JOURNAL OF ALGORITHMS 4, 303–309 (1983)

# The Complexity of Finding Generalized Paths in Tournaments

PAVOL HELL* AND MOSHE ROSENFELD†

*Simon Fraser University, Burnaby, Canada, and Ben Gurion University, Beer Sheva, Israel*

Received June 17, 1981; revised June 3, 1982

Given a tournament with $n$ vertices, we consider the number of comparisons needed, in the worst case, to find a permutation $v_1v_2\dots v_n$ of the vertices, such that the results of the games $v_1v_2, v_2v_3, \dots, v_{n-1}v_n$ match a prescribed pattern. If the pattern requires all arcs to go forward, i.e., $v_1 \to v_2, v_2 \to v_3, \dots, v_{n-1} \to v_n$, and the tournament is transitive, then this is essentially the problem of sorting a linearly ordered set. It is well known that the number of comparisons required in this case is at least $cn \lg n$, and we make the observation that $O(n \lg n)$ comparisons suffice to find such a path in any (not necessarily transitive) tournament. On the other hand, the pattern requiring the arcs to alternate backward–forward–backward, etc., admits an algorithm for which $O(n)$ comparisons always suffice. Our main result is the somewhat surprising fact that for various other patterns the complexity (number of comparisons) of finding paths matching the pattern can be $cn \lg^{\alpha} n$ for any $\alpha$ between 0 and 1. Thus there is a veritable spectrum of complexities, depending on the prescribed pattern of the desired path. Similar problems on complexities of algorithms for finding Hamiltonian cycles in graphs and directed graphs have been considered by various authors, [2, pp. 142, 148, 149; 4].

## DEFINITIONS

A *tournament* is an orientation of a complete graph. Any two vertices (players) $v, w$ are adjacent by exactly one arc, i.e., either $v \to w$ ($v$ beats $w$), or $w \to v$ ($v$ loses to $w$); there are no ties. Throughout the paper $T_n$ denotes an arbitrary tournament with vertices $1, 2, \dots, n$. A tournament is *transitive* if the relation $v \to w$ is transitive. Clearly, a transitive tournament is linearly ordered by the relation $v \to w$, and there is (up to isomorphism) a unique transitive tournament on $1, 2, \dots, n$; it has $i \to j$ if and only if $i > j$, and it is denoted by $TT_n$.

*With support from the President's Research Grant, Simon Fraser University, 1981.
†Supported by the N.S.E.R.C. through Grant A 4792.



0196-6774/83 $3.00
Copyright © 1983 by Academic Press, Inc.
All rights of reproduction in any form reserved.

The determination, for arbitrary players $v, w$, of which of the cases $v \to w, w \to v$ actually occurs (either by "playing" the game between $v$ and $w$, or by "looking up" the result of that game), is called a *comparison*. Note that the term comparison is not intended to convey any connotation of an underlying linear order; comparisons are made in arbitrary tournaments.

Let $v_1v_2 \ldots v_n$ be a permutation of the vertices of $T_n$. The *pattern* of $v_1v_2 \ldots v_n$ is the 01-sequence $s_1s_2 \ldots s_{n-1}$ in which $s_i = 1$ if $v_i \to v_{i+1}$ and $s_i = 0$ otherwise. Any 01-sequence of length $n - 1$ is a potential pattern of some permutation of vertices of $T_n$. Given a 01-sequence $s_1s_2 \ldots s_{n-1}$ and a tournament $T_n$, we are interested in finding a permutation of the vertices of $T_n$ with the pattern $s_1s_2 \ldots s_{n-1}$. (We say that the permutation *realizes* $s_1s_2 \ldots s_{n-1}$.) More specifically, for each 01-sequence $s_1s_2 \ldots s_{n-1}$ we seek an algorithm which will, for any $T_n$, find a permutation realizing $s_1s_2 \ldots s_{n-1}$, and require the fewest number of comparisons (in the worst case). A permutation with the pattern 11..1 (or 00..0) is called a directed Hamiltonian path, a permutation with the pattern 010..10 (or 010..01, 101..10, 101..01) an antidirected Hamiltonian path [5, 8], and, in general, a permutation with a fixed pattern, a generalized Hamiltonian path. As we observed above, finding a directed Hamiltonian path in $T_n$ is a generalization of the sorting problem (which may be viewed as finding a directed Hamiltonian path in a transitive $T_n$).

