The worst parallel Hanoi graphs

Ernst L. Leiss a, *, Isaac Mackey b

a University of Houston, United States
b University of Virginia, United States

A R T I C L E   I N F O

Article history:
Received 26 October 2016
Received in revised form 14 September 2017
Accepted 17 September 2017
Available online 21 September 2017
Communicated by A. Fink

Keywords:
Towers of Hanoi
Parallel moves
Hanoi graphs
Maximal number of moves

A B S T R A C T

The “Towers of Hanoi” is a problem that has been extensively studied and frequently generalized. In particular, it has been generalized to be played on arbitrary directed graphs and using parallel moves of two types. We ask what is the largest number of parallel moves, in either of the two models, that is required to move n disks from the starting node to the destination node. Not all directed graphs allow solving this problem; we will call those graphs that do Hanoi graphs. In previous work, we settled the question of what are the worst sequential Hanoi graphs, that is, those graphs that require the largest number of sequential moves. We also demonstrated that the characterization of sequential Hanoi graphs carries over the parallel Hanoi graphs. Here, we determine the worst Hanoi graphs provided parallel moves are allowed. It turns out that for one of the two models of parallel moves, the worst graphs are quite different from the worst sequential graphs, while in the other model of parallelism, there is little difference with the sequential situation.

© 2017 Elsevier B.V. All rights reserved.

1. Background

The Towers of Hanoi game is a classical problem that is frequently (ab)used in data structures and algorithms classes to illustrate the power of recursion. For a comprehensive review of the literature on this problem, see [12] and [5]. In [6], the game was generalized to be played on graphs; specifically, there is a finite directed graph \( G = (V, E) \) with two distinguished nodes \( S \) and \( D \), there are \( n \) disks of \( n \) different sizes on node \( S \) such that no larger disk may lie on top of a smaller disk, and the objective is to move the \( n \) disks from \( S \) to \( D \) subject to the following rules:

1. Only one disk may be moved at a time and only along an edge in \( G \).
2. A disk is always placed on top of all the disks on the node where it is moved and no larger disk may ever be placed on top of a smaller disk.

If the problem can be solved for a given graph \( G \) for all \( n ≥ 1 \), the Hanoi problem is called solvable. If for a given graph the associated Hanoi problem is not solvable, the Hanoi problem is called finite. For finite Hanoi problems see [7] and [2].

There is a rather elegant characterization of all those graphs with solvable Hanoi problems.

Theorem 0 ([6]). A Hanoi problem with graph \( G = (V, E) \) is solvable if and only if there exist three different nodes \( v_1, v_2, \) and \( v_3 \) in \( V \) such that:

* Support under NSF grants IIS-1359199 (REU Site Program) and DGE-1433817 is acknowledged.
* Corresponding author.
E-mail addresses: coscel@cs.uh.edu (E.L. Leiss), iam6ep@virginia.edu (I. Mackey).

https://doi.org/10.1016/j.tcs.2017.09.012
0304-3975/© 2017 Elsevier B.V. All rights reserved.
1. There exists a path in $G$ from $S$ to one of the three nodes $v_1, v_2, v_3$.
2. There exists a path in $G$ from one of the three nodes $v_1, v_2, v_3$ to $D$.
3. There exist paths from $v_1$ to $v_2$, from $v_2$ to $v_3$, and from $v_3$ to $v_1$ in $G$.

Thus, whether a sequential Hanoi problem is solvable depends exclusively on the graph on which it is played and on the choice of nodes $S$ and $D$. We will call a directed graph $G = (V, E)$ with distinguished nodes $S$ and $D$ in $V$ a sequential Hanoi graph $(G, S, D)$ if it satisfies the conditions of Theorem 0.

