A RESTRICTED LAGRANGEAN APPROACH TO THE TRAVELING SALESMAN PROBLEM

by

Egon Balas*6c Nicos Christofides**

DRC-70-4-79

September 1979

^Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, PA 15213

**Imperial College of Science and Technology
London, England

This report was prepared as part of the activities of the Management Science Research Group, Carnegie-Mellon University, under Grant MCS76-12026 A02 of the National Science Foundation and Contract N0014-75-C-0621 NR 047-48 with the U.S. Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U.S. Government.

620.0042
C27d
R-70-4-H

Abstract

We describe an algorithm for the asymmetric traveling salesman problem (TSP) using a new, restricted Lagrangean relaxation based on the assignment problem (AP). The Lagrange multipliers are constrained so as to guarantee the continued optimality of the initial AP solution, thus eliminating the need for repeatedly solving AP in the process of computing multipliers. We give several polynomially bounded procedures for generating valid inequalities and taking them into the Lagrangean function with a positive multiplier without violating the constraints, so as to strengthen the current lower bound. Upper bounds are generated by a fast heuristic whenever possible. When the bound-strengthening techniques are exhausted without matching the upper with the lower bound, we branch by using two different rules, according to the situation: the usual subtour breaking disjunction, and a new disjunction based on conditional bounds. We discuss computational experience on 120 randomly generated asymmetric TSP's with up to 325 cities, the maximum time used for any single problem being 82 seconds. Though the algorithm discussed here is for the asymmetric TSP, the approach can be extended to the symmetric TSP by using the 2-matching problem instead of AP.

1. Outline of the Approach

The traveling salesman problem (TSP), i.e., the problem of finding a minimum-cost tour (or hamiltonian circuit) in a directed graph $G = (N,A)$, can be formulated as the problem of minimizing

$$(1) \qquad \sum_{i\in N}\sum_{j\in N} c_{ij}x_{ij}$$

subject to

$$(2) \qquad \begin{cases} \sum_{j\in N} x_{ij} = 1, & i\in N \\ \sum_{i\in N} x_{ij} = 1, & j\in N \end{cases}$$

$$(3) \qquad x_{ij} \in \{0,1\}, \quad i,j\in N$$

$$(4) \qquad x \text{ is a tour.}$$

For $(i,j) \in A$, c_{ij} is the cost associated with the arc (i,j); for $(i,j) \notin A$, $c_{ij} = \infty$.

Conditions (3), (4) can be replaced by

$$(5) \qquad x_{ij} \geq 0, \quad i,j\in N$$

$$(6) \qquad \sum_{i\in N}\sum_{j\in N} a_{ij}^t x_{ij} \geq a_0^t, \quad t\in Q$$

where (6) is a set of inequalities which, together with (2) and (5), define the convex hull of all tours in G.

If S and T are node sets, we denote $(S,T) = \{(i,j) \in A \mid i\in S, j\in T\}$. For any problem P, we denote by v(P) the value of (an optimal solution to) P.

We describe an arc premium/penalty-based branch and bound method for solving TSP, which uses

(a) a new Lagrangean relaxation of TSP and a restricted Lagrangean problem derived from it, which has constraints on the multipliers;

(b) several procedures for generating inequalities which can be taken into the Lagrangean function with a positive multiplier (premium or penalty), without violating the constraints or changing the multipliers generated earlier;

(c) a new branching rule based on disjunctions derived from conditional bounds.

We first outline the method, then discuss its various components in detail.

The assignment problem (1), (2), (5) associated with TSP will be denoted by AP. It is well known that any integer solution to AP is either a tour (hamiltonian circuit), or a collection of subtours (a union of disjoint circuits). The Lagrangean problem mentioned under (a) is an assignment problem obtained from AP by applying premia or penalties to certain arcs, in a way which is equivalent to taking into the objective function, in a Lagrangean fashion, some of the constraints (6).

From the set of inequalities (6) we extract a subset

$$(7) \qquad \sum_{i\in N}\sum_{j\in N} a^t_{ij} x_{ij} \geq a^t_0, \quad t \in T \subset Q$$

and call LP the linear program (1), (2), (5), (7). Though the set Q is at least exponential in |N|, empirical evidence as well as theoretical considerations indicate that there are relatively small subsets T of Q such that the value of the corresponding LP comes very close to (or coincides with) that of TSP.

Using Lagrangean relaxation on (7) and denoting by w the vector of Lagrange multipliers, we obtain the problem L(w,x), equivalent to LP, of finding $w \geq 0$ to maximize z(w), where

$$(1') \qquad z(w) = \min \sum_{i\in N}\sum_{j\in N} \Big(c_{ij} - \sum_{t\in T} w_t a^t_{ij}\Big) x_{ij} + \sum_{t\in T} w_t a^t_0$$

subject to

$$(2) \qquad \begin{cases} \sum_{j\in N} x_{ij} = 1, & i\in N \\ \sum_{i\in N} x_{ij} = 1, & j\in N \end{cases}$$

$$(5) \qquad x_{ij} \geq 0, \quad i,j\in N.$$

The Lagrangean relaxation $L(w,x)$ of TSP can be used to generate lower bounds on $v(TSP)$. While $L(w,x)$ may yield very strong bounds indeed, depending on the choice of the inequalities (6), its solution via, say, subgradient optimization, requires a considerable computational effort, including the solution of the assignment problems associated with every vector w generated during the procedure. Instead, we consider a restriction $RL(w,x)$ of the Lagrangean relaxation $L(w,x)$. Let $\bar{x}$ be an optimal solution to AP, i.e., to the assignment problem with cost function (1). $RL(w,x)$ is then the problem of finding $w \geq 0$ to maximize $z(w)$ defined by (1'), (2) and (5), subject to

$$(8) \qquad u_i + v_j + \sum w_t a_{ij}^t \begin{cases} = c_{ij} & \text{if } \bar{x}_{ij} > 0 \\ \\ \leq c_{ij} & \text{if } \bar{x}_{ij} = 0 \end{cases}$$

for some $u,v \in R^n$.

Problem $RL(w,x)$ has two properties which make it useful towards solving TSP.

First, any $\hat{u}$, $\hat{v}$ and $\hat{w}$ satisfying (8) and $\hat{w} \geq 0$ is a feasible solution to the linear program dual to LP; therefore the objective function value of this dual linear program is a lower bound on $v(LP)$, hence on $v(TSP)$, i.e., we have

Proposition 1. If $\hat{u}$, $\hat{v}$ and $\hat{w} \geq 0$ satisfy (8), then

$$(9) \qquad \sum_{i \in N} \hat{u}_i + \sum_{j \in N} \hat{v}_j + \sum_{t \in T} \hat{w}_t a_0^t \leq v(TSP).$$

Second, while $\bar{x}$ remins an optimal solution to the assignment problem with the modified objective function (1'), the changes brought about by the penalties/premia $\hat{w}_t$, $t \in T$, are likely to create new, alternative optima.

Whenever such an alternative optimal solution $\hat{x}$ to the assignment problem (1'), (2), (5) turns out to be a tour, it has the following property.

Proposition 2. If $\hat{x}$ satisfies with equality the inequality (7) indexed by t for all $t \in T$ such that $\hat{w}_t > 0$, $\hat{x}$ is an optimal tour.

Proof, $\hat{x}$ and $(\hat{u},\hat{v},\hat{w})$ are feasible solutions to LP and its dual, respectively. From the definition of $\hat{x}$, we have

$$\hat{u}_i + \hat{v}_j + \sum_{t\in T} \hat{w}_t a^t_{ij} = c_{ij}$$

whenever $\hat{x}_{ij} > 0$. This, together with the condition of the Proposition, means that $\hat{x}$ and $(\hat{u},\hat{v},\hat{w})$ satisfy the complementary slackness conditions. Thus $\hat{x}$ is an optimal solution to LP, hence an optimal tour.||

We start by solving the assignment problem AP in the free variables . Next we use several different procedures for generating an increasing sequence of lower bounds on v(TSP), by successively identifying inequalities (7) that

(i) are not satisfied by the current solution $\bar{x}$ to AP, and

(ii) admit a positive multiplier w_t which, together with the multipliers already assigned, satisfies (8);

and by setting the multipliers w_t each time to the greatest positive value compatible with (ii). At any given stage, the admissible graph $G_Q = (N,A_Q)$ is the spanning subgraph of G containing those and only those arcs with zero reduced cost, i.e.,

$$A_Q = \{(i,j)\in A \mid u_i + v_j + \sum_{t\in T} w_t a^t_{ij} = c_{ij}\}.$$

When no more inequalities (7) satisfying conditions (i) and (ii) can be found, we store the bound given by (9) and try to find a tour in the admissible graph. If a tour is found which satisfies with equality all inequalities associated with positive multipliers, it is optimal for the given subproblem. If a tour is found which violates this condition for some inequalities, attempts are made at finding new inequalities which satisfy the condition and admit positive multipliers. If successful, these attempts strengthen the lower bound, and they may also eliminate the inequalities that are slack. In any case, the value of the tour (in the original costs c_{ij}) provides an upper bound on v(TSP), while (9) provides a lower bound for the current subproblem; and we branch. Finally, if no tour is found in $G_{\underset{\sim}{Q}}$, we add arcs to $G_{\underset{\sim}{Q}}$ in the order of increasing reduced costs until a tour is found in the resulting graph. The cost of this tour again provides an upper bound on v(TSP), while (9) still provides a lower bound for the current subproblem; and we branch.

The assignment problems are solved by the Hungarian method, the same method is used to recalculate the reduced costs whenever some u_i and v_j have to be changed. The constraints (7) are "subtour-breaking" inequalities and combinations of the latter with*some of the equations (2), but they are used here in a novel way. The bounding procedures are all polynomial-time algorithms, considerably more efficient (in terms of improvement obtained versus computational effort) than earlier approaches (like [3]), as evidenced by the computational results of section 5. Searching the admissible graph $G_{\underset{-}{Q}}$ for a tour is accomplished by a specialized implicit enumeration procedure, with a cut-off rule. Finally, for branching we use two different rules, one which derives a disjunction from a conditional bound [1], and one which breaks up a subtour.

A preliminary version of our approach, with fewer and less sophisticated bounding procedures, was discussed in [2].

2. Bounding Procedures

At any stage of the procedure, the reduced costs

$$\bar{c}_{ij} = c_{ij} - u_i - v_j - \sum_{t\in T'} w_t a_{ij}^t$$

will be defined relative to the subset $T' \subset T$ of inequalities already introduced into the function (1').

We use three types of inequalities (7), and we will denote by T_1, T_2 and T_3 the corresponding subsets of T.

For $t\in T$, let $\emptyset \neq S_t \subset N$ and $\bar{S}_t = N \setminus S_t$. An arc set of the form

$$K_t = (S_t, \bar{S}_t)$$

is called a (directed) cutset.

Clearly, the inequalities

(7a) $$\sum_{(i,j)\in K_t} x_{ij} \geq 1, \qquad t\in T_1$$

are satisfied by every tour, and so are the inequalities

(7b') $$\sum_{i\in S_t} \sum_{j\in S_t} x_{ij} \leq |S_t| - 1, \qquad t\in T_2$$

or, to preserve the direction of the inequality,

(7b) $$- \sum_{i\in S_t} \sum_{j\in S_t} x_{ij} \geq 1 - |S_t|, \qquad t\in T_2 .$$

For a given set S_t, the "subtour-breaking" inequalities (7a) and (7b) are equivalent: (7b) can be obtained from (7a) by subtracting the sum of the equations

$$\sum_{j\in N} x_{ij} = 1, \quad i\in S_t,$$

and (7a) can be obtained from (7b) by the reverse operation. Nevertheless, the presence of inequalities associated with the same set S_t in both subsets (7a) and (7b) need not be avoided, since it may enrich the set of dual vectors (u,v,w) satisfying (8) and $w \geq 0$.