There is a longstanding conjecture, [8], which appears to be quite difficult, that for $n \geqslant 8$, any tournament $T_n$ realizes all patterns $s_1s_2 \ldots s_{n-1}$. This is known to hold for directed and antidirected Hamiltonian paths for all $n \geqslant 8$ ([5, 8]), and for all generalized Hamiltonian paths when $n$ is a power of 2, [3]. Moreover, the conjecture has been verified for all $n$ and 01-sequences $s_1, s_2 \ldots s_{n-1}$ in which the $i$th block has at least $i + 1$ elements [1]. (A *block* of a 01-sequence is a maximal set of consecutive digits of the same kind.) Theorem 3, below, extends these results to a further class of generalized Hamiltonian paths.

We use the convention that $\lg x = \log_2 x$. The symbol $T[v_1, v_2, \ldots, v_k]$ denotes a *subtournament* of the tournament $T$ induced by the vertices $v_1, v_2.., v_k$; i.e., it is the tournament on $v_1, v_2, \ldots, v_k$ in which the outcomes of all games are identical to the corresponding outcomes in $T$.

## RESULTS

THEOREM 1. *A directed Hamiltonian path in $T_n$ can be found by making no more than $n \lg n$ comparisons.*

*Proof.* The following straightforward algorithm ("generalized binary insertion sort") finds a directed Hamiltonian path in $T_n$ with at most $n \lg n$ comparisons. Having found a directed path $v_1 \to v_2 \to \cdots \to v_k$ among the first $k$ vertices of $T_n$, consider the vertex $k+1$: We can use binary search to find $i$ such that $v_i \to k+1$ and $v_{i+1} \leftarrow k+1$, or $k+1 \to v_1$ or $v_k \to k+1$, using at most $\lg k$ comparisons. This results in a directed path $v_1 \to v_2 \to .. \to v_i \to k+1 \to v_{i+1} \to .. \to v_k$ among the first $k+1$ vertices.

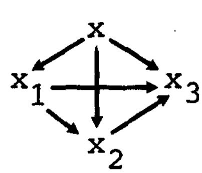
Since an algorithm to find a directed Hamiltonian path in a transitive tournament will sort the set of players linearly ordered by the relation $v \to w$, it must take $cn \lg n$ comparisons in the worst case [6, 7]. □

THEOREM 2. *Any antidirected Hamiltonian path in $T_n$ can be found by making no more than $3n + c_0$ comparisons.*

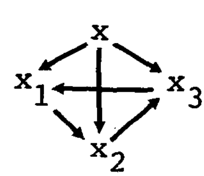
*Proof.* Instead of giving an iterative description of the algorithm, as in the previous proof, we shall describe it recursively. Let $a_n$ be the number of comparisons sufficient to construct any antidirected Hamiltonian path in $T_n$. Given a tournament $T_{n+1}$, we choose a vertex $x$, and let $T_n' = T_{n+1} \setminus \{x\}$.

CASE 1. $n \equiv 1 \pmod 2$. With $a_n$ comparisons we find an antidirected Hamiltonian path $v_1 \leftarrow v_2 \to \cdots \leftarrow v_{n-1} \to v_n$ in $T_n'$. With one additional comparison we can check the outcome of the game $(v_1, v_n)$; we may assume without loss of generality that $v_1 \to v_n$. Now we perform an additional comparison between $x$ and $v_1$: if $x \to v_1$ we have $x \to v_1 \leftarrow v_2 \to \cdots \to v_n$ and if $x \leftarrow v_1$ we have $v_2 \to v_3 \leftarrow \cdots \to v_n \leftarrow v_1 \to x$. Thus the pattern 1010... 01 is realized at the cost of no more than $a_n + 2$ comparisons. Of course, the pattern 0101..10 is realized by the reversal of any permutation realizing 1010..01. Hence $a_{n+1} \leqslant a_n + 2$.