The transition from the original problem (on three pegs) to one involving arbitrary directed graphs is a major generalization. Here, we examine another generalization, namely one that involves parallelism. More specifically, we want to retain the rules outlined above but permit moving more than one disk at a time. Below, we describe in detail the two models that arise naturally in this context. Informally, we may think of agents that move disks from one node (peg) to another along a directed edge. In the sequential problem, there is exactly one agent. In the parallel models, we assume there are several agents which can move disks. These movements for each agent must obey the standard rules for Hanoi games but two or more agents might move disks at the same time. This would correspond to processors carrying out computations in parallel [see [8]]. The two models we assume here differ in the way agents may interact; both models arise naturally when considering parallelism, but in one we assume that the two (or more) agents must operate entirely independently of each other (the one-step model) while in the other model an agent may move a disk to a node from which another agent removed a disk (provided all the other requirements are satisfied).

Specifically, in [10], we introduced the two models of parallel moves. In the one-step model, a move consists of a single step; therefore two moves may be done in parallel under this model only if both destination nodes of the disks are not occupied by smaller disks. If there are more than two agents, any two of them must satisfy these requirements. Thus, given two sequential moves

$$v_1 - d_1 \rightarrow v_2 \quad \text{and} \quad v_3 - d_2 \rightarrow v_4,$$

they can be done in parallel under the one-step model iff

(a) $d_1 \neq d_2$,
(b) $d_1$ is the top disk on $v_1$,
(c) $d_2$ is the top disk on $v_3$,
(d) $v_2$ does not contain a disk smaller than $d_1$,
(e) $v_4$ does not contain a disk smaller than $d_2$,
(f) $v_2 \neq v_3$,
(g) $v_2 \neq v_4$, and
(h) $v_4 \neq v_1$.

Note that the first three requirements imply the necessary condition that $v_1 \neq v_3$. Therefore, the nodes $v_1, v_2, v_3, v_4$ must be four distinct nodes.

In the two-step model, a move consists of the lifting of the disk from the node on which it resides and then the placing of that disk on the destination node. It follows that we may have the following situation: disk $d_1$ resides on node $v_1$ and disk $d_2$ resides on node $v_2$; $d_1$ is to move from $v_1$ to $v_2$ and $d_2$ is to move from $v_2$ to node $v_3$. Assuming all other requirements are satisfied, in the two-step model these two moves may be carried out in parallel, since first both disks are lifted from their respective nodes thereby removing these disks, and in the second step the two disks are moved to their respective destination nodes. It should be clear that in the one-step model, this would not be possible. Therefore, not all of the eight conditions for the one-step model apply to the two-step model. Thus, two sequential moves

$$v_1 - d_1 \rightarrow v_2 \quad \text{and} \quad v_3 - d_2 \rightarrow v_4$$

can be done in parallel under the two-step model iff

(a) $d_1 \neq d_2$,
(b) $d_1$ is the top disk on $v_1$,
(c) $d_2$ is the top disk on $v_3$,
(d) if $v_2 \neq v_3$, then $v_2$ does not contain a disk smaller than $d_1$, if $v_2 = v_3$, then $v_2$ does not contain a disk smaller than $d_1$ once $d_2$ is removed,
(e) if $v_4 \neq v_1$, then $v_4$ does not contain a disk smaller than $d_2$, if $v_4 = v_1$, then $v_4$ does not contain a disk smaller than $d_2$ once $d_1$ is removed, and
(f) $v_2 \neq v_4$.

An important move that can be carried out in the two-step model but not in the one-step model is the swap move. Briefly, if two nodes $i$ and $j$ contain $p$ and $q$ as their top disks respectively and there are the edges $(i, j)$ and $(j, i)$ in the graph, then the disks $p$ and $q$ can be interchanged in one move in the two-step model (provided this does not place a larger
disk on top of a smaller one). This swap move is not possible in the one-step model. The swap move turns out to be very important in moving disks efficiently in the two-step model.

It follows (see [10]) that every sequential move can be viewed as a one-step parallel move, and every one-step parallel move can be viewed as a two-step parallel move, but the converse does not apply: Two-step parallel moves result in the fewest number of moves, followed by one-step parallel moves, with the largest number of moves required when moves must be sequential. In particular, we showed in [10] that there exist parallel moves that cannot be replaced by sequential moves. This raised the question whether the characterization of sequential Hanoi graphs (Theorem 0) carries over to parallel Hanoi graphs, of either type. In [10], we showed that any parallel Hanoi graph (under either model) is also a sequential Hanoi graph. Since the converse is obvious, the set of sequential Hanoi graphs is identical to the set of parallel Hanoi graphs.

