \pi^m\| \to 0$. By (6)

$$\|\pi^{m+1} - \pi^m\| = \lambda_m \frac{\bar{w} - w(\pi^m)}{\|v_{k(\pi^m)}\|},$$

so that $\bar{w} - w(\pi^m) \to 0$. Hence $w\left(\lim_{m\to\infty} \pi^m\right) = \bar{w}$, and the limit point is in $P_{\bar{w}}$. Clearly it is on the boundary, since no π^m is in $P_{\bar{w}}$.

Lemma 3 can be used to evolve various strategies for determining a point where $w(\pi)$ is close to its maximum value. In the present paper we forgo a general discussion of such strategies, and restrict ourselves to examining the strategy used in the traveling-salesman computations. There t_m is simply taken to be a constant $\bar{t}$ for all m. In retrospect this seems naive, but the computations preceded (and, in large measure, prompted) the theoretical investigation of the ascent process.

Theorem 1:

The iteration (3), with $t_m = \bar{t}$ for all m, satisfies:

$$\sup_m w(\pi^m) \geq \max_\pi w(\pi) - \tfrac{1}{2}\bar{t} \limsup_{m\to\infty} \|v_{k(\pi^m)}\|^2.$$

Proof:

We argue by contradiction. Let $L = \limsup_{m\to\infty} \|v_{k(\pi^m)}\|^2$ and assume that for all m,

$$w(\pi^m) < \hat{w} < \max_{\pi} w(\pi) - \tfrac{1}{2}\bar{t}L \tag{7}$$

Choose $\bar{w} = \hat{w} + \frac{1}{2}\bar{t}L$ and consider lemma 3. The condition $\lambda_m \leq 2$ becomes

$$\frac{\bar{t}\,\|v_{k(\pi^m)}\|^2}{\bar{w} - w(\pi^m)} \leq 2$$

or $\bar{t}\|v_{k(\pi^m)}\|^2 \leq 2(\hat{w} - w(\pi^m)) + \bar{t}L$. By (7) and the definition of L, this is true for all sufficiently large m.

To apply lemma 3, we must also verify that

$$\lambda_m = \frac{\bar{t}\,\|v_{k(\pi^m)}\|^2}{\bar{w} - w(\pi^m)}$$

is bounded away from zero. It is necessary to check that

(a) $v_{k(\pi^m)} \neq 0$ for all m;

and

(b) $w(\pi^m)$ is uniformly bounded below.

(a) by lemma 1, $(\bar{\pi} - \pi^m)\cdot v_{k(\pi^m)} \geq w(\bar{\pi}) - w(\pi^m)$. Hence $v_{k(\pi^m)} = 0 \Rightarrow 0 \geq w(\bar{\pi}) - w(\pi^m)$ for all $\bar{\pi}$; i.e., $w(\pi^m) = \max_{\pi} w(\pi) > \hat{w}$, a contradiction.

(b) by lemma 2,

$$0 < \frac{\bar{t}\,\|v_{k(\pi^m)}\|^2}{\bar{w} - w(\pi^m)} < 2$$

implies that $\{\pi^m\}$ is Féjer-monotone relative to $P_{\bar{w}}$, and hence $\{\pi^m\}$ lies in a bounded set, so that $w(\pi^m)$ is bounded below.

Thus the hypotheses of lemma 3 are satisfied; but the conclusion of lemma 3 violates (7), and the desired contradiction is reached.

We have found that, when the iteration (3) is applied to the traveling-salesman problem with $t_m = \bar{t}$, the 1-trees produced tend to resemble tours very closely; i.e., a great many vertices have degree 2, so that $\|v_k\|^2$ tends to be small. Thus it is reasonable to suppose that $\limsup_{m\to\infty} \|v_{k(\pi^m)}\|^2$ is a small integer. In any case, the iteration scheme (3) has proved very successful, as will be seen in section 4. In fact, this experience with the traveling-salesman problem indicates that some form of the relaxation method may be superior to the simplex method for linear programs involving a very large number of inequalities.

