On Optimal Solutions for the Bottleneck Tower of Hanoi Problem*

Yefim Dinitz and Shay Solomon

Department of Computer Science, Ben-Gurion University of the Negev
Beer-Sheva 84105, Israel
{dinitz,shayso}@cs.bgu.ac.il

Abstract. We study two aspects of a generalization of the Tower of Hanoi puzzle. In 1981, D. Wood suggested its variant, where a bigger disk may be placed *higher than* a smaller one if their size difference is less than k. In 1992, D. Poole suggested a natural disk-moving strategy for this problem, but only in 2005, the authors proved it be optimal in the general case. We describe the family of all optimal solutions to this problem and present a closed formula for their number, as a function of the number of disks and k. Besides, we prove a tight bound for the diameter of the configuration graph of the problem suggested by Wood. Finally, we prove that the *average* length of shortest sequence of moves, over all pairs of initial and final configurations, is the same as the above diameter, up to a constant factor.

1 Introduction

The classic Tower of Hanoi (ToH) puzzle is well known. It consists of three pegs and disks of sizes $1, 2, \ldots, n$ arranged on one of the pegs as a "tower": in decreasing, bottom-to-top size. The goal of the puzzle is to transfer all disks to another peg, placed in the same order. The rules are to move a single disk from (the top of) one peg to (the top of) another one, at each step, subject to the divine rule: to never have a larger disk above a smaller one.

The goal of the corresponding mathematical problem, which we denote by $HT = HT_n$, is to find a sequence of moves ("algorithm") of a minimal length ("optimal"), solving the puzzle. We denote the pegs naturally as *source*, *target*, and *auxiliary*, while the size of a disk is referred as its name. The following algorithm γ_n is taught in introductory CS courses as a nice example of a recursive algorithm. It is known and easy to prove that it solves HT_n in $2^n - 1$ disk moves, and is the unique optimal algorithm for it.

- If n is 1, move disk n from *source* to *target*.
- Otherwise:
  - recursively perform $\gamma_{n-1}(source, auxiliary)$;
  - move disk n from *source* to *target*;
  - recursively perform $\gamma_{n-1}(auxilary, target)$.

* Partially supported by the Lynn and William Frankel Center for Computer Science.

Jan van Leeuwen et al. (Eds.): SOFSEM 2007, LNCS 4362, pp. 248–259, 2007.

In the recent two decades, various ToH type problems were considered in the mathematical literature. Many algorithms were suggested, and extensive related analysis was performed. As usual, the most difficult, far not always achievable task is showing that a certain algorithm is optimal, by providing the matching *lower bound*. A distinguished example is the Frame-Stewart algorithm (of 1941), solving the generalization of the ToH problem to four or more pegs. It is simple, and an extensive research was conducted on its behavior, since then. However, its optimality still remains an open problem; the proof of its *approximate* optimality [5] was considered a breakthrough, in 1999. This paper contributes to the difficult sub-area of the ToH research—optimality proofs.

In 1981, D. Wood [6] suggested a generalization of HT, characterized by the k*-relaxed placement rule*, $k \geq 1$: *If disk j is placed higher than disk i on the same peg (not necessarily neighboring it), then their size difference $j - i$ is less than k.* In this paper, we refer it as the *Bottleneck Tower of Hanoi problem* (following D. Poole [4]), and denote it $BTH_n = BTH_{n,k}$. Now, there are more than one legal way to place a given set of disks on the same peg, in general; we refer the decreasing bottom-to-top placement of all disks on the same peg as the *perfect* disk configuration. If k is 1, we arrive at the classic ToH problem.

In 1992, D. Poole [4] suggested a natural algorithm for BTH_n and declared its optimality. However, his (straightforward) proof is done under the fundamental assumption that before the last move of disk n to the (empty) target peg, *all* other $n-1$ disks are gathered on the spare peg. This situation is far not general, since before the last move of disk n, from some peg X to the target peg, any set of the disks $n-1, n-2, \ldots, n-k+1$ may be placed below disk n on peg X. In 1998, S. Beneditkis, D. Berend, and I. Safro [1] gave a (far not trivial) proof of optimality of Poole's algorithm for the first non-trivial case $k = 2$ only. In 2005, the authors proved it for the general case, by different techniques (see [3]).

X. Chen et al. [2] considered independently a few ToH problems, including the bottleneck ToH problem. They also suggested a proof of optimality of Poole's algorithm, based on another technical approach.

Poole's algorithm is based on an optimal algorithm for another related problem of "moving somehow", under the k-relaxed placement rule: *To move m disks* $[1..m]$*, placed entirely on one peg, to another peg, in any order.* This algorithm is denoted by $\beta_m = \beta_m(source, target)$, and is as follows:

- If m is at most k, move all disks from *source* to *target* one by one.
- Otherwise:
  - recursively perform $\beta_{m-k}(source, auxiliary)$;
  - move disks $[(m-k+1)..m]$ from *source* to *target* one by one;
  - recursively perform $\beta_{m-k}(auxilary, target)$.