In both sequential and parallel models, one might be interested in graphs that are “good” (require few moves to move \( n \) disks from the starting node to the destination node) or “bad” (requiring many moves). Thus, it is natural to ask what are the best or the worst graphs. It turns out that the first question, what are the best graphs, is a long-standing open problem, even for sequential graphs: While it is generally believed that the Frame–Stewart conjecture gives an optimal solution for the complete graph on \( m \) nodes, to date nobody has been able to prove this, even for \( m = 4 \). For this reason, it is unlikely that one will be able to provide a satisfactory answer to this question for the parallel models. However, the question what are the worst sequential graphs can be answered quite comprehensively:

**Theorem 1** ([9]). For every \( m \geq 3 \) and for any Hanoi graph \( (G, S, D) \) with \( m \) nodes, there exists an integer \( n_{m,0} \geq 1 \) such that for all \( n \geq n_{m,0} \), the number of moves required to move \( n \) disks from \( S \) to \( D \) in \( G \) is no greater than \( w_m(n) = 3^n + n(m - 3) - 1 \). This upper bound is attained by the Hanoi graph \( (W_m, 1, m) \) where

\[
W_m = \{(1, \ldots, m), \{(1, 2), (2, 1), (2, 3), (3, 2)\} \cup \{(i, i + 1) \mid i = 3, \ldots, m - 1\}\}.
\]

The graph \( W_m \) can be graphically represented as \( S = 1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \rightarrow \ldots \rightarrow m = D \); here the nodes 1, 2, and 3 form a graph that requires precisely \( 3^0 - 1 \) sequential moves while the remaining nodes 4, \ldots, \( m \) form a tail that is attached to node 3.

In [10], it was shown that the graph \( W_3 \) (no tail) allows moving \( n \) disks from node \( S = 1 \) to node \( D = 3 \) in no more than \( O(3^{n/2}) \) two-step parallel moves. Specifically, if we denote the number of two-step parallel moves to move \( n \) disks in \( W_3 \) by \( T(n) \), then we have \( T(2k) = 2 \cdot 3^k - 2 \) and \( T(2k - 1) = 4 \cdot 3^{k-1} - 2 \). Note that this is based on the algorithm given in that paper; therefore strictly speaking \( T(n) \) is an upper bound. No claim is made that this algorithm is optimal although we conjecture that it is.

For later use, we define three classes of directed graphs, \( C_m, W_m, \) and \( Z_m, m \geq 3 \). All three have \( m \) nodes, and we always assume in the following that \( S = 1 \) and \( D = m \). \( C_m \) is simply the cyclic graph on \( m \) nodes,

\[
C_m = \{(1, \ldots, m), \{(s, s + 1) \mid s = 1, \ldots, m - 1\} \cup \{(m, 1)\}\}.
\]

Informally, \( W_m \) is the graph \( W_3 \) on nodes 1, 2, and 3, with a tail from node 3 to node \( m \); \( Z_m \) is the cyclic graph \( C_3 \) on the three nodes 1, 2, and 3, with a tail from node 3 to node \( m \). Formally, we have

\[
W_m = \{(1, \ldots, m), \{(1, 2), (2, 1), (2, 3), (3, 2)\} \cup \{(i, i + 1) \mid i = 3, \ldots, m - 1\}\}.
\]

\[
Z_m = \{(1, \ldots, m), \{(1, 2), (2, 3), (3, 1)\} \cup \{(i, i + 1) \mid i = 3, \ldots, m - 1\}\}.
\]

2. The worst parallel Hanoi graphs

Motivated by the result for sequential moves (Theorem 1), we study the worst parallel Hanoi graphs. In other words, we determine those graphs for which, under either of the two parallel models, the number of moves required to move \( n \) disks from node \( S \) to node \( D \) is maximal among all graphs with \( m \) nodes. Given the result of [10] that every parallel Hanoi graph is a sequential Hanoi graph, we can recycle a good deal of the results from [9]. For completeness, we reproduce the salient points in the next few paragraphs. For more details, see the original paper. We note that the following arguments do not depend on the type of moves involved. Therefore they apply to either type of parallel moves even though the original paper focused on sequential moves.

