KM 27(1988)
pag 47 - 66

A BRANCH AND BOUND METHOD FOR THE TRAVELING SALESMAN PROBLEM ON A ROAD NETWORK

J.N.M. van Loon *

ABSTRACT

A Branch and Bound algorithm for the exact solution of the Traveling Salesman problem is presented. The problem is defined directly on the (incomplete) road network as an Integer Linear Program. Solutions can be determined with the aid of "Transportation-like" tableaux,so without knowing the underlying simplex tableau explicitly. Extra tour constraints can be generated and treated in these tableaux because we split them up in such a way that unimodularity of the coefficient matrix is preserved. This gives rise to the Branch and Bound process. The algorithm is tested on a 13-node network.

* Vakgroep Wisk. en Op. Res., KMA Postbus 90.154 4800 RG Breda; Tel.076 -273163

1. INTRODUCTION

In the classical form of the Traveling Salesman Problem (TSP) each node of a complete network must be passed through exactly once.The shortest closed tour that meets this demand is looked for. However in many applications a lot of work is saved by working directly on the incomplete network. Therefore we define the TSP on a road network as follows (see[1]):

RTSP:Find the shortest closed tour in a directed network that visits all nodes at least once.

Such a tour may contain cycles and pass through a node more than once.

Example:Consider the network in fig.1.
The one and only tour passes
two times through the nodes 2 and 3.

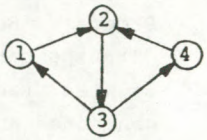


fig.1

In section 2 the RTSP is formulated as an Integer Linear Program (ILP) and extra tour constraints are introduced. A Transportation Tableau format(TT format) is dealt with in section 3 and illustrated with an example. Next a Branch and Bound(BandB) procedure is set up in section 4, followed by some computational arrangements in the tableau in section 5. The example is completed in section 6 and some considerations are given in section 7. Finally the results for a 13-node network are mentioned.

2. PROBLEM FORMULATION

Let $G=(N,A,c)$ be a directed network with node set $N=(j|j=1,..,n)$, arc set $A=\{(i,j)\}$ and distances $c[i,j]$. The RTSP can be formulated as the following ILP:

$$\text{Min} \sum_{(i,j)\in A} c[i,j]x[i,j]$$

(2.1) $$\text{s.t.} \sum_{(i,j)\in A} x[i,j] = \sum_{(j,k)\in A} x[j,k] \quad \text{for all } j \in N$$

(2.2) $$\sum_{(j,k)\in A} x[j,k] \geq 1 \quad \text{for all } j \in N$$

(2.3) $x[i,j] \geq 0$ and integer for all $(i,j) \in A$

(2.4) solution $\{x[i,j]\}$ builds a tour,

where $x[i,j]$=the number of times the tour passes through arc (i,j).

Theorem 1: The coefficient matrix of the ILP (2.1-2.3) is unimodular.

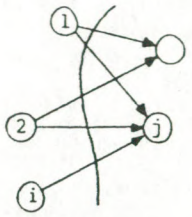
Proof: It is well-known that problems with network structure have unimodular coefficient matrices (see[2]). Without constraint set (2.2) the ILP has network structure: in each column of the matrix the non-zero elements +1 and -1 appear. Addition of an inequality from (2.2) does not change this character, because the only 1-elements in such a row form exactly the +1 part of the corresponding row in set (2.1).

By theor.1 it is clear that the basic solutions of the ILP, e.g. found by the (dual) simplex method, are integer. Moreover we can determine and present these solutions in TT format. If the solution, by chance, satisfies the tour condition (2.4) then the RTSP is solved. In most cases however the tour is not closed because of unadmittable subtours. To counter this we add one extra tour condition, based on the current solution in the following way. Letting N1 and N2 be the node sets corresponding to an arbitrary couple of subtours the following inequality must hold (see fig.2):

(2.5) $\sum x[i,j] >= 1$

iEN1; jEN2 N1^N2=Ø

all(i,j)EA N1UN2=N



cut fig.2

In order to preserve unimodularity of the matrix this inequality should be split up in "one-node" cuts and treated one after another (for ease of notation let the concerning nodes of N1 be 1,2,....):

(2.6) $\sum_j x[1,j] >= 1$ or $\sum_j x[2,j] >= 1$ or .....

This can be organised in a BandB process. (The solution sets are not disjoint! See section 7.2).

