THE ART OF INTEGER PROGRAMMING - RELAXATION

D. E. Bell

November 1973

WP-73-8

THE ART OF INTEGER PROGRAMMING - RELAXATION

David Bell
November 1973

When faced with a difficult problem, the integer programmer is apt to take the common approach of finding a related easier problem and solving that instead. In other disciplines this means approximating the data, making simplifying assumptions, etc.; in integer programming, the idea is to find a relaxation of the original problem.

Let
$$Z = \min_{x \in P} f(x) \tag{1}$$

be the original problem. If Q is a set containing P, then the problem

$$Z^* = \min_{x \in Q} f(x) \tag{2}$$

is said to be a relaxation of (1). The key to this approach may be highlighted by the following theorem.

THEOREM If x_Q solves (2) then

(i) $x_Q \in P$ implies x_Q solves (1).

(ii) $Z^* = f(x_Q) \leq Z$.

Thus, having solved (2), there is a simple test to see if (1) has been solved automatically (is $x_Q \in P$?) and if this is not the case, the effort of solving (2) is not wasted, for it provides a lower bound, $f(x_Q)$, for Z, the optimal value of (1). This is most useful in branch and bound procedures (see [1]).

Two questions spring readily to mind in connection with this idea of relaxing:

1. How should Q be chosen?

2. What can be done if $x_Q \notin P$?

Naturally, the more the problem is relaxed (making Q very large), the less likely it is that $x_Q \varepsilon P$. On the other hand, the larger Q is, the easier the relaxation is likely to be to solve, a clear case of a tradeoff in values. The hope is that for some Q, problem (2) is a well solved easy problem closely approximating (1).

The remainder of the paper gives three examples of relaxations being used in integer programming.

A. The Travelling Salesman Problem

Given a set of cities with known distances between them (some perhaps infinite), the salesman's aim is to set out from home (city no. 1 say) and visit all the other cities and return home having covered the minimum possible distance. In most practical examples, this can be done without visiting any city twice and it will be assumed that this is the case.

The problem is extremely difficult and no straightforward algorithm has been put forward to solve it. Approximate answers are easily obtained, the exact answer is not.

The set P in this case is the set of all tours, that is, all possible sequences of cities. How may we find a good relaxation set Q? Consider the following problem based on the same set of cities.

Suppose that no roads connect these cities and the government wishes to lay a system of roads which connects all the cities together but uses a minimum total distance of road. This problem is very simple indeed, solved by the greedy algorithm (see [2]). Having observed that the solution will include no circuits (for then one road must be redundant), the idea is to build the shortest road, then the next shortest road and so on, subject only to the requirement that no road should be built if it completes a circuit.

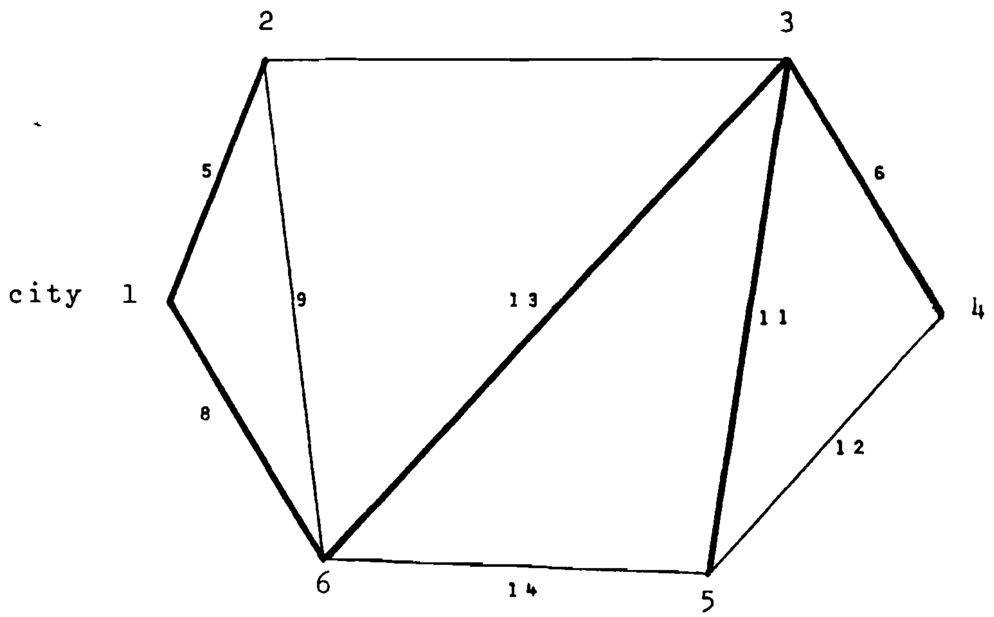

Figure 1

Define a Q-tour to consist of any two roads to city 1 plus a connected circuitless system of roads on the remaining cities.