If \( (G, S, D) \) is a parallel Hanoi graph, then the provisions of Theorem 0 must be satisfied (by [10]). Specifically, there exist three different nodes \( v_1, v_2, \) and \( v_3 \) in \( G \) such that:

1. There exists a path in \( G \) from \( S \) to one of the three nodes \( v_1, v_2, v_3 \).
2. There exists a path in \( G \) from one of the three nodes \( v_1, v_2, v_3 \) to \( D \).
3. There exist paths from \( v_1 \) to \( v_2 \), from \( v_2 \) to \( v_3 \), and from \( v_3 \) to \( v_1 \) in \( G \).
For any Hanoi graph, there may be several triples \((v_1, v_2, v_3)\) satisfying these provisions. Given any such triple \((v_1, v_2, v_3)\), consider the set of all nodes that are involved in the three paths, from \(v_1\) to \(v_2\), from \(v_2\) to \(v_3\), and from \(v_3\) to \(v_1\). Consider now a smallest graph \(H = (U, F)\) that contains nodes \(v_1\), \(v_2\), and \(v_3\) as well as any nodes involved in the paths from \(v_1\) to \(v_2\), from \(v_2\) to \(v_3\), and from \(v_3\) to \(v_1\), with the edges \(F\) being precisely the edges involved in these three paths.

If \(U = \{v_1, v_2, v_3\}\), then either the nodes form a graph isomorphic to \(W_3\) or \(H\) contains \(C_3\) as a subgraph. In this case the incoming chain corresponds to the first path from \(S\) to one of the three nodes and the outgoing chain to the second path from one of the three nodes to \(D\).

Suppose now that there is at least one other node in \(U\). By assumption, this node \(u\) is involved in a least one of the three paths (from \(v_1\) to \(v_2\), from \(v_2\) to \(v_3\), or from \(v_3\) to \(v_1\)).

If the node \(u\) is involved in exactly one path, say from \(v_1\) to \(v_2\), then we distinguish two cases: Either the other node \(v_3\) is involved in the path from \(v_1\) to \(v_2\), or it is not involved. If \(v_3\) is involved in that path, then it is either between \(v_1\) and \(u\), or it is between \(u\) and \(v_2\). In the first subcase there is now a new, strictly smaller graph that does not contain \(v_1\) but contains a \(C_3\) as a minor: it consists of the nodes \(v_3\), \(u\), and \(v_2\), plus all nodes along the paths from \(v_3\) to \(u\), from \(u\) to \(v_2\), and from \(v_2\) to \(v_3\). This is a contradiction to the assumed minimality of \(H\). In the second subcase, it follows that \(H\) contains a \(C_4\) as a minor.

If \(u\) is involved in exactly two paths, in that from \(v_1\) to \(v_2\) and either that from \(v_2\) to \(v_3\), or that from \(v_3\) to \(v_1\), then either \(v_1\), \(u\), and \(v_3\) induce a \(C_3\) minor in \(H\), or \(v_3\), \(u\), and \(v_2\) induce a \(C_3\) minor in \(H\). Again, this is in contradiction to the assumed minimality of \(H\).

Finally, if \(u\) is involved in all three paths, any three of the four nodes \((u, v_1, v_2, v_3)\) will induce a \(W_3\). This is again a contradiction, since the minimal \(H\) contains at least four nodes.

We conclude that there are exactly three possible outcomes:

1. \(H = W_3\).
2. \(H = C_3\).
3. \(H = C_r\) for \(r > 3\).