Theorem 2 :The extended constraint set is unimodular.

Proof: Since each "one-node" cut is of the form (2.2) the same arguments from the proof of theor.1 can be used again.

The (dual) simplex manipulations for the extended problem can be carried out again in TT format. If the new optimal solution still contains subtours another extra inequality of the shape (2.5) is added,otherwise the tour solution is kept in mind and a new branch is searched and so forth. This procedure is treated in section 4. Finally we define the slack variables for the ILP as follows:

(2.7) $py[j] = \sum x[j,k] - 1 >= 0$ corr. with node or row j of (2.2)

$pz[j] = \sum x[j,k] - \sum x[i,j] = 0$ corr. with node or row j of (2.1)

When py[j]>0 the tour passes through node j more than once.

3. SIMPLEX METHOD AND TT FORMAT

The ILP without tour condition (2.4) can be tackled with the dual simplex method . The dual program is:

$$\text{Max} \sum_{j \in N} y[j]$$

s.t.

(3.1) $y[i]+z[i]-z[j]=c[i,j]$ for columns corr. to basic variables

(3.2) $y[i]+z[i]-z[j]<=c[i,j]$ for all other columns

(3.3) $y[j]>=0$ and $z[j]$ arbitrary,

where $y[j]$ and $z[j]$ are the dual variables resp. for the constraint sets (2.2) and (2.1).

The set (2.1) is linearly dependent,so one of the variables $z[j]$ can be chosen zero. The use of the TT format can best be shown in an example:

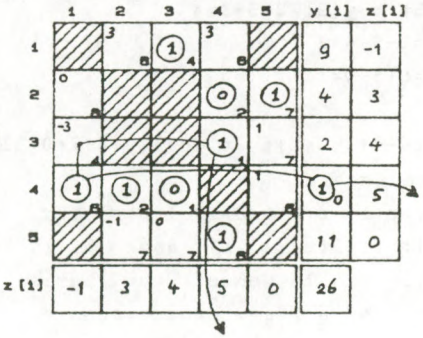
Example:

Consider the directed network in fig.3. A tour solution is indicated by black arrow heads:

$x[1,3]=x[3,4]=x[4,2]=x[2,5]=x[5,4]=x[4,1]=1$ and $py[4]=1$.

Since we need $2n-1=9$ basic variables in a simplex solution, let us try also the variables $x[2,4]$ and $x[4,3]=0$ as such. This solution is arranged in a transportation tableau, see table 1.

Table 1



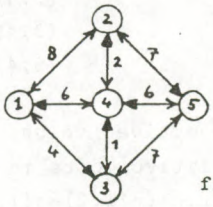
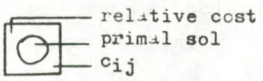


fig. 3



The tableau is extended with extra columns for the dual

variables and a bottom row only for ease of calculations. When a primal slack variable py[i] is basic, its value is given in the y[i]-column and is encircled. Note that the corresponding dual y[i] is zero in that case, because of the complementarity condition. Let us start with z[5]=0. Next the dual solution can be determined uniquely, unless ties are present in the primal solution, applying the equations (3.1) over basic cells. Begin with column 5 where z[5]=0. Row 2 has assignments in columns 5 and 4 so that

$$
\begin{aligned}
& y[2]+z[2]-z[4]=c[2,4] \\
\text{(appl.1)} \qquad & y[2]+z[2]-z[5]=c[2,5] \\
& \text{---------- minus} \\
& z[5]-z[4]=c[2,4]-c[2,5] \\
\text{hence} \quad & z[4]=c[2,5]-c[2,4]=7-2=5
\end{aligned}
$$

The second characteristic step in determining the dual solution makes use of a primal basic cell:e.g. in cell (4.3) holds:

$$
\begin{aligned}
& y[4]+z[4]-z[3]=c[4,3]=1 \text{ and since } z[4] \text{ is known:} \\
\text{(appl.2)} \qquad & z[3]=y[4]+z[4]-c[4,3]=0+5-1=4
\end{aligned}
$$

Next appl.1 can be used in row 4 yielding

$$z[1]=4-(6-1)=-1 \text{ and } z[2]=4-(2-1)=3$$

Finally all the remaining y[i] values can be found:

$$
\begin{aligned}
\text{via cell } (1,3) &: y[1]+z[1]-z[3]=c[1,3] \text{ gives } y[1]=9 \\
(2,4) \text{ or } (2,5) &: y[2]=2+5-3=4 \\
(3,4) &: y[3]=1+5-4=2 \\
(5,4) &: y[5]=6+5-0=11
\end{aligned}
$$