3. The branch-and-bound procedure

Since the ascent method does not necessarily compute a maximum point of $w(\pi)$, and since, in any event, $\max_{\pi} w(\pi)$ may be less than C^*, our algorithm for the traveling-salesman problem combines the ascent method with a branch-and-bound procedure.

Typically, a branch-and-bound procedure successively partitions the set of feasible solutions into subsets and calculates a lower bound on the cost of solutions in each subset. In the case of the traveling-salesman problem such partitioning is usually accomplished by including and excluding sets of edges [2]. The derived problem which results from including the set of edges X and excluding the set Y is of the same form as the original problem, and our method of computing bounds applies to it. Following [7], let $T(X, Y)$ be the set of all 1-trees which include the edges in X and exclude the edges in Y, and let

$$w_{X,Y}(\pi) = \min_{k \in T(X,Y)} [c_k + \pi \cdot v_k] .$$

Then $w_{X,Y}(\pi)$ is a lower bound on the cost of any tour for the derived problem.

The state of the branch-and-bound procedure at any point is specified by a list of derived problems with their associated bounds. A typical entry is of the form $(X, Y, \pi, w_{X,Y}(\pi))$. Initially the list consists of the single entry $(\phi, \phi, 0, w(0))$, where 0 denotes the zero n-vector.

At a general step, an entry $(X, Y, \pi, w_{X,Y}(\pi))$ of least bound is selected and the iteration

$$\pi^{m+1} = \pi^m + \bar{t}v_{k(\pi^m)}$$

is applied, where $k(\pi^m)$ is the index of a minimum-weight 1-tree in $T(X, Y)$ relative to the weights $(c_{ij} + \pi_i^m + \pi_j^m)$. Two outcomes are possible:

(a) for some m, $w_{X,Y}(\pi^m) \geq \bar{C}$, where $\bar{C}$ is an upper bound on the cost of an optimum tour. In this case, the derived problem may be discarded.

(b) for some h, $\max_{0 \leq l \leq ph-1} w_{X,Y}(\pi^l) = \max_{0 \leq l \leq p(h+1)-1} w_{X,Y}(\pi^l)$; i.e., no improvement occurs for a block of p iterations, where p is a parameter of the program. A π' is then chosen such that

$$w_{X,Y}(\pi') = \max_{0 \leq l \leq ph-1} w_{X,Y}(\pi^l)$$

and branching is performed; i.e., new entries of the form $(X_i, Y_i, \pi', w_{X_i,Y_i}(\pi'))$ are created, where $X_i \supseteq X$, $Y_i \supseteq Y$ and every tour in $T(X, Y)$ is in one of the sets $T(X_i, Y_i)$.

The following branching strategy was adopted. The edges which have not yet been included or excluded are ordered according to the amount by which the bound would increase if the edge were excluded. Then the resulting sequence is $e_1, e_2, \ldots, e_r$ where

$$w_{X, Y\cup\{e_1\}}(\pi') \geq w_{X, Y\cup\{e_2\}}(\pi') \geq \ldots \geq w_{X, Y\cup\{e_r\}}(\pi')$$

this ordering can easily be accomplished as a by-product of the "greedy" algorithm for the computation of minimum weight 1-trees [6], [10]. Then

$$\begin{array}{ll}
X_1 = X & Y_1 = Y \cup \{e_1\} \\
X_2 = X \cup \{e_1\} & Y_2 = Y \cup \{e_2\} \\
X_3 = X \cup \{e_1, e_2\} & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
\cdot & \cdot \\
X_q = X \cup \{e_1, e_2, \ldots, e_{q-1}\} & Y_q = Y \cup R_i
\end{array}$$

where q is the smallest index for which there exists a vertex i such that X does not contain two edges incident on i, but X_q does. R_i consists of all edges incident on i not in X_q. It is legitimate to include R_i in Y_q, since any tour which includes two edges incident upon i excludes all the other edges incident on i. There may exist two such vertices i and j, in which case $Y_q = Y \cup R_i \cup R_j$.

