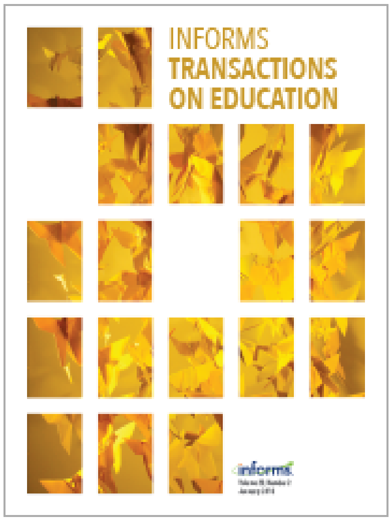


INFORMS Transactions on Education

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Solving the U2 Brainteaser with Integer and Dynamic Programming

Jack Brimberg, Bill Hurley,

Vol. 7, No. 3, May 2007, pp. 223–227
ISSN 1532-0545 | 07 | 0703 | 0223

INFORMS
Transactions on Education

informs®

DOI 10.1287/ited.7.3.223

Solving the U2 Brainteaser with Integer and Dynamic Programming

Jack Brimberg, Bill Hurley
Department of Business Administration, Royal Military College of Canada,
Kingston, Ontario, Canada {Jack.Brimberg@rmc.ca, hurley_w@rmc.ca}

1. Introduction

There are a number of well-known puzzles that involve transporting groups of people across water under interesting restrictions. Among these, the missionary-cannibal problem is perhaps the most famous:

> Three missionaries and three cannibals must cross a river in a boat that holds only two people. If the cannibals outnumber the missionaries on either bank, they will eat the missionaries. How can the missionaries manage the crossing without being eaten?

The interested reader is referred to Peterson (2004) for variations of the problem. A more recent version involves the rock band U2:

> U2 has a concert that starts in 17 minutes and they must all cross a bridge to get there. All four men begin on the same side of the bridge. It is night and they have only one flashlight. A maximum of two people can cross at one time. Any party who crosses, either 1 or 2 people, must have the flashlight with them. The flashlight must be walked back and forth; it cannot be thrown or returned in any other way. Each bandmember walks at a different speed as follows:

Bono:	1 minute
Edge:	2 minutes
Adam:	5 minutes
Larry:	10 minutes

> A pair must walk together at the rate of the slower man's pace. For example, if Bono and Larry walk across first, it will take them 10 minutes to get to the other side of the bridge. If Larry then returns with the flashlight, a total of 20 minutes will have passed and you have failed the mission. You must show how to get the bandmembers across in time to start the concert.

Dozens of versions of this problem have appeared under different names, such as the Bridge-Crossing Puzzle, the Bridge and Torch Problem, and the Flashlight Puzzle. Sillke (2001) gives a history of the problem on his web page. Sniedovich (2007) uses dynamic programming to solve an extended version where more than two persons at a time can cross the bridge. His web page includes an on-line interactive module programmed in Java Script for visualizing solutions and finding the best one. There have been suggestions that Microsoft tests prospective employees with this problem.

For the most part, structured solutions to puzzles of this sort have come from the artificial intelligence literature. However operations research has also made a contribution. The reader is referred to a very interesting series of papers appearing in this journal, e.g., Chlond (2002, 2003, 2004, 2005), Chlond and Toase (2002, 2003), and Chlond and Akyol (2003), which use integer programming to solve puzzles, and Sniedovich (2002, 2003) where dynamic programming is used. In this paper, we present a new integer programming formulation to solve the U2 problem. We also show that it is equivalent to a shortest path problem on a network, which may be solved by a simple dynamic program. To the best of our knowledge, this is the only instance of a puzzle where both techniques, integer and dynamic programming, are applied together.

We believe this puzzle has value in an introductory OR course for a couple of reasons. First, we think it is an excellent modeling exercise. At RMC in our business program, we tend to stress formulation and modeling skills rather than solution skills. In this regard the U2 problem is nice because it is difficult to formulate. And for some reason our students, particularly our graduate students, spend a lot of time on it. There is something about these seemingly simple puzzles that engage students. It is as though not being able to solve them is a significant mark of failure.

And second, the U2 problem is a good example of the use of an analysis phase preceding the construction of the model itself. The problem has some structure which allows the model formulation to be simplified considerably. If students recognize this, they are able to reap the rewards of their ingenuity.