Objective value :primal:4+7+1+6+2+6=26,dual:9+4+2+0+11=26
The relative costs in the NB-cells are:
cell (1,2):r[1,2]=c[1,2]-(y[1]+z[1]-z[2])=8-(9-1-3)=3 etc.
This solution is not optimal: cells (3,1) and (5,2) have negative relative costs. Let x[3,1] enter the basis. We construct a new basic solution in a way, analogous to the transportation method. Determine a closed path,starting from

cell (3,1) with alternating + and - signs,as indicated in table 1. Note that we need sometimes elements from the py[i]-column and that the sign does not change in that case! Choosing arbitrarily x[4,1] to leave the basis we get the new solution as indicated in table 2,with object value 26-3*1=23. Unfortunately this is not a tour solution!(see fig.4).

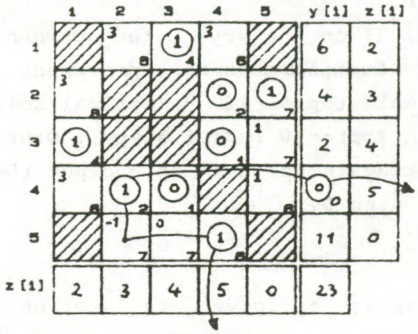


Table 2

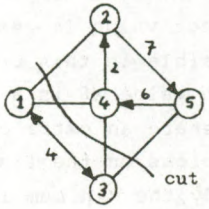


fig. 4

4. BRANCH AND BOUND PROCEDURE

Table 2 of the example is not optimal and the solution contains two subtours. We should decide how to proceed. Improving the solution probably will give no tour solutions,but we are not sure. Therefore we must start the BandB part up from the optimal solution of the ILP. In this section the global procedure is outlined. Suppose the problem has a tour solution.

Global solution procedure:

Stepl. Determine,by hand or otherwise,a good primal feasible solution as a start for the simplex method.

2. Build this solution up to a basic solution.
3. Improve this solution by the simplex method,up to the optimal solution.
4. If the optimal solution,object value x0,is a tour then set xopt=x0 and go to step 6,if not add an extra constraint of the

shape (2.5) and split it up according to (2.6) giving k problems on the BandB list.
Choose an upper bound on the object value,say xopt,perhaps found in applying the foregoing steps as some tour solution.

5. Take one problem from the list. If the list is empty go to 6.
Express the extra constraint in terms of the NB variables of the current tableau. The solution of the extended problem is not primal but dual feasible.
Use the dual simplex method. If on the way to the optimum the object value is >=xopt then stop:there is no improvement possible in this branch,repeat step 5. If the optimal solution, with value x0 is a tour then xopt:=x0,repeat step 5. Otherwise generate an extra constraint again,split it up and put the new problems on the list.Repeat step 5.
6. Ready,the minimum is xopt.

Details about this procedure and its use are given in section 6. First we illustrate the procedure with the example from section 3.

Example:

Consider table 2. The relative cost of cell (5,2) is negative. The closed path is indicated in table 2. We assign a zero and py[4] leaves the basis. The new solution is given in table 3. The step is a degenerate one. The tableau is now optimal,the solution is the same as indicated in fig.4. There is an alternative as appears from r[4,5]=0. It leads to the same solution but in reversed order for the nodes 2,4 and 5. The object value is x0=23. None of the solutions forms a tour.

	1	2	3	4	5	y[i]	z[i]
1		3	(1)	2		7	2
2	3			(0)	(1)	3	4
3	(1)			(0)	1	1	5
4	2	(1)	(0)		0	1	5
5		(0)	1	(1)		11	0
z[i]	2	4	5	5	0	23	

Table 3

Up till now we have gone through the steps 1 to 4 of the global procedure. In step 4 we generate the extra constraint (see fig.4)

$$x[1,2]+x[1,4]+x[3,4]+x[3,5]>=1$$

and split it up in the "one-node" constraints

$$x[1,2]+x[1,4]>=1 \text{ or } x[3,4]+x[3,5]>=1$$

Let us start with the second one,because c[3,4] is minimal among the c[i,j] involved. But to be able to use this extra constraint in the current tableau we should first consider some computational and notational aspects of the TT format.