The entries generated in the process are of three types: those which are never chosen as an entry of least bound (type $>$), those which are chosen but eliminated in the ascent process because of a bound exceeding $\overline{C}$ (type o) and those which are selected and lead to branching. The entire process can be viewed as a rooted tree whose nodes are all the entries generated. The type $>$ and type o entries correspond to frontier nodes, and the other entries correspond to interior nodes with descendants corresponding to the entries created from them by branching.

4. Computational results

In this section we report our computational experience with the algorithm given in the present paper. The problems tried, ranging in size from 20 cities to 64 cities, came from various sources. Several appear in the open literature, and are identified here by appropriate references. The problem designated "Mezei" was transmitted to us as a challenge by J. Mezei. In problems designated "random (M)," where M is a positive integer, the distances c_{ij} are drawn independently from a discrete uniform distribution over $\{0, 1, ..., M\}$ (with $c_{ij} = c_{ji}$, of course). In problems designated "random Euclidean (M)," the n points were placed randomly in a square of side M, and c_{ij} is the Euclidean distance between points i and j. The term "$p \times q$ knight's tour" refers to a problem in which the cities are the squares of a $p \times q$ chessboard, with distance 0 between squares a knight's move apart, and ∞ otherwise. Three problems were specially constructed. In problem "Tutte," the matrix (c_{ij}) is determined from the "Tutte graph" (cf. fig. 9) as follows:

$$c_{ij} = \begin{cases} 0 \text{ if } i \text{ and } j \text{ are adjacent} \\ \infty \quad \text{otherwise.} \end{cases}$$

The object of the computation in this case was to verify the nonexistence of a Hamilton circuit. In problem "Slant" (c_{ij}) is the sum of two matrices A and B; the first is of the form $a_{ij} = \sigma_i + \sigma_j$, where the σ_i are drawn from the uniform distribution over [0, 2000], and B is a matrix whose elements are drawn from the uniform distribution over [0, 200], except for the "slant" positions corresponding to the tour 1, 2, 3, ..., 64, which contain 0. Problem "Join" was obtained by combining the Dantzig problem and the 22-city random Euclidean problem in imitation of a construction due to Lin [11], p. 2258.

In addition to the cost matrix (c_{ij}), each run required three parameters to be specified: $\bar{t}$, which governs the step size in the ascent method; p, a parameter which determines when the ascent process is terminated and a branch created; and $\bar{C}$, the upper bound on C^* which is used to eliminate derived problems.

Any heuristic procedure (see, for example, [3], [9] or [11]) may be used to obtain $\bar{C}$.

In the following tables, all computing times refer to the IBM 360/91. Table 1 lists those examples in which the bound resulting from applying the ascent method to the initial entry $(\phi, \phi, 0, w(0))$ is equal to C^* (and hence, clearly, no gap exists; i.e., $\max_{\pi} w(\pi) = C^*$). In these examples branching was necessary only to verify this fact and exhibit an optimal tour.

Table 2 gives details of the other runs. The value of $w(\pi)$ after the initial ascent is invariably a close lower bound on C^*.

Some experience with ascent computations using a step size which is varied dynamically during the computation suggests that the bounds

Table 1
Problems where solution was obtained in initial ascent

Problem	n	$\bar{t}$	p	Time (secs)
Croes [3]	20	1	20	4
Random (400)	20	1	25	6
8 × 3 knight's tour *	24	1	25	0
Held and Karp [8]	25	1	20	12
Mezei	26	1	20	22
Random (1800)	30	1	25	19
Slant	64	4,1 **	30	182

* There is no closed knight's tour on a 8 × 3 board.
** $\bar{t}$ was changed during the computation.