Poole's algorithm, denoted by $\alpha_n = \alpha_n(source, target)$ is as follows:

- perform $\beta_{n-1}(source, auxiliary)$;
- move disk n from *source* to *target*;
- perform $\beta_{n-1}(auxilary, target)$.

In [4], it was erroneously stated that β_m and α_n are unique optimal solutions for the corresponding problems. Let us show an example of an optimal solution to BTH_7, for $k = 2$, distinct from α_7. It has the same template as α_7, but uses another optimal "somehow" algorithms, instead of β_6. In the following description, a configuration of BTH_7 on the three pegs is depicted by three parentheses, containing the disk numbers for pegs A, B, and C, from bottom to top; each configuration is obtained from the previous one by one or more moves. The difference from α_7 begins from the eighth configuration, marked by !!.

(7654321)()() → (76543)(12)() → (765)(12)(34) → (765)()(3421) →
(7)(56)(3421) → (712)(56)(34) → (712)(564)(3) → (71)(5642)(3) !! →
(71)(56423)() → (7)(564231)() → ()(564231)(7) → ()(56423)(71) →
(3)(5642)(71) → (3)(564)(712) → (34)(56)(712) → (3421)(56)(7) →
(3421)()(765) → (34)(12)(765) → ()(12)(76543) → ()()(7654321)

In this paper, we find the *family of all optimal solutions* to the Bottleneck Tower of Hanoi Problem, and present a closed formula for their number.

Consider now a generalization of BTH_n, where the prescribed initial order of disks on peg A and their final order on peg C are not by decreasing size, but arbitrary. It is easy to see that α_n is far not always legal w.r.t. the k-relaxed placement rule. A natural arising question is what is the length of the *longest* one among all shortest sequences of moves, over all pairs of initial and final configurations, that is what is the diameter of the configuration graph of BTH_n. We prove a tight bound for the diameter, up to a constant factor. We also prove a stronger result: that the *average* length of shortest sequence of moves, over all pairs of initial and final configurations, is the same as the above diameter, up to a constant factor (for the cases $n \leq k$ and $n > 3k$). We believe that finding *exact* bounds for these problems is difficult even for the degenerate case $n \leq k$.

2 Definitions and Notation

A configuration of a disk set D is called *gathered*, if all disks in D are on the same peg. Such a configuration is called *perfect*, if D is an initial interval of naturals, and the order of disks (on a single peg) is decreasing. For any configuration C of D and any $D' \subseteq D$, the *restriction* $C|_{D'}$ is C with all disks not in D' removed.

A move of disk m from peg X to peg Y is denoted by the triplet (m, X, Y); the third peg, $Z \neq X, Y$, is referred as the *spare peg* of (m, X, Y). For a disk set D, the configuration of $D \setminus \{m\}$ is the same before and after such a move; we refer it as the *configuration of* $D \setminus \{m\}$ *during* (m, X, Y).

A *packet-move*, P, of D is a sequence of moves transferring the entire set of disks D from one peg to another. W.r.t. P, the former peg is called *source*, the latter *target*, and the third peg *auxiliary*. The *length* $|P|$ of P is the number of moves in it. If both initial and final configurations of P are perfect, we call P a *perfect-to-perfect* (or *p.t.p.*, for short) packet-move of D.

For better mnemonics, the entire set of disks $[1..m]$ is divided into $\lceil \frac{m}{k} \rceil$ *blocks* $B_i = B_i(m)$: $B_1 = [(m-k+1)..m]$, $B_2 = [(m-2k+1)..(m-k)]$, …, $B_{\lceil \frac{m}{k} \rceil} =$

$[1..(m - (\lceil \frac{m}{k} \rceil - 1) \cdot k)]$. Note that the set of disks in any block is allowed to be placed on the same peg in an arbitrary order. For any $m \geq 1$, let D_m denote $[1..m]$, and $Small(m)$ denote $D_m \backslash B_1(D_m)$. In the above notion, BTH_n concerns finding the shortest perfect-to-perfect packet-move of D_n.

A configuration is called *well-separated* if it satisfies the condition that at each peg, the disks in any block are placed continuously. Notice that β_m applied to a gathered well-separated configuration of m disks is legal, and results in the configuration, where the first block of disks is at the bottom of *target* in the *reverse order* w.r.t. its initial order at the bottom of *source*, while the rest of the disks are above it in their original order. As well, β_m applied to the latter configuration is legal and results in the original disk ordering.

We say that a move-sequence S *contains* a move-sequnece S' if S' is a subsequence of S. Several move-sequences S_i, $1 \leq i \leq r$, contained in S, are called *disjoint* if the last move in S_i precedes the first move in S_{i+1}, for each $1 \leq i \leq r-1$; the case when also $|S| = \sum_{i=1}^{r} |S_i|$ holds is denoted by $S = S_1 \circ S_2 \ldots \circ S_r$.