5. SOME COMPUTATIONAL ARRANGEMENTS IN THE TABLEAU

5.1 If the extra constraint contains only NB variables we can choose the one with the minimum relative cost,find the closed path and force it to unity. E.g. the constraint x[1,2]+x[1,4]>=1 in table 3 yields x[1,4]=1. The dual simplex method can be used in case of an infeasible solution. No variable is leaving the basis,in fact the slack of the extra constraint does!In our example the closed path would be (1,4)-(5,4)-(5,2)-(4,2)-(4,3)-(1,3)-(1,4). The solution would remain feasible.See section 6 for an application.

5.2 If the extra constraint contains at least one basic variable we must express it in terms of the current solution. This can easily

be executed in the tableau,in the same way as we treat the objecive equation in order to find the relative costs in the transportation tableau. Let us illustrate this with the constraint x[3,4]+x[3,5]>=1 in our example. Because it corresponds to row 3 we define a slack ps[3]:

x[3,4]+x[3,5]-ps[3]=1 or ps[3]=-1+x[3,4]+x[3,5]>=0

where x[3,4] is basic but zero and x[3,5] is NB in the current solution. Suppose now x[3,4]+x[3,5] is the object function,then we would proceed as follows. Create extra columns and a bottom row,see table 4.

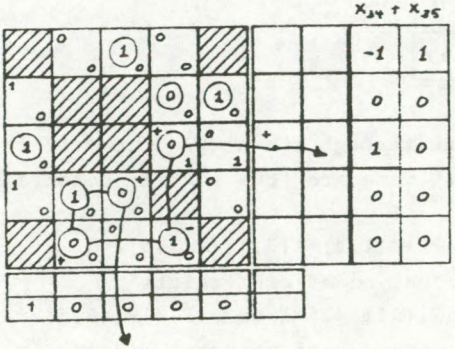


Table 4

The only non zero coefficients are indicated in cell (3,4) and (3,5). Calculate the dual values and the relative costs for the NB cells, starting with z[5]=0 as before. Table 5 gives the resulting simplex row coefficients and the corresponding relative costs.

NB variables	x12	x14	py1	x21	py2	x35	py3	x41	x45	py4	x53	py5	ps3
extra row	0	0	-1	1	0	0	1	1	0	0	0	0	-1
rel. costs				3			1	2					

Table 5.

Possible entries for the pivot in the dual simplex method are:x[2,1], x[4,1] and py[3]. Look at the corresponding relative costs in table 3 and choose the minimum: py[3] enters the basis. Determine the closed path and get to the new solution in table 6. Note that no variable leaves the basis,in fact ps[3] does.

	1	2	3	4	5	y[i]	z[i]	s[i]
1						8	1	
2						3	4	
3						1	5	1
4						1	5	
5						11	0	
z[i]	1	4	5	5	0	24		

Table 6

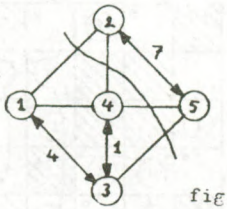


fig. 5

5.3 In further computations we should keep in mind that x[3,4]+x[3,5]>=1 must stay satisfied. Therefore we mark these cells in the tableau with a dot. The corresponding primal slack is zero,but we must create a dual variable s[3] for this extra constraint and indicate it in another extra column at row 3. It should fill the gap in the dual constraint, e.g. in the basic cell (3,4):

$$y[3]+z[3]-z[4]+s[3]=1$$

Also in a NB dotted cell,e.g. cell (3,5) must hold:

$$y[3]+z[3]-z[5]+s[3]<=7$$

In finding the dual values start as usual,but neglect the dotted cells. Afterwards we adjust via s[3]:cell (3,4):s[3]=5+0-5+1=1 and in cell (3,5):r[3,5]=7-(0+5-0+1)=1.

5.4 In the dual object function we must add the dual value corresponding to the extra constraint. The solution in table 6. is optimal,but does not build a tour. The solution should satisfy for example

$$x[2,1]+x[2,4]+x[5,3]+x[5,4]>=1$$

which is equivalent with the set of "one-node" inequalities

$$x[2,1]+x[2,4]>=1 \text{ or } x[5,3]+x[5,4]>=1$$

Because of c[2,4]=2 we try first x[2,1]+x[2,4]>=1 in the BandB process. According to 5.2,x[2,4] is a basic variable,so we look for the row coefficients of this constraint. In table 7. the 0-coefficients are left out. The non zero part of the row is given in table 8.