Table 2
Problems which required branching

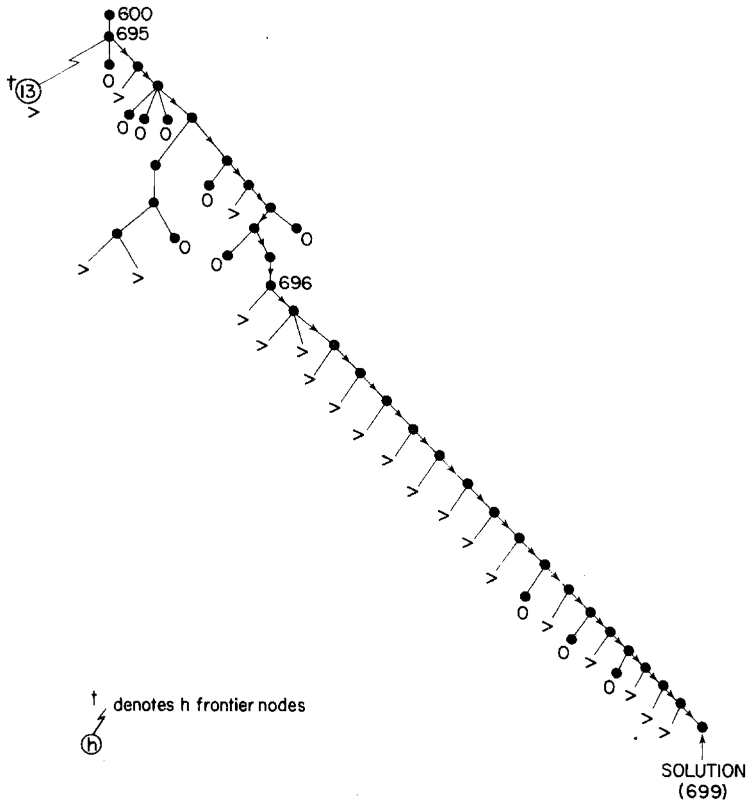
Problem	n	$\bar{t}$	p	C^*	$w(\pi)$ after initial ascent	Type >	Type o	Interior	Time (min)
Random Euclidean (200)	22	1	20	765	763	16	16	22	0.16
Random (700)	25	1	25	2025	2015	21	15	25	0.30
10 × 3 knight's tour	30	1	25	0	0	30	13	31	0.33
Dantzig [4]	42	1	25	699	695	34	11	28	0.90
Tutte	46	1	100	> 0	0	33	284	112	15.00
Held and Karp [8]	48	1	15	11,461	11,430	48	19	48	1.40
8 × 6 Knight's tour	48	1	25	0	0	49	85	85	2.66
Karg and Thompson [9]	57	1	30	12,955	12,906	79	128	182	13.00
Join	64	1	30	1,335	1,321	93	57	115	8.40
Random Euclidean (1500)	64	2	30	9,937	9,924	56	26	57	5.50
Random (1500)	64	1/3	30	9,971	9,970	57	12	44	4.30
8 × 8 Knight's tour	64	1	50	0	0	64	31	72	6.93

computed in the initial ascent are extremely close to $\max_\pi w(\pi)$, so that discrepancies between this bound and C^* are due almost entirely to the "gap" $C^* - \max_\pi w(\pi)$.

The small numbers of nodes in the search trees attest further to the tightness of the bound $\max_\pi w(\pi)$ and the effectiveness of the ascent method in approximating it. In fact, it is possible for us to do something which has never been done before – to present in their entirely the search trees for large combinatorial problems of this type (cf. figs. 2, 3, 4, 5, 6, 7, 8 and 10).

Possible improvements in computing time

The computation times given in tables 1 and 2 appear high considering the simplicity of the search trees. This may be to some extent intrinsic, but considerable improvement can be expected through better



† denotes h frontier nodes

Fig. 2. 42 – city (Dantzig).