We now distinguish the case of the one-step model and the case of the two-step model. It turns out that depending on the type of parallelism assumed, the results are quite different. This will be shown in the following subsections.

2.1. One-step parallel moves

It is a direct consequence of the definition of one-step moves that any one-step moves on \(C_3\) necessarily are all sequential moves. This is because in order to do two sequential moves in parallel in this model, four distinct nodes must be available, which is not the case in \(C_3\). Consequently, the number of one-step parallel moves required to move \(n\) disks from \(S = 1\) to \(D = 3\) on \(C_3\) is equal to the number of sequential moves.

In order to show the main result of this section, determining the worst graphs under the one-step model, we will first give the number of moves required to move \(n\) disks from \(S = 1\) to \(D = 3\) on \(C_3\). Then we show that asymptotically the number of moves on any \(C_r\) for \(s > 3\) is smaller than this number. We also show that the number of moves on \(W_3\) is asymptotically larger than the number of moves on \(C_3\). Then we tie everything together to obtain the result that \(W_m\) is asymptotically the worst Hanoi graph on \(m\) nodes, under the one-step parallel model.

We note that the number of sequential moves for the graph \(C_3\) has been precisely determined by Atkinson [1] (see also [4]) and the number of sequential moves for \(C_r\) for \(s > 3\) has been completely determined in [3]. Thus, some of the results outlined below follow directly from these two papers; the discussion below is provided only to make the exposition self-contained. Note however that we are dealing with parallel moves, so the results of these papers have to be adapted since they were derived for sequential moves only.

Let \(c_{3,2}(n)\) denote the number of sequential moves required to move \(n\) disks from node 1 to node 3 in \(C_3\). In other words, this is the number of moves of an optimal sequential algorithm. Following [11], one derives that \(c_{3,2}(1) = 2\), \(c_{3,2}(2) = 7\), and for \(n > 2\), \(c_{3,2}(n) = 2c_{3,2}(n - 1) + 2c_{3,2}(n - 2) + 3\). Denote the quantity \(1 + 3^{1/2}\) by \(\alpha\). It follows that \(2\alpha + 2 = \alpha^2\). Then one can show that \(c_{3,2}(n) > \alpha^n\) for all \(n > 2\). This is evidently true for \(n = 3, 4\) and follows by induction on \(n\):

\[
c_{3,2}(n) = 2c_{3,2}(n - 1) + 2c_{3,2}(n - 2) + 3 > 2\alpha^{n-1} + 2\alpha^{n-2} + 3 = \alpha^n + 3 > \alpha^n.
\]

Note that \(\alpha\) is approximately 2.732.

Now consider the cyclic graph \(C_s\) on \(s\) nodes, for \(s > 3\). We first show that on \(C_4\), we need asymptotically fewer moves than on \(C_3\). Note that for this purpose, we do not need an optimal algorithm. It suffices to specify some algorithm that requires fewer moves than \(c_{3,2}(n)\). Here is such an algorithm consisting of two functions, \(M_{4,3}(n)\) and \(M_{4,2}(n)\), where the second subscript indicates how many hops the stack of disks is to be moved. We use the notation \(i : u \rightarrow v\) to describe the move of disk \(i\) from node \(u\) to node \(v\). We use the symbol \(\parallel\) to indicate that moves can be carried out in parallel. It is apparent that we can carry out moves in parallel on \(C_4\), something that was not possible on \(C_3\) in the one-step model.
Theorem 2. The graph with \( m \) nodes that requires asymptotically the largest number of one-step parallel moves to move \( n \) disks from node \( S = 1 \) to node \( D = m \) is the graph \( W_m \). The number of moves required by this graph is \( 3^n + m - 4 \).
Algorithm. If \( n = 1 \):

1. \( 1 : S \rightarrow 2 \).
2. \( 1 : 2 \rightarrow D \).

If \( n = 2 \):

1. \( 1 : S \rightarrow 2 \).
2. \( 1 : 2 \rightarrow D \) \( \parallel 2 : S \rightarrow 2 \).
3. \( \text{Swap}(1, 2) \).
4. \( 1 : 2 \rightarrow D \).