Finally, for any $k \in N$, $S_t \subset N \setminus \{k\}$ and $\bar{S}_t = N \setminus S_t$, the arc sets

$$K'_t = (S_t, \bar{S}_t \setminus \{k\}) \quad \text{and} \quad K''_t = (\bar{S}_t \setminus \{k\}, S_t)$$

are (directed) cutsets in the subgraph $\langle N \setminus \{k\} \rangle$ of G induced by $N \setminus \{k\}$.

Proposition 3. The inequalities

(7c) $$\sum_{(i,j) \in K'_t \cup K''_t} x_{ij} \geq 1, \qquad t \in T_3$$

are satisfied by every tour.

Proof. Every $x \in \{0,1\}^n$ that violates (7c) corresponds to a subgraph G' of G which is either disconnected, or contains an articulation point k; hence G' cannot be a tour.||

Actually, more can be said about the family (7c):

Proposition 4. In the presence of the constraint set (2), for every $k \in N$ and $S_t \subset N \setminus \{k\}$, the inequality (7c) is implied by the two subtour-breaking inequalities associated with the node sets $S_t \cup \{k\}$ and S_t respectively, i.e., by

$(7b)_1$ $$-\sum_{i \in S_t \cup \{k\}} \sum_{j \in S_t \cup \{k\}} x_{ij} \geq -|S_t|$$

and

$(7b)_2$ $$-\sum_{i \in S_t} \sum_{j \in S_t} x_{ij} \geq 1 - |S_t|.$$

Proof. The inequality (7c) is the sum of $(7b)_1$, $(7b)_2$ and the equations

$$\sum_{j \in N} x_{ij} = 1, \qquad i \in S_t$$

$$\sum_{i \in N} x_{ij} = 1, \qquad j \in S_t.||$$

The components of w associated with the inequalities (7a), (7b), and (7c) will be denoted by λ, μ and ν respectively.

2.1. <u>Bounding Procedure 1</u> starts by searching for an inequality (7a) which satisfies conditions (i), (ii) of section 1, i.e., is violated by $\bar{x}$ and can be assigned a positive multiplier without making any of the reduced costs negative. Clearly, these conditions are satisfied for the inequality (7a) defined by a cutset K_t, if and only if

$$(10) \qquad K_t \cap A_0 = \emptyset,$$

where A_0 is the arc set of the admissible graph G_0.

To find K_t satisfying (10), we choose any node $i \in N$ and form its reachable set $R(i)$ in G_0. If $R(i) = N$, there is no cutset $(S,\bar{S})$ with $i \in S$, satisfying (10), so we choose another $i \in N$. If for some $i \in N$, $R(i) \neq N$, then $K_t = (S,\bar{S})$ satisfies (10) for $S = R(i)$. Furthermore,

$$(11) \qquad \lambda_t = \min_{(i,j) \in K_t} \bar{c}_{ij}$$

is clearly the largest value that can be assigned to the corresponding multiplier without making some reduced costs negative. We thus assign λ_t the above value and set

$$\bar{c}_{ij} \leftarrow \bar{c}_{ij} - \lambda_t, \quad (i,j) \in K_t,$$

i.e., we apply a premium of λ_t to each arc of the cutset K_t. As a result of this, the arcs for which the minimum in (11) is attained, become admissible, and we add them to A_0, thus enlarging the admissible graph G_0.

Next, we extend the reachable set $R(i)$ of node i by using the new arcs of G_0 and either find $R(i) = N$, or locate another cutset K_t satisfying (10). If $R(i) = N$, again we choose another node.

This procedure ends when $R(i) = N$, $\forall\, i \in N$. At that stage G_0 is strongly connected, and

$$K \cap A_0 \neq \emptyset$$

for all cutsets $K = (S,\bar{S})$, $S \subset N$.

Proposition 5. Bounding Procedure 1 stops after generating at most $\frac{1}{2}$"(h-l)(h+2) cutsets, where h is the number of subtours in $\bar{x}$.

Proof Starting with node *i^* belonging to subtour S_1, every cutset adds to G_0 an ate which includes into R(i^) a new subtour. After generating at most h-1 cutsets, RCi^-N. Now starting with node i_2 belonging to subtour *S_2t$$_l$* and proceeding to find R(i_2), again at most h-1 cutsets can be generated. However, since we now have i_2c R(i^) and i^e R(i_2), the number of strong components of the current graph G_Q is at most h-1. Thus, continuing to find R(i_3) for some node i_3 belonging to a subtour S_3, S^S^Sj, at most h-2 cutsets can be generated, and since the vertices of S_1, S_2 and S_3 now form a strong component, the number of strong components in the current graph G_0 is at most h-2. Continuing in the same way, the number of cutsets generated by the procedure (until $\bar{G}_o$ becomes strongly connected) is at most

$$(h-1) + (h-1) + (h-2) + (h-3) + ... + 1 » \tfrac{1}{2}(h-l)(h+2).|]$$

Since the optimal dual variables $\bar{u}$^ $\bar{v}_j$ associated with $\bar{x}$ are not changed by this procedure, and since

$$\sum_{i \in N} \bar{u}_i + \sum_{j \in N} \bar{v}_j - c\bar{x} » v(AP),$$

if T_x is the index set of the inequalities generated by Bounding Procedure 1, the lower bound obtained for the current subproblem is, from Proposition 1,

(12) $$B_1 » v(AP) + \sum_{t \in T_1} \lambda_t .$$

2-2- Bounding Procedure 2 starts by searching for an inequality (7b) which is violated by $\bar{x}$ and admits a positive penalty without changing any of the X_ti t eT^. If S^,...,Sk are the node sets of the h subtours of $\bar{x}$, every inequality (7b) defined by S_t, t - l,...,h, is violated by $\bar{x}$; but a positive penalty ^ can be applied without violating the condition that $\bar{x}_{ij}$ > 0 implies $\bar{c}_{ij}$ » 0, only by changing the values of some u_i and v_j, and only if an additional condition is satisfied. This condition can best be expressed in terms of the assignment tableau used in conjunction with the Hungarian method. A <u>line</u> of the tableau is a row or a column, a <u>cell</u> of the tableau is the intersection of a row and a column. Cells correspond to arcs and are denoted the same way.

Let S_t be the node set of a subtour of $\bar{x}$, let

$$A_t = \{(i,j) \in A_0 \mid i,j \in S_t\}$$

and

$$A'_t = \{(i,j) \in A_t \mid \bar{x}_{ij} > 0\}.$$

<u>Proposition 6</u>. A positive penalty can be applied to the arcs with both ends in S_t if and only if there exists a set C of lines such that

(i) every $(i,j) \in A'_t$ is covered by exactly one line in C,

(ii) every $(i,j) \in A_t \setminus A'_t$ is covered by at most one line in C,

(iii) no $(i,j) \in A_0 \setminus A_t$ is covered by any line in C.

If such a set C exists, and it consists of row set I and column set J, then the <u>[illegible]</u> applicable penalty is

$$\pi_t = \min_{(i,j) \in M} \bar{c}_{ij}, \tag{13}$$

where

$$M = (I,J) \cup (I,\bar{S}_t) \cup (\bar{S}_t,J). \tag{14}$$

<u>Proof</u>. Sufficiency. Suppose there exists a line set C, consisting of row sets I and column sets J, satisfying conditions (i), (ii), (iii). Then adding an amount $\pi > 0$ to $\bar{c}_{ij}$ for all $(i,j) \in (S_t,S_t)$, as well as to all $\bar{u}_i$, $i \in I$, and all $\bar{v}_j$, $j \in J$, produces a set of reduced costs $\bar{c}'_{ij}$ such that $\bar{c}'_{ij} = 0$ for all $(i,j) \in A'_t$, since $C = I \cup J$ satisfies (i). Further, from property (ii) of the set C, $\bar{c}'_{ij} \geq 0$, $\forall (i,j) \in A_t \setminus A'_t$; and from

(iii), $Z'_{ij}. = \bar{c}_{ij}. = 0$, V (i,j) « $A_0\backslash A_t$. Thus the only reduced costs that get diminished as a result of the above modification, are those associated with arcs (i,j) «$_A$ for which either (a) nothing is added to $\bar{c}_{j,j}$ and p, is added to $\bar{u}_\pm$ or to $\bar{v}_{j5}$ or (0) ^ is added to $Z_{\pm j}$. and to both $\bar{u}_t$ and $\bar{v}$^. The two sets of arcs for which («) holds are (I,§$_t$) and (§$_t$,J); whereas the arc set for which 0) holds is (I,J). The union of these three arc sets is M defined by (14). Thus a positive penalty at most equal to ^ defined by (13) can be applied to the arc set (S_t,S_t) in the above described manner without producing any negative reduced costs.

Necessity. Suppose a penalty n > 0 can be applied to the arc set (S_t,S_t). Since adding n > 0 to $Z_{\pm y}$ V (i,j) e (S_t,S_t), produces positive reduced costs for all (i,j) eA$_{fc}$, in order to obtain reduced costs $\bar{c}_{ij}$. = 0 for all (i,j) eA^, one must increase by p, the sum $\bar{u}_i + \bar{v}_j$ for all (i,j)eA$'_t$. It is easy to see that if this can be done, then it can be done by adding n to $\bar{u}_£$ or $\bar{v}_j$ (but not to both), for every (i.jjeA$^\wedge_t$, hence there exists a set C of lines satisfying condition (i) . Further, if (i,j) S A ^ , then n cannot be added to both $\bar{u}_i$ and $\bar{v}_j$ without creating $\bar{c}_{ij}$< 0, hence C must satisfy (ii). Finally, if (i»j) eA$_Q$\A$_t$, then p, cannot be added to either $\bar{u}_i$ or $\bar{v}_j$ without making $\bar{c}_{ij}$!. < 0, hence condition (iii) must also hold.||

Given the node set S_t of a subtour, we have to check whether a set of lines C satisfying (i), (ii), (iii) exists. This can be done as follows.

First, every row i eS_t such that (i,j) «A$_Q$ for some j eN\S$_t$, and every column j «S$_£$ such that (i,j) eA$_Q$ for some i eN\S$_t$, can be ruled out as a candidate for entering C. Let R and K be the index sets of such rows and columns respectively,, and let

$$I_1 = \{i \in N | (i,j) \in A_j' \text{ and } j \in K\},$$

$$J_1 = \{j \in N | (i,j) \in A_t' \text{ and } i \in R\}.$$

Since by (i) every cell of A'_t must be covered by at least one line in C, C must contain $I_1 \cup J_1$. For the same reason, if

(15) $$A'_t \cap R \cap K \neq \emptyset$$

then no positive penalty can be applied to the arc set $(S_t, \bar{S}_t)$.

Since by (i) and (ii) every cell of A'_t must be covered by at most one line in C, if

(16) $$A'_t \cap I_1 \cap J_1 \neq \emptyset,$$

then again no positive penalty can be applied to the arc set $(S_t, \bar{S}_t)$. Now assume neither (15) nor (16) holds. Then if (I_1, J_1) covers A'_t, we set $C = I_1 \cup J_1$ and we are done; otherwise we use the Hungarian algorithm to complete the search for a cover satisfying (i), (ii), (iii). If such a cover exists, the Hungarian algorithm finds it, and μ_t given by (13) can be applied as a penalty; otherwise the Hungarian method finds a cover which violates some of the conditions (i), (ii), (iii), in which case no positive penalty can be applied.

If T_2 is the index set of the inequalities (7b) which admit positive penalties μ_t, we have the following

Proposition 7.

(17) $$B_2 = B_1 + \sum_{t\in T_2} \mu_t$$

is a lower bound on the value of the current subproblem.