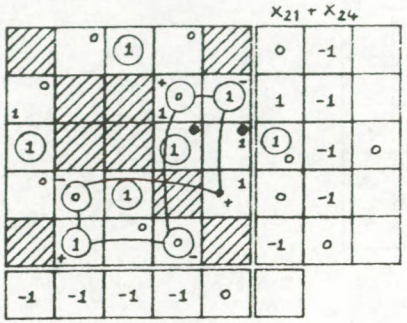


Table 7

NB var	x35	x45	py2	ps2
Extra row	1	1	1	-1
Rel costs	1	0	3	

Table 8

----- exta row coeff NB var

----- extra row coeff if ≠0

So x[4,5] enters the basis. Unfortunately the new solution in table 9 is infeasible again. The extra constraint in the dotted cells (2,1) and (2,4) is now satisfied. Of course the relative costs do not change,due to the zero relative cost of x[4,5].Next the infeasibility in cell (4,2) or (5,4) must be tackled. Note that in the dotted cells the s[i] values are taken into account. The row coefficients of x[4,2] are given in table 9 as well.

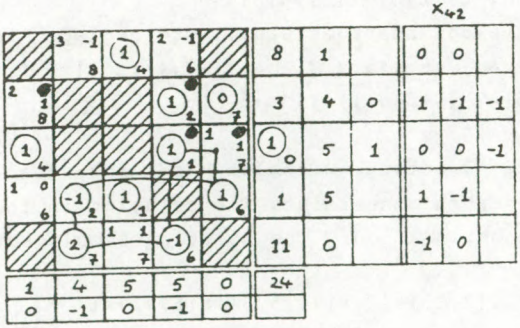
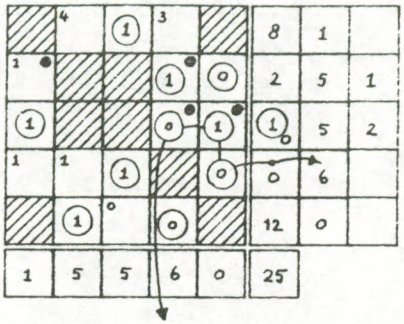


Table 9

5.5 The closed path may pass through two dotted cells in the same row,since then in the new solution the extra constraint is satisfied again. Let us choose x[3,5] to enter the basis and x[4,2] must leave (dual method!),giving the solution in table 10. Which is

an optimal tour solution! An alternative tour (13452431) can be found via y[4]=0.



	4	1	3		8	1	
1			1	0	2	5	1
1			0	1	1	5	2
1	1	1		0	0	6	
	1	0	0		12	0	
1	5	5	6	0	25		

Table 10

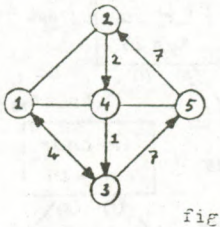


fig. 6

6. EXAMPLE

In this section we will illustrate the BandB procedure with a full exposition of our example.

step 1. In section 3 we started the example with the tour solution (1342541) and py[4]=1. This solution produces also an initial upper bound on the object value: xopt=26. However a tour solution is not necessary to begin with, mostly it is even disadvantageous as we shall see later.

steps 2. and 3. In section 4, table 3 the optimum solution was found.

step 4. It was not connected, so the extra condition

x[1,2]+x[1,4]+x[3,4]+x[3,5]>=1

was added, yielding the branching in fig.7.

step 5. Next we solved the ILP with x[3,4]+x[3,5]>=1. Two subtours were found (fig.5) and

x[2,1]+x[2,4]+x[5,3]+x[5,4]>=1

was split up, giving two feasible tours with xopt=25 in the branch with x[2,1]+x[2,4]>=1 (table 10 and fig 6).
Following the LIFO rule we then search the branch with x[5,3]+x[5,4]>=1, starting from table 6.

