数理解析研究所講究録
947 巻 1996 年 150-161

A Recursive Algorithm for the Bottleneck Reve's Puzzle

A.A.K. Majumdar[1]
Department of Business Administration, Aichi University,
370 Kurozasa, Miyoshi-cho, Aichi 470-02, Japan

A. Halder
Department of Mathematics, Dhaka University,
Dhaka 1000, Bangladesh

Abstract

This paper gives the dynamic programming formulation of the bottleneck Reve's puzzle, in which the movements of the discs are restricted by the bottleneck size, b, where $b\,(\geq 1)$ is a preassigned integer. We have derived some local-value relationships, and based on these, we have given an algorithm that enables us to determine the optimal value function as well as the related parameters without appealing to the dynamic programming equations.

Key words : Bottleneck Reve's puzzle, dynamic programming, standard and legal positions, narrowness of a tower, bottleneck size, optimal partition numbers.

AMS (1991) Subject Classification : 90C39,05A99

1 Introduction

We consider the following problem, henceforth called the *Bottleneck Reve's Puzzle:* Given are four pegs, S, P_1, P_2 and D, and $n\,(\geq 1)$ discs, labelled $D_1, D_2, ..., D_n$ in increasing order, so that

[1]On leave of absence from the Department of Mathematics, Jahangirnagar University, Savar, Dhaka 1342, Bangladesh.

$$D_n > D_{n-1} > \cdots > D_2 > D_1.$$

Initially, the discs rest on the source peg, S, in a *tower in standard position* in small-on-large ordering (with D_n at the bottom and D_1 at the top). For any collection T of discs (on any peg), the *narrowness* of T, denoted by $N(T)$, is defined to be the label-index of the smallest disc of T, that is

$$N(T) = \min\{i : D_i \in T\},$$

with

$$N(\phi) = \infty \ \ (\phi \text{ being the empty set}).$$

The problem is to shift the tower of n (≥ 1) discs in standard position on S to the standard position on the destination peg, D, in minimum number of moves (using the auxiliary pegs, P_1 and P_2), where each move can transfer only the topmost disc from one peg to another under the condition that a disc D_i may not be placed on a tower of discs T if

$$i > N(T) + b - 1,$$

where b (≥ 1) is a preassigned integer, called the *bottleneck size.*

Any arrangement of the discs on the four pegs that can be obtained without violating the conditions of the bottleneck Reve's puzzle, will be called a *legal position.* For $b = 1$, the above problem reduces to the Reve's puzzle, introduced by Dudeney [1], and later studied by Roth [6], Hinz [2] and Majumdar [3]. When $b = n$, any disc can be placed on any other disc. It may be mentioned here that any disc can be placed on an empty peg, and further that, b is the maximum size of an "inverted tower" that can be formed without violating the given conditions of the problem.

The problem with three pegs was first posed by Wood [7]. Later, the problem was taken up by Poole [5], but in a different setting. For the 4-peg generalization, we have adopted the version of Poole.

In the next section, we give the Dynamic Programming (DP) formulation of the bottleneck Reve's puzzle, from which some local-value relationships are obtained. These local-value relationships are then exploited to give a recursive algorithm for the bottleneck Reve's puzzle in Section 3.

2 Dynamic Programming Formulation

For the 3-peg bottleneck Tower of Hanoi problem with n (≥ 1) discs and the bottleneck size b (≥ 1), let $g_3(n, b)$ denote the minimum number of moves required to shift the tower of n discs from its starting position to a legal (but not necessarily standard) position on another peg, and let $M_3(n, b)$ denote the minimum number of moves required to solve the bottleneck Tower of Hanoi problem. Then, $g_3(n, b)$ and $M_3(n, b)$ satisfy the following relationships, established by Poole [5].

Lemma 2.1 : For n (≥ 1) and b (≥ 1),

$$\begin{aligned}
(1) \quad g_3(n,b) &= 2g_3(n-b,b)+b, \; n \geq b, \\
g_3(n,b) &= n \text{ for all } 0 \leq n \leq b, \\
(2) \quad M_3(n,b) &= 2g_3(n-1,b)+1, \; n \geq 1, \\
M_3(0,b) &= 0 \text{ for all } b \geq 1.
\end{aligned}$$

The complete solution to the bottleneck Tower of Hanoi problem, due to Poole [5], is given in the following theorem.