For any sequence of moves S of D and any $D' \subseteq D$, the *restriction of S to D'*, denoted by $S|_{D'}$, is the result of omission from S all moves of disks not in D'. Note that any restriction of a legal sequence of moves is legal as well, and if $D' \subseteq D$, a restriction of a packet-move of D to D' is a packet-move of D'. Clearly, if D is partitioned into D' and D'', then $|P| = |P|_{D'}| + |P|_{D''}|$.

Let us denote the length of β_m by b_m. It is known from [4,3] that if $m = sk+r$, where $0 \leq r < k$, then $b_m = (k + r) \cdot 2^s - k$. We will use the following result.

Theorem 1 ([3]). *Under the rules of BTH, the length of any packet-move of D_m is at least b_m.*

3 Exploration of the Configuration Graph

3.1 Main Results

The Diameter. We define the *Configuration Graph* for $BTH_{n,k}$ as the directed graph $G^{conf} = G^{conf}_{n,k} = (V, E)$, where V is the set of all the possible configurations of D_n on the three pegs, under the k-relaxed placement rule, and an edge $e = (u, v)$ is in E if $u, v \in V$ and u and v are reached one from the other by a single move. Let us denote the diameter of $G^{conf}_{n,k}$ by $Diam(n, k)$.

Our first result provides tight bounds for the diameter, up to a constant factor.

Theorem 2 (proof is omitted)

$$Diam(n,k) = \begin{cases} \Theta(n \cdot \log n) & \text{if } n \leq k \\ \Theta(k \cdot \log k + (n-k)^2) & \text{if } k < n \leq 2k \\ \Theta(k^2 \cdot 2^{\frac{n}{k}}) & \text{if } n > 2k \ . \end{cases}$$

The Average. Let us denote by $Avg(n, k)$ the average number of moves required to get from one configuration to another, taken over all pairs of configurations. The following theorem strengthens the first and the asymptotic cases of Theorem 2, asserting that $Avg(n,k)$ is the same as $D(n,k)$, up to a constant factor.

Theorem 3

$$Avg(n,k) = \begin{cases} \Theta(n \cdot \log n) & if\ n \leq k \\ \Theta(k^2 \cdot 2^{\frac{n}{k}}) & if\ n > 3k \end{cases} .$$

The following remains open.

Conjecture 1
For $k < n \leq 2k$, $Avg(n,k) = \Theta(n \log n + (n-k)^2)$.
For $2k < n \leq 3k$, $Avg(n,k) = \Theta(k^2)$.

3.2 Proof of Theorem 3

We first consider the degenerate case $n \leq k$. By the similar case of Theorem 2, any pair of disk configurations is reachable one from the other by a sequence of $O(n \log n)$ moves. Thus, the following lemma suffices.

Lemma 1. *Let $n \leq k$. The average number of moves required to get from one configuration to another, taken over all pairs of configurations, is $\Omega(n \cdot \log n)$.*

Proof. It suffices to prove that, for any configuration C, the average number of moves required to get from C to any other configuration is $\Omega(n \cdot \log n)$. We construct a BFS tree (the tree of shortest paths) of G^{conf} rooted at C, T_C, and note that the maximum degree of such a tree is six. A tree is called *6-ary* if its maximal degree is six. We call a 6-ary tree T $full$ if the number of vertices in each layer i, except for, maybe, the last layer, is 6^{i-1}; in this case, the depth of T is $h = \lceil \log_6(5 \cdot |V| + 1) \rceil$.

In order to bound the average distance from the root to a vertex in the tree T_C, we prove that the minimum argument for this value among all 6-ary trees, is a full 6-ary tree, and show that for such a tree, this value is $\Omega(n \cdot \log n)$ (details are omitted). □

Now we turn to the case $m > k$.

A move M in a move-sequence S is called *switched* either if it is the first move in S, or if the spare peg of M is different from the spare peg of its preceded move in S. A disk is called *switched* w.r.t. S if it participates in at least one switched move. We define the number of switched disks required to get from one configuration C to another C' as the minimal number of switched disks in a move sequence with the initial configuration C and the final configuration C'.

Lemma 2 (joint with N. Solomon). *Let $m > k$. The average number of switched disks in $B_1(m) \cup B_2(m)$ required to get from one configuration of D_m to another, taken over all pairs of configurations of D_m, is $\Omega(k)$.*

Proof. We may assume that $k > 10$, since otherwise the proof of the lemma is immediate. Let C_{init} be some configuration of D_m. We will show that the average number of switched disks in $B_1(m) \cup B_2(m)$ required to get from C_{init} to another configuration, taken over all configurations of D_m, is $\Omega(k)$, which suffices.