2. Solving with an Integer Program

First some terminology. We identify the members of the band by their times to cross the bridge. Hence "1" is Bono, "2" is Edge, and so on. At any particular point in time, we call the subset of individuals who are on the concert side of bridge the *Concert Side Set*; the subset of those on the other side is termed the *Wrong Side Set*.

It would appear sensible to select the person to return the flashlight from the Concert Side to the Wrong Side as the bandmember with the shortest time to get back across the bridge. For instance, if Edge, Adam, and Larry are on the Concert Side with the flashlight, Edge would be chosen to carry the flashlight back to the Wrong Side. We state this observation formally as follows:

Property 1. Selecting a member to return the flashlight to the Wrong Side who is not the fastest member in the current Concert Side Set cannot lead to a better solution.

Proof. Consider any sequence where the fastest member is not chosen each time to return the flashlight to the Wrong Side. Select one of these violations, i.e. member j carries the flashlight back with crossing time t_j, while member i could have been chosen with crossing time $t_i < t_j$.

Now consider the same sequence, except at the point in time when member j is returning the flashlight, we chose member i instead. Following this point there are two possibilities:

1. In the original sequence, j crosses back to the Concert Side with another member k while i remains on the Concert Side without moving. Then in the new sequence, i again takes the place of j thus crossing with member k to the Concert Side. All other moves in the two sequences are the same. We see that a net saving in the total elapsed time occurs with the new sequence, given by,

$$T(\text{original}) - T(\text{new}) = t_j + \min(t_j, t_k) - (t_i + \min(t_i, t_k)) \geq t_j - t_i > 0. \quad (1)$$

2. In the original sequence, i returns to the Wrong Side with the flashlight before j makes a move back to the Concert Side. Then in the new sequence, simply replace i with j, keeping all remaining moves the same. The total elapsed time of each sequence will now be the same, i.e.,

$$T(\text{original}) - T(\text{new}) = 0. \quad (2)$$

We may proceed in this fashion to undo all violations, with each exchange either reducing the total elapsed time of the sequence or keeping it the same. This completes the proof.

This result allows us to simplify the model formulation considerably as follows. We conceive the *first cycle* as the actions corresponding to the first two bandmembers going across the bridge and the quicker bandmember bringing the flashlight back to the Wrong Side. Based on this notion of first cycle, we define $x(1,2)$ to be 1 if bandmembers 1 and 2 are the first to cross the bridge and 0 otherwise. In a similar way we define

$$x(1,5),\ x(1,10),\ x(2,5),\ x(2,10),\ x(5,10). \quad (3)$$

The *second cycle* is defined as the remaining crossings to get all of the bandmembers to the Concert Side. Given that bandmember k is on the Concert Side after the first cycle, we define $y(i,j\mid k)$ to be 1 if bandmembers i and j are the next to cross and 0 otherwise. At this point there will be three bandmembers on the Concert Side, i, j, and k. The bandmember to return the flashlight and get the last bandmember will be the one with the shortest time to cross among the three available. Hence we have 9 additional decision variables:

$$\begin{gathered} y(1,5\mid 2), y(1,10\mid 2), y(5,10\mid 2), \\ y(1,2\mid 5), y(1,10\mid 5), y(2,10\mid 5), \\ y(1,2\mid 10), y(1,5\mid 10), y(2,5\mid 10). \end{gathered} \quad (4)$$

The coefficients in the objective function give the elapsed time in each cycle. For example, if bandmembers 1 and 5 cross the bridge first ($x(1,5)=1$), the elapsed time of the first cycle is

$$\max(1,5)+1=6. \quad (5)$$

If member 5 is on the Concert Side at the end of cycle 1 and members 1 and 10 cross next ($y(1,10\mid 5)=1$), the elapsed time of the second cycle is

$$\max(1,10)+1+\max(1,2)=13, \quad (6)$$

since bandmember 1 will then return to pick bandmember 2 for the last crossing.