A Q-tour may be completed in Figure 1 by adding the edge 2 - 6. Now let "Q" be the set of Q-tours.

Proposition Every tour is a Q-tour. Thus, P is a subset of Q, and the problem of finding a minimum Q-tour is a relaxation of the travelling salesman problem. Unfortunately, in Figure 1 it can be seen that if road 2 - 6 is added, the minimum Q-tour is not a tour, that is, $x_Q \notin P$. What can be done?

Suppose that a toll is imposed for entering or leaving a city. This means that if Ti is the toll for town i, the effective cost of travelling from city i to city j increases by Ti + Tj. Note that since the salesman must enter and leave each city exactly once, he has no choice but to pay an extra ΣTi no matter which route he takes, hence his optimal route is unaltered by imposing the tolls. However, this will affect the minimum Q-tour.

Since the aim is to have two roads leading into each city, the idea is to put a high toll on those cities with more than two roads in the optimal Q-tour (cities 3, 6 in Figure 1) and a low toll for those with only one road (4, 5).

Is it possible to find a system of tolls such that $x_Q \varepsilon P$?

For example, if $T_3 = 2$ and $T_6 = 4$, the problem in Figure 1 becomes

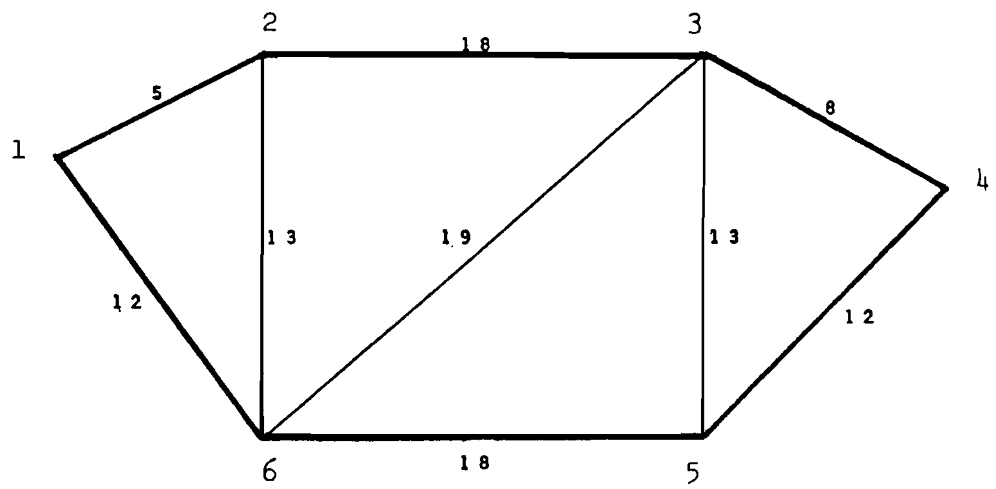

Figure 2

giving the minimum tour as 1 - 2 - 3 - 4 - 5 - 6 with a cost of 61. Note that the previous minimum Q-tour cost 52, a lower bound.

It has been shown [3, 4] that this method often works but that some networks do not have a suitable system of tolls. Branch and bound procedures are used in these cases.

B. Cutting Planes

The standard linear integer program is

$$\begin{aligned} \text{minimize} \quad & \bar{c}W \\ \text{s.t.} \quad & AW = b \\ & W \geq 0 \\ & W \equiv 0 \end{aligned} \tag{3}$$

where '≡' stands for equality modulo 1. It will be assumed here that $\bar{c}$, A, $\bar{b}$ are each integral, where A is an mx n+m matrix. Solving AW = b for m of the variables in terms of the remaining n, yields a problem written entirely in terms of those n variables

$$\begin{aligned} \min \quad & cx \\ & Nx \leq b \\ & Nx \equiv b \\ & x \geq 0 \quad \text{integral} \end{aligned} \tag{4}$$

The set P in this case is all integral values of x satisfying the constraints of (4).