Theorem 2.1 : Given $n \geq 1$ and $b \geq 1$, let

$$n = bq + r \text{ (with } q \in \{0, 1, ...\}, 0 \leq r < b).$$

Then,

$$\begin{aligned} (1)\quad g_3(n,b) &= (b+r)2^q - b, \\ (2)\ M_3(n,b) &= \begin{cases} (2b-1)(2^q-1), & \text{if } r=0 \\ (b+r-1)2^q - 2b + 1, & \text{if } r \neq 0 \end{cases} \end{aligned}$$

Now, given $n\,(\geq 1)$ and $b\,(\geq 1)$, let $g_4(n,b)$ be the minimum number of moves required to shift the tower of n discs from its starting position to a legal position on another peg (using all the 4 pegs), and let $M_4(n,b)$, be the minimum number of moves required to solve the bottleneck Reve's puzzle. Then, the dynamic programming equations (DPE) satisfied by $g_4(n,b)$ and $M_4(n,b)$ are given in the following lemma.

Lemma 2.2 : For $n\,(\geq 1)$ and $b\,(\geq 1)$,

$$\begin{aligned} (1)\quad g_4(n,b) &= \min_{0\leq k\leq n-b} 2\{g_4(k,b) + g_3(n-b-k,b)\} + b,\ n \geq b, \\ g_4(n,b) &= n \text{ for all } 0 \leq n \leq b, \\ (2)\ M_4(n,b) &= \min_{0\leq l\leq n-1} 2\{g_4(l,b) + g_3(n-l-1,b)\} + 1,\ n \geq 1, \\ M_4(0,b) &= 0 \text{ for all } b \geq 1. \end{aligned}$$

Proof : (1) In order to find the DPE satisfied by $g_4(n,b)$, we note that the transfer of the tower from its starting position on S to a legal position on another peg, say, P_1, may be affected as follows:

Step 1 : First, move the topmost k (smallest, consecutive) discs from S to a legal position on P_2 (using all the four pegs). This involves a minimum $g_4(k,b)$ number of moves.

Step 2 : Next, shift the (consecutive) $n-b-k$ discs from S to the legal position on D, using the three pegs available, in (minimum) $g_3(n-b-k,b)$ number of moves.

Step 3: Then, transfer the remaining b (largest) discs from S to P_1 in an inverted tower (in b number of moves).

Now, in the next two steps, following *Step 1* and *Step 2* in reverse order so as to shift the discs from D and P_2 respectively to P_1, we get the desired tower, in (minimum)

$$2\{g_4(k,b) + g_3(n-b-k,b)\} + b$$

number of moves, where $k\,(0 \le k \le n-b)$ is to be determined so as to minimize the above expression.

(2) In order to find the DPE satisfied by $M_4(n,b)$, the steps followed are:

Step 1 : First, transfer the topmost l discs from S to the legal position on some auxiliary peg, say, P_1, (using all the four pegs), in (minimum) $g_4(l,b)$ number of moves.

Step 2 : Next, move the $n-l-1$ discs from S to the legal position on P_2, using the three pegs available, in (minimum) $g_3(n-l-1,b)$ number of moves.

Step 3 : Shift the largest disc, D_n, thus freed, from S to D.

In the next two steps, *Step 1* and *Step 2* are followed in reverse order so as to move the discs from P_2 and P_1 respectively to D. Then, the total number of moves involved is

$$2\{g_4(l,b) + g_3(n-l-1,b)\} + 1$$

and $l\,(0 \le l \le n-1)$ is to be chosen so that the above expression is minimized.

All these complete the proof of the lemma. □

It may be mentioned here that

$$M_4(n,1) = g_4(n,1) \text{ for all } n \ge 1,$$

and it has been shown by Majumdar [4] that the above scheme is optimal for the problem with $b=1$. However, for $n > b \ge 2$, it remains to show that the schemes followed to find the DPE satisfied by $g_4(n,b)$ and $M_4(n,b)$ are indeed optimal.

For $n, b\,(\geq 1)$ fixed, let

$$G(n,k,b) = 2\{g_4(k,b) + g_3(n-b-k,b)\} + b;\ 0 \leq k \leq n-b, \quad (2.1)$$

$$F(n,l,b) = 2\{g_4(l,b) + g_3(n-l-1,b)\} + 1;\ 0 \leq l \leq n-1. \quad (2.2)$$

Furthermore, let

$$k_{min}(n,b) = \min\{k : 0 \leq k \leq n-b,\ g_4(n,b) = G(n,k,b)\}, \quad (2.3)$$

$$k_{max}(n,b) = \max\{k : 0 \leq k \leq n-b,\ g_4(n,b) = G(n,k,b)\}, \quad (2.4)$$

$$l_{min}(n,b) = \min\{l : 0 \leq l \leq n-1,\ M_4(n,b) = F(n,l,b)\}, \quad (2.5)$$

$$l_{max}(n,b) = \max\{l : 0 \leq l \leq n-1,\ M_4(n,b) = F(n,l,b)\}, \quad (2.6)$$

with

$$k_{min}(b,b) = k_{max}(b,b) = 0 = l_{min}(1,b) = l_{max}(1,b) \text{ for all } b \geq 1. \quad (2.7)$$