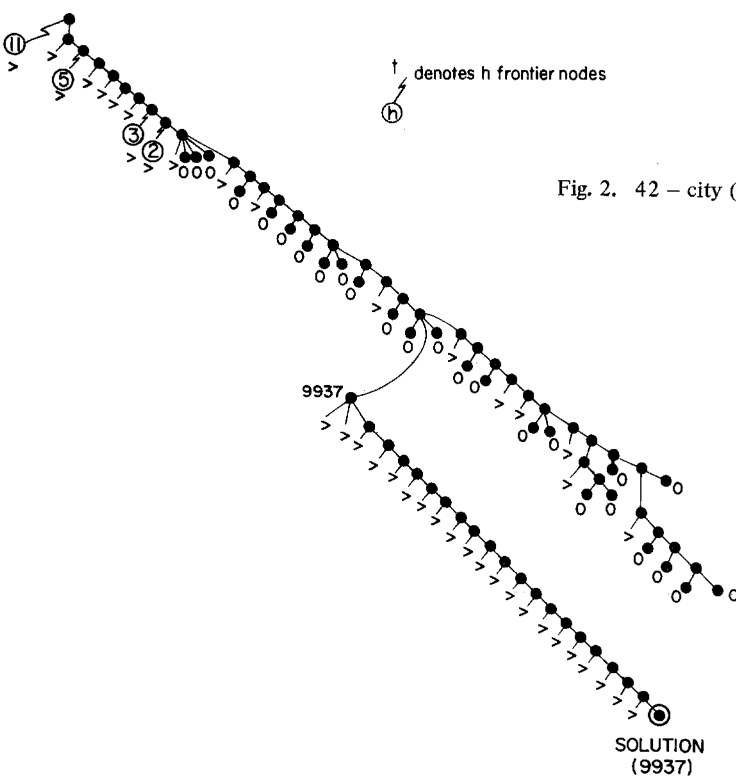


Fig. 3. 64 – city (random Euclidean).

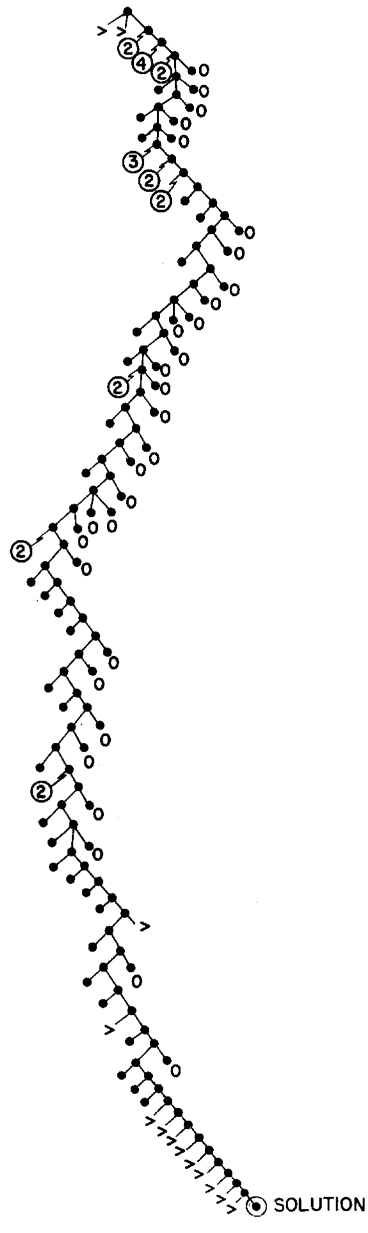


Fig. 4. (8 × 8) 64 – city knights tour.

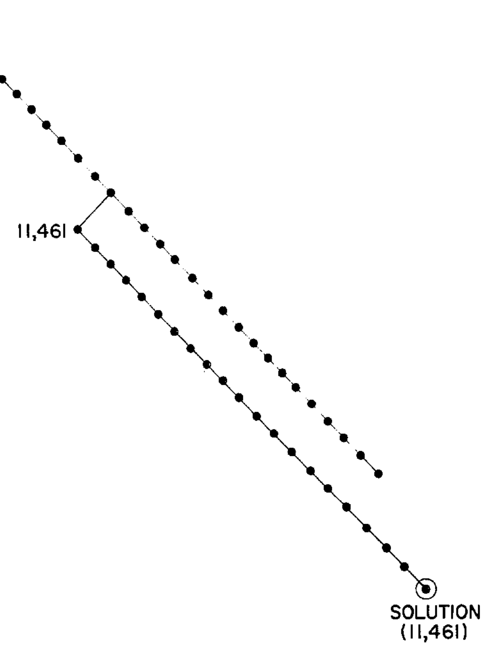


Fig. 5. 48 – city (Held and Karp) Type > and type 0 nodes omitted.