A relaxation which is well solved (by the simplex method [5]) is that formed by ignoring the integrality constraints in (4) namely Nx ≡ b and x integral. This results in a linear program with optimal solution x_Q. Suppose x_Q is not integral or does not satisfy $Nx_Q \equiv b$ or both?

Let us suppose (and it is reasonable) that the m variables eliminated between (3) and (4) were the L.P. optimal basic variables, so that $x_Q = 0$ (and thus is integral) and hence that if $x_Q \notin P$, then $Nx_Q \equiv 0 \not\equiv b$.

Let $N^* \equiv N \qquad 0 \leq N^* < 1$

$b^* \equiv b \qquad 0 \leq b^* < 1$.

then $Nx \equiv b$

is equivalent to $N^*x \equiv b^*$.

Since $x \geq 0$, $Nx \equiv b$ are necessary conditions for $x \varepsilon P$ it must be that all $x \varepsilon P$ satisfy

$$N^*x \geq b^* . \tag{5}$$

But $N^*x_Q = 0 \not\equiv b$, and hence

$N^*x_Q \not\geq b^*$.

Let $\bar{Q}$ be all those elements of Q which satisfy (5), then

(i) $\bar{Q}$ contains P,

(ii) $\bar{Q}$ does not contain x_Q.

Hence, when the new problem ($\bar{Q}$) is solved, a new solution $x_{\bar{Q}}$ is found. If this is not in P, the process may be repeated. This procedure has been shown to converge (Gomory in [5] or see [6]).

C. The Group Problem

Remaining with (4), a second relaxation is to ignore the constraints $Nx \leq b$ leaving the relaxed problem

$$\begin{aligned} \min \quad & cx \\ & N^*x \equiv b^* \\ & x \geq 0 \quad \text{integer} \end{aligned} \tag{6}$$

where, with the assumption of the missing variables being L.P. optimal, $c \geq 0$.

Now, as it happens, ([7], [8]) the column vectors N*, b* generate a finite abelian group [9], say

$$G = \{g_0, g_1, \ldots g_k\} .$$

Consider the following network of k + 1 nodes corresponding to the group elements. Include a directed arc from node i to node j with cost c_k if $g_i + g_k \equiv g_j$.

Suppose, for example, the problem is

$$\begin{aligned} \min \quad & 3x_1 + 4x_2 \\ \text{s.t.} \quad & \tfrac{2}{5}x_1 + \tfrac{4}{5}x_2 \equiv \tfrac{1}{5} \qquad (7) \\ & x_1 \; x_2 \geq 0 \text{ integer.} \end{aligned}$$

The network then is

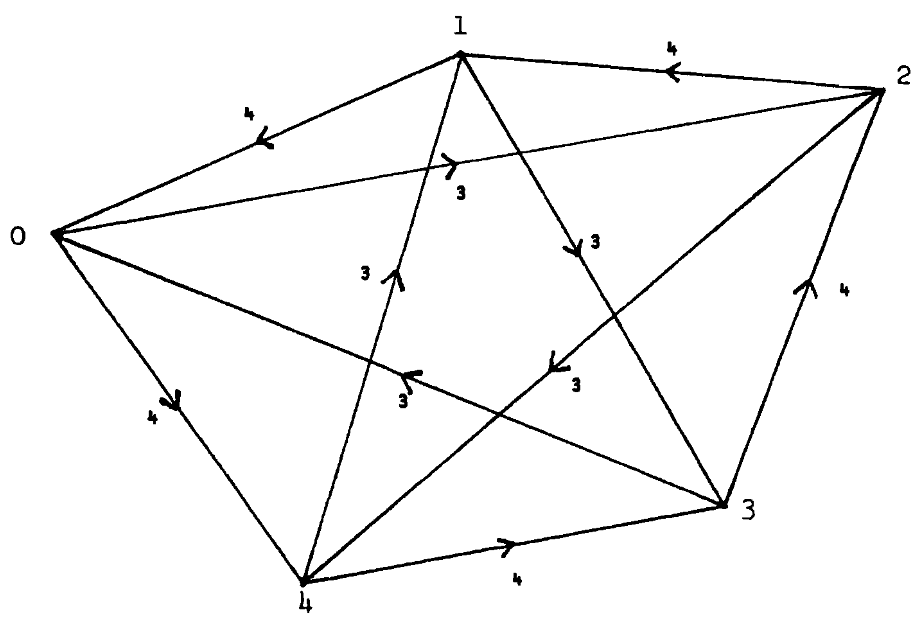

Figure 3

Now note that problem (7) is equivalent to finding the shortest route in the network from node 0 to node 1, and in general to the node equivalent to b*. The problem of finding a shortest route in a network is well solved and relatively easy. In Figure 3 it is 7 with two routes 0-4-1 and 0-2-1, which correspond to the solution $x_1 = 1$, $x_2 = 1$ to problem (7). The solution so obtained must be tested for feasibility in P ($x_Q \varepsilon P$?) in this case