CASE 2. $n$ is even. We choose five players and find among them three players yielding the same outcome in their game with $x$. We further check the outcomes of the three games these players played among themselves. Without loss of generality we may assume that $x$ won the three games. Hence the two possible outcomes are as follows:




Case A




Case B

Let $T_{n-3}' = T_{n+1} \setminus \{x, x_1, x_2, x_3\}$. And let

$$v_1 \leftarrow v_2 \to v_3 \leftarrow \cdots \to v_{n-3} \qquad (v_1 \to v_{n-3})$$

be an antidirected path that (with the additional arc) took $a_{n-3}+1$ comparisons to find. If $x_2 \to v_1$ then

$$x_1 \leftarrow x \rightarrow x_3 \leftarrow x_2 \rightarrow v_1 \leftarrow v_2 \rightarrow \cdots \rightarrow v_{n-3}$$

is an antidirected Hamiltonian path in $T_{n+1}$. If $v_1 \to x_2$ then take

$$v_2 \rightarrow v_3 \leftarrow \cdots \rightarrow v_n \leftarrow v_1 \rightarrow x_2 \leftarrow x \rightarrow x_3 \leftarrow x_1, \qquad \text{(Case A)}$$

$$v_2 \rightarrow v_3 \leftarrow \cdots \rightarrow v_n \leftarrow v_1 \rightarrow x_2 \leftarrow x \rightarrow x_1 \leftarrow x_3. \qquad \text{(Case B)}$$

This again assures the realization of the pattern 1010... 10 with no more than $a_{n-3}+11$ comparisons. If we need to realize the pattern 0101... 1 we need only make one additional comparison, of the two end vertices. For example, in Case A, if $v_2 \to x_1$, then $x_1 \leftarrow v_2 \rightarrow v_3 \leftarrow \cdots \rightarrow v_n \leftarrow v_1 \rightarrow x_2 \leftarrow x \rightarrow x_3$ will realize 0101..1; a similar argument applies in the other cases.

Thus $a_{n+1} \leqslant a_{n-3}+12$. It follows that $a_n \leqslant 3n + c_0$ where $c_0$ accounts for some starting value.

In [5, 8] it was shown that antidirected Hamiltonian paths exist in all tournaments $T_n$ except for three tournaments of sizes 3, 5, and 7. Thus when $n \leqslant 11$ antidirected Hamiltonian paths will have to be formed by another method. These costs are incorporated in the constant $c_0$. Observe also, that one has to make at least $n-1$ comparisons to realize a given pattern in $T_n$, thus the order of magnitude $cn$ is best possible.

THEOREM 3. *Let $2k+15 < n$. A generalized Hamiltonian path with the pattern*

$$010\ldots10\underbrace{11\ldots1}_{k \text{ times}}$$

*can be found in $T_n$ making no more than $cn + k \lg k$ comparisons.*

*Proof.*

*Step* 1. Choose a vertex $x$ and some $2k+15$ other vertices. Select $k+8$ of these vertices which yield the same outcome in their game with $x$. Call these vertices $a_1, a_2, \ldots, a_k, b_1, \ldots, b_8$. This step will take no more than $2k+15$ comparisons.

*Step* 2. Among the eight vertices $\{b_1, \ldots, b_8\}$ determine a transitive subtournament on four vertices as follows: Find among $b_2, b_3, \ldots, b_8$, four vertices that yield the same outcome in their game with $b_1$. Suppose, without loss of generality, that these four vertices are $\{b_2, b_3, b_4, b_5\}$ (at most seven comparisons). Find, among $b_3, b_4, b_5$ two vertices with the same outcome in their game with $b_2$, and check the outcome of the game between these two players. (At most four comparisons). It is easily seen that altogether we made at most 11 comparisons and the four players selected determine a $TT_4$. Without loss of generality we assume that the vertices selected are $\{b_1, b_2, b_3, b_4\}$ with $b_i \to b_j$ if $i < j$.

*Step* 3. Denote the vertices of $T_n \setminus \{x, a_1, \ldots, a_k, b_1, \ldots, b_4\}$ by $\{c_1, \ldots, c_m\}$. Observe that because of the form of the pattern, $n - k - 1$ and hence $m$ is odd. We find an antidirected path in $T[c_1, \ldots, c_m]$ making no more than $3m + c$ comparisons (Theorem 2). Let the path be

$$c_1 \leftarrow c_2 \rightarrow \cdots \leftarrow c_{m-1} \rightarrow c_m. \tag{3.1}$$