Proof. Whenever a penalty $\mu_t > 0$ is applied to an arc set $(S_t, \bar{S}_t)$ associated with a constraint (7b), the cost function of AP is modified, and the value of the solution $\bar{x}$, hence also the value of a solution to the dual of $\overline{AP}$, the assignment problem with the modified costs, is increased by $(|S_t|-1)\mu_t$. Thus, after applying $|T_2|$ penalties μ_t, the value of an optimal solution $(\hat{u},\hat{v})$ to the dual of $\overline{AP}$ is

$$\sum_{i\in N} <_{i} + \sum_{j\in N} \hat{v}_{j} - v(AP) + \sum_{t\in T_2} |S_t|\mu_t .$$

Using Proposition 1, and noting that a!$_b$ • 1 for t e l,_ and a*$_0$= 1 - |s$_t$| for t e T$_2$, we obtain the lower bound

$$B_{2} \text{--} \; \text{EE ffi}_{i} \text{:+} \; SS \; *_{j} + \sum_{t\in T_1} X_t + \sum_{t\in T_2} (l-|S_t|)\mu_c .$$

$$- v(AP) + \sum_{t\in T_1} \lambda_t + \sum_{t\in T_2} n_t$$

$$= B_1 + \sum_{t\in T_2} \mu_t . \|$$

2.3. <u>Bounding Procedure 3</u> searches for inequalities of the form (7c) which are violated by $\bar{x}$ and admit a positive multiplier v_t without requiring changes in the multipliers assigned earlier. This is done by checking for each node whether it is an articulation point of G_Q. If node k is an articulation point, i.e., if the subgraph ^N - [k ^ of G_Q is disconnected, with S_{fc} as one of its components, then denoting K^ » (S_t,?$_t$\{k}) and K''_t » (?$_t$\[k},S_t) we have

$$K\pounds n A_Q - 0, \quad K\pounds n A_Q » 0.$$

Thus we can apply a positive premium to the arcs in the pair of cutsets ¥?$_t$., K^*_t, whose value is

(18) $$v_t » \min_{(i,j)\in K'_t \cup K''_t} \bar{c}_{ij} .$$

If T_3 is the index set of ail those inequalities (7c) found to admit a positive multiplier, at the end of Bounding Procedure 3 we have (from

Proposition 1 and (17)) the lower bound

$$B_3 = B_2 + \sum_{t\in T_3} \nu_t$$

(19) $$= v(AP) + \sum_{t\in T_1} \lambda_t + \sum_{t\in T_2} \mu_t + \sum_{t\in T_3} \nu_t .$$

If at any time during the Bounding Procedure the current lower bound matches (or exceeds) the upper bound given by the value of the best available tour, the current subproblem is fathomed and we turn to another node of the search tree. Otherwise, after obtaining the bound B_3 we try to find a tour in G_0.

2.4. <u>Example 1</u>. Consider the 9-city TSP whose cost matrix is shown in Table 1.

$[c_{ij}]=$	1	2	3	4	5	6	7	8	9
1	X	2	8	11	15	12	12	[illegible]	13
2	8	X	4	12	18	14	12	14	17
3	6	9	X	15	20	17	13	10	17
4	13	15	17	X	5	8	11	15	16
5	10	14	16	3	X	16	12	15	13
6	7	7	11	9	14	X	3	7	9
7	5	7	[illegible]	2	4	11	X	2	9
8	4	10	1	[illegible]	[illegible]	10	13	X	4
9	9	5	[illegible]	11	8	2	7	7	X

<u>Table 1</u>

The solution to AP has value 31. The reduced cost matrix $[\bar{c}_{ij}]$ is shown in Table 2 and the solution $\bar{x}$ is given by $\bar{x}_{ij} = 1$ for those (i,j) corresponding to boxes in that matrix, $\bar{x}_{ij} = 0$ otherwise. The corresponding admissible graph is shown in Fig. 1.

$$[\bar{c}_{ij}] = \begin{array}{c|ccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline
1 & X & \boxed{0} & 6 & 9 & 13 & 10 & 10 & 9 & 11 \\
2 & 4 & X & \boxed{0} & 8 & 14 & 10 & 8 & 10 & 13 \\
3 & \boxed{0} & 3 & X & 9 & 14 & 11 & 7 & 4 & 11 \\
4 & 8 & 10 & 12 & X & \boxed{0} & 3 & 6 & 10 & 11 \\
5 & 7 & 11 & 13 & \boxed{0} & X & 13 & 9 & 12 & 10 \\
6 & 4 & 4 & 8 & 6 & 11 & X & \boxed{0} & 4 & 5 \\
7 & 3 & 5 & 4 & 0 & 2 & 9 & X & \boxed{0} & 6 \\
8 & 0 & 6 & 3 & 5 & 8 & 6 & 9 & X & \boxed{0} \\
9 & 7 & 3 & 7 & 9 & 6 & \boxed{0} & 5 & 5 & X \\ \hline
\end{array}$$

Table 2

Bounding procedure 1. Cutset K_1 = ({1,2,3}, {4,5,6,7,8,9} admits $\lambda_1 = 4$, and cutset K_2 = ({4,5}, {1,2,3,6,7,8,9}) admits $\lambda_2 = 3$. The lower bound, from (12), becomes $B_1 = 31 + 4 + 3 = 38$. The new reduced cost matrix is shown in Table 3 and the corresponding admissible graph in Fig. 2.

$$[\bar{c}_{ij}] = \begin{array}{c|ccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ \hline
1 & X & \boxed{0} & 6 & 5 & 9 & 6 & 6 & 5 & 7 \\
2 & 4 & X & \boxed{0} & 4 & 10 & 6 & 4 & 6 & 9 \\
3 & \boxed{0} & 3 & X & 5 & 10 & 7 & 3 & 0 & 7 \\
4 & 5 & 7 & 9 & X & \boxed{0} & 0 & 3 & 7 & 8 \\
5 & 4 & 8 & 10 & \boxed{0} & X & 10 & 6 & 9 & 7 \\
6 & 4 & 4 & 8 & 6 & 11 & X & \boxed{0} & 4 & 5 \\
7 & 3 & 5 & 4 & 0 & 2 & 9 & X & \boxed{0} & 6 \\
8 & 0 & 6 & 3 & 5 & 8 & 6 & 9 & X & \boxed{0} \\
9 & 7 & 3 & 7 & 9 & 6 & \boxed{0} & 5 & 5 & X \\ \hline
\end{array}$$

Table

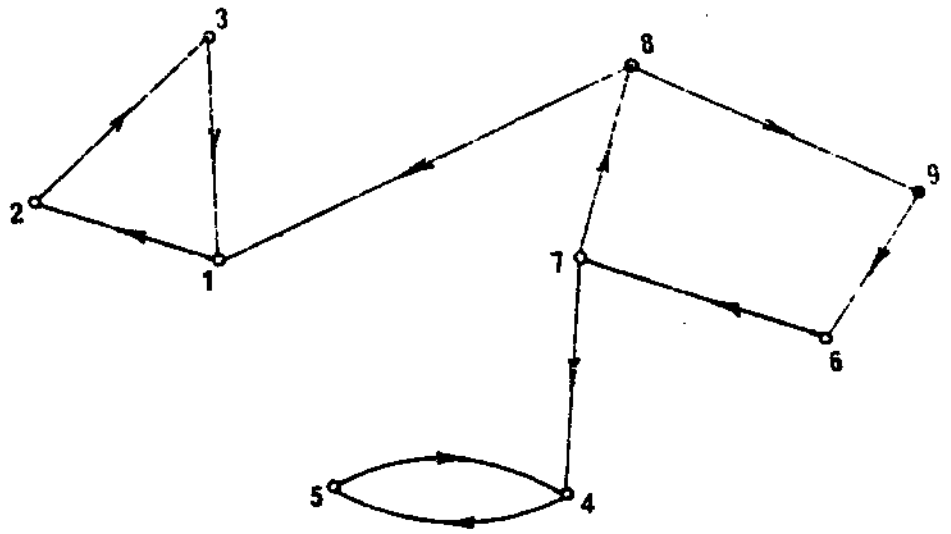

Fig 1, Graph G^ defined by the AP solution.

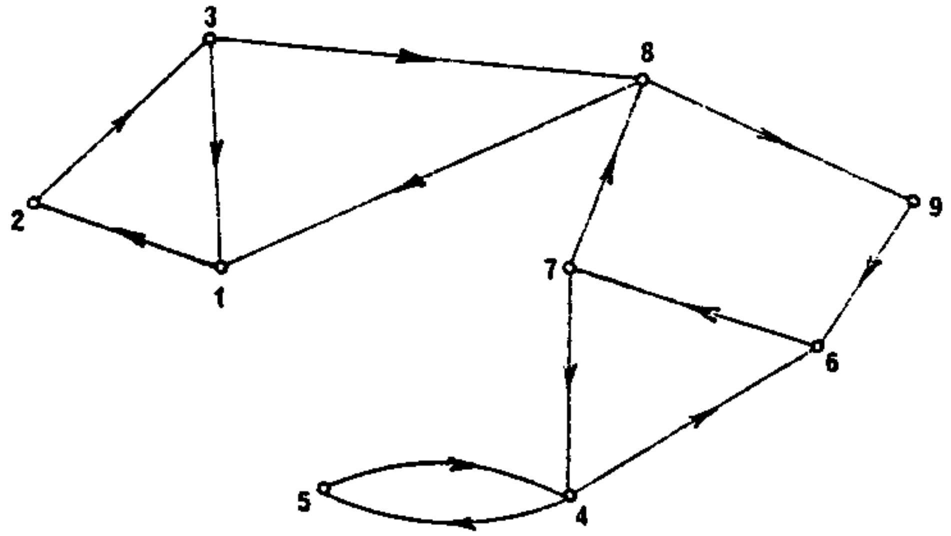

Fig 2. *G** after bounding procedure 1.

Bounding procedure 2. The subtours of the AP solution are (1,2,3), (4,5) and (6,7,8,9). Subtours (1,2,3) and (6,7,8,9) do not admit positive values of λ_t. However, inequality (7b) for subtour (4,5) is

$$-(x_{45} + x_{54}) \geq -1,$$

and a set C of lines of the matrix of Table 3 satisfying the conditions of Proposition 6 for this subtour is given by: (row 5, column 5). From this set C we compute $p_t = 2$. From Proposition 7, the lower bound becomes $B_2 = 38 + 2 = 40$.

The new reduced cost matrix $[\bar{c}_{ij}]$ is shown in Table 4, and the corresponding admissible graph in Fig. 3.

$$[\bar{c}_{ij}] = \begin{array}{c|ccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 3 & 9 \\
\hline
1 & X & \boxed{0} & 6 & 5 & 7 & 6 & 6 & 5 & 7 \\
2 & 4 & X & \boxed{0} & 4 & 8 & 6 & 4 & 6 & 9 \\
3 & \bar{A} & 3 & X & 5 & 8 & 7 & 3 & 0 & 7 \\
4 & 5 & 7 & 9 & X & \boxed{0} & 0 & 3 & 7 & 3 \\
5 & 2 & 6 & 8 & \boxed{0} & X & 3 & 4 & 7 & 5 \\
6 & 1 & 4 & 8 & 6 & 9 & X & \boxed{0} & 4 & 5 \\
7 & 3 & 5 & 4 & 0 & 0 & 9 & X & \boxed{0} & 6 \\
3 & 0 & 6 & 3 & 5 & 6 & 6 & 9 & X & 3 \\
9 & 7 & 3 & 7 & 9 & 4 & \boxed{0} & 5 & 5 & X \\
\hline
\end{array}$$

Table 4

Bounding procedure 3. Vertex 8 is an articulation point of the admissible graph of Fig. 3. The cutsets corresponding to this articulation point are $K_1 = (\{1,2,3\}, \{4,5,6,7,9\})$ and $K_2 = (\{4,5,6,7,9\}, \{1,2,3\})$. Applying (18) to Table 4, we obtain $v_t=2$, corresponding to element (5,1). From (19) the lower bound becomes $B_3= 40 + 2 = 42$. The new reduced cost matrix and the corresponding admissible graph are shown in Table 5 and Fig. 4 respectively.

