1 LINEAR PROGRAMMING MODELS

Getting children to school:
Using linear programming to decide the 'best' school bus routes
(This extension material accompanies the text on p. 563 of your book.)

In Ankara, Turkey a number of buses pick up children from their homes and take them to an elementary school in the city centre. Each bus has a particular route to/from the school that is the same for both picking up and dropping off the students. Obviously each bus has a limit to how many students it can take. It is also desirable to restrict the distance each bus has to cover on a trip to 25 kilometres, so that the children's journeys are not too long.

Each bus stop is visited by only one bus, so that all the children there are picked up or dropped off at once.

At present 26 buses, each with a capacity of 33 children, pick up 519 students and the routing is planning intuitively. The routes tend to form a 'star' shape coming out from the centre of the city, most buses heading off to one destination only.

The bus company would like to reduce costs as much as possible.

The analysts started by deciding where the bus stops should be. To do this, they grouped the home locations of all the students into 29 sub-regions of the city that had roughly equal numbers of children. They used the centre of each sub-region as the pick-up point. They then calculated the distances between each pair of points in metres. Each bus was taken to have a fixed cost of about 18.6 million Turkish lire and then incur a per-metre cost of 300 Turkish lire. The existing system used 26 buses and on this basis cost a total of 556.8 million lire.

So how did the analysts model this situation as a linear programme?

First, they defined the decision variables. Let's label each bus stop arbitrarily 1, 2,… b. We can call the school 'bus stop 0' and it will be quite useful say that there is a fictitious final stop to which the buses go at the end of each route, called f. So we have a network with $b + 2$ stops or 'nodes'. Now consider the route or 'arc' which goes from the ith one of these to the jth, where i and j can each take any value from 0, 1, 2, … , b, f. Now suppose we introduce a set of decision variables x_{ij} which can only take the values 0 or 1. x_{ij} takes the value 1 in a feasible solution to the problem if one of the buses goes directly from node i to node j. However, if no bus goes directly from i to j, $x_{ij} = 0$. We could write the whole set of xs in a grid, as shown below for 5 buses. Each row corresponds to a node, and its entries to arcs *from* that node. In a similar way, each column corresponds to the arcs *to* a node. In the grid below, one bus goes from school to bus stop 2, and then from bus stop 2 to the finishing node. A second bus starts at school, goes to bus stop 3 first, from 3 to 5 and then 5 to 4 before finishing.

	0	*1*	*2*	*3*	*4*	*5*	*f*
0= school	0	0	1	1	0	0	0
1	0	0	0	0	0	1	0
2	0	0	0	0	0	0	1
3	0	0	0	0	0	1	0
4	0	0	0	0	0	0	1
5	0	0	0	0	1	0	0
f	0	0	0	0	0	0	0

To solve the linear programme we would like the set of 0s and 1s which minimises the costs. However, the 0s and 1s can't just go anywhere in the grid, there are restrictions to the values the *x*s can take. For instance, if a bus arrives at a stop, it must then leave it. There are several types of restriction.

Only a certain number of buses are available, say k, so at most k buses can leave the school and the total of the first row is at most k.

All these buses must end their route so the total of the last column is at most k.

Each of the intermediate stops, i.e. those other than school or the final stop, must have exactly one bus leaving it. So the total of each of the corresponding rows must be exactly 1. In a similar way, each of the immediate stops must have exactly one bus arriving at it , so the total of each of these columns must be exactly 1.

Whilst this establishes viable routes we haven't yet considered the number of children at each bus stop or the capacity of each bus. Also, no bus route must be longer than a fixed number of kilometres. Here the analysts used some rather clever constraints. For any arc i to j, the number of children who will be on the bus after picking up from j can be related linearly to the number who were on the bus after i, using the corresponding x_{ij} and the number of students at stop j. The number on the bus after any stop must be less than the total capacity of a bus. In this way, a set of linear constraints for bus capacity is formed. In a broadly similar way, the length of each route after arc ij can be related to the length up to i using x_{ij} and the distance between i and j, and restricted to less than the desired total distance for each route.

The upshot is that all the constraints on the values which the *x*s might take are linear.

We haven't yet spoken of the objective function. It will usually be possible to assign a cost, possibly related to distance to each arc of a route, say c_{ij}. Also, the use of each bus is likely to incur a fixed cost, say f.

The cost of using arc *i* to *j* is therefore $c_{ij}x_{ij}$. When that arc is used $x_{ij} = 1$ and the corresponding cost is incurred. When $x_{ij} = 0$, it isn't. The fixed cost multiplied by the number of buses (the total of the *x*s in the first row of the table above) plus the sum of the $c_{ij}x_{ij}$ is linear.

In Turkey, the analysts solved this programme on a fast computer using commercial optimisation software. The optimal solution used 18 buses at a total cost of 397 million lire, a reduction of some 29%. The total distance was decreased from 246 km to 210 km but the average distance travelled by each bus increased from 9.5 km to 11.7 km.

The pattern of the proposed routes was less 'star'-shaped than before, some buses following more 'zig-zag' routes.

For full details of this study take a look at:

T. Bektas and Seda Elmastas (2007) 'Solving school bus routing problems through integer programming', *Journal of the Operational Research Society*, 58, 1599–1604.