Without loss of generality we may assume that $c_1 \to c_m$ (1 comparison). We now make the following comparisons: $\{(c_1, b_1), (c_1, b_2), (c_1, b_3), (c_m, b_1), (c_m, b_2), (c_m, b_3)\}$ (six comparisons). If $c_1$ or $c_m$ loses one of these games we construct one of the following antidirected paths:

$$\begin{aligned}
&(\text{if } b_1 \to c_1)\colon c_m \leftarrow c_{m-1} \rightarrow \cdots \leftarrow c_2 \rightarrow c_1 \leftarrow b_1 \rightarrow b_4 \leftarrow b_2 \rightarrow b_3,\\
&(\text{if } b_3 \to c_m)\colon c_1 \leftarrow c_2 \rightarrow \cdots \leftarrow c_{n-1} \rightarrow c_m \leftarrow b_3 \rightarrow b_4 \leftarrow b_1 \rightarrow b_2,\\
&(\text{if } b_2 \to c_m)\colon c_1 \leftarrow c_2 \rightarrow \cdots \leftarrow c_{m-1} \rightarrow c_m \leftarrow b_2 \rightarrow b_4 \leftarrow b_1 \rightarrow b_3.
\end{aligned} \tag{3.2}$$

The other three cases are identical due to the symmetry of the path (3.1). Therefore we assume that $c_1$ and $c_m$ win all the six games. We now check the games $(c_2, b_1)$ and $(c_2, b_3)$ (2 comparisons).

$$(\text{if } c_2 \to b_1)\colon b_1 \leftarrow c_2 \rightarrow \cdots \leftarrow c_{m-1} \rightarrow c_m \leftarrow c_1 \rightarrow b_3 \leftarrow b_2 \rightarrow b_4, \tag{3.3}$$

$$(\text{if } c_2 \to b_3)\colon b_3 \leftarrow c_2 \rightarrow \cdots \leftarrow c_{m-1} \rightarrow c_m \leftarrow c_1 \rightarrow b_2 \leftarrow b_1 \rightarrow b_4, \tag{3.4}$$

$$(\text{otherwise})\colon c_3 \leftarrow \cdots \leftarrow c_{m-1} \rightarrow c_m \leftarrow c_1 \rightarrow b_2 \leftarrow b_1 \rightarrow c_2 \leftarrow b_3 \rightarrow b_4. \tag{3.5}$$

In all four cases, (3.2)–(3.5), we end up with an antidirected path with the two last vertices in the set $\{b_1, b_2, b_3, b_4\}$. These steps take no more than $3m + c_0 + 9$ comparisons.

*Step* 4. If in step 1, $x$ was a "winner" (i.e., $x$ beats the $k + 8$ chosen vertices), we first determine a directed Hamiltonian path in $T[a_1, \ldots, a_k]$. This will use no more than $k \lg k$ comparisons (Theorem 1). Let the path be $a_1 \to a_n \to \cdots \to a_k$. Pick the antidirected path obtained in step 3 and consider

$$\cdots \rightarrow b_j \leftarrow x \rightarrow a_1 \rightarrow a_2 \rightarrow \cdots \rightarrow a_k \qquad (b_j \in \{b_1, b_2, b_3, b_4\}).$$

This permutation realizes the given pattern.

If $x$ was a "loser," let $b_j$ be as above. We first find a Hamiltonian path in $T[b_j, a_1, \ldots, a_k]$. This will be achieved in no more than $(k+1)\lg(k+1)$ comparisons (Theorem 1). Let this path be $g_0 \to g_1 \to \cdots \to g_k$. To demonstrate the final step, assume for example that the path (3.4) was obtained in step 3. Then:

$$b_3 \leftarrow c_2 \to \cdots \leftarrow c_{m-1} \to c_m \leftarrow c_1 \to b_2 \leftarrow b_1 \to x$$
$$\leftarrow g_0 \to g_1 \to \cdots \to g_k$$

realizes the given pattern. The other cases can be handled in the same way. Thus, the given pattern can be realized making no more than $cn + k\lg k$ comparisons (where $c$ is a constant).

To establish lower bounds of the same order of magnitude as the upper bounds implied by Theorem 3, we shall consider the following situation. Consider a linearly ordered set of $n$ elements, and suppose we wish to find a sorted sequence of $k$ elements. Obviously, we can choose $k$ elements and sort them, using $O(k\lg k)$ comparisons. It might seem possible to use the flexibility of choosing the elements to be sorted along the way of the sorting process, to lower the number of comparisons. That, however, is not the case:

LEMMA. *Any binary-comparison-based algorithm to find a sorted sequence of length $k$ in a linearly ordered set of $n$ elements requires, in the worst case $ck\lg k$ comparisons.*