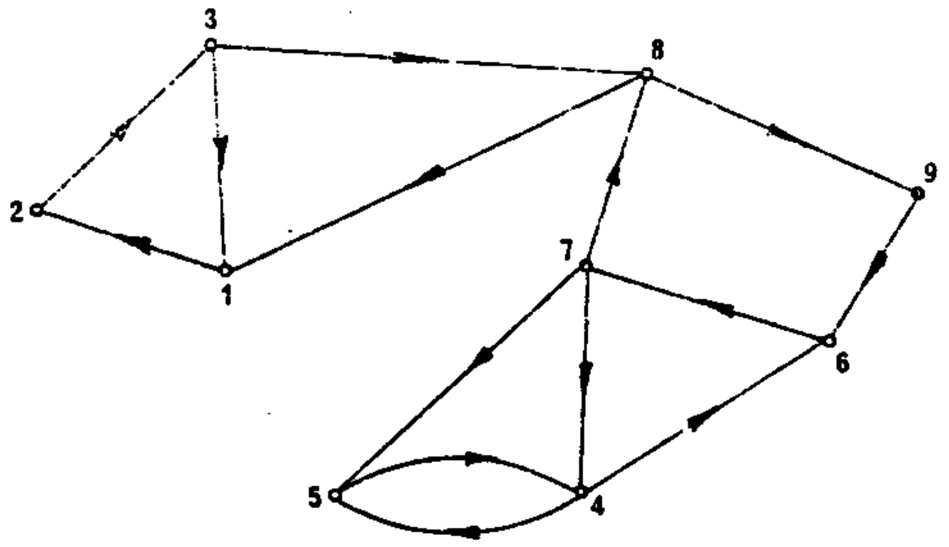

Fig. 3. $\underset{\sim}{G}$ after bounding procedure 2.

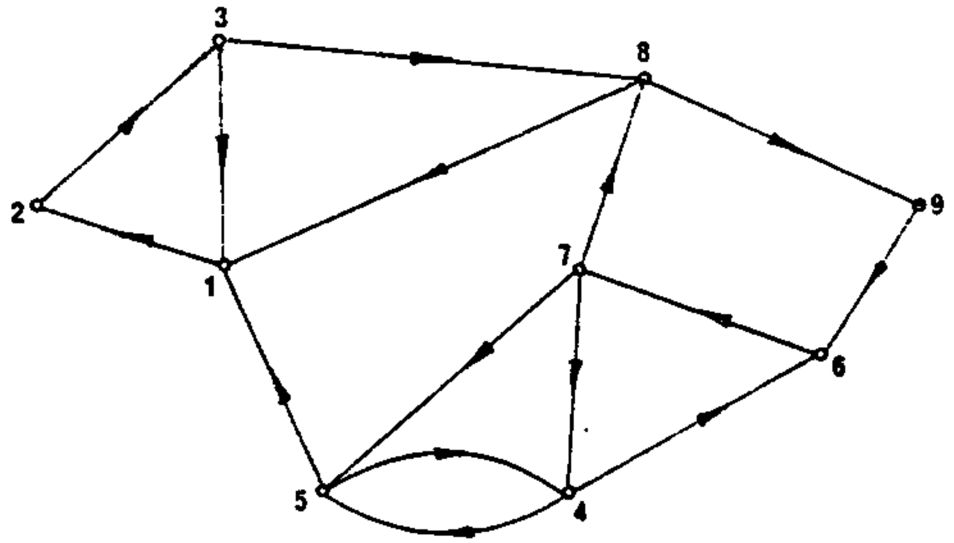

Fig. 4. $\underset{\sim}{G}_n$ after bounding procedure 3.

$$[c_{ij}] = \begin{array}{c|ccccccccc|}
 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 3 & 9 \\
\hline
1 & X & \boxed{0} & 6 & 3 & 5 & 4 & 4 & 5 & 5 \\
2 & 4 & X & \boxed{0} & 2 & 6 & 4 & 2 & G & 7 \\
3 & \boxed{0} & 3 & X & 3 & 6 & 5 & 1 & 0 & 5 \\
4 & 3 & 5 & 7 & X & \boxed{0} & 0 & 3 & 7 & 8 \\
5 & 0 & 4 & 6 & \boxed{0} & X & 8 & 4 & 7 & 5 \\
6 & 2 & 2 & 6 & 6 & 9 & X & m & 4 & 5 \\
7 & 1 & 3 & 2 & 0 & 0 & 9 & X & a & 6 \\
3 & 0 & 6 & 3 & 5 & 6 & 6 & 9 & X & \boxed{0} \\
9 & 5 & 1 & 5 & 9 & 4 & \boxed{0} & 5 & 5 & X \\
\hline
\end{array}$$

Table 5

3. Finding a Tour and Improving the Bound

Establishing whether a graph contains a tour (i.e., is hamiltonian) is, from the point of view of worst-case analysis, of the same order of difficulty as finding an optimal tour. However, for the vast majority of all possible graphs, the first problem is incomparably easier than the second one. We use a specialized implicit enumeration procedure, the multi-path method of [4] , ch. 10, for finding a tour in G_Q if one can be found without exceeding a given time limit. Let $\hat{x}$ denote the solution associated with such a tour. If $\hat{x}$ satisfies with equality all inequalities associated with positive multipliers (i.e., if the tour defined by $\hat{x}$ crosses exactly once each cutset K_t, $t \in T_1$, contains exactly $|S_t| - 1$ arcs with both ends in S_t for all sets S_t, $t \in T_2$, and contains exactly one arc of each pair of cutsets, K'_t, K''_t, $t \in T_3$), then $\hat{x}$ defines an optimal tour for the current subproblem, and the latter is fathomed.

For example, after bounding procedure 3, when G_Q is the graph of Fig. 4, the following tour is detected in G_Q: H*(1,2,3,8,9,6,7,4,5,1). This tour satisfies with equality all four constraints with positive associated Lagrange multipliers; i.e., H contains exactly one arc of each of the cutsets K_1 and K_2, contains exactly one arc of the subtour {(5,4), (4,5)}, and contains exactly one arc of the set $K'_1 \cup K''_1$. Thus, H is an optimal solution to the TSP and B_3= 42 is its value.

If an inequality that is slack for $\hat{x}$ belongs to one of the sets (7a) or (7b), we attempt to strengthen the current lower bound by introducing some new inequalities (of the same type) that are tight for $\hat{x}$, and that admit positive multipliers. If the attempt is successful, it may also result in the removal of the inequality that is slack for $\hat{x}$.

3.1. Bounding Procedure 4. Suppose the inequality (7a) associated with the cutset K_t is slack for $\hat{x}$, i.e., the tour $\hat{H}$ defined by $\hat{x}$ intersects K_{fc} in more than one arc, and let $\hat{H}TK_t$ - £(1^j^ ,..., (i_p, j_p) }. For every (i_r, j_r) e $\hat{H}$OK_t , let S^r be a set of nodes containing j_r and such that, denoting $\bar{S}^r$ • N\S^r, the cutset K_{tr}- $(5^T, S^r)$ contains no other arc of $\hat{H}$ than (*-$_r$J$_r$)-

Then the inequalities

$$\underset{(i,j)\in K_{tr}}{Z} x_{ij} \geq 1, \quad r - 1,\dots,p$$

are all satisfied with equality by $\hat{x}$. Since every K_{tr} contains an arc with zero reduced cost, namely the arc (i_r, j_r) also contained in K_t, the above inequalities do not admit a positive premium, unless the premium X_t applied to K_t is reduced. If this is done, however, then a positive premium may be applicable to several of the sets K_{tr}, and the sum of these premia may well exceed the amount by which X_t must be reduced, i.e., an improvement of the lower bound may be obtained. The conditions under which this is possible are stated in the next two propositions.

Proposition 8. The tour $\hat{H}$ intersects the cutset K_{tr} • (§^{r},S^r) only in the arc $(i_r>i_v)>$ if a«* °»ly if the arcs of $\hat{H}$ with both ends in S^r form a path whose first node is j_r.

Proof. Let $\hat{H}$ = (i(l),...,i(n)}, and without loss of generality, assume d_r»J_r) * [i(l),i(2)}. Now suppose either S^r = (j(2)}, or the arcs of $\hat{H}$ with both ends in S^r form a path (i(2),... ,i(k) }. Then $\hat{H}$ intersects the cutset K^ = ($\bar{S}$^S*) in the single arc [i(l),i(2)] « (i_r,j_r).

Conversely, suppose S^r # [j(2)} and the arcs of $\hat{H}$ with both ends in S^r either form a path P whose first node is not j(2), or do not form a path. In the first case, $\hat{H}$fTK^ = [i(h) ,i(h+l)], where i(h+l) is the first aode of P. In the second, the arcs of $\hat{H}$ with both ends in S^r form k paths P_1,...,P_k, with $k \geq 2$; and £rK_{tr} --CKh^.Khj+l)]$_f$... ,[1(1^) ,10^+1)]$_f$ where l(h_r+l) is the first node of P_r, r =* 1,... ,k. \ \

Proposition 9. A positive premium can be applied to the cutset K_{tr} (provided that λ_t is decreased) if and only if

(20) ^ V$^{n\,A}$0 * ^{0}'

If R ^ 0 is the set of those r e {l,...,p} for which (20) holds, the maximum premium applicable to each K^., reR, is

(21) X^r » min [\., $\min\limits_{(i,j)\in}$ $\bar{c}$. .) > 0;

provided the premium X_t applied to K_t is replaced by

(22) . $\hat{X}_t$ « X_t - $\max\limits_{rcR}$ X^r.

This replaces the current lower bound B by

(23) B^7 » B + $\sum\limits_{rcR}$ X^r - $\max\limits_{reR}$ X^r.

Proof, A decrease in X_t increases the reduced costs of all arcs of K_t; hence makes it possible to apply a positive premium to the arcs of those, and only those, cutsets K_{tr} satisfying condition (20) The maximum size of the premium on $I\hat{}_r$ is X^r defined by (21), positive for those r for which (20) holds. The premia X^T are however applicable only if X_t is diminished by the amount of the largest premium applied to the arcs of any cutset K^ i.e., only if $X._t$ is replaced by X_t of (22) (otherwise some of the reduced costs become negative). If this is done, the current lower bound B is replaced by

$$B^7 - B + \sum_{reR} X^T + (\hat{X}_t - X_t)$$

which yields (23) after substituting for $\backslash_t$.||

<u>Bounding Procedure 4</u> looks for cutsets K_t to which a premium $X_t > 0$ had been applied and which are intersected by the tour $\hat{H}$ in more than one arc. For each arc (i_r,j_r) of $\hat{fl}$ that belongs to such a cutset K_t, we try to find a set S^r of nodes containing j_r and satisfying the conditions of Propositions 8 and 9; i.e., such that the arcs of $\hat{H}$ with both ends in S^r form a path whose first node is j_r, and that the cutset $K\hat{}_r \bullet (\bar{S}^r,S^r)$ satisfyies (20). As a matter of practicality,we first try $|s^r j * 2$, then $|S^r| » 3$ etc., until either we find a set which satisfies (20) or we find out that none exists. Taking the candidate sets in this order makes sense, since smaller sets S^r define smaller cutsets K^ over which one takes the minimum in (22) to define the premia $\backslash^r$.

