Information Processing Letters 44 (1992) 241–243
North-Holland

21 December 1992

The towers of Hanoi problem with parallel moves

Jer-Shyan Wu and Rong-Jaye Chen

Department of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu 30050, Taiwan, ROC

Communicated by L. Boasson
Received 2 September 1992
Revised 12 October 1992

Abstract

Wu, J.-S. and R.-J. Chen, The towers of Hanoi problem with parallel moves, Information Processing Letters 44 (1992) 241–243.

This paper proposes a variant of the towers of Hanoi problem allowing parallel moves. An algorithm for this problem is presented and proved to be optimal.

Keywords: Towers of Hanoi; recurrence relations; algorithms

1. Introduction

The towers of Hanoi problem is well-known and discussed in [1–4]. There are three pegs (A, B, C), and n disks of different sizes are placed in small-on-large ordering on peg A. The object of the problem is to move all the n disks from peg A to peg B in original order. The rules of disk movements are as follows:

Rule 1. Only one disk can be moved at a time.

Rule 2. No disk is ever placed on a smaller one.

This paper proposes a variant of the towers of Hanoi problem allowing *parallel moves*. Every top disk may be simultaneously moved from its peg and placed on another peg at a given time. The constraint is that no more than one disk can be placed on the same peg. In other words, four types of moves including *single move, exchange, consecutive move* and *circular move* can be made. These moves are presented in Fig. 1.

Correspondence to: Dr. J.-S. Wu, Department of Computer Science and Engineering, National Chiao Tung University, 1001 TA Hsueh Road, Hsinchu 30050, Taiwan, ROC. Email: jswu@algo1.csie.nctu.edu.

2. The optimal algorithm

In the following we propose an algorithm for the variant of the towers of Hanoi problem allowing parallel moves and prove its optimality.

Notation:

A, B, C:	A is FROM peg, B is TO peg and C is SPARE peg.
$A(d_1, d_2, \ldots)$:	peg A with disks $d_1, d_2, \ldots$, from top to bottom; similarly for pegs B, C.
$A(0)$:	peg A with no disk; similarly for pegs B and C.
$R(A, B, C)$:	state of pegs A, B and C.
$f(n)$:	the minimal number of disk moves for the problem with n disks.

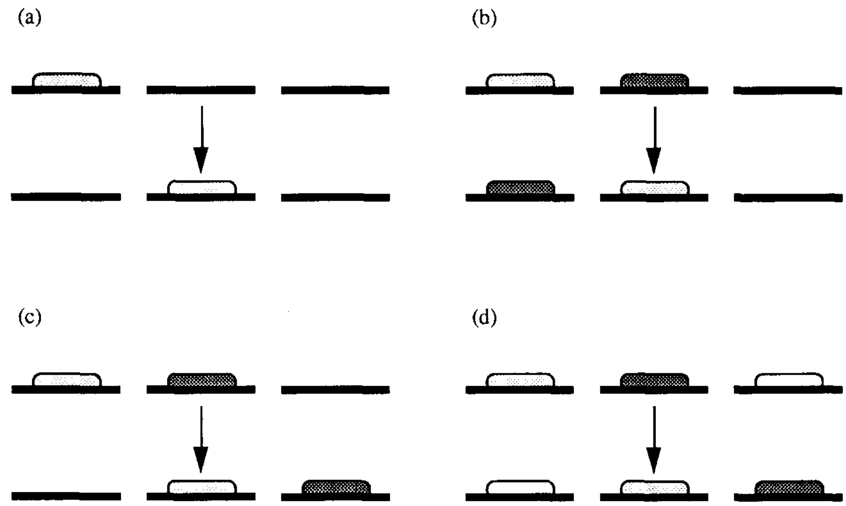

Fig. 1. Four moves: (a) single move, (b) exchange, (c) consecutive move, (d) circular move.

In this section, an optimal algorithm is derived and proved to be correct. To proceed, we need the following lemmas.

Lemma 1. *The number of disk moves for transforming* $R(A(1,\dots,n), B(0), C(0))$ *into* $R(A(0), B(1,\dots,n), C(0))$, *for* $n \geqslant 4$, *is at least* $2f(n-2)+1$.

Proof. The transformation of $R(A(1,\dots,n), B(0), C(0))$ into $R(A(0), B(1,\dots,n), C(0))$ can be divided into three steps:

Step 1 shows the transformation of $R(A(1,\dots,n), B(0), C(0))$ into the state immediately before disk n is moved,

Step 2 shows the move of disk n from peg A to peg B,

Step 3 shows the transformation of the state immediately after disk n is moved into $R(A(0), B(1,\dots,n), C(0))$.