If \( n \geq 3 \):

1. Move the top-most \( n - 2 \) disks recursively from \( S \) to \( D \).
2. \( n - 1 : S \rightarrow 2 \) [in parallel with the last move of (1)].
3. \( \text{Swap}(n, n - 1) \).
4. \( n - 1 : S \rightarrow 2 \).
5. Move the \( n - 2 \) disks recursively from \( D \) to \( S \).
6. \( n - 1 : 2 \rightarrow D \).
7. \( \text{Swap}(n - 1, n) \).
8. \( n - 1 : 2 \rightarrow D \) [in parallel with the first move of (9)].
9. Move the \( n - 2 \) disks recursively from \( S \) to \( D \).

Here \( \text{Swap}(i, j) \) indicates that disks \( i \) and \( j \) are interchanged. This is of course only possible if there is an edge from the node holding disk \( i \) to the node holding disk \( j \), and also an edge from the node holding disk \( j \) to the node holding disk \( i \). Moreover, all other requirements about moves in Hanoi games must be satisfied, in particular, that no larger disk may ever be placed on top of a smaller disk. It is easily verified that these conditions are satisfied when we use the \( \text{Swap} \) operation in this algorithm. The importance of the swap move becomes apparent when one determines the computational complexity of the algorithm.

For the number of moves \( T(n) \) of this algorithm, we obtain the following recurrence relation:

\[
T(1) = 2 \\
T(2) = 4 \\
T(n) = 3T(n - 2) + 4 \quad \text{for } n > 2.
\]

One can easily verify that for even \( n = 2k \), \( T(2k) = 2 \cdot 3^k - 2 \); for odd \( n = 2k - 1 \), \( T(2k - 1) = 4 \cdot 3^{k-1} - 2 \). Consequently, we can move \( n \) disks using two-step parallel moves in no more than \( O(3^{n/2}) \) moves. Note that in contrast to the sequential situation where we know that \( 3^n - 1 \) moves are required, in the parallel case we only have an upper bound: It is conceivable (although not very likely) that there is a better parallel algorithm. In fact, we conjecture that this algorithm is optimal under the two-step model for this graph.

In order to show the main result of this section, determining the worst graphs under the two-step model, we proceed as follows in the remainder of this subsection: We will show that \( C_3 \) requires asymptotically strictly more two-step parallel moves than any \( C_s \) for \( s > 3 \). Furthermore, \( W_3 \) requires strictly fewer moves than \( C_3 \). This implies that \( Z_m \) requires asymptotically strictly more moves than \( W_m \). Finally, we conclude that \( Z_m \) is asymptotically the worst Hanoi graph under the two-step parallel model.

We first derive an optimal algorithm to move \( n \) disks from \( S = 1 \) to \( D = 3 \) in \( C_3 \) under the two-step model. We observe that in order to be able to do \( n : 1 \rightarrow 2 \) (which must be carried out at some point in any algorithm since eventually disk \( n \) must come to rest on node 3), all other disks must necessarily be on node 3. Then, for the move \( n : 2 \rightarrow 3 \) to be possible, all other disks must necessarily be on node 1. (Again, this move must occur at some point in any algorithm.) Finally, so that all disks end up on node 3, all disks 1 through \( n - 1 \) must be moved from node 1 to node 3. Since all these movements are forced, this provides a lower bound on the number of moves. Since the algorithm below attains precisely this lower bound, it follows that it is optimal.

\[ N_{3,2}(n) : \]

\[
n = 1: \ 1 : 1 \rightarrow 2. 1 : 2 \rightarrow 3. \\
n = 2: \ 1 : 1 \rightarrow 2. 1 : 2 \rightarrow 3 \parallel 2 : 1 \rightarrow 2. 1 : 3 \rightarrow 1 \parallel 2 : 2 \rightarrow 3. 1 : 1 \rightarrow 2. 1 : 2 \rightarrow 3. \\
n > 2: \ N_{3,2}(n - 1), \ n : 1 \rightarrow 2. N_{3,1}(n - 1), \ n : 2 \rightarrow 3. N_{3,2}(n - 1). \]
Note that the move \( n : 1 \rightarrow 2 \) can be done in parallel with the last move of the preceding recursive call, as can be the move \( n : 2 \rightarrow 3 \).