If no sets S^r with the desired properties is found, we take another arc of $H^* D K_t$. Otherwise we compute X^r, the premium to be applied to the cutset $K\hat{}_r$, using (22); and then take the next arc of $\hat{H} D K_t$. When all arcs of HHK^have been examined, we compute the new value X_t of the premium applicable to the arcs of K_t, as given in (22), and replace λ_t by $\hat{\lambda}_t$. All this replaces the reduced costs $\bar{c}_{ij}$ by

$$\bar{c}'_{ij} = \begin{cases} \bar{c}_{ij} - \lambda^r & (i,j) \in K_{tr} \setminus K_t,\ r \in R \\ \bar{c}_{ij} - \lambda^r + \max\limits_{s \in R} \lambda^s & (i,j) \in K_{tr} \cap K_t,\ r \in R \\ \bar{c}_{ij} + \max\limits_{s \in R} \lambda^s & (i,j) \in K_t \setminus (\bigcup\limits_{s \in R} K_{ts}) \\ \bar{c}_{ij} & \text{all other } (i,j) \in A \end{cases}$$

and the lower bound B by B′ defined in (23). If $\hat{\lambda}_t > 0$, the inequality associated with K_t (which is slack for $\hat{x}$) continues to be represented in the Lagrangean form (1′). If, however, $\hat{\lambda}_t = 0$, i.e., $\max\limits_{r \in R} \lambda^r = \lambda_t$, then the inequality corresponding to K_t is removed from (1′) and we have succeeded in replacing this constraint, slack for the solution $\hat{x}$, with a set of inequalities that are all tight for $\hat{x}$.

If T_1^+ is the index set of those inequalities (7a) that are slack for $\hat{x}$ and for which $|R| \neq \emptyset$, and if we attach a subscript t to the index sets R associated with each cutset K_t and to the premia λ^r indexed by R, then at the end of Bounding Procedure 4 we have the lower bound

$$(24) \qquad B_4 = B_3 + \sum_{t \in T_1^+} \sum_{r \in R_t} \lambda_t^r - \sum_{t \in T_1^+} \max_{r \in R_t} \lambda_t^r .$$

3.2. Example 2. Consider the reduced cost matrix of Table 6 resulting from the solution of AP for an 8-city TSP. The value of the AP solution is v(AP) = 50. The admissible graph is given in Fig. 5.

$[\bar{c}_{ij}] =$

	1	2	3	4	5	6	7	8
1	X	$\boxed{0}$	5	7	3	7	3	6
2	3	X	0	8	4	6	9	5
3	9	5	X	$\boxed{0}$	5	7	3	4
4	$\boxed{0}$	7	6	X	6	5	6	9
5	6	4	5	6	X	$\boxed{0}$	2	5
6	7	0	8	9	$\boxed{0}$	X	6	4
7	8	9	7	6	6	8	X	$\boxed{0}$
8	5	4	5	0	8	2	$\boxed{0}$	X

Table 6

Applying bounding procedure 1 and taking cutset $K_1 = ([1,2,3,4]\ \{5,6,7,8\})$ we obtain $X_1 = 3$ and hence $B_1 = 53$. The reduced cost matrix $[\bar{c}_{ij}]$ is shown in Table 7 and the admissible graph G_0 in Fig. 6.

$[\bar{c}_{ij}] =$

	1	2	3	4	5	6	7	8
1	X	0	5	7	0	4	5	3
2	3	X	0	8	1	3	6	2
3	9	5	X	$\boxed{0}$	2	4	0	1
4	$\boxed{0}$	7	6	X	3	2	3	6
5	6	4	5	6	X	$\boxed{0}$	2	5
6	7	0	5	9	0	X	6	4
7	8	9	7	6	6	8	X	$\boxed{0}$
8	5	4	5	0	8	2	$\boxed{0}$	X

Table 7

Bounding procedure 1 is complete and bounding procedures 2 and 3 are unsuccessful in improving the bound beyond $B_1 = 53$.

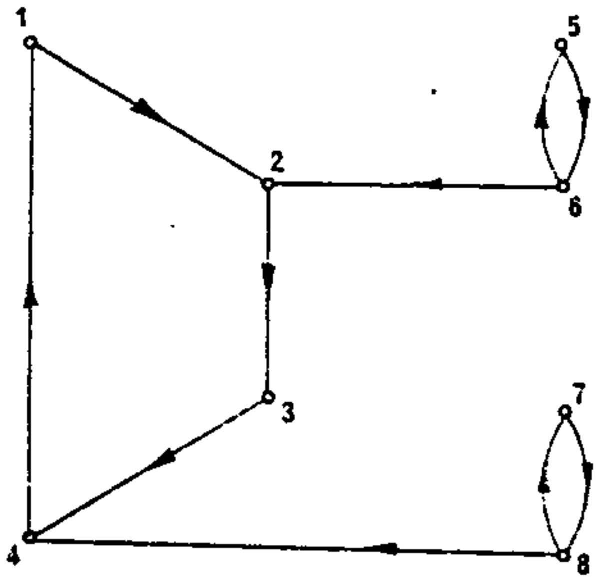

Fig. 5. Graph G_Q defined by the AP solution.

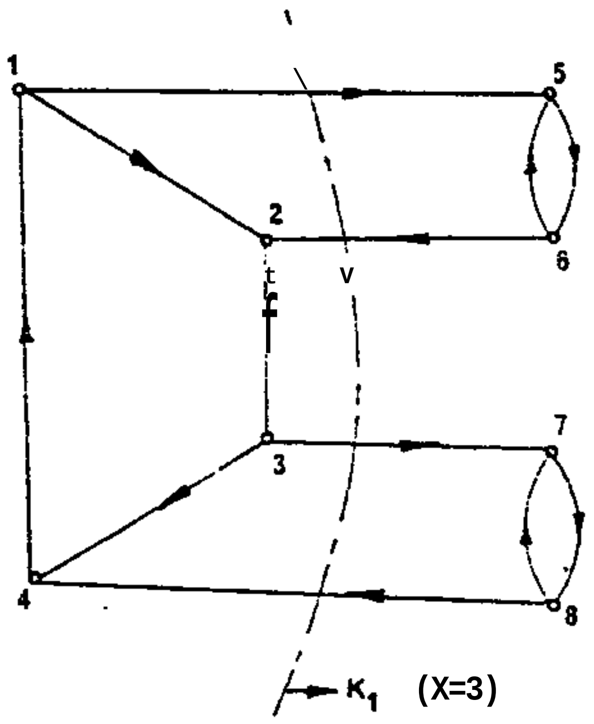

Fig. 6. G_Q after bounding procedures 1, 2 and 3.

A tour H = (1,5,6,2,3,7,8,4,1) is detected in G_0, but H contains two arcs of the cutset K_1, and is therefore not necessarily optimal.

Applying bounding procedure 4 to L_1 we identify the cutsets $K_{1,1}$ = ([1,2,3,4,5,6], {7,8}) with $\lambda_1^1 = 2$, and $K_{1,2}$ = ({1,2,3,4,7,8}, {5,6}) with $\lambda_1^2 = 2$, while reducing the multiplier associated with cutset K_1 from $\lambda_1 = 3$ to $\lambda_1 = 1$.

The new reduced cost matrix is given in Table 8 and the associated admissible graph is shown in Fig 7.

$$[\bar{c}_{ij}] = \begin{array}{c|cccccccc|} & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ \hline 1 & X & \boxed{0} & 5 & 7 & 0 & 4 & 5 & 3 \\ 2 & 3 & X & 0 & 8 & 1 & 3 & 6 & 2 \\ 3 & 9 & 5 & X & 0 & 2 & 4 & 0 & 1 \\ 4 & 0 & 7 & 6 & X & 3 & 2 & 3 & 6 \\ 5 & 6 & 4 & 5 & 6 & X & 0 & 0 & 3 \\ 6 & 7 & 0 & 8 & 9 & \boxed{0} & X & 4 & 2 \\ 7 & 3 & 9 & 7 & 6 & 4 & 6 & X & 0 \\ 8 & 5 & 4 & 5 & 0 & 6 & 0 & 0 & X \\ \hline \end{array}$$

Table 8

The lower bound is now improved from 53 to B_4 = 53 + 2 + 2 - 2 = 55, as given by (24).

Next we turn to the inequalities (7b).

3.3. Bounding Procedure 5. Suppose the inequality (7b) defined by the node set S_t is slack for $\hat{x}$, i.e., the tour $\hat{H}$ contains fewer than $|S_t|-1$ arcs of the set (S_t,S_t). Let G_t be the subgraph of G induced by the arc set $\hat{H} \cap (S_t,S_t)$, i.e., the graph consisting of those arcs of the tour $\hat{H}$ with both ends in S_t, and the end-nodes of these arcs. Note that $\hat{H} \cap (S_t,S_t)$ may be empty, since it is possible for a tour to contain all

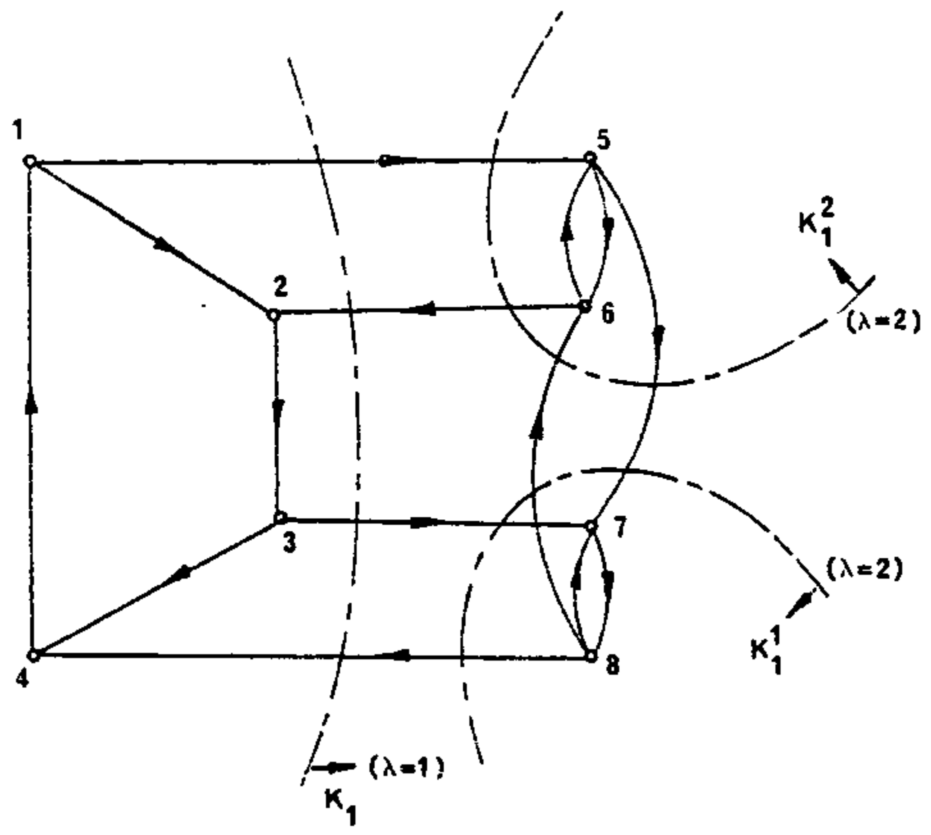

Fig. 7. $\underset{\smile}{G_n}$ after bouading procedure 4.

nodes in S_t without containing any arc with both ends in S_tJ and when this is the case, no new inequalities can be derived from *S^.*

Assume now that $\hat{H}$ *f) (S^S_t)* *£* 0, and let $C^1,\ldots,C^s$ be the (connected) components of G_t. For q$Q « 0»-»->s}, let S^q and A^q denote the node set and arc set, respectively, of C^q. By construction, each C^q is an open (directed) path, with $2 \leq |S^q| \leq |S_t| - 1$ and $|A^q|$ * Js^q| - 1; hence x satisfies with equality each of the inequalities

$$S_q \ S \ q x.j <\_|s^q! - i \quad . \quad qcQ$$
$$i e S^q \ j e S^q \ ^{1J}$$

or, to put them in the form (7b),

(25) $$- £ \quad £ \quad x . >\_1 - |S^q| \quad , \quad qcQ .$$
$$i e S^q \ j \ c \ S^q \ ^{1J}$$

Since $(S^q,S^q)(lA_o$ ^ 0 ,VqcQ , these inequalities do not admit a positive penalty (without a change in the dual variables $\bar{u},\bar{v}$) , unless the penalty M^*_t associated with S_t is reduced. If, however, this can be done, then each of the inequalities (25) admits a positive penalty and the current lower bound may be strengthened. The next proposition states the conditions for this.