In the final state of Step 1, only disk n is on peg A, and no more than one disk is on peg B while all others are on peg C (i.e. at least $n-2$ disks are on peg C). Hence the number of disk moves for Step 1 is at least $f(n-2)$. Step 2 takes one move and Step 3 is similar to Step 1. Therefore, we get the least number of disk moves for this problem: $2f(n-2)+1$, for $n \geqslant 4$. □

Now, we propose an algorithm for this problem taking $2f(n-2)+1$ disk moves, for $n \geqslant 4$.

Lemma 2. *To transform* $R(A(1,\dots,n), B(0), C(0))$ *into* $R(A(0), B(1,\dots,n), C(0))$, *for* $n \geqslant 4$, *the following algorithm*,

Step 1: *transform* $R(A(1,\dots,n), B(0), C(0))$ *into* $R(A(n-1, n), B(1), C(2,\dots, n-2))$;

Step 2: *transform* $R(A(n-1, n), B(1), C(2,\dots,n-2)$ *into* $R(A(n), B(n-1), C(1,\dots,n-2))$;

Step 3: *transform* $R(A(n), B(n-1), C(1,\dots,n-2))$ *into* $R(A(n-1), B(n), C(1,\dots,n-2))$;

Step 4: *transform* $R(A(n-1), B(n), C(1,\dots,n-2))$ *into* $R(A(1), B(n-1, n), C(2,\dots,n-2))$;

Step 5: *transform* $R(A(1), B(n-1, n), C(2,\dots, n-2))$ *into* $R(A(0)), B(1,\dots,n), C(0))$,

takes exactly $2f(n-2)+1$ *disk moves.*

Proof. The minimal number of disk moves for transforming $R(A(1,\dots,n), B(0), C(0))$ into $R(A(n-1, n), B(0), C(1,\dots,n-2))$ is $f(n-2)$ and it is clear that we can arrange the last move in this transformation to be the transfer of disk 1 from peg B to peg C. Hence the minimal number of disk moves for Step 1 is exactly $f(n-2)-1$. Step 2, Step 3 and Step 4 make a consecutive move, exchange and consecutive move in that order. Finally, Step 5 is similar to Step 1, and its minimal number of disk moves is also $f(n-2)-1$.

Therefore, the number of disk moves for this algorithm is

$$[f(n-2)-1]+1+1+1+[f(n-2)-1]$$
$$=2f(n-2)+1. \qquad \square$$

Applying the above two lemmas, we present the optimal algorithm as follows.

Theorem 3. *To transform* $R(A(1,\ldots,n), B(0), C(0))$ *into* $R(A(0), B(1,\ldots,n), C(0))$, $n \geqslant 4$, *the optimal algorithm is*

Step 1: *transform* $R(A(1,\ldots,n), B(0), C(0))$ *into* $R(A(n-1, n), B(1), C(2,\ldots,n-2))$;

Step 2: *transform* $R(A(n-1, n), B(1), C(2,\ldots, n-2))$ *into* $R(A(n), B(n-1), C(1,\ldots, n-2))$;

Step 3: *transform* $R(A(n), B(n-1), C(1,\ldots,n-2))$ *into* $R(A(n-1), B(n), C(1,\ldots,n-2))$;

Step 4: *transform* $R(A(n-1), B(n), C(1,\ldots,n-2))$ *into* $R(A(1), B(n-1, n), C(2,\ldots,n-2))$;

Step 5: *transform* $R(A(1), B(n-1, n), C(2,\ldots,n-2))$ *into* $R(A(0), B(1,\ldots,n), C(0))$.

Further, the minimal number of disk moves is $3\cdot 2^{(n-1)/2}-1$ *for n odd and* $2\cdot 2^{n/2}-1$ *for n even.*

Proof. By Lemma 1, the number of disk moves for this problem is at least $2f(n-2)+1$, and we have presented an algorithm that makes exactly $2f(n-2)+1$ moves in Lemma 2. Applying these two lemmas, we see that the algorithm is optimal.

It is trivial that the minimal numbers of disk moves for $n = 1, 2, 3$ are 1, 3, 5, respectively. We solve the recurrence relation $f(n) = 2f(n-2)+1$, for $n \geqslant 4$, and get $f(n) = 3\cdot 2^{(n-1)/2}-1$, for n odd and $f(n) = 2\cdot 2^{n/2}-1$, for n even. $\square$

References

[1] M.D. Atkinson, The cyclic towers of Hanoi, *Inform. Process. Lett.* **13** (1981) 118–119.

[2] P. Buneman and L. Levy, The towers of Hanoi problem, *Inform. Process. Lett.* **10** (1980) 243–244.

[3] P.J. Hayes, A note on towers of Hanoi problem, *Comput. J.* (1977) 282–285.

[4] S. Minsker, The towers of Hanoi rainbow problem: coloring the rings, *J. Algorithms* **10** (1989) 1–19.