In the following lemmas, we give some local-value relationships satisfied by the optimal value functions $g_4(n,b)$ and $M_4(n,b)$, and the optimal partition numbers $k_{min}(n,b)$, $k_{max}(n,b)$, $l_{min}(n,b)$ and $l_{max}(n,b)$.

Lemma 2.3 : (1) For $b\,(\geq 1)$ fixed, both $g_4(n,b)$ and $M_4(n,b)$ are strictly increasing in $n\,(\geq 1)$.

(2) For $n\,(\geq 1)$ fixed, each of $g_4(n,b)$ and $M_4(n,b)$ is non-increasing in $b\,(\geq 1)$.

Proof : Part (1) can be proved by induction on n, while part (2) can be established by induction on n and b. The details are omitted. □

Lemma 2.4 : For any $n\,(\geq 1)$ and $b\,(\geq 1)$, let

$$P(n,b) = k_{min}(n,b)/k_{max}(n,b)/l_{min}(n,b)/l_{max}(n,b).$$

Then,

$$\begin{aligned}
&(1) \quad P(n,b) \leq P(n+1,b) \leq P(n,b)+1,\\
&(2) \quad g_4(n+1,b) - g_4(n,b) = 2^s \text{ for some integer } s \geq 0,\\
&(3) \quad M_4(n+1,b) - M_4(n,b) = 2^t \text{ for some integer } t \geq 1,\\
&(4) \quad g_4(n+1,b) - g_4(n,b) \leq g_4(n+2,b) - g_4(n+1,b)\\
&\qquad\qquad \leq 2\{g_4(n+1,b) - g_4(n,b)\},\\
&(5) \quad M_4(n+1,b) - M_4(n,b) \leq M_4(n+2,b) - M_4(n+1,b)\\
&\qquad\qquad \leq 2\{M_4(n+1,b) - M_4(n,b)\}.
\end{aligned}$$

Proof : We prove parts (2) and (4) together with

$$k_{min}(n,b) \leq k_{min}(n+1,b) \leq k_{min}(n,b)+1. \qquad (A)$$

We note that $G(n+1,k,b)$ is not minimized at $k = k_{min}(n,b) - 1$, for, otherwise, we would have the following chain of inequalities

$$\begin{aligned}
&2\ \{g_3(n+2-b-k_{min}(n,b),b) - g_3(n+1-b-k_{min}(n,b),b)\}\\
&< g_4(n+1,b) - g_4(n,b)\\
&\leq 2\{g_3(n+1-b-k_{min}(n,b),b) - g_3(n-b-k_{min}(n,b),b)\}
\end{aligned}$$

which,therefore, would lead to a contradiction. Continuing in this way, we can show that $G(n+1,k,b)$ is minimized at none of the values $k = k_{min}(n,b) - 1$, $k_{min}(n,b) - 2, \ldots$, and consequently, we get the l.h.s. inequality of (A).

The proof in the remaining cases is by induction on n. The results are true for $n = 1$. So, we assume the validity of the results for N with $1 \leq N \leq n$. Furthermore, it is sufficient to consider the case when $k_{min}(n+1,b) \geq k_{min}(n,b)+1$. Now, since

$$\begin{aligned}
&2\ \{g_4(k_{min}(n+1,b),b) - g_4(k_{min}(n+1,b)-1,b)\}\\
&\leq g_4(n+1,b) - g_4(n,b)\\
&\leq 2\{g_4(k_{min}(n,b)+1,b) - g_4(k_{min}(n,b),b)\}
\end{aligned}$$

the induction hypothesis gives

$$k_{min}(n+1,b) \leq k_{min}(n,b)+1 \Rightarrow k_{min}(n+1,b) = k_{min}(n,b)+1.$$

Thus, the proof of (A) is complete.