Consider some configuration C' of D_m. For each peg X, denote by d_X the highest disk from $B_1(m)$ on peg X, if any, and by $D_X(C')$ the set of all disks which reside on X lower than d_X; note that all of them belong to $B_1(m) \cup B_2(m)$. We define $B_{1,2}(C') := D_A(C') \cup D_B(C') \cup D_C(C')$, and note that $|B_{1,2}(C')| \geq k - 3$, since $(B_1(m) \setminus \{d_A, d_B, d_C\}) \subseteq B_{1,2}(C')$. Let us divide the entire set of disks $B_{1,2}(C')$ into triads and a residue of size at most six, according to their placement at C', as follows. Each triad consists of three consecutive disks placed on the same peg, X, from below d_X downwards, whereas there may remain a residue of size at most two, close to the bottom of each peg. Let $l_{C'}$ denote the number of such triads; note that $l_{C'} = \Omega(k)$.

We say that a triad is *switched* w.r.t. a move-sequence S from C_{init} to C', if at least one disk in that triad is switched w.r.t. S. A triad is called *cheap*, if the disks in it at C_{init} are consecutive on some peg, and either preserve their order at C', or reverse it; otherwise, it is called *expensive*.

Claim. Any expensive triad is switched w.r.t. any move-sequence from C_{init} to C'.

Proof. Let S be a move-sequence from C_{init} to C' and let S^{-1} be the symmetric move-sequence of S, from C' to C_{init}.

Let τ be an expensive triad in C' w.r.t. C_{init}. We claim that during S, at least one disk in τ performs a switched move. Assume for contradiction that τ is not switched w.r.t. S. It follows that in S, for each disk d in τ, any move of it from some peg X to another peg Y, is preceded by the move from peg X to peg Y of the disk sitting on d. It follows that in S^{-1}, for each disk d in τ, any move of it from peg Y to peg X is followed by the move from peg Y to peg X of the disk on which d was sitting. Recall that at the initial configuration of S^{-1}, C', the three disks sit on each other. This property is preserved during S^{-1}, since whenever some disk in τ moves from some peg Y to another peg X, the two other disks in τ should move from peg Y to peg X immediately afterwards. Since each such triple of moves inverses the order of τ, at the final configuration C_{init} of S^{-1}, the three disks sit on each other in either their initial order at C' or in the reversed order, yielding a contradiction to the choice of τ as expensive w.r.t. C_{init}. □

Denote the set of all configurations of D_n by $\mathcal{C}$ and define $l := \min(l_{C'} | C' \in \mathcal{C})$. We show that for at least half of the configurations, C'', at least $\lfloor \frac{l}{2} \rfloor = \Omega(k)$ switched disks in $B_{1,2}(C'')$ are required to get from C_{init} to C'', which suffices.

For any configuration C', let $e(C')$ denote the number of expensive triads in C' w.r.t. C_{init}. By the above claim, in any move sequence with the initial configuration C_{init} and the final configuration C', there are at least $e(C')$ switched triads. The following claim completes the proof of Lemma 2.

Lemma 3. $|\{C' | C' \in \mathcal{C}$ *and* $e(C') \geq \lfloor \frac{l}{2} \rfloor\}| \geq \lceil \frac{|\mathcal{C}|}{2} \rceil$.

Proof. Denote $\{C'|C' \in \mathcal{C}$ and $e(C') < \lfloor \frac{l}{2} \rfloor\}$ by $S1$ and $\{C'|C' \in \mathcal{C}$ and $e(C') \geq \lfloor \frac{l}{2} \rfloor\}$ by $S2$. Clearly, $S1 \bigcup S2 = \mathcal{C}$. Therefore, showing that $|S1| \leq |S2|$ provides the required result. For this, we construct now an injection, $h : S1 \rightarrow S2$, which will suffice. Let $\hat{C}$ be a configuration in $S1$, s.t. $e(\hat{C}) < \lfloor \frac{l}{2} \rfloor$.

Before describing $h(\hat{C})$ in detail, let us outline the basic tool. In each triad as defined above, we change the disk order, but *not* by just swapping the top-most and the bottom-most disks in it. Note that since each triad consists of three consecutive disks, if such a transformation does not violate the k-relaxed rule *inside* the triad, then it does not cause the entire new configuration to contradict this rule. Besides, since each such transformation rearranges disks inside a triad only, the configuration resulting from any sequence of such transformations defines the *same* set of unordered triads.

It is easy to see that any transformation as above converts any cheap triad in $\hat{C}$ to an expensive one in $h(\hat{C})$. Therefore, $e(h(\hat{C})) \geq l(\hat{C}) - e(\hat{C}) \geq \lfloor \frac{l(\hat{C})}{2} \rfloor \geq \lfloor \frac{l}{2} \rfloor$, that is $h(\hat{C}) \in S2$.

Then, it would remain to show that h is an injection. For this, it would suffice to show that h restricted to each triad is an injection.