Let F_t be the set of those arcs of 6 having both ends in S_t, but not both ends in the same set S^q , for any q e Q; i.e., let

$$F_c - (S_t,S_t) - \bigcup_{qcQ} (S^q,S^q) .$$

<u>Proposition 10</u>. A penalty μ^*_t > 0 can be applied to each of the arc sets (S^q,S^q), qeQ (provided that the penalty μ_t is decreased), if and only if

(26) $$F_t \cap A_Q * 0 .$$

If (26) holds, then

(27) $$\mu_t^* = \min_{(i,j)\in F_t} \bar{c}_{ij} > 0 \ ,$$

and the penalty $|\pm_t^*$ can be applied to each arc set (S^q,S^q) provided that the penalty $u>_t$ applied to (S_t,S_t) is replaced by l^*_t - $M»_t^*$. This replaces the current bound B by

(28) $$B' \leftarrow B + (|Q| - 1)*_t^* \ .$$

<u>Proof</u>. Since (S^q,S^q) C (S_t,S_t) and (S^q,S^q) n A_Q * 0 , $Vq_\epsilon Q$, a positive penalty $\wedge_t^*$ can be applied to any (and all) of the arc sets (S^q,S^q) if and only if the penalty n_{fc} applied to the arc set (S_t,S_t) can be reduced by the same amount $y,_t^*$. This, however, is possible if and only if no arc in the set F_{fc} has a zero reduced coat (i.e., condition (26) is satisfied) and n_t^* does not exceed the reduced cost of any arc in F_t . When these conditions are present, all arc sets (S^q,S^q), q«Q , can be penalized by the amount $|\pm_t^*$ specified in (27), provided the penalty $\wedge_t$ on the arc set (S_t,S_t) is replaced by ^ - $j\pm_t^*$. The effect of all this on the lower bound is to add y_t^* as many times as the number |Q| of components of G_t, and to subtract JJ_t^* once; i.e., to add to the current bound B the amount

$(|Q| - D*_t^* \bullet$ II

<u>Bounding Procedure 5</u> takes an inequality (7b) that is slack for £, forms the associated arc set F_{fc} defined above, and checks condition (26). If (26) is not satisfied, nothing can be done, and the procedure goes to the next ineqxiality that is slack for $\hat{x}$. if (26) holds, we calculate $»_t^*$ given by (27), and penalize by p_t^* all arc sets (S^q,S^q),qeQ, defined by the components of the graph G_t; while replacing the penalty $y,_t$ on the arcs of (S_t,S_t), by $M»_t$ - n_t^* . This replaces the reduced costs $\bar{c}_{ij}$ by

$$\bar{c}'_{ij} = \begin{cases} \bar{c}_{ij} - \mu^*_t , & (i,j) \in F_t \\ \bar{c}_{ij} , & (i,j) \in A \setminus F_t , \end{cases}$$

and the current lower bound B by B" defined by (28). If y^*_t < $^\wedge_t$, the inequality (7b) defined by the vertex set S_t (which is slack for $\hat{x}$) continues to be part of the Lagrangean expression (I^7); if, however, $M»^*_t$ = U^, i.e. the penalty associated with the inequality in question becomes zero, then we have succeeded in replacing this inequality in (I^7) by a set of other constraints that are all tight for $\hat{x}$.

Next the procedure goes to another inequality (7b) that is slack for x. When all such inequalities have been examined, let T_2 be the index set of of those among them for which condition (26) was satisfied, and for each t € T^+_2, let $|Q_t|$ be the number of components of the graph G_t. Bounding Procedure 5 then produces the lower bound

(29) B_5 - B_4 + E $_+$<|QJ$_t$-Ol**$_t$ •
teTj .

Finally, we turn to the inequalities (7c).

3.4. Bounding Procedure 6, Suppose the inequality (7c) associated with the articulation point k and the cutsets K^» (S^ S^ \ [k}), $K^{t/}$= ($?^t$\{k},S^t) is slack for x, i.e., the tour H defined by x contains more than one arc of the set K^UK^t, and let H fl (K^UK^) - l(i_rj^),..., ($i_{pf}$$J_p$)}. For every ($i_r$,$J_r$)e£n (K^UK^), r * l,..-,p, we will specify a node set S^rCN\{k} such that,denoting *S** » N\$S^r$ and K^ « (S^r,$?^r$\Ck}), K^$_r$ « (s'XCk}),S^r), the only arc of H contained in K^UK^ is (i_r,J_r).

<u>Proposition IX</u>. The only arc of $\hat{H}$ contained in $K'_{tr} \cup K''_{tr}$ is (i_r, j_r), if and only if $S^r = S\setminus\{k]$, where S is the node set of one of the two paths $P_X = \{k,\ldots,i_r\}$ and P_2 - £$j_r,\ldots,k\}$ in $\hat{H}$.

<u>Proof</u>. Assume S^r « $S\setminus\{k]$, where S is the node set of P_1 or P_2. In the first case, $\hat{H}$ O K^ - $\{(i_r,J_r)\}$ and $\hat{H}$ O K^ = 0; in the second, $\hat{H}$ d K^ = 0 and $\hat{H}$HK^$_r$ - $\{(i_r,J_r)\}$. In both cases, $(i_r$»$J_r)$ is the only arc of $\hat{H}$ contained in $K'_{tr} \cup K''_{tr}$.

Conversely, let (i_r, j_r) be the only arc of $\hat{H}$ contained in $K'_{tr} \cup K''_{tr}$. Then either [(i^j^} * $\hat{H}$ O K^ and $\hat{H}$PIK^. - 0, or $((i_r,J_r)\}$ - $\hat{H}$OK^, and
^ / * r
tr
HflK = 0. In the first case, H enters S from k rather than from some node of S^Mk}, since HflK^ = 0; and it exits S^r exactly once, through i_y;
1
hence $S^r = S\setminus(k\}$, where S is the node set of P-. In the second case, H exits S^r through an arc whose front end is k, rather than some node of $S^T\setminus\{k]$, since HflK^ = 0; and it enters S^r exactly once, through j^r; hence S^r - $S\setminus\{k\}$, where S is the node set of P_2.||

Thus, if the node sets S^r, r » l,...,p, satisfy the conditions of Proposition 11, then the inequalities

$$\sum_{(i,j)\in K'_{tr} \cup K''_{tr}} x_{ij} \geq 1 \quad , \quad r = 1,\ldots,p$$

are all satisfied by $\hat{x}$ with equality. The next proposition states the conditions under which a positive premium can be applied to the sets $K'_{tr} \cup K''_{tr}$.

<u>Proposition 12</u>. A positive premium can be applied to the arc set $K'_{tr} \cup K''_{tr}$ if and only if

(30) $$[(K'_{tr} \cup K''_{tr})\setminus K_t] \cap A_0 = \emptyset.$$

If R ^ 0 is the set of those re{l,...,p} ,for which (30) holds, the maximum premium applicable to each $K'_{tr} \cup K''_{tr}$, reR, is

(31) $$v^r = \min\{v_t, \min_{(U)eK} \bar{c}_{1J}\},$$

where K » <$^{K}t_r{}^{UK}t''_r$)NjCt; provided the premium v_t applied to K_t is replaced by

(32) $$\hat{v}_c = v_c - \max_{reR} v^r.$$

This replaces the current lower bound B by

(33) $$B' = B + \sum_{reR} v^r - \max_{reR} v^r.$$

Proof. Analogous to the proof of Proposition 9.||

Bounding Procedure 6 looks for indices t€£$_3$ for which a positive premium v_t has been applied to the arc set K^UK^ , and for which $\hat{H}$n(K^UK^) » {(i_1,j_1),...,(i_p,j_p)}, with $p \geq 2$. Given such a t«3> for each re{l,...,p} we use the node set of the path P_r = (k,...,i_r) in $\hat{H}$, after removing from it node k, to derive an arc set of the form K^$_r$UK^ defined in Proposition 11. We then check whether K^UK^ satisfies (30), and if so, we calculate the premium v^r to be applied to K'_{tr} UK$''_{tr}$; otherwise we move to the next re(l,...,p}. When all arcs of $\hat{H}$n(K^UK^) have been examined, we compute the value $\hat{v}_t$ of the premium applicable to the arcs of K_t, as given by (32), and replace v_t by G_t. All this replaces the reduced costs $\bar{c}_{ij}$ by

$$\bar{c}'_{ij} = \begin{cases} \bar{c}_{ij} - \nu^r & (i,j)\in(K'_{tr}\cup K''_{tr})\setminus K_t,\ r\in R \\ \bar{c}_{ij} - \nu^r + \max_{s\in R} \nu^s & (i,j)\in(K'_{tr}\cup K''_{tr})\cap K_t,\ r\in R \\ \bar{c}_{ij} + \max_{s\in R} \nu^s & (i,j)\in K_t\setminus \bigcup_{s\in R}(K'_{ts}\cup K''_{ts}) \\ \bar{c}_{ij} & \text{all other } (i,j)\in A \end{cases}$$

and the lower bound B by B' defined in (33).

As in the case of Procedures 4 and 5, if $\hat{\nu}_t = 0$, the inequality associated with $K'_t \cup K''_t$ is removed from the Lagrangean function (1′), otherwise it stays there with the new premium.

Let T_3^+ be the index set of those inequalities (7c) that are slack for $\hat{x}$ and for which $|R| \neq \emptyset$, and let us attach a subscript t to the index set R associated with $K'_t \cup K''_t$ and to the premia ν^r indexed by R. At the end of Bounding Procedure 6 we then have the lower bound

$$(34) \qquad B_6 = B_5 + \sum_{t\in T_3^+} \sum_{r\in R_t} \nu_t^r - \sum_{t\in T_3^+} \max_{r\in R_t} \nu^r.$$

Naturally, if at any stage of the bounding procedures described above the lower bound for the current subproblem matches the upper bound on v(TSP) given by the value of the best tour at hand, the current subproblem is fathomed.

At this point we may find ourselves in one of two possible situations: (α) we have found a tour in G_0, and used it to obtain the lower bound B_6 on the value of the current subproblem; or (β) the attempt to find a tour was unsuccessful, and B_3 is the best lower bound we have for the current subproblem. In case (β), we define $G_\epsilon = (N, A_\epsilon)$, with $A_\epsilon = \{(i,j)\in A \mid \bar{c}_{ij} \leq \epsilon\}$, where the $\bar{c}_{ij}$ are the current reduced costs and ϵ is the smallest number for which we are able to find a tour in G_ϵ within the given time limit. In either case, we denote by $\hat{H}$ the tour at hand, by i$\hat{c}$ the associated solution, by $\bar{c}\backslash_{ij}$, (i,j)cA, the last set of reduced costs, and by B the lower bound for the current subproblem. Obviously e$\hat{x}$, where c is the original cost vector, is an upper bound on v(TSP), and the best such upper bound at each stage will be denoted by B*.

3.5. Computational complexity of the bounding procedures. Each of the six bounding procedures discussed in sections 2 and 3 is polynomially bounded. For each of then except for the first one, the number of operations required in the worst case is $0(n^3)$, where n is the number of cities. For procedure 1, this number if $0(n^4)$. Solving the assignment problem at the start also requires at most $0(n^3)$ operations.