As for constraints, we first need to make sure that 2 members cross the bridge to begin the process. Hence

$$x(1,2)+x(1,5)+x(1,10) + x(2,5)+x(2,10)+x(5,10)=1. \quad (7)$$

Next there is a "conservation of flow" constraint for each of bandmembers 2, 5, and 10:

$$\begin{gathered} x(1,2)=y(1,5\mid 2)+y(1,10\mid 2)+y(5,10\mid 2) \\ x(1,5)+x(2,5)=y(1,2\mid 5)+y(1,10\mid 5)+y(2,10\mid 5) \\ x(1,10)+x(2,10)+x(5,10) = y(1,2\mid 10)+y(1,5\mid 10)+y(2,5\mid 10) \end{gathered} \quad (8)$$

The complete program is

$$
\begin{aligned}
\min\ & 3x(1,2)+6x(1,5)+11x(1,10)+7x(2,5)\\
&+12x(2,10)+15x(5,10)+16y(1,5\,|\,2)\\
&+16y(1,10\,|\,2)+14y(5,10\,|\,2)+13y(1,2\,|\,5)\\
&+13y(1,10\,|\,5)+14y(2,10\,|\,5)+13y(1,2\,|\,10)\\
&+8y(1,5\,|\,10)+9y(2,5\,|\,10)\\
\text{s.t.}\ & x(1,2)+x(1,5)+x(1,10)+x(2,5)\\
&+x(2,10)+x(5,10)=1 \qquad (9)\\
& x(1,2)-y(1,5\,|\,2)-y(1,10\,|\,2)-y(5,10\,|\,2)=0\\
& x(1,5)+x(2,5)-y(1,2\,|\,5)\\
&-y(1,10\,|\,5)-y(2,10\,|\,5)=0\\
& x(1,10)+x(2,10)+x(5,10)-y(1,2\,|\,10)\\
&-y(1,5\,|\,10)-y(2,5\,|\,10)=0\\
& x(i,j),y(i,j\,|\,k)\in\{0,1\},\quad \text{for all } i,j,k.
\end{aligned}
$$

The solution, obtained using the Excel solver, is

$$
\begin{aligned}
x^*(1,2)&=1,\\
y^*(5,10\,|\,2)&=1,
\end{aligned}
\qquad (10)
$$

with all other variables at 0. This tells us to send 1 and 2 across first with 1 returning. Then 5 and 10 go across *with* 2 *returning*. Finally 1 and 2 cross. All bandmembers would then be on the Concert Side. The optimal value of the objective function is $3+14=17$ minutes. Hence the band can get to the Concert Side in 17 minutes.

There is a second optimal solution. We could have bandmember 2 return to the Wrong Side with the flashlight at the end of the first cycle and still get 17 minutes. Note that this corresponds to case 2 in the proof of Property 1.

3. Solving with a Dynamic Program

Unlike some other puzzles (see the references) that have been formulated and solved as integer programs, the U2 Brainteaser is also amenable to a dynamic programming approach. This provides a good way of introducing the "principle of optimality" that often applies in decision processes. We view this decision process as a sequence of decisions resulting in a sequence of changing states. In our case, the subset of bandmembers on the Concert Side changes after each decision. Initially the subset is empty, and at the end of the process it contains all the bandmembers. The principle of optimality requires that a partial sequence of consecutive decisions leading from any intermediate state A to a subsequent state B must be optimal if the entire sequence is to be optimal. This allows a "divide and conquer" approach to the problem that can lead to some interesting discussions in the classroom.

The problem at hand has three decision points, or stages (using the terminology of dynamic programming), given in chronological order by the numbers 1, 2, 3. At stage i, we have $i-1$ members on the Concert Side resulting in a particular state A_i, and $5-i$ members still remaining on the Wrong Side given by the complement

$$A_i^C=\{1,2,3,4\}\backslash A_i. \qquad (11)$$

Suppose at this stage, two members r, $s\in A_i^C$ are sent across to the Concert Side. By Property 1, we may select the fastest member in the new subset $A_i\cup\{r,s\}$ to return the flashlight to the Wrong Side without making the end solution worse. The new state at stage $i+1$ is now fixed at