$$Nx_Q \leq b \ .$$

A variety of methods exist for proceeding if $x_Q \notin P$, [10, 11, 12], an example of which is to add the cutting plane

$$cx \geq cx_Q$$

to problem (4) and to repeat the process.

REFERENCES

1. R. Garfinkel and G. Neuhauser, "Integer Programming," 1972.
2. J. Edmonds, "Matroids and the Greedy Algorithm," Mathematical Programming 1, 127-136, 1971.
3. M. Held and R.M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees," Operations Research 18, 1138-1162, 1970.
4. M. Held and R.M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees: Part II," Mathematical Programming 1, 6-25, 1971.
5. G.B. Dantzig, Linear Programming and Extensions, Princeton University Press, Princeton, New Jersey, 1963.
6. D.E. Bell, "The Resolution of Duality Gaps in Discrete Optimization," M.I.T. Operations Research Center, Technical Report No. 81, August 1973.
7. R.E. Gomory, "Some Polyhedra Related to Combinatorial Problems," Linear Algebra 2, 451-558, 1969.
8. L.A. Wolsey, "Group Representation Theory in Integer Programming," Technical Report No. 41, Operations Research Center, Massachusetts Institute of Technology, 1969.
9. G.D. Mostow, J.H. Sampson and J.P. Meyer, Fundamental Structures of Algebra, McGraw-Hill, New York, 1963.
10. M.L. Fisher and J.F. Shapiro, "Constructive Duality in Integer Programming," to appear in the SIAM Journal on Applied Mathematics.
11. D.E. Bell and M.L. Fisher, "Improved Bounds for Integer Programs Using Intersections of Corner Polyhedra," University of Chicago Graduate School of Business Working Paper, October 1973.
12. D.E. Bell, "Improved Bounds for Integer Programs: A Supergroup Approach". Preliminary Draft IIASA, October 1973.

13. J.D.C. Little, K.G. Murty, D.W. Sweeney and C. Karel, "An Algorithm for the Traveling Salesman Problem," Operations Research 11, 972-989, 1963.

14. G.A. Gorry, W.D. Northup and J.F. Shapiro, "Computational Experience with a Group Theoretic Integer Programming Algorithm," Mathematical Programming V4, 171-192.

APPENDIX

Computer Times

These times have been collected from various sources to give an indication of the rate of solution. With different codes, different machines and in different years no comparisons should be attempted.

A. The Traveling Salesman Problem

The toll procedure of Held and Karp gave the exact solution to those problems starred below. The remainder were continued by Branch and Bound. The machine was an IBM 360/91.

Number of Cities	Time (seconds)
* 20	4
* 20	6
22	10
* 25	12
25	18
* 26	22
* 30	19
30	20
42	54
46	900
48	84
48	160
57	780
* 64	182
64	504
64	330
64	258
64	418

Many of the above problems were "challenges" to the system so it could be expected to perform rather better on average problems.

Little, Murty, Sweeney and Karel in 1963 (six years earlier) give <u>average</u> times for randomly generated problems (these tend to be easier) on an IBM 7090 of:

Cities	Seconds
10	.72
20	5
30	59
40	500

B. No Information

C. <u>Group Problem</u>

Gorry, Northup and Shapiro report the following times using the group theoretic approach:

Rows	Columns	L.P. sec.	Total sec.	Machine
12	116	1.34	14	UNIVAC 1108
14	32	0.07	2.4	IBM 360/85
36	72	5.56	112	IBM 360/67
57	132	6.86	33	UNIVAC 1108
86	195	12.82	29	UNIVAC 1108
313	482	35.14	193	IBM 360/85
176	2385	49.26	67	IBM 360/85
5	54	0.35	0.4	IBM 360/85
27	641	4.38	5.3	IBM 360/85
26	383	10.08	136	IBM 360/85
50	65	0.39	4	IBM 360/85