At every node of the search tree, the bounding procedures are applied once (after solving the assignment problem, if necessary) in the order 1,2,3. If at that point the node was still not fathomed (i.e., the lower bound is still below the current upper bound), an attempt is made at finding a tour in G_Q. Though there is no algorithm guaranteed to accomplish this in polynomial time, we let our implicit enumeration procedure run only for a fixed amount of time, that is an input parameter defined as a linear function of n. If a tour is found, bounding procedures 4,5,6 are applied in that order; otherwise we branch.

In conclusion, the amount of work performed at any given node of the search tree is $0(n^4)$ in the worst case.

4. Branching Rules

Before branching, we attempt to fix some variables by using the bounds B and B*. Let

$$Q_0 = \{(i,j)\in A \mid \bar{c}_{ij} \geq B^* - B\}.$$

It is not hard to show (see [1]) that, if the reduced costs $\bar{c}_{ij}$ are derived from the same dual solution and Lagrange multipliers as the lower bound B (as is the case here), then any solution x to TSP such that $cx < B^*$ must satisfy the condition $x_{ij} = 0, \forall(i,j)\in Q_0$. Hence we set $x_{ij} = 0, (i,j)\in Q_0$ for the current subproblem and its descendants, i.e., we replace A by $A\backslash Q_0$.

Next we describe two branching rules, which we use intermittently. The first rule derives a disjunction from a conditional bound [1]; the second rule derives one from a subtour-breaking inequality.

4.1. A <u>disjunction from a conditional bound</u> can be obtained as follows. Consider a family of sets $Q_k \subset A$, $k = 1,\dots,p$, such that $\bar{c}_{ij} > 0, \forall(i,j)\in Q_k$, $k = 1,\dots,p$. Then if the inequalities

$$\sum_{(i,j)\in Q_k} x_{ij} \geq 1 \quad , \quad k = 1,\dots,p$$

were added to the constraint set of LP, the lower bound B could be improved by choosing appropriate multipliers for these inequalities. Further, if this improved bound (termed conditional, because of the hypothetical nature of the inequalities) matches the upper bound B*, then every solution better than the one associated with B* violates at least one of the above inequalities; i.e., satisfies the disjunction

(35) $$\bigvee_{k=1}^{p} (x_{ij} = 0, \forall(i,j)\in Q_k).$$

To implement this principle, we first remove from the Lagrangean function (1′) all those inequalities (7a) and (7c) that are slack for $\hat{x}$ while the associated multiplier w_t is positive. If T^+ is the index set of these inequalities, this removal amounts to replacing B by

$$\hat{B} = B - \sum_{t\in T^+} w_t$$

and $\bar{c}_{ij}$ by

$$\hat{c}_{ij} = \bar{c}_{ij} + \sum_{t \in T} a^t_{ij} w_t$$

for $(i,j) \in A$.

Next, we choose a minimum-cardinality arc set $S \subset \hat{H}$ such that

$$(36) \qquad \sum_{(i,j) \in S} \hat{c}_{ij} \geq B^* - \hat{B}.$$

The existence of such $S \subset \hat{H}$ would be guaranteed if we removed from (1') all inequalities (7) that are slack for $\hat{x}$ (see [1] for a proof). However, removing the inequalities (7b) would either produce negative reduced costs, or would require a recalculation of the u_i and v_j. To avoid this recalculation, we restrict ourselves to the removal of inequalities (7a) and (7c), taking the risk of not being able to find a set $S \subset \hat{H}$ satisfying (36). Whenever this happens, we apply the second branching rule, to be discussed below.

Given that (36) holds, let $S = \{(i_1,j_1),\ldots,(i_p,j_p)\}$. We then construct a $p \times |A|$ 0-1 matrix $D = (d^k_{ij})$ by setting $d^k_{ij} = 1$ in each column (i,j) for as many indices $k \in \{1,\ldots,p\}$ as possible, subject to the conditions

$$(37) \qquad d^k_{i_k j_k} = 1, \quad k = 1,\ldots,p$$

and

$$(38) \qquad \sum_{k=1}^{p} d^k_{ij} \hat{c}_{i_k j_k} \leq \hat{c}_{ij}, \qquad (i,j) \in A.$$

These constraints leave some freedom for choosing the entries d^k_{ij} of each column $(i,j) \in A$, which we use to make the number of 1's in each row as close to equal as possible.

Proposition 13. Every solution x to TSP such that $cx < B^*$ satisfies the disjunction

(39) $$\bigvee_{k=1}^{p} (x_{ij} = 0,\ \forall (i,j)\in Q_k),$$

where

(40) $$Q_k = \{(i,j)\in A \mid d_{ij}^{k} = 1\}\ ,\quad k=1,\dots,p.$$

Outline of proof (see [1] for details). If $\tilde{x}$ violates (39), it satisfies

(41) $$\sum_{(i,j)\in Q_k} x_{ij} \geq 1\ ,\quad k=1,\dots,p.$$

Adding (41) to the constraint set of LP and assigning the multiplier (dual variable) $\hat{c}_{i_k j_k}$ to the kth inequality (41), yields the lower bound

$$\hat{B} + \sum_{k=1}^{p} \hat{c}_{i_k j_k} = \hat{B} + \sum\sum_{(i,j)\in S} \hat{c}_{ij}$$

$$\geq B^*$$

where the last inequality follows from (36). Hence $c\tilde{x} \geq B^*$. ||

The disjunction (39) creates p subproblems. In the kth subproblem we have $x_{ij} = 0$, $(i,j)\in Q_k$, and since $(i_k, j_k)\in \hat{H}\cap Q_k$, the tour $\hat{H}$ becomes infeasible for each of the subproblems. On the other hand, the current solution to AP remains feasible for each of the subproblems.

4.2. A disjunction from a subtour breaking inequality is obtained in the usual way; i.e., if S is the arc set of a subtour of the AP solution, then every solution to TSP satisfies the disjunction:

(42) $$\bigvee_{(i_k,j_k)\in S} (x_{i_k j_k} = 0 \text{ and } x_{i_t j_t} = 1,\ \forall t \leq k-1)$$

At an arbitrary node of the branch and bound tree, a subset $S^f c5$ of the arc set S (of the subtour selected for branching) may already have been fixed to be in the solution. In this case set S in disjunction (42) is replaced by S\S'. Branching on (42) creates $|s\backslash S^f|$ subproblems- For each of these subproblems, the AP solution to the parent problem becomes infeasible.

In choosing the arc set S for the disjunction (42), it is desirable to give preference to subtours (of the current AP solution) having either a minimum number of arcs (min |S|),or a minimum number of free arcs (min $|s\backslash S^v|$). In the computational tests discussed in the next section we used the first of these two criteria.

As to the two disjunctions (39) and (42), an efficient procedure must use them intermittently, since (39) can on occasion be considerably stronger than (42), while at other times it can be much weaker. We tried several rules for mixing them, and the one actually used in the tests is discussed in the next section.

S. Implementation and Computational Experience.

Our algorithm was programmed in FORTRAN IV for the CDC 7600 and tested on a set of 120 randomly generated asymmetric TSP's of sizes varying between 50 and 325 cities. Here we discuss some features of the implementation, give the computational results, and interpret them.

5.1. Use of sparsity. Unlike in the case of those symmetric TSP's whose costs are baaed on distances and can therefore be generated whenever needed from the 2n coordinates of the cities, in the case of the asymmetric TSP one has to explicitly store the costs, whose number in case of a complete graph is n(n-l). However, our procedure derives both lower and upper bounds on the value of the problem and, as discussed at the beginning of section 4, provides a valid criterion for setting to 0 certain variables. As it will be discussed below, the number of variables that can be fixed at 0 before the first branching is usually very high. Therefore at that point we actually remove from the graph all those arcs whose variables can be fixed at 0, and from then on we work with a graph (usually quite sparse) represented by a list of nodes and a list of arcs with their costs. Additional fixing of variables (at 0 or 1) later in the procedure is handled differently (see below).

5.2, <u>Solution of the AP's</u>. At every node of the search tree, a subset of variables is fixed at 0, another subset is fixed at 1, and the current problem is the one in the free variables. A variable x_{ij} is set to 1 by adding a large number $M > 0$ to all c_{ik}, $k = 1,\ldots, n$, $k \neq j$. A variable x_{ij} is set to 0 by adding M to c_{ij}. The reason for not simply removing the arc from the graph, as done before the first branching, is that (a) the variable and its cost may be needed later on another branch; (b) the transition from the old AP solution to the one for the new subproblem is easier this way.

All AP's are solved by the Hungarian algorithm modified as follows:
(i) At every subproblem, we start with a solution derived from the solution of the predecessor problem. In particular, the growing of an alternating tree (in search of an augmenting path during the application of the Hungarian algorithm) starts with a matching (i.e., a set of independent zeroes in the reduced cost matrix) derived from the solution to the predecessor of the current subproblem. A single augmenting path is almost always sufficient to solve the current AP.

(ii) Since the Hungarian algorithm is a dual procedure, it can be terminated prematurely whenever the value of the objective function exceeds the value of the current upper bound.

5.3. Branching and node selection. The two types of branching discussed in section 4 are used intermittently according to the following rule. A branching of type 1 (based on disjunction (39)) is performed whenever a set of arcs SC$\hat{H}$, S « $\{(i_1 Jj),\ldots,(i_p,j_p)\}$, can be found, such that

(i) inequality (36) is satisfied;

(ii) |s| $\leq$ ^ + 1, where p is the condinality of the smallest subtour in the current AP solution; and

(iii) at least n/3 variables can be fixed at 0 on each branch.

Whenever any of the above conditions is violated, a branching of type 2 (using disjunction (42)) is performed.

The node selection rule used in the code is to ehoose a successor of the current node whenever available, and otherwise to select a node k for which the following evaluation attains its minimum:

$$\text{Boo-new} - v(AP)] \cdot \frac{s(k)-1}{s(0)-s(k)}.$$

Here B(k) is the lower bound for subproblem k, v(AP) is the value of the (initial) AP, while s(0) and s(k) are the number of subtours in the solutions to the initial AP and the current one (at node k), respectively. The integer s(k) is used as a measure of the "distance" of the AP solution at node k from an optimal tour.

5.4. Information stored for each subproblem. All subproblems are stored on a linked list in order of increasing lower bounds. For each subproblem k the following information is stored:

- The AP solution.
- The value of the associated bound.
- A pointer to the father node of node k.
- A code to indicate the type of branching (one of the 2 types described above) that produced node k.
- The number of sons of the father of node k.
- The rank (index) of node k among its brothers.
- If the type of branching that produced node k was based on disjunction (39), we store the arcs in $S = \{(i_1,j_1),\ldots,(i_p,j_p)\}$. If it was based on disjunction (42), then a pointer gives the subtour in the AP solution corresponding to S in (42).
- A list of the operations (in coded and ordered form) which produced the current matrix $[c_{ij}]$ from the matrix for the predecessor node. (This is not strictly necessary but speeds up considerably the backtracking process).

5.5. Computational results. The above described code was run on the CDC 7600 to solve 120 randomly generated test problems whose associated (directed) graphs are complete and whose cost coefficients were drawn from a uniform distribution of the integers in the range [1, 1000]. The problems belong to 12 classes based on size, with $n = 50, 75, \ldots, 300, 325$, and with 10 problems in each class. Table 9 summarizes the results. These results are quite remarkable, in that the number of nodes generated is surprisingly small, and seems to increase only slightly faster than the problem size (number of cities). This is also illustrated on Fig. 8, where the slope of the curve is only slightly steeper for $200 \leq n \leq 325$ than for $50 \leq n \leq 200$. Note, also, that the maximum time required to solve any one of the 120 problems was 82 seconds.

Table 9. Computational results on random asymmetric TSP's.