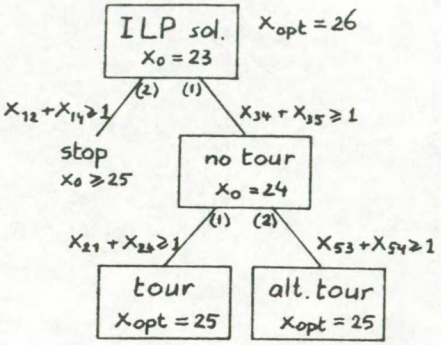
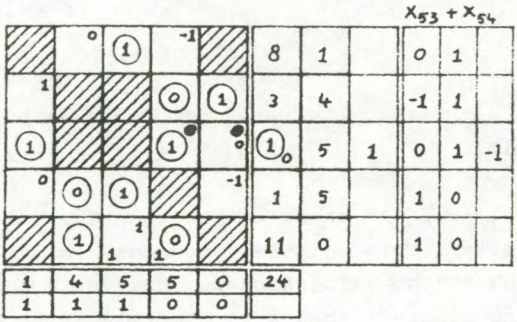


fig. 7

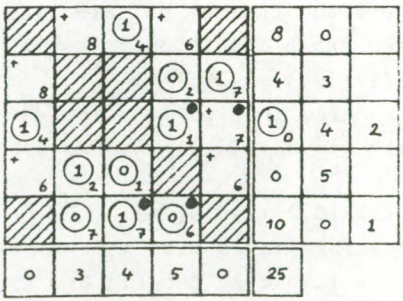
NB var	x21	x53	py4	py5	ps5
Extra row	1	1	1	1	-1
Rel costs	2	1	1	11	

Table 12



	0	1	-1		8	1		0	1	
1			0	1	3	4		-1	1	
1			1	0	1	5	1	0	1	-1
0	0	1		-1	1	5		1	0	
	1	1 1	0		11	0		1	0	
1	4	5	5	0	24					
1	1	1	0	0						

Table 11



	8 4	6		8	0	
8		0 2	1 7	4	3	
1 4		1 1	7	1 0	4	2
6	1 2	0 1	6	0	5	
	0 7	1 7	0 6	10	0	1
0	3	4	5	0	25	

Table 13

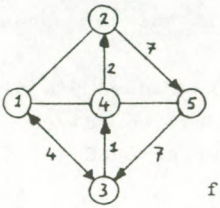


fig. 8

Let us choose x[5,3] according to table 12. Now we could stop in this branch because x0>=25 and xopt=25. But let us continue and find the alternative optimum in table 13 and fig.9.
Next we go back to the ILP solution in table 3 and add x[1,2]+x[1,4]>=1,giving the left branch of the tree in fig.7. This is an illustration of the remark in 5.1 because both x[1,2] and x[1,4] are NB variables. The best one:x[1,4] with r[1,4]=2 enters the basis giving the solution in table 14. We could stop because of the new optimum becoming at least 23+2=25>=xopt.
Moreover the solution is not connected,so we cannot improve in this branch. There is no problem left on the list.

step 6.The optimum value is xopt=25.

		0	1		7	2	2
			0	1	3	4	
1			0		1	5	
	0	1			1	5	
	1		0		11	0	
2	4	5	5	0	25		

Table 14

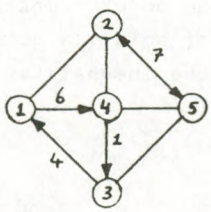


fig. 9

7. SOME CONSIDERATIONS

7.1 Starting solution for the ILP.

In practice mostly a feasible solution,whether it is a tour or an AP solution,can easily be found,using the directed road network. One can imagine that an AP solution is cheaper,in the mean,than a tour solution. So it is a better start in the way up to the optimum ILP solution,which must be found. However we also need a good "upper bound" tour. If the corresponding basic simplex solution appears to be optimal, the BandB procedure is not needed at all. So some effort should be made to find a good tour. But

perhaps an AP solution or both do not exist. The network may be so complex that it is not an easy task to find one or other. Therefore we need a good procedure for this search. The next question is how to find the or a basic simplex tableau corresponding with such a "hand"-solution. An AP solution is heavily degenerate and should be extended with suitably chosen 0-assignments. A tour solution could have less than (2n-1) positive assignments, but sometimes more, in which case apparently extra constraints are needed to build the basic solution. Another possibility is that redundant cycles occur in the hand-solution. Can these be recognised and deleted? Working with AP solutions means degeneracy. But advanced primal AP methods can be used in order to overcome these troubles. Conclusion: much can be gained by not starting the ILP problem simply with the dual simplex method.