\[
N_{3,1}(n):
\]
\[
n = 1: \quad 1 : 1 \rightarrow 2.
\]
\[
n = 2: \quad 1 : 1 \rightarrow 2, \ 1 : 2 \rightarrow 3 || 2 : 1 \rightarrow 2, \ 1 : 3 \rightarrow 1, \ 1 : 1 \rightarrow 2.
\]
\[
n > 2: \quad N_{3,2}(n-1). \ n : 1 \rightarrow 2. \ N_{3,2}(n-1).
\]

Again, the move \( n : 1 \rightarrow 2 \) can be done in parallel with the last move of the preceding recursive call.

We therefore obtain the following recurrence relations for the number of moves:

\[
d_{3,2}(1) = 2, \quad d_{3,2}(2) = 5, \quad \text{and} \quad d_{3,2}(n) = 2d_{3,2}(n-1) + d_{3,1}(n-1).
\]
\[
d_{3,1}(1) = 1, \quad d_{3,1}(2) = 4, \quad \text{and} \quad d_{3,1}(n) = 2d_{3,2}(n-1).
\]

This implies that

\[
d_{3,2}(1) = 2, \quad d_{3,2}(2) = 5, \quad \text{and} \quad d_{3,2}(n) = 2d_{3,2}(n-1) + 2d_{3,2}(n-2).
\]

We now show that \( d_{3,2}(n) > \alpha^n/2 \) for all \( n \geq 1 \), where \( \alpha \) as before is \( 1 + 3^{1/2} \) and satisfies \( \alpha^2 = 2\alpha + 2 \). This is evidently true for \( n = 1, 2 \); for \( n > 2 \), we have

\[
d_{3,2}(n) = 2d_{3,2}(n-1) + 2d_{3,2}(n-2) > 2[\alpha^{n-1}/2 + \alpha^{n-2}/2] = \alpha^n/2.
\]

Moreover, it also follows that \( d_{3,2}(n) < \alpha^n \): This is true for \( n = 1, 2 \), and follows in general inductively:

\[
d_{3,2}(n) = 2d_{3,2}(n-1) + 2d_{3,2}(n-2) < 2\alpha^{n-1} + 2\alpha^{n-2} = \alpha^n.
\]

Consider now the graph \( C_4 \). It suffices to show that there is an algorithm that allows moving \( n \) disks from 1 to 3 in fewer than \( d_{3,2}(n) \) moves, for all \( n \geq n_1 \), for some \( n_1 \). There is no need to have an optimal algorithm for this. Instead of giving the algorithm explicitly, we give only the recurrence relations from which the details of the algorithm can easily be inferred.

We denote by \( d_{4,i}(n) \) the number of moves this method uses to move \( n \) disks by \( i \) hops \( (i = 2, 3) \), from the starting node 1 to the destination node 4. We have:

\[
d_{4,3}(1) = 3; \quad d_{4,3}(2) = 7; \quad d_{4,3}(3) = 11; \quad d_{4,3}(4) = 15; \quad d_{4,3}(n) = d_{4,3}(n-1) + 2d_{4,2}(n-1)
\]

and

\[
d_{4,2}(1) = 2; \quad d_{4,2}(2) = 6; \quad d_{4,2}(3) = 10; \quad d_{4,2}(4) = 14; \quad d_{4,2}(n) = 2d_{4,2}(n-1).
\]

We remind the reader that more than two disks may be moved. To illustrate this, consider the case of \( d_{4,3}(3) = 11 \). Here we have:

Nodes

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>123</td>
<td>23</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>23</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>123</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

