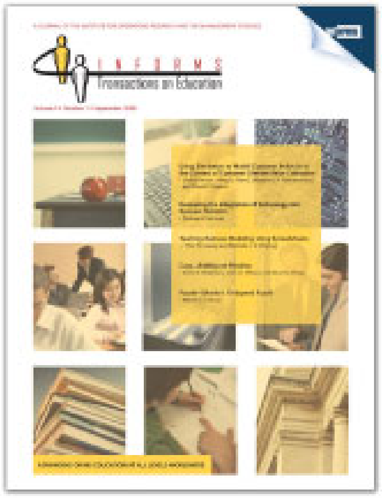

INFORMS Transactions on Education

Puzzle—Fairground Attractions

Martin J. Chlond,

Vol. 12, No. 2, January 2012, pp. 114–115
ISSN 1532-0545 (online)

INFORMS Transactions on Education

informs

http://dx.doi.org/10.1287/ited.1110.0084

Puzzle

Fairground Attractions

Martin J. Chlond
Lancashire Business School, University of Central Lancashire, Preston PR1 2HE, United Kingdom,
mchlond@uclan.ac.uk

A side show at Coney Island claiming to be "The Squarest Game on the Beach" is described in (Loyd 1914, p. 8) as follows.

> There were ten little dummies which you were to knock over with baseballs. The man said: "Take as many throws as you like at a cent apiece and stand as close as you please. Add up the numbers on all the men that you knock down and when the sum amounts to exactly fifty, neither more nor less you get a genuine 25 cent Maggie Cline cigar with a gold band around it."

The numbers on the ten dummies were 3, 6, 9, 12, 15, 30, 21, 25, 27, 30.

This is a classical knapsack problem with binary variables and is modeled with the constraint

$$\sum_{i\in N} v_i x_i = 50,$$

with v_i as the value of dummy i, $x_i = 1$ if dummy i is knocked over, 0 otherwise, and $N = 1, \ldots, 10$.

We merely need to achieve the target score and hence no objective function is required.

A similar puzzle, also in Loyd (1914), requires an archer to achieve a score of exactly 100. The rings on the target have values 16, 17, 23, 24, 39, and 40, respectively, but in this case more than one arrow may hit a particular ring. This may be solved using the above model with the modification that the variables are defined as integers rather than as binary variables.

The following puzzle from Love (2011) is in the same spirit as the above but provides us with a more interesting formulation exercise.

> At a fairground stall there are three piles of cans. You get three throws and you may only knock off the top can of a pile. The second throw counts double and the third triple. How do you achieve a score of exactly 50?

The values on each of the cans are shown in Figure 1.

Define the set $N = 1, \ldots, n$ where n is the number of cans in each pile, the number of piles and also the number of throws allowed, in this case $n = 3$.

Decision variables, defined over the range $\forall i \in N$, $j \in N$, $k \in N$, are as follows:

$x_{i,j,k} = 1$ if can i from pile j is hit by throw k, 0 otherwise.

Define parameters as follows:

$$v_{i,j} = \text{numerical value of can } i \text{ in pile } j.$$

The following constraints ensure that the conditions of the puzzle are fulfilled.

Achieve the target score:

$$\sum_{i\in N}\sum_{j\in N}\sum_{k\in N} k v_{i,j} x_{i,j,k} = 50.$$

Each can is hit no more than once:

$$\sum_{k\in N} x_{i,j,k} \le 1 \quad \forall i \in N,\ j \in N.$$

Each shot hits a single can:

$$\sum_{i\in N}\sum_{j\in N} x_{i,j,k} = 1 \quad \forall k \in N.$$

The first shot hits the top row:

$$\sum_{j\in N} x_{1,j,1} = 1.$$

Thereafter only hit the top can of a pile:

$$x_{i,j,k} \le \sum_{l=1}^{k} x_{i-1,j,l} \quad \forall i \in 2, \ldots, n,\ j \in N,\ k \in 2, \ldots, n.$$

Despite the fact that this problem presents a more challenging formulation, it may be solved quite easily by inspection. This is due to the fact that at each throw there are only three possible targets, that is, the top can of each pile. A brute force search can be car-

Figure 1 **Cans Puzzle**

8	10	7
10	7	9
7	9	8

Figure 2 **Maximize Score**

8	15	4	7
19	7	18	10
6	5	8	11
12	21	9	14

Figure 3 **Target = 100**

1	36	10	7
21	10	7	9
10	7	9	22
7	9	12	28

ried out in the form of a three stage decision tree with 27 end nodes.

The model may also be used to check for uniqueness. For example, the solution to the above puzzle is $x_{1,3,1} = 1$, $x_{1,1,2} = 1$, and $x_{2,3,3} = 1$, that is, the first shot hits the seven from the third pile, the second shot hits the eight from the first pile and the final shot hits the nine from the third pile. Hence, the constraint $x_{1,3,1} + x_{1,1,2} + x_{2,3,3} \leq 2$ will allow us to check whether other optimal solutions exist.

The puzzle may be generalised to larger sizes and we may also require that a score must be maximised rather than a target score achieved. The above model may be used to help create such puzzles and also to check solutions for uniqueness. Additional examples (Figures 2 and 3) are presented for the edification of the reader.

GNU Linear Programming Kit (GLPK 2010) MathProg implementations of the above formulations are included with this article (squaregame.mod, archery.mod, cans.mod).

Supplementary Material

Files that accompany this paper can be found and downloaded from http://ite.pubs.informs.org/.

References

GLPK. 2010. Accessed December 7, http://www.gnu.org/software/glpk/.

Love, T. 2011. Mathematical puzzles. Accessed December 14, http://www2.eng.cam.ac.uk/~tpl/maths/puzzles.html.

Loyd, S. 1914. *Cyclopedia of Puzzles*. The Lamb Publishing Company, New York. Available at http://www.mathpuzzle.com/loyd/.