7.2 Disjoint solution sets.

The "one-node" constraints in (2.6) do not necessarily yield disjoint solution sets. These can be achieved as follows. Suppose we treat the inequalities in the order of (2.6):

$$\sum_{jEJ1} x[1,j]>=1 \quad \text{or} \quad \sum_{jEJ2} x[2,j]>=1 \quad \text{and } x[1,j]=0 \text{ for } jEJ1 \quad \text{or} \quad \sum_{jEJ3} x[3,j]>=1 \quad \text{and } x[1,j]=0 \text{ for } jEJ1 \quad \text{and } x[2,j]=0 \text{ for } jEJ2 \quad \text{or } \ldots$$

where $Ji=\{i/iENl \text{ and } (i,j)EA\}$

Example:

Considering the situation in fig.5 and table 6. we get

$x[2,1]+x[2,4]>=1$ or $x[5,4]+x[5,3]>=1$ and $x[2,1]=x[2,4]=0$

or equivalently

$x[1,2]>=1$ or $x[4,2]+x[4,5]>=1$ or $x[3,5]>=1$ and $x[1,2]$ and $x[1,2]=x[4,2]=x[4,5]=0$

These disjointedness (DIS)-conditions can be used in several ways, e.g.

(i) only when an extra constraint contradicts a DIS-condition: the branch can be cut off.

(ii) the DIS-condition is fully worked up in the process, see section 8 for an application.

7.3 Extra constraints with slacks.

When, in applying the dual simplex method,a slack variable py[j] becomes negative in a primal basic solution we should express it in the NB variables by using the defining row in (2.7):

$$py[j]=x[j,k]-1$$

Apparently all x[j,k]=0,but some of these could be basic variables. We must have x[j,k]>=1,so we can treat this in the same way as the extra constraints in section 5.2. An extra slack variable ps[i] can be treated in a similar way.

7.4 Equivalence of inequalities.

The extra constraint (2.5) is equivalent with the "opposite" one in the reverse direction through the same cut:

$$\sum_{iEN2;jEN1} x[i,j]>=1$$

This can easily be proven using the set of equations (2.1).

Example:

In fig.4 we see:

Node 1: x[1,2]+x[1,4]=x[2,1]+x[4,1]+x[3,1]-x[1,3]

Node 3: x[3,4]+x[3,5]=x[4,3]+x[5,3]+x[1,3]-x[3,1]

---

x[1,2]+x1,4]+x[3,4]+x[3,5]=x[2,1]+x[4,1]+x[4,3]+x[5,3]

When expressed in the NB variables of the current tableau we get the same extra constraint(see also table 5):

$$-py[1]+x[1,2]+x[1,4]+x[2,1]+x[4,1]+py[3]>=1$$

But the result for the BandB process is different:

On one side:x[1,2]+x[1,4]>=1
or x[3,4]+x[3,5]>=1

On the other:x[2,1]>=1
or x[4,1]+x[4,3]>=1
or x[5,3]>=1

So even in this symmetric network we have a different number of branches. It depends on the current solution and the distances (costs) involved whether there will be significant difference in the efficiency of the process. In general one should choose such that the number of branches is as small as possible and such that the distinction in costs is as great as possible. Of course one should begin with searching the most promising branch.

8. APPLICATION TO A 13-NODES NETWORK

See fig.10 for the road network. In first instance a straightforward program in BASIC was used to solve this problem. Extra constraints were added to the usual simplex tableau in an interactive way. Later on we also solved the problem by hand using the TT-format. A possible BandB tree is given in fig.11. Finding a good primal feasible solution and building it up to a basic solution by adding appropriate zeroes was an easy task. In fact we did this during the determination of the dual values y[i] and z[i],choosing the cheaper cells if possible. In five (three degenerate) steps the optimal AP solution was found. The consequences of using the DIS-conditions of section 7.2 are indicated in the tree. For larger problems the gain is expected to be considerable. Perhaps the net profit decreases climbing higher up in the tree.