Now, we define the disk rearrangement, as above, of each triad τ. The only possible cases, allowed by the k-relaxed rule, are as follows:

– The disks in τ are allowed to be in an arbitrary order. Then, we swap the two top-most disks in τ.
– The two bigger disks should be below the smallest one, in an arbitrary order. Then, we swap the two bigger disks.
– The two smaller disks should be above the biggest one, in an arbitrary order. Then, we swap the two smaller disks.
– The biggest disk should be below the smallest one, while the intermediate disk d is allowed to be at any place. If d is above all or in the middle, we swap it with the disk that it sits on; otherwise (i.e. when it is bottom-most), we move it above the two other disks.

Note that the case, where only the decreasing order of disks in τ is allowed, is impossible, since $\tau \subseteq B_1(m) \cup B_2(m)$.

It is easy to show that in any one of the above cases, the resulting ordered triad τ' allows to restore the *unique* triad τ, whose transformation as above results in τ'. The required result follows. □

□

Proposition 1. *For $n > 3k$, $Avg(n,k) = \Theta(k^2 \cdot 2^{\frac{n}{k}})$.*

Proof. By Theorem 2, it suffices to prove that $Avg(n,k) = \Omega(k^2 \cdot 2^{\frac{n}{k}})$. By Lemma 1, the average number of switched disks in $B_1(n) \cup B_2(n)$ required to get from one configuration of D_n to another, taken over all pairs of configurations, is $\Omega(k)$. Clearly, the number of switched disks in $B_1(n) \cup B_2(n)$ required to get from one configuration of D_n to another is at most $2k$.

It follows that there exist constants $c_1 > 1$ and $c_2 < 2$, s.t. $\frac{1}{c_1}$ out of all the pairs of configurations of D_n require at least $c_2 \cdot k$ switched disks in $B_1(n) \bigcup B_2(n)$, in order to get from one to the other. We denote $\frac{c_2}{2}$ by c_3.

Note that if there are at least $c_2 \cdot k = 2c_3 \cdot k$ switched disks, then at least $\lceil c_3 \cdot k \rceil$ of them belong to the set $B_1(n) \bigcup B_2(n) \setminus \{n-2k+1, \ldots, n-2k+\lfloor c_3 \cdot k \rfloor\}$. It follows that at least $\frac{1}{c_1}$ out of all the pairs of configurations of D_n require at least $\lceil c_3 \cdot k \rceil$ switched disks in $B_1(n) \bigcup B_2(n) \setminus \{n-2k+1, \ldots, n-2k+\lfloor c_3 \cdot k \rfloor\}$ in order to get from one to the other.

We note that any move-sequence of D_n that contains N switched moves of disks in $B_1(n) \bigcup B_2(n) \setminus \{n-2k+1, \ldots, n-2k+\lfloor c_3 \cdot k \rfloor\}$ requires N packet-moves of $Small(n-2k+\lfloor c_3 \cdot k \rfloor\}$. By Theorem 1, it follows that at least $\lceil c_3 \cdot k \rceil \cdot b_{n-3k+\lfloor c_3 \cdot k \rfloor}$ moves are made.

Recall that $b_m \geq m$, for any $m \geq 1$, and if $m = qk + r$, where $0 \leq r < k$, then $b_m = (k+r) \cdot 2^q - k$. We distinguish two cases. If $3k \leq n \leq 4k$, then $\lceil c_3 \cdot k \rceil \cdot b_{n-3k+\lfloor c_3 \cdot k \rfloor} \geq \lceil c_3 \cdot k \rceil \cdot b_{\lfloor c_3 \cdot k \rfloor} \geq \lceil c_3 \cdot k \rceil \cdot \lfloor c_3 \cdot k \rfloor = \Omega(k^2)$. If $n \geq 4k$, then $\lceil c_3 \cdot k \rceil \cdot b_{n-3k+\lfloor c_3 \cdot k \rfloor} = \Omega(k \cdot k \cdot 2^{\frac{n-3k}{k}}) = \Omega(k^2 \cdot 2^{\frac{n}{k}})$. The Proposition follows. □

4 Family of All Optimal Solutions

4.1 Local Definitions and Problem Domain

In this sub-section we describe the *family of all optimal solutions to* BTH_n, and present a closed formula for their number. We use the following result, based on the description of an optimal algorithm α_n (see Section 1 for the definitions and the description).

Corollary 1 ([3]). *The only difference of an arbitrary optimal algorithm for* BTH_n *from* α_n *could be in choosing another optimal algorithms, instead of* β_{n-1}*, for the two included optimal "somehow" packet-moves of* D_{n-1}*.*

Denote by $Somehow\text{-}pt_{X \to Y}(m, R)$ the family of all optimal "somehow" packet-moves of D_m with the initial perfect configuration on peg X, s.t. the final configuration of D_m on another peg Y is R; recall that they are of length b_m each. Let us denote by F_m the set of all such possible final configurations. We introduce also the family $Somehow\text{-}pt^{-1}_{Y \to X}(m, R)$ consisting of the move sequences, symmetric to those in $Somehow\text{-}pt_{X \to Y}(m, R)$. Obviously, it consists of the move-sequences optimal among those with the gathered initial configuration R on peg Y and the perfect final configuration on another peg X.