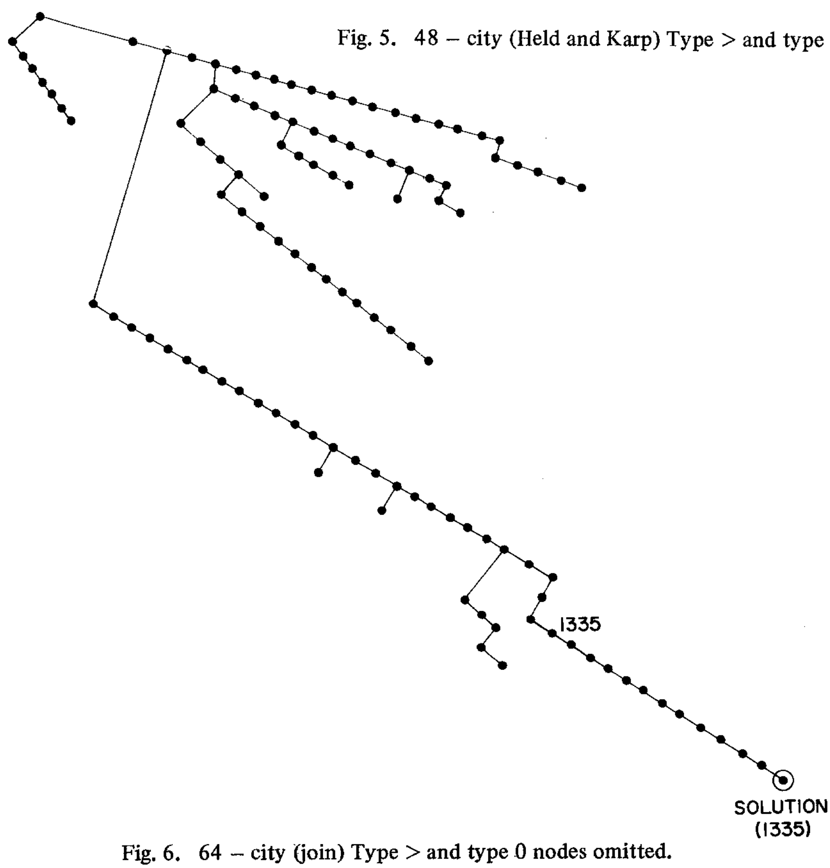


Fig. 6. 64 – city (join) Type > and type 0 nodes omitted.