$$A_{i+1}=A_i\cup\{r,s\}\backslash\{f\}, \qquad (12)$$

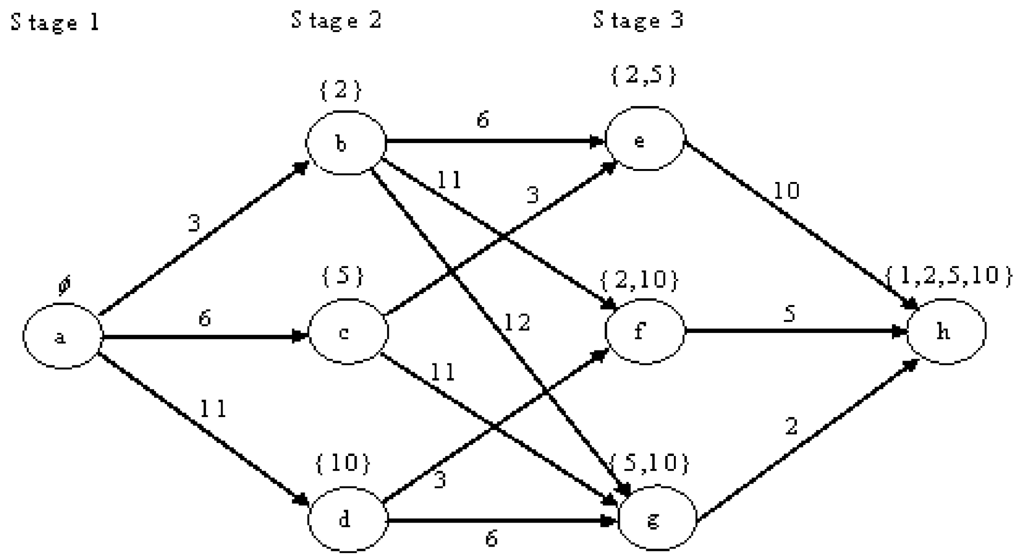
where member f has been selected according to Property 1. We see that the net effect of decision i is to add one more member to the Concert Side. Recalling that t_j denotes the crossing time of member j, the elapsed time (or cost) associated with this decision is given by $\min(t_r,t_s)+t_f$.

The decision process may be visualized as a network as shown in Figure 1. The nodes represent different states, and are grouped by stage number i. The directed edges are used to connect a node to a successive one at the next stage. A weight (or distance or cost) is assigned to each edge equal to the elapsed time associated with that decision. In some cases, there may be more than one edge connecting a pair of nodes. For example, the state $A=\{5\}$ may be reached by sending 1 and 5 across or 2 and 5, with costs of 6 and 7, respectively. However it is obvious that we only need to retain the edge with the shortest time. It is important to stress in the classroom that Property 1 allows us to simplify this graph considerably.

The problem is now equivalent to finding the shortest path from node a to node h on the graph. Due to the sequential structure of the graph, we may use a simple recursive relation derived from the principle of optimality. For example, if the shortest path from node a to node h goes through node b, the sub-path connecting node b to node h must also be a shortest path between these two nodes. Letting $D_{i+1}^*(k)$ denote the shortest distance from node k in state $i+1$ to the destination node, S_{i+1} be the subset of nodes grouped in state $i+1$, j be a node in state i, and $t(j,k)=$ the distance of edge (j,k) ($t(j,k)=+\infty$, if there is no edge joining j to k), the recursive relation may be summarized as follows:

$$D_i^*(j)=\min_{k\in S_{i+1}}\{t(j,k)+D_{i+1}^*(k)\}. \qquad (13)$$

Figure 1 The Directed Graph for the U2 Brainteaser



We proceed backwards (from right to left), and retain the adjacent node (k^*) identified on the shortest path at each stage in order to re-build this path from source to destination at the end of the recursion. Since the decision at stage $i = 3$ is obvious (send the two remaining members across), we begin with stage $i = 2$.

$i = 2$:

$$D_2^*(b) = \min(6+10, 11+5, 12+2) = 14 \quad (k^* = g)$$
$$D_2^*(c) = \min(3+10, 11+2) = 13 \quad (k^* = e \text{ or } g) \qquad (14)$$
$$D_2^*(d) = \min(3+5, 6+2) = 8 \quad (k^* = f \text{ or } g)$$

$i = 1$:

$$\begin{aligned} D_1^*(a) &= \min(3 + D_2^*(b), 6 + D_2^*(c), 11 + D_2^*(d)) \\ &= \min(17, 19, 19) \\ &= 17 \quad (k^* = b). \end{aligned} \qquad (15)$$

Thus the shortest path is found to be

$$a \to b \to g \to h, \qquad (16)$$

with a total distance of 17. This is the same solution obtained by the integer program.