Theorem 4 is the main result of this section. Note that its item 1 follows from Corollary 1, by the definitions of $Somehow\text{-}pt_{X \to Y}(m, R)$, $Somehow\text{-}pt^{-1}_{Y \to X}$ (m, R) and F_m.

Theorem 4. *1. The family of all optimal perfect-to-perfect packet-moves of* D_n *from peg* A *to peg* C *is*

$$\begin{aligned} Opt_{A \to C}(n) = \{S_1 \circ (n, A, C) \circ S_2 \mid \exists R \in F_{n-1} : \\ S_1 \in Somehow\text{-}pt_{A \to B}(n-1, R), \\ S_2 \in Somehow\text{-}pt^{-1}_{B \to C}(n-1, R)\} . \end{aligned}$$

2. $|Opt_{A \to C}(n)| = (\binom{k+r}{k} - \binom{k+r}{k+1})^{\lceil 2^{\lceil \frac{n-1}{k} \rceil - 2} \rceil - 1}$*, where* $r = n - 1 \bmod k$*.*

The description provided by the first item of this theorem becomes explicit using the description of the family *Somehow-pt*$_{X \to Y}(m, R)$ given in Proposition 2 below.

When studying *Somehow-pt*$_{X \to Y}(m, R)$, we assume $m > k \geq 2$. The case $k = 1$ is disregarded, since it has been proved long ago that there exists a unique optimal solution for the classical problem of Hanoi. We assume that $m > k$, since otherwise it is obvious that the packet-move that moves all disks, one by one, from one peg to another, is the unique optimal solution to the problem.

In the sequel, we would see that the last two blocks of D_m (see Section 2 to recall the division into blocks), $B_{\lceil \frac{m}{k} \rceil}$ and $B_{\lceil \frac{m}{k} \rceil - 1}$, behave differently than the other blocks in the following sense. For any packet-move in *Somehow-pt*$_{X \to Y}$ (m, R), no disk-moves of two different blocks are interleaved, except for, maybe, interleaving disk moves of these two special blocks.

We use the following definitions to distinguish the last two blocks from the other blocks. A configuration is called *almost-well-separated* if it satisfies the condition that at each peg, the disks in any block except for, maybe, the last two blocks, are placed continuously. An almost-well-separated configuration is called *almost-perfect*, if the two following conditions hold: 1. On each peg, the disks in each block are in either the increasing or the decreasing order. 2. If $B_{\lceil \frac{m}{k} \rceil}$ and $B_{\lceil \frac{m}{k} \rceil - 1}$ are gathered on the same peg, then the disks in $B_{\lceil \frac{m}{k} \rceil}$ are in the decreasing order.

Let $m = sk + r_m$, where $0 \leq r_m \leq k - 1$, and let $q = k + r_m$. Clearly, $B_{\lceil \frac{m}{k} \rceil} \cup B_{\lceil \frac{m}{k} \rceil - 1} = D_q$. Denote by R_q the unique almost-perfect configuration of D_q on some peg, where the k bigger disks are in the increasing order and the r_m smaller disks are in the decreasing order. A gathered configuration of D_q is called *perfect-mixed* if the k bigger disks, as well as the r_m smaller disks are in the decreasing order.

In order to investigate *Somehow-pt*$_{X \to Y}(m, R)$, for each R in F_m, we extend our discussion to a more coarse grain family of packet-moves. Denote by $\mathcal{S}_{X \to Y}(m)$ the family of all optimal packet-moves of D_m, whose initial configuration is *almost-perfect* gathered on peg X and whose final configuration is gathered on another peg Y.

Proposition 2. *1. An arbitrary packet-move in* $\mathcal{S}_{source \to target}(m)$ *with an initial almost-perfect configuration* $Init$ *can be described as follows.*

- *If* ($m \leq 2k$*, and hence* $q = m$)
  - *If* ($Init$ *is perfect-mixed*)
    - *perform /* named* $From\text{-}PM(Init, source, target)$ **/:*
      - *move disks in* $Init$ *from source one by one, so that disks in* $B_1(m)$ *go to target and disks in* $B_2(m)$ *to auxiliary;*
      - *move all disks in* $B_2(m)$ *from auxiliary to target one by one;*
  - *Otherwise /** $Init$ *is* R_m **/*
    - *perform /* named* $To\text{-}PM(source, target, R')$ **/, for an arbitrary perfect-mixed configuration* R' *of* D_m*:*
      - *move all disks in* $B_2(m)$ *from source to auxiliary one by one;*