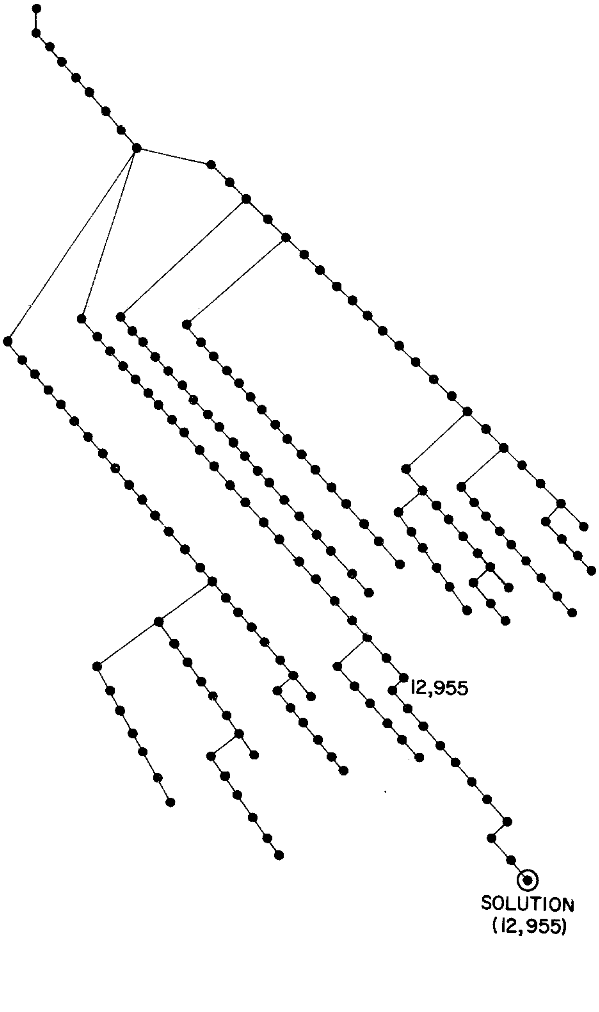


Fig. 7. 57 – city (Karg and Thompson) Type > and type 0 nodes omitted.

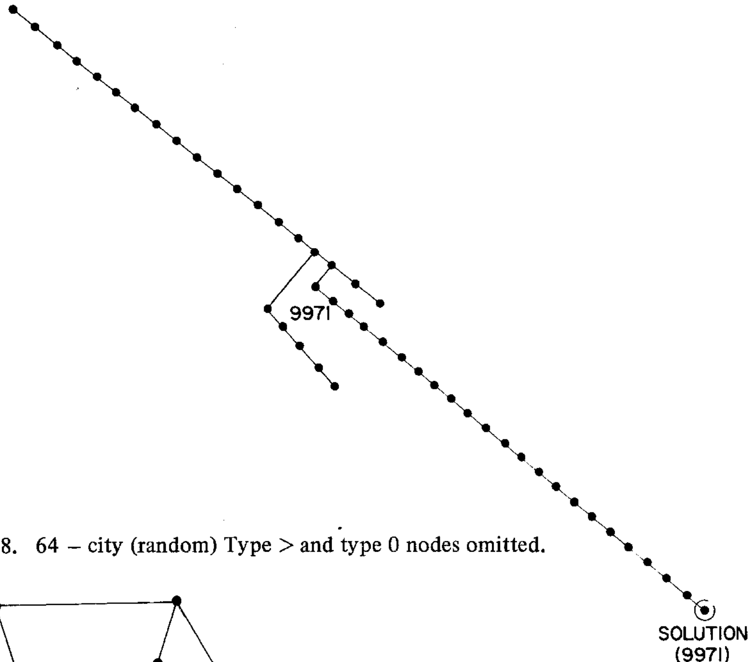


Fig. 8. 64 – city (random) Type > and type 0 nodes omitted.

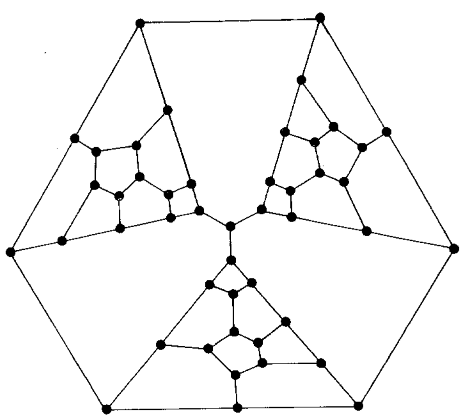


Fig. 9. Tutte graph.