To prove part (2), we first note that

$$g_4(n+1,b) - g_4(n,b) = \begin{cases} 1, & \text{if } 1 \leq n \leq b-1 \\ 2, & \text{if } n = b. \end{cases}$$

For $n \geq b+1$, by (A), either

$$\begin{aligned} g_4(n+1,b) - g_4(n,b) &= 2\{g_3(n+1-b-k_{min}(n,b),b) \\ &- g_3(n-b-k_{min}(n,b),b)\}, \end{aligned}$$

or,

$$g_4(n+1,b) - g_4(n,b) = 2\{g_4(k_{min}(n,b)+1,b) - g_4(k_{min}(n,b),b)\},$$

and in either case, we get the desired result, in the first case the result following by virtue of Theorem 2.1(1), and in the second case, the result follows by virtue of the induction hypothesis.

To prove part (4), we consider the following four cases that may arise:

Case 1 : $k_{min}(n+2,b) = k_{min}(n+1,b) = k_{min}(n,b)$,

Case 2 : $k_{min}(n+2,b) = k_{min}(n+1,b)+1,\ k_{min}(n+1,b) = k_{min}(n,b)$,

Case 3 : $k_{min}(n+2,b) = k_{min}(n+1,b) = k_{min}(n,b)+1$,

Case 4 : $k_{min}(n+2,b) = k_{min}(n+1,b)+1,\ k_{min}(n+1,b) = k_{min}(n,b)+1$.

In all the Cases 1-4, the proof follows readily; for example, in Case 4, by virtue of the induction hypothesis, we have the following chain of inequalities

$$
\begin{aligned}
& g_4(n+2,b) - g_4(n+1,b) \\
= \;& 2\{g_4(k_{min}(n,b)+2,b) - g_4(k_{min}(n,b)+1,b)\} \\
\geq \;& 2\{g_4(k_{min}(n,b)+1,b) - g_4(k_{min}(n,b),b)\} \\
= \;& g_4(n+1,b) - g_4(n,b).
\end{aligned}
$$

Similarly,

$$g_4(n+2,b) - g_4(n+1,b) \;\leq\; 2\{g_4(n+1,b) - g_4(n,b)\}.$$

All these complete the induction. □

Corollary 2.1 : For any $n, b\,(\geq 1)$,

(1) $G(n+1,k,b)$ is minimized at $k = k_{min}(n,b)+1$, $k_{min}(n+2,b)$, $k_{max}(n,b)$, $k_{max}(n+2,b)-1$,

(2) $F(n+1,l,b)$ is minimized at $l = l_{min}(n,b)+1$, $l_{min}(n+2,b)$, $l_{max}(n,b)$, $l_{max}(n+2,b)-1$,

Proof: We prove part (1) only.

To prove that $G(n+1,k,b)$ is minimized at $k = k_{min}(n,b)+1$, we need only consider the case when

$$k_{min}(n+1,b) = k_{min}(n,b).$$

Now, if $G(n+1,k,b)$ does not attain its minimum at $k = k_{min}(n,b)+1$, then we get the following chain of inequalities

$$
\begin{aligned}
& 2\ \{g_4(k_{min}(n,b),b) - g_4(k_{min}(n,b)-1,b)\} \\
< \;& g_4(n+1,b) - g_4(n,b) \\
< \;& 2\{g_4(k_{min}(n,b)+1,b) - g_4(k_{min}(n,b),b)\}.
\end{aligned}
$$

Then, by Lemma 2.4(2),

$$2^s < g_4(n+1,b) - g_4(n,b) < 2^{s+1} \text{ for some integer } s \geq 1,$$

which violates Lemma 2.4(2) itself.

Hence, $G(n+1, k, b)$ must be minimized at $k = k_{min}(n, b) + 1$.

The proof of the remaining parts is similar. □

We conclude this section with the following lemma, which states that, for $n\,(\geq 1)$ and $b\,(\geq 1)$ fixed, in determining the values of $g_4(n, b)$ $(M_4(n, b))$, it is sufficient to keep track of $k_{min}(n, b)$ and $k_{max}(n, b)$ $(l_{min}(n, b)$ and $l_{max}(n, b))$ only.

Lemma 2.5 : For $n\,(\geq 1)$ and $b\,(\geq 1)$ fixed,

(1) $G(n, k, b)$ is minimized at all k with $k_{min}(n, b) \leq k \leq k_{max}(n, b)$,

(2) $F(n, l, b)$ is minimized at all l with $l_{min}(n, b) \leq l \leq l_{max}(n, b)$.

Proof : We prove part (1) only. Proof for part (2) is similar.