  - *move to target the disks, in the bottom-to-top order of R', one by one from the peg on which it resides;*
- *Otherwise: /* $m > 2k$ */*
  - *perform an arbitrary packet-move in $\mathcal{S}_{source \to auxiliary}(m-k)$ with the initial configuration $Init|_{D_{m-k}}$; let $Temp$ denote its final configuration.*
  - *move disks $[(m-k+1)..m]$ from X to Y one by one;*
  - *perform an arbitrary packet-move in $\mathcal{S}_{auxiliary \to target}(m-k)$ with the initial configuration $Temp$;*

2. *For the case $m \leq 2k$, the unique packet-move in Somehow-pt$_{X \to Y}(m, R)$ is From-PM$(Init, source, target)$.*
3. *For the case $m > 2k$, in any packet-move as in item 1, the contained packet-moves of D_q alternate between From-PM and To-PM types.*
4. *For the case $m > 2k$, an arbitrary packet-move P in Somehow-pt$_{source \to target}$ (m, R) can be described as in item 1, except that for the last packet-move of D_q (that finishing P), the perfect-mixed configuration R' is not arbitrary, but $R|_{D_q}$.*

4.2 Proof of Proposition 2

Fact 5 ([3]) *1. During a move (m, X, Y), all disks in Small(m) are on the spare peg $Z \neq X, Y$.*

2. *If some sequence of moves S begins from a configuration, where disk m and Small(m) are gathered on X, and finishes at a configuration, where disk m and Small(m) are gathered on Y, $X \neq Y$, then it contains two disjoint packet-moves of Small(m): one (from X) before the first move of disk m and another (to Y) after its last move.*

Lemma 4 (proof is omitted). *For any P in $\mathcal{S}_{X \to Y}(m)$, P contains 2^{i-1} disjoint packet-moves of $\bigcup_{j \geq i} B_j(m)$, for each $1 \leq i \leq \lceil \frac{m}{k} \rceil$.*

The optimal length b_m of the packet-moves in $\mathcal{S}_{X \to Y}(m)$, together with Lemma 4 yield:

Corollary 2. *For any P in $\mathcal{S}_{X \to Y}(m)$ and any $1 \leq i \leq \lceil \frac{m}{k} \rceil$, each disk in $B_i(m)$ moves exactly 2^{i-1} times, during P.*

The following lemma is the central statement for proving Proposition 2.

Lemma 5 (proof is omitted). *For any packet-move in $\mathcal{S}_{X \to Y}(m)$, every configuration reached during its execution is almost-perfect.*

An easy consequence is that, in $\mathcal{S}_{X \to Y}(m)$, the final configuration of any packet-move of $\bigcup_{j \geq i} B_j(m)$, for each $1 \leq i \leq \lceil \frac{m}{k} \rceil$, is almost-perfect. Therefore, the third item for the case $m > 2k$ in the description of $\mathcal{S}_{X \to Y}(m)$ is well-defined.

Corollary 3 (proof is omitted). *Item 1 of Proposition 2 is valid for the case $m > 2k$.*

By Lemma 5 and Corollary 3, it follows that packet-moves in $\mathcal{S}_{X\to Y}(m)$ with the same initial almost-perfect configuration are not very different one from another. Such a difference is reflected only in interleaving moves of disks of the two last blocks of D_m, $B_{\lceil \frac{m}{k} \rceil}$ and $B_{\lceil \frac{m}{k} \rceil -1}$, in packet-moves of their union. In the sequel, we investigate the possibilities of such interleaving of moves.

Recall that $B_{\lceil \frac{m}{k} \rceil} \cup B_{\lceil \frac{m}{k} \rceil -1}$ is D_q, where $m = sk + r_m$, $0 \le r_m \le k-1$, and $q = k + r_m$.

By Lemma 4 and the optimality of packet-moves in $\mathcal{S}_{X\to Y}(m)$, any packet-move P in $\mathcal{S}_{X\to Y}(m)$ contains $2^{\lceil \frac{m}{k} \rceil -2}$ disjoint packet-moves of D_q, of length b_q each. (Note that if all of these packet-moves are fixed as β_q, the resulting packet-move is β_m.) Hence, the study of $\mathcal{S}_{X\to Y}(m)$ is reduced to the study of the family of all optimal solutions to the following problem, and its cardinality.

Problem 1. Let $q = k + r_m$, s.t. $1 \le r_m \le k$, $l \ge 0$. Describe an optimal packet-move P of D_q with an initial almost-perfect configuration, which is a composition of 2^l disjoint packet-moves of D_q, of length b_q each.

Lemma 6 (proof is omitted). *For any perfect-mixed configuration R of D_q, holds:*

1. *$From\text{-}PM(R, source, target)$ is the unique optimal packet-move of D_q with the initial configuration R. Its final configuration is R_q.*
2. *$To\text{-}PM(source, target, R)$ is the unique optimal packet-move of D_q with the initial configuration R_q and the final configuration R.*

Validity of items 1 and 2 of Proposition 2 follows.