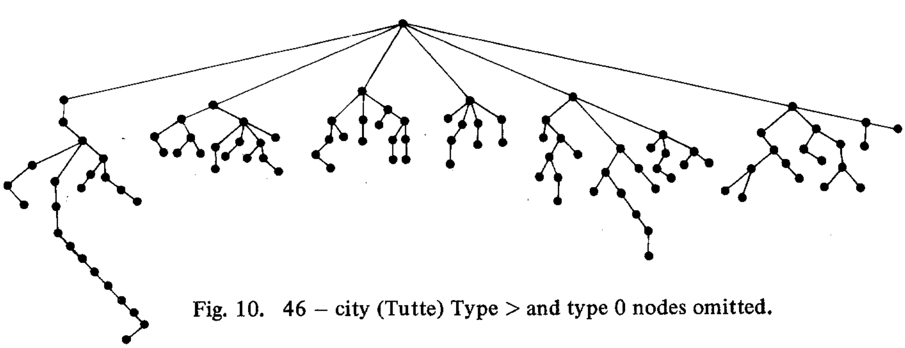


Fig. 10. 46 – city (Tutte) Type > and type 0 nodes omitted.

engineering of the time-consuming ascent procedure. We sketch a few possibilities:

a) A better starting point for the initial ascent can be obtained by solving the assignment problem for the matrix (c_{ij}) and setting $\pi_i^0 = -\frac{1}{2}(u_i + v_i)$, where the u_i and v_j give an optimal solution to the dual of the assignment problem:

$$\text{Maximize } \Sigma\, u_i + \Sigma\, v_j$$

$$\text{Subject to } u_i + v_j \leq c_{ij}\,.$$

It is easily verified that, with this choice, $w(\pi^0)$ is greater than or equal to the cost of an optimal solution to the assignment problem.

b) Keeping t_m, which governs the step size, constant in the ascent method is very crude. Limited experience indicates that it may be far preferable to use a variable step size based upon lemma 3. Similarly, it has been observed that keeping constant the parameter p, which governs the termination of the ascent process, leads to much unnecessary computation. This could be avoided using a more flexible rule.

c) There are two competing ways to compute minimum spanning 1-trees: the method [5] used in the program always requires $O(n^2)$ computation steps; a second method [10] is much faster on the average, but presupposes that the c_{ij} are sorted into ascending numerical order. This method may be significantly preferable, since the ascent method requires a sequence of minimum spanning 1-tree calculations, with the data only slightly changed between calculations, so that the necessary sorting would not need to be time consuming.

Acknowledgements

The authors wish to express their appreciation to Mrs. Linda Ibrahim who programmed the algorithm reported on in this paper. We also wish to thank Philip Wolfe for a number of illuminating observations concerning the ascent method. The existence of references [1] and [12] came to our attention through a serendipitous conversation with Alan Hoffman.

References

[1] S. Agmon, "The relaxation method for linear inequalities," *Canadian Journal of Mathematics* 6 (1954) 382–392.

[2] M. Bellmore and G.L. Nemhauser, "The traveling salesman problem: a survey," *Operations Research* 16 (1968) 538–558.

[3] G.A. Croes, "A method for solving traveling salesman problems," *Operations Research* 6 (1958) 791–812.

[4] G.B. Dantzig, D.R. Fulkerson and S.M. Johnson, "Solution of a large scale traveling salesman problem," *Operation Research* 2 (1954) 393–410.

[5] E.W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik* 1 (1959) 269–271.

[6] D. Gale, "Optimal assignments in an ordered set: an application of matroid theory," *Journal of Combinatorial Theory* 4 (1968) 176–180.

[7] M. Held and R.M. Karp, "The traveling-salesman problem and minimum spanning trees," *Operations Research* 18 (1970) 1138–1162.

[8] M. Held and R.M. Karp, "A dynamic programming approach to sequencing problems," *Journal of the Society for Industrial and Applied Mathematics* 10 (1962) 196–210.

[9] L.L. Karg and G.L. Thompson, "A heuristic approach to solving traveling salesman problems," *Management Science* 10 (1964) 225–248.

[10] J.B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," *Proceedings of the American Mathematical Society* 2 (1956) 48–50.

[11] S. Lin, "Computer solution of the traveling salesman problem," *Bell System Technical Journal* 44 (1965) 2245–2269.

[12] T. Motzkin and I.J. Schoenberg, "The relaxation method for linear inequalities," *Canadian Journal of Mathematics* 6 (1954) 393–404.