Class	n	Average no. of nodes	Computing time (CDC 7600 seconds)		Percentage of nodes fathomed by						Percentage of time spent on						
			Average	Maximum	B_0	B_1	B_2	B_3	H	$B_{4,5,6}$	B_0	B_1	B_2	B_3	H	$B_{4,5,6}$	Other
1	50	12.3	.20	.88	25.2	29.1	6.9	10.4	21.4	7.0	5.3	15.5	26.0	15.3	1.2	20.1	16.6
2	75	26.6	.29	.93	26.7	26.6	10.3	10.1	19.8	6.5	5.3	15.4	26.9	12.6	0.8	20.7	18.3
3	100	39.1	.71	1.41	21.8	30.7	9.5	14.9	16.5	6.6	4.9	15.1	27.8	14.1	0.8	23.1	15.2
4	125	42.7	1.13	2.07	19.6	34.4	12.0	13.8	16.5	3.7	4.8	16.0	23.9	17.3	1.1	26.1	10.8
5	150	45.7	1.97	3.30	19.6	28.5	15.1	12.9	13.3	10.6	3.9	16.3	24.8	14.9	1.3	26.6	12.2
6	175	58.3	4.18	6.68	20.1	28.7	17.4	13.7	15.4	4.7	4.6	16.9	28.4	14.8	0.9	19.3	15.1
7	200	63.4	6.06	19.33	14.7	33.9	14.9	17.2	12.1	7.6	4.1	16.5	29.7	16.7	1.6	19.1	12.3
8	225	84.1	10.44	18.65	11.1	29.6	22.6	16.8	8.8	11.1	3.8	16.3	29.1	16.4	1.8	20.1	12.5
9	250	88.5	13.65	17.43	12.5	29.7	17.2	21.9	9.2	9.5	3.3	17.6	27.7	17.3	1.6	20.2	12.3
10	275	106.4	21.74	68.86	9.3	25.4	20.5	19.3	8.5	7.0	2.9	18.5	27.9	16.5	1.9	18.9	13.4
11	300	124.1	38.37	55.15	10.7	28.9	19.1	23.8	6.9	10.6	3.1	18.9	29.1	14.1	2.1	20.0	12.7
12	325	141.8	49.66	81.57	7.7	32.3	18.7	21.6	7.8	11.9	2.6	20.1	30.3	16.0	2.1	17.1	11.8

Notes. n: number of cities

B_0: lower bound obtained by solving the current AP and adding to v(AP) a penalty derived from $[\bar{c}_{ij}]$

B_1, B_2, B_3: lower bound obtained by procedures 1,2 and 3, respectively

H: upper bound obtained by finding a tour in G_0 satisfying the condition of Proposition 2

$B_{4,5,6}$: lower bound obtained by procedures 4 to 6

Other: branching, node selection, updating, etc.

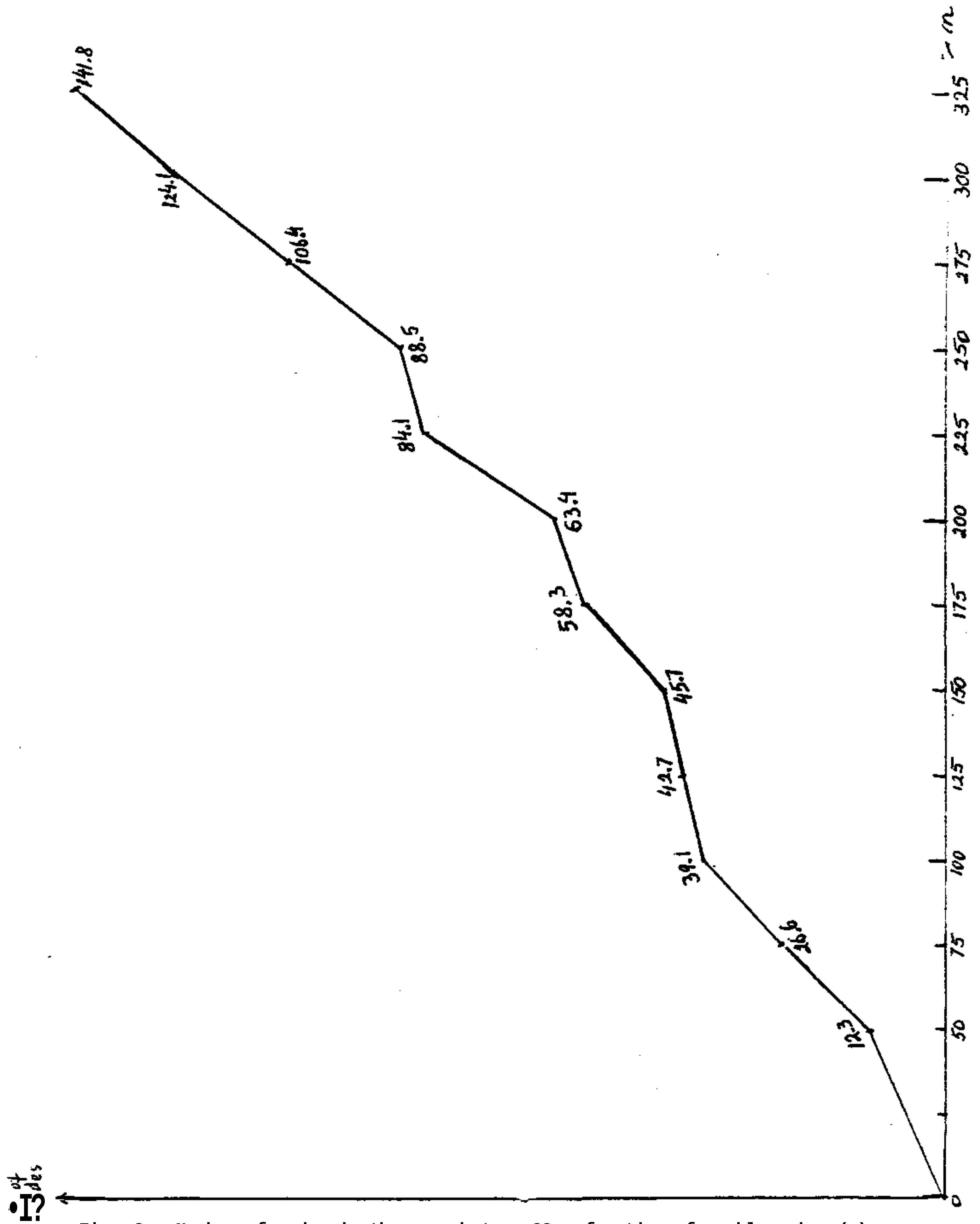

Fig. 8. Number of nodes in the search tree 33 a function of problem size (a).

Since the average cost of the various bounding procedures is not proportional to their usefulness, we have tested each of the bounding procedures individually and in subsets to see whether their use pays off. The outcome of our tests was that using all 6 bounding procedures is more efficient than using any subset in any combination.

Another remarkable feature of the approach discussed here is the large number of arcs that can be removed from the graph (of variables that can be permanently fixed at 0) at the root node of the search tree, as a result of the test discussed at the beginning of section 4. This is shown in Table 10. The fact that such a high proportion of the arcs can be removed before branching shows the power of the bounding procedures used in our approach. For a comparison, if only the bound obtained from AP were used, then the percentage of variables removed in problem classes 1, 2 and 3 would be on the average 87% (and this percentage does not seem to increase with problem size). Thus, for problem class 3, for example, our bounding procedures reduce the number of arcs remaining in the graph from the 13% that would be left by the AP bound, to 2.9%.

Table 10. Percent of arcs removed on the average at the root node.

Problem class	1	2	3	4	5	6	7	8	9	10	11	12
$\frac{\text{Arcs removed (average)}}{\text{Total arcs}} \times 100$	95.3	96.4	97.1	97.3	97.5	97.6	97.9	98.1	98.4	98.3	98.6	98.7

In connection with the two branching rules, it is important to mix them judiciously. While rule 1 (disjunction (39)) often allows one to fix more variables than rule 2 (disjunction (42)), if used as the only branching rule it yields inferior results, since occasionally it is very bad. The mixing strategy used in the above runs (and discussed under 5.3) has resulted in rule 1 being used only at the upper levels of the search tree (often at level 1, or 1 and 2, only). To compare the results obtained by using this strategy with those obtainable by using rule 2 only, we ran 4 of the 12 problem sets (i.e., 40 of the 120 problems) with branching rule 2 only. Table 11 compares the results.

Table 11. Comparison of branching rules

Class	a	Average no. of nodes		Computing time (CDC 7600 sec)	
		Rule 2	Mix of rules 1 and 2	Rule 2	Mix of rules 1 and 2
1	50	10.3	12.3	.29	.20
3	100	31.9	39.1	2.10	.71
5	150	36.8	45.7	4.60	1.97
7	200	49.9	63.4	11.68	6.06

Note that although branching rule 2 tends to produce a smaller number of nodes than the mixed strategy described in section 5.3, it also tends to require about twice as much time than the latter. This is because the disjunction (39) (rule 1) creates nodes for which the AP solution at the father node remains feasible, and for which a large number of variables can be fixed at 0 — two features that make such nodes easy to fathom.

6. The Symmetric Case

Our algorithm can of course be applied to symmetric TSP's as it is, but it would not be efficient for such problems in its present form. This is so because of the well known fact that AP's associated with symmetric TSP's tend to have optimal solutions involving a large number of subtours of length two. However, our approach can easily be adapted to the symmetric case by replacing the assignment problem with the 2-matching problem as the basic relaxation of the TSP. We are in che process of developing such an algorithm for the symmetric TSP.

References

[1] E. Balas: "Set Covering with Cutting Planes from Conditional Bounds." MSRR #399, Carnegie-Mellon University, July 1976.

[2] E. Balas and N. Christofides: "A New Penalty Method for the Traveling Salesman Problem." Paper presented at the Ninth International Symposium on Mathematical Programming, Budapest, August 1976.

[3] N. Christofides: "Bounds for the Traveling Salesman Problem." Operations Research, 20, 1972, p. 1044-1055.

[41 N. Christofides: Graoh Theory: An Algorithmic Approach. Academic Press, 1976.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER Technical Report No. 439	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Restricted Lagrangean Approach to the Traveling Salesman Problem	5. TYPE OF REPORT & PERIOD COVERED Technical Report July 1979
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Egon Balas and Nicos Christofides	8. CONTRACT OR GRANT NUMBER(s) N0014-75-C-0621 NR 047-048 MCS76-12026 A02
9. PERFORMING ORGANIZATION NAME AND ADDRESS Graduate School of Industrial Administration Carnegie-Mellon University Pittsburgh, PA 15213	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Personnel and Training Research Programs Office of Naval Research (Code 434) Arlington, VA 22217	12. REPORT DATE 13. NUMBER OF PAGES 41
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) Unclassified
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

traveling salesman problem, Hamiltonian circuits, directed graphs, Lagrangean, penalty methods, branch and bound

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

We describe an algorithm for the asymmetric traveling salesman problem (TSP) using a new, restricted Lagrangean relaxation based on the assignment problem (AP). The Lagrange multipliers are constrained so as to guarantee the continued optimality of the initial AP solution, thus eliminating the need for repeatedly solving AP in the process of computing multipliers. We give several polynomially bounded procedures for generating valid inequalities and taking them into the Lagrangean function with a positive multiplier without

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified (over)

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

violating the constraints, so as to strengthen the current lower bound. Upper bounds are generated by a fast heuristic whenever possible. When the bound-strengthening techniques are exhausted without matching the upper with the lower bound, we branch by using two different rules, according to the situation: the usual subtour breaking disjunction, and a new disjunction based on conditional bounds. We discuss computational experience on 120 randomly generated asymmetric TSP's with up to 325 cities, the maximum time used for any single problem being 82 seconds. Though the algorithm discussed here is for the asymmetric TSP, the approach can be extended to the symmetric TSP by using the 2-matching problem instead of AP.