*Proof.* A binary-comparison-based algorithm can be represented by a binary decision tree $T$ [6, 7]. Each leaf of the tree $T$ corresponds to an output of the algorithm, i.e., to a permutation of some $k$ of the $n$ elements. There is, of course, no need for all permutations of all subsets of size $k$, to be present as outputs. We claim, however, that at least $k!$ outputs must appear in the tree. To prove the claim, we note that for each output, i.e., a permutation of some $k$ of the $n$ elements, there are precisely $\binom{n}{k} \cdot (n-k)! = n!/k!$ linear orders on the $n$ elements, for which the output is correct. Hence if there are fewer than $k!$ different outputs to our algorithm, then some of the $n!$ possible linear orders will not be correctly answered by any of the outputs. Therefore $T$ has at least $k!$ leaves and hence height at least $\lg k! \geqslant ck\lg k$. On the other hand, the height of $T$ is the worst case number of comparisons of the algorithm represented by $T$.

THEOREM 4. *Let $p$ be a pattern in which the longest block has $k$ elements. Any algorithm to find a permutation with pattern $p$ requires at least $ck\lg k$ comparisons in the worst case.*

*Proof.* Any algorithm to find a permutation with pattern $p$ can be used to find a sorted sequence of length $k$ in a set linearly order by $<$. It suffices to view the set as a tournament with $i \to j$ if and only if $i < j$. From the permutation with pattern $p$ we can obviously recover a sorted sequence of length $k$. Hence the lower bound follows from the Lemma.

COROLLARY. *For every $\alpha$, $0 \leqslant \alpha \leqslant 1$, and all sufficiently large $n$, there exists a pattern $p_n$ such that finding, in $T_n$, a permutation with pattern $p_n$ requires at least $c_1 n(\lg n)^\alpha$ comparisons, and can be done with no more than $c_2 n(\lg n)^\alpha$ comparisons.*

*Proof.* Theorems 1 and 2 take care of the cases $\alpha = 0$ and $\alpha = 1$. Let $k = \lfloor n/(\lg n)^{1-\alpha} \rfloor$ and note that

$$\frac{n(\lg n)^\alpha}{3} \leqslant \frac{n \lg n - (1-\alpha) n \lg\lg n - \lg 2}{2(\lg n)^{1-\alpha}} \leqslant k \lg k$$

$$\leqslant \frac{n \lg n - (1-\alpha) n \lg\lg n}{(\lg n)^{1-\alpha}} \leqslant n(\lg n)^\alpha$$

for all large $n$. Let $p_n$ be the sequence described in Theorem 3. The upper bound follows from Theorem 3, the lower bound from Theorem 4.

Some of the patterns investigated in [1, 3] have had their presence in any $T_n$ confirmed by proofs which do not appear to yield efficient algorithms. It would be of interest to consider the complexity of algorithms to find generalized Hamiltonian paths with such patterns. We would not be surprised if some patterns required more than $cn \lg n$ comparisons.

## ACKNOWLEDGMENTS

The authors wish to express their thanks to J. Horton and D. Kirkpatrick for stimulating conversations concerning the above topics.

## REFERENCES

1. B. ALSPACH AND M. ROSENFELD, Realization of certain generalized paths in tournaments, *Discrete Math.*, **34** (1981), 199–202.
2. J. C. BERMOND, Hamiltonian graphs, *in* "Selected Topics in Graph Theory" (L. W. Beineke, R. J. Wilson, Eds.), pp. 127–167, Academic Press, New York, 1978.
3. R. FORCADE, Parity of paths and circuits in tournaments, *Discrete Math.* **6** (1973), 115–118.
4. D. GOUYON BEAUCHAMPS, Un algorithme de recherche de circuit Hamiltoniens dans les graphes 4-connexes planaires, *in* "Problème Combinatoires et Théorie des Graphes" (J. C. Bermond *et al.*, Eds.), pp. 185–187, C.N.R.S., Paris, 1978.
5. B. GRÜNBAUM, Antidirected Hamiltonian paths in tournaments, *J. Combinatorial Theory* (*B*) **11** (1971), 249–257.
6. E. HOROWITZ AND S. SAHNI, "Fundamentals of Computer Algorithms," Computer Science Press, Potomac, Md., 1978.
7. D. KNUTH, "The Art of Computer Programming," Vol. 3, Addison–Wesley, Reading, Mass., 1973.
8. M. ROSENFELD, Antidirected Hamiltonian paths in tournaments, *J. Combinatorial Theory* (*B*) **12** (1972), 93–99.