Without loss of generality, we may assume that

$$k_{max}(n, b) = k_{min}(n, b) + M \text{ for some } M \geq 2. \tag{2.8}$$

Now, if $G(n, k, b)$ is not minimized at $k = k_{min}(n, b) + 1$, then we would have the following inequality :

$$\begin{aligned} &g_4(k_{min}(n, b) + 1, b) - g_4(k_{min}(n, b), b) \\ &> g_3(n - b - k_{min}(n, b), b) - g_3(n - b - 1 - k_{min}(n, b), b). \end{aligned} \tag{B}$$

In this case, $G(n, k, b)$ cannot be minimized at $k = k_{min}(n, b) + 2$, for otherwise,

$$\begin{aligned} &g_4(k_{min}(n, b) + 2, b) - g_4(k_{min}(n, b) + 1, b) \\ &< g_3(n - b - 1 - k_{min}(n, b), b) - g_3(n - b - 2 - k_{min}(n, b), b). \end{aligned}$$

which, together with (B), would lead to a contradiction. Proceeding in this way, we can conclude that $G(n, k, b)$ cannot be minimized at $k = k_{min}(n, b) + N$ for all $N \geq 2$, contradicting our assumption (2.8).

Hence, $G(n, k, b)$ must be minimized at $k = k_{min}(n, b) + 1$. Continuing the argument, we get the desired result. □

3 Recursive Algorithm

Starting with (2.7), the Lemma 2.4 enables us to calculate $k_{min}(n,b)$ ($k_{max}(n,b)$) and $l_{min}(n,b)$ ($l_{max}(n,b)$), and hence, $g_4(n,b)$ and $M_4(n,b)$ respectively for given n and b. The following algorithm calculates $g_4(n,b)$, $k_{min}(n,b)$, $l_{max}(n,b)$ and $M_4(n,b)$ for given $n, b\,(\geq 1)$. In determining the values of $g_4(n,b)$ and $M_4(n,b)$, we have made use of Corollary 2.1. Since explicit expressions of these quantities are not available, this algorithm is the only one to calculate the optimal partition numbers as well as the optimal value functions.

Algorithm : kmin(n,b), g4(n,b), lmax(n,b), M4(n,b)

```
/*  n,b ( >1 ) - given integers */
    kmin(b,b) = 0
    lmax(1,b) = 0
       for  j  from  0  to  b  do
          g3(j,b) = j
          g4(j,b) = j
/*  Determination of kmin(n,b)  and g4(n,b)  */
       for i  from  b+1 to n do
          g3(i,b) = 2*g3(i-b,b) + b
          k = kmin(i-1,b)
          g1 = g4(k,b) + g3(i-b-k,b)
          g2 = g4(k+1,b) + g3(i-b-k-1,b)
          g4(i,b) = 2*g2 + b
             if (g1=g2) then kmin(i,b) = k
             else   kmin(i,b) = k+1
/*  Determination of lamx(n,b)  and M4(n,b)  */
       for i  from  2 to n do
```

```
l = lmax(i-1,b)
M1 = g4(l,b) + g3(i-l-1,b)
M2 = g4(l+1,b) + g3(i-l-2,b)
M4(i,b) = 2*M1 + 1
   if (M1=M2) then lmax(i,b) = l+1
   else   lmax(i,b) = l
```

Acknowledgement

One of the authors, A.A.K. Majumdar, is grateful to Professor Mitsushi Tamaki of the Aichi University, Japan, for his co-operation and encouragement. Thanks are also due to the AIEJ and the Ministry of Education, Japan, for the grant of scholarship under the Monbusho Follow-up Fellowship Program.

References :

1. Dudeney H.E. (1907), *The Canterbury Puzzles.* Thomas Nelson & Sons.

2. Hinz A.M. (1989), An Iterative Algorithm for the Tower of Hanoi with Four Pegs. *Computing,* **42**, 133 - 140.

3. Majumdar A.A.K. (1994), The Generalized Four-Peg Tower of Hanoi Problem. *Optimization,* **29**, 349 - 360.

4. Majumdar A.A.K. (1994), Frame's Conjecture and the Tower of Hanoi Problem with Four Pegs. *Indian J. Math.,* **36**(3), 215 - 227.

5. Poole D. (1992), The Bottleneck Tower of Hanoi Problem. *J. Recreational Math.,* **24**(3), 203 - 207.

6. Roth T. (1974), The Tower of Brahma Revisited. *J. Recreational Math.* **7**(2), 116 -119.

7. Wood D. (1981-82), The Towers of Brahma and Hanoi Revisited. *J. Recreational Math.,* **14**(1), 17 - 24.