4. A Final Comment

Upon further examination of the U2 solution, one might wonder if there is a general procedure which finds an optimal solution for an arbitrary number of bandmembers and crossing times. One such procedure is as follows:

Step 1—Send the two fastest members in the group (say numbers 1 and 2) across to the Concert Side; the fastest (number 1) returns the flashlight to the Wrong Side.

Step 2—Of the remaining members on the Wrong Side, send the two slowest across to the Concert Side; the second fastest member (number 2) returns the flashlight to the Wrong Side.

Step 3—Go back to Step 1, and repeat the process until no one is left on the Wrong side.

If there is time, this question could be put to students either as a follow on homework exercise or as part of an in-class discussion. Some students might argue that the above procedure always finds an optimal solution, while others may not be sure. Students should be encouraged to work with simple examples to get a better feel for the problem. Some of them may come up with a counterexample. If not, suggest to them that they examine the same U2 problem except that Edge's time is now 4 minutes instead of 2. Using the crossing times as before to identify the bandmembers, the solution given by the procedure is

giving a total crossing time of

$$\{1,4\} - \{1\} + \{5,10\} - \{4\} + \{1,4\} \qquad (17)$$
$$(4+1) + (10+4) + (4) = 23 \text{ minutes}. \qquad (18)$$

However, the optimal solution,

$$\{1,4\} - \{1\} + \{1,5\} - \{1\} + \{1,10\}, \qquad (19)$$

or path $a \to b \to e \to h$ in Figure 1, has a total time of 21 minutes. Thus, the given procedure is good but not optimal.

Rote (2002) provides an in-depth analysis of the general bridge-crossing problem on his web page. The problem is converted to one of finding a subgraph with special properties, and where candidate solutions are identified, one of which must be optimal.

For the case $n = 4$, the two candidate solutions are the ones found above.

References

Chlond, M. J. 2002. The riddle of the pilgrims. *INFORMS Trans. Ed.* **2**(2), http://ite.pubs.informs.org/Vol2No2/Chlond/.

Chlond, M. J., C. M. Toase. 2002. IP modeling of chessboard placements and related puzzles. *INFORMS Trans. Ed.* **2**(2), http://ite.pubs.informs.org/Vol2No2/ChlondToase/.

Chlond, M. J. 2003. The traveling space telescope problem. *INFORMS Trans. Ed.* **3**(1), http://ite.pubs.informs.org/Vol3No1/Chlond/.

Chlond, M. J., O. Akyol. 2003. A nimatron. *INFORMS Trans. Ed.* **3**(3), http://ite.pubs.informs.org/Vol3No3/ChlondAkyol/.

Chlond, M. J., C. M. Toase. 2003. IP modeling and the logical puzzles of raymond smullyan. *INFORMS Trans. Ed.* **3**(3), http://ite.pubs.informs.org/Vol3No3/ChlondToase/.

Chlond, M. J. 2004. Fantasy OR. *INFORMS Trans. Ed.* **4**(3), http://ite.pubs.informs.org/Vol4No3/Chlond/.

Chlond, M. J. 2005. Classroom exercises in IP modeling: Su doku and the log pile. *INFORMS Trans. Ed.* **5**(2), http://ite.pubs.informs.org/Vol5No2/Chlond/.

Peterson, I. 2004. Tricky crossings. *Muse* (September) 34–35.

Rote, G. 2002. Crossing the bridge at night. Accessed at http://page.mi.fu-berlin.de/~rote/Papers/pdf/.

Sillke, T. 2001. Crossing the bridge in an hour. Accessed at http://www.mathematik.uni-bielefeld.de/~sillke/PUZZLES/crossing-bridge.

Sniedovich, M. 2002. OR/MS Games: 2. Towers of Hanoi. *INFORMS Trans. Ed.* **3**(1), http://ite.pubs.informs.org/Vol3No1/Sniedovich/index.php.

Sniedovich, M. 2003. OR/MS games: 3. Counterfeit coin problem. *INFORMS Trans. Ed.* **3**(2), http://ite.pubs.informs.org/Vol3No2/Sniedovich/.

Sniedovich, M. 2007. The bridge and torch problem. Accessed at http://www.tutor.ms.unimelb.edu.au/bridge/.