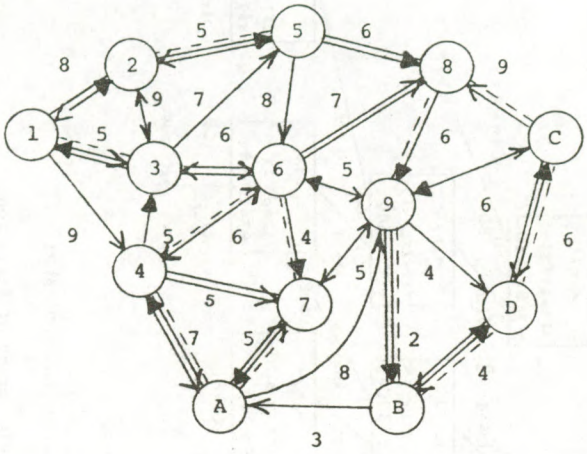
REFERENCES

[1] B.Fleischmann:"A cutting plane procedure for the TSP on road networks",E.J.O.R. Vol.21 (1985) p.307-317.

[2] R.S.Garfinkel,G.L.Nemhauser:"Integer programming",J.Wiley (1972).

[3] J.N.M.van Loon:"A primal method for the A.P.", Kwantitatieve methoden",Vol.5 (1984) nr.14,p.134-141 (in english).

Ontvangen: 06-07-1987
Geaccepteerd: 07-10-1987



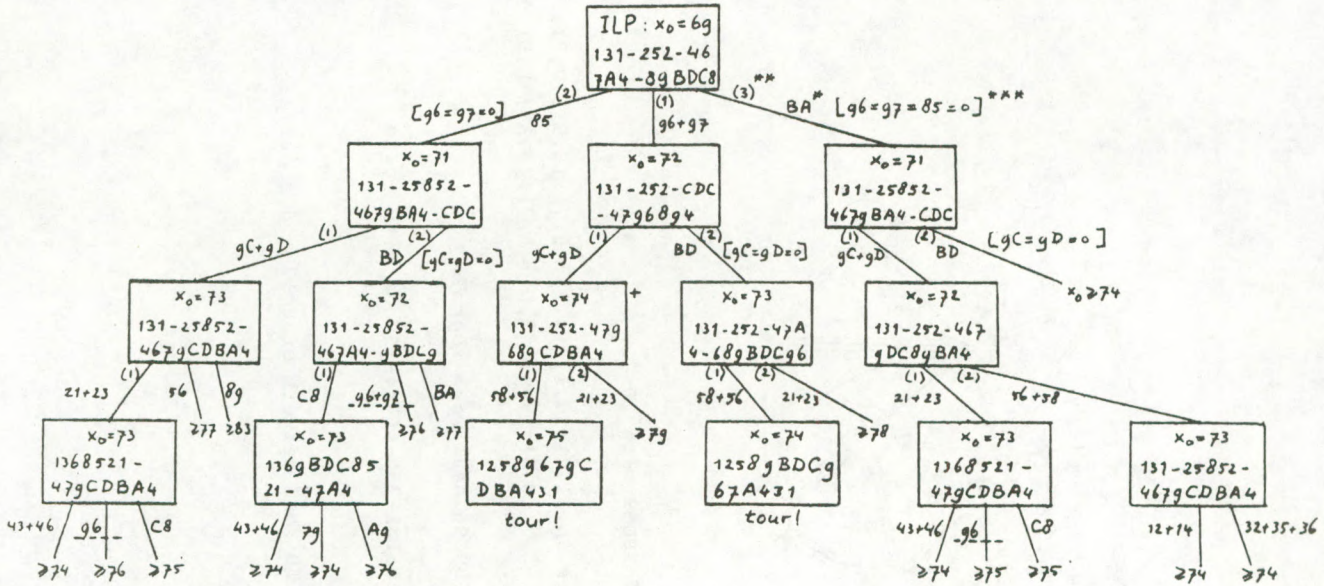
═══ start:1368521-47A4-9BDC9 with x0=72

0-assignments in (1,2);(3,1);(4,3);(6,7);(6,9);(7,9);
(8,9);(9,6);(9,D);(A,7);(B,A);(D,B)

▶ optimal tour:12589BDC967A431 with xopt=74

– – – optimal AP-solution:131-252-467A4-89BDC8 with x0=69

fig.10 Road network of a 13-nodes example.



* BA means: x[B, A] ≥ 1

** (3): order of execution

*** [.] : extra DIS-conditions: if used see lower left corner of this page

---- contradicts condition in this branch: stop

\+ when the order of execution was changed, we could have stopped here!

Fig.11: B and B tree of 13-nodes example.