Denote the family of all optimal solutions to Problem 1 by $OPT(q, 2^l)$.

Lemma 7 (proof is omitted). *For any member in $OPT(q, 2^l)$, if its initial configuration is perfect-mixed (resp., R_q), then:*

1. *The final configuration of any odd-numbered (resp., even-numbered) packet-move contained in it is R_q.*
2. *The final configuration of any even-numbered (resp., odd-numbered) packet-move contained in it is a perfect-mixed configuration of D_q.*

Validity of items 3 and 4 of Proposition 2 follows.

4.3 Counting the Optimal Solutions to BTH_n

In this sub-section, we prove Item 2 of Theorem 4.

By Item 1 of Theorem 4 together with Item 2 of Proposition 2, it follows that in the case $n \le 2k+1$ holds $|Opt_{A\to C}(n)| = 1$, which corresponds to Item 2 of Theorem 4. Thus, we may henceforth assume that $n > 2k+1$.

By the results of the previous sub-section, the members of $OPT(q, 2^l)$ correspond bijectively to the sequences of 2^{l-1} perfect-mixed configurations of D_q. Hence, the members of $Somehow\text{-}pt_{A\to B}(n-1, R)$ and those of $Somehow\text{-}pt^{-1}_{B\to C}$ $(n-1, R)$ correspond bijectively to the sequences of $\lceil 2^{\lceil \frac{n-1}{k} \rceil -3} \rceil - 1$ perfect-mixed configurations of D_q. Therefore, by Theorem 4(1), the members of $Opt_{A\to C}(n)$ correspond bijectively to the sequences of $2(\lceil 2^{\lceil \frac{n-1}{k} \rceil -3} \rceil - 1) + 1 = \lceil 2^{\lceil \frac{n-1}{k} \rceil -2} \rceil - 1$ perfect-mixed configurations of D_q.

Our next goal is to describe the family of all perfect-mixed configurations of D_q and to prove that their number, denoted by $f(k,r)$, equals $\binom{k+r}{k} - \binom{k+r}{k+1}$. This equality and the above correspondence will yield item 2 of Theorem 4.

Consider some perfect-mixed configuration, denoted by M. We denote by M_i the set of disks in $Small(q)$, higher than $q-i+1$ at M but lower than $q-i$, for each $1 \le i \le k-1$, and denote the set of disks in $Small(q)$, higher than $q-k+1$ at M by M_k.

The fact that $Small(q-i)$ is higher than $q-i$ at any perfect-mixed configuration of D_q, M, together with the fact that the disks in each one of $Small(q)$ and $B_1(q)$ are in decreasing order implies that $M_i \le i$, for each $1 \le i \le k$, and that $\bigcup_{i=1}^{k} M_i = Small(q)$. The equality in question holds by the case $c = 0$ of the following proposition.

Proposition 3. *Let t, n and c be three non-negative integers s.t. $n \le t + c$. Denote the number of non-negative integer solutions that satisfy the two following conditions, as a function of n, t and c by $\phi(t,n,c)$.*
1. $\sum_{i=0}^{t} x_i = n$.
2. For each $0 \le j \le n-c : \sum_{i=0}^{j} x_i \le j + c$.
Then, $\phi(t,n,c) = \binom{n+t}{t} - \binom{n+t}{t+c+1}$.

Proposition 3 is proved by a complete induction on t, based on the fact that for all natural values of n, t and c s.t. $n \le t + c$, holds $\phi(n,t,c) = \sum_{i=0}^{c} \phi(t-1, n-i, c-i+1)$ and using the Pascal Triangle equality $\sum_{i=0}^{c} \binom{i+x}{x} = \binom{c+x+1}{x+1}$.

References

1. Beneditkis, S. and Safro, I.: Generalizations of the Tower of Hanoi Problem. Final Project Report, supervised by D. Berend, Dept. of Mathematics and Computer Science, Ben-Gurion University (1998)
2. Chen, X., Tian, B., and Wang, L.: Santa Claus' Towers of Hanoi. Manuscript (2005)
3. Dinitz, Y. and Solomon, S.: Optimal Algorithms for Tower of Hanoi Problems with Relaxed Placement Rules. To appear in Proc. of the 17th International Symp. on Algorithms and Computation (ISAAC'06), Kolkata, India (December 2006)
4. Poole, D.: The Bottleneck Towers of Hanoi Problem. J. of Recreational Math. **24** 3 (1992) 203–207.
5. Szegedy, M.: In How Many Steps the k Peg Version of the Towers of Hanoi Game Can Be Solved? Symposium on Theoretical Aspects of Computer Science **1563** (1999) 356
6. Wood, D.: The Towers of Brahma and Hanoi Revisited. J. of Recreational Math. **14** 1 (1981-1982) 17-24