From the recurrence relations for \( d_{4,3}(n) \) and \( d_{4,2}(n) \), we obtain

\[
d_{4,3}(1) = 3; \quad d_{4,3}(2) = 7; \quad d_{4,3}(3) = 11; \quad d_{4,3}(4) = 15; \quad d_{4,3}(n) = d_{4,3}(n-1) + 4d_{4,3}(n-2).
\]

We now claim that \( d_{3,2}(n) > d_{4,3}(n) \) for all \( n \geq 3 \). To this end, we first show that \( d_{4,3}(n) \leq \beta^n \) for all \( n \geq 3 \) where \( \beta \) is again the quantity \((17^{1/2} + 1)/2\) which satisfies \( \beta^2 = \beta + 4 \). This is evidently true for \( n = 3, 4 \) and follows generally by induction:

\[
d_{4,3}(n) = d_{4,3}(n-1) + 4d_{4,3}(n-2) \leq \beta^n - 1 + 4\beta^{n-2} = \beta^n.
\]
Now we have
\[ d_{4,3}(n) \leq \beta^n < \alpha^n / 2 < d_{3,2}(n) \text{ for all } n \geq 11. \]

Thus, \( C_3 \) requires more moves than \( C_4 \). Using the same simulation trick as before, we derive from this that \( C_s \) for any \( s > 3 \) requires fewer moves than \( C_3 \), since we can always simulate \( C_s \) on \( C_4 \) and the gap between \( \beta^n \) and \( \alpha^n / 2 \) in the above inequality always allows us to subsume the constant factor that arises from the simulation. Finally, we observe that the tails of \( Z_m \) and \( W_m \) contribute very little towards the final total of moves, namely exactly \( m - 3 \). As before in the case of one-step parallel moves, it is clear that only the last disk, disk 1, must be moves along the tail in a way that cannot be done in parallel with the other moves. Therefore, we can summarize the results for the two-step parallel moves:

**Theorem 3.** The graph with \( m \) nodes that requires asymptotically the largest number of two-step parallel moves to move \( n \) disks from node \( S = 1 \) to node \( D = m \) is the graph \( Z_m \). The number of moves required by this graph is given by \( d_{3,2}(n) + m - 3 \). Moreover, letting \( \alpha = 1 + 3^{1/2} \), we have \( \alpha^n / 2 < d_{3,2}(n) < \alpha^n \), for all \( n \geq 1 \).

Since \( \alpha \) is approximately 2.732, it follows that the two-step model requires asymptotically for its worst graphs still fewer moves than the one-step model for its worst graphs. Recall from Theorem 2, that the required number of moves was \( O(3^n) \) when keeping the number \( m \) of nodes constant, which incidentally is the same as for sequential moves, while for two-step parallel moves we need no more than \( O(\alpha^n) \).

3. Conclusion

We have studied parallelism in the context of the Towers of Hanoi game, generalized to be played on directed graphs. In analogy to the sequential situation which was studied in [5], we determined which graphs with \( m \) nodes require the largest number of parallel moves to move \( n \) disks from a starting node \( S \) to a destination node \( D \), subject to the usual rules of Hanoi games. We considered two different models of parallelism, the one-step and the two-step models. It turns out that the situation when using one-step parallel moves is virtually identical to the sequential case: Asymptotically, precisely those graphs that are worst in the sequential case are also worst in the one-step parallel case. These graphs are given by \( W_m = \{(1, \ldots, m), (\{1, 2\}, (2, 3), (3, 2)) \cup (\{i, l + 1\} | i = 3, \ldots, m - 1)\} \), for \( m \geq 3 \): the number of moves required by \( W_m \) is \( 3^n + m - 4 \). However, under the two-step model, the situation is quite different: Asymptotically, the graphs that require the largest number of moves are \( Z_m = \{(1, \ldots, m), (\{1, 2\}, (2, 3), (3, 1)) \cup (\{i, l + 1\} | i = 3, \ldots, m - 1)\} \), for \( m \geq 3 \); the number of moves required by this class of graphs is \( O(\alpha^n) \) where \( \alpha = 1 + 3^{1/2} \) (\( \alpha \approx 2.732 \)).

References