Hardness of Finding Kings and Strong Kings

Ziad Ismaili Alaoui
Department of Computer Science, University of Liverpool, UK

Nikhil Mande
Department of Computer Science, University of Liverpool, UK

Abstract

A king in a directed graph is a vertex v such that every other vertex is reachable from v via a path of length at most 2. It is well known that every tournament (a complete graph where each edge has a direction) has at least one king. Our contributions in this work are:

- We show that the query complexity of determining existence of a king in arbitrary n-vertex digraphs is $\Theta(n^2)$. This is in stark contrast to the case where the input is a tournament, where Shen, Sheng, and Wu [SICOMP'03] showed that a king can be found in $O(n^{3/2})$ queries.
- In an attempt to increase the "fairness" in the definition of tournament winners, Ho and Chang [IPL'03] defined a *strong king* to be a king k such that, for every v that dominates k, the number of length-2 paths from k to v is strictly larger than the number of length-2 paths from v to k. We show that the query complexity of finding a strong king in a tournament is $\Theta(n^2)$. This answers a question of Biswas, Jayapaul, Raman, and Satti [DAM'22] in the negative.

A key component in our proofs is the design of specific tournaments where every vertex is a king, and analyzing certain properties of these tournaments. We feel these constructions and properties are independently interesting and may lead to more interesting results about tournament solutions.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Theory of computation → Oracles and decision trees; Mathematics of computing → Combinatorics

Keywords and phrases Tournaments, kings, query complexity

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2025.36

1 Introduction

A tournament is a complete directed graph. Motivated to find a reasonable notion of a "most-dominant chicken" in a flock of chickens (where the "pecking order" is represented as a tournament), Landau [10] introduced the notion of a *king* in a tournament. A king in a tournament is a vertex k such that for every other vertex v, either there is an edge from k to v, or there exists a u such that there are edges from k to u and from u to v. That is, a king is a vertex such that every other vertex is reachable from it via a path of length at most 2. The notion of a king is a natural generalisation of that of a source: a vertex is a source if every other vertex is reachable from it via a path of length 1. While it is easy to see that a tournament need not have a source, it is also not hard to show that every tournament has at least one king [10, 13]. Soon after Maurer's popular science article about kings in tournaments [13], Reid [16] showed the existence of tournaments in which every vertex is a king. Tournament solutions/winners are also a natural object of study in social choice theory. See, for instance, [5] and the references therein. The monograph by Moon [14] sparked a line of research on tournaments, their solutions, and their structural properties.

A natural computational model to study the complexity of finding tournament solutions and related problems is that of *query complexity*. In this setting, an algorithm has access to the input in the form of *queries*: in one step an algorithm may query the direction of the edge between a pair of vertices of its choosing (in case of an undirected graph problem, the algorithm may query the presence/absence of an edge between a pair of vertices of its choosing). The algorithm's goal is to minimize the number of queries made in the worst case,

and all other operations are treated as free. There is a rich literature on the study of graph problems in the setting of query complexity, see, for example, [18, 17, 21, 7, 3, 5, 12] and the references therein.

The focus of this paper is on finding a king in a digraph. Given that every tournament has a king as mentioned earlier, it is natural to ask: how efficiently can a king in an n-vertex tournament be found? Note that this is equivalent to determining whether a digraph has a *radius*[1] of at most 2. The study of this question was initiated by Shen, Sheng, and Wu [19], who exhibited an algorithm with query complexity $O(n^{3/2})$, and they also showed a non-matching lower bound of $\Omega(n^{4/3})$. To date, these are the best-known bounds on the complexity of finding a king in a tournament, and closing the gap remains an intriguing open problem. There have been some very recent works showing better query bounds for variants of the task of finding a king, or studying the complexity of finding kings in models other than deterministic query complexity [2, 9, 12, 11, 1]. Our first main contribution in this paper to show that if we allow for non-tournament inputs, then the query complexity of even determining the existence of a king shoots up.

▶ **Theorem 1.** *The randomized query complexity of determining the existence of a king in an n-vertex digraph is $\Theta(n^2)$.*

Relevant to our work is the notion of a *strong king* in a tournament. This notion was defined by Ho and Chang [8] in an attempt to increase the "fairness" of the notion of a winner in a tournament. Let $\delta(u, v)$ denote the number of length-2 paths from u to v. A king k is said to be *strong* if for all w with $w \to k$, we have $\delta(k, w) > \delta(w, k)$. To see why this notion increases the fairness of the definition of a winner, consider the following scenario: say we have a tournament in which a vertex k has a single out-neighbour u, and all other vertices are in-neighbours of k. Further suppose every vertex other than k and u is an out-neighbour of u. The vertex k is a king in this tournament, but not necessarily a "fair" one, since it "defeats" only one vertex. However, it is not hard to see that k is not a strong king. The study of strong kings in tournaments has also gained some attention [4, 15]. Biswas et al. [2, Section 7] asked if one could devise an algorithm to find a strong king in an n-vertex tournament with cost $o(n^2)$. Our second main contribution in this paper is to answer this in the negative.

▶ **Theorem 2.** *The randomized query complexity of finding a strong king in an n-vertex tournament is $\Theta(n^2)$.*

1.1 Organization of the Paper

In Section 2, we review basic preliminaries and formalize the computational questions and models of interest. In Section 3, we recall two known constructions of tournaments where every vertex is a king. In Section 4, we show certain properties of these constructions and use these properties to conclude our main results, Theorem 1 and Theorem 2 (restated formally as Theorem 14 and Theorem 18, respectively). Section 4 contains the main technical novelty of this paper. Finally, in Section 5, we recall our main results and discuss potential directions for future work.

[1] The radius of a directed graph is the minimum eccentricity of any vertex, where the eccentricity of a vertex is defined to be the greatest distance from it to any other vertex.

2 Preliminaries

2.1 Basic Concepts and Notation

For a positive integer n, let $[n] = \{1, 2, \ldots, n\}$ and define $[n]_0 = \{0\} \cup [n-1]$. We use S_n to denote the set of permutations on the elements in $[n]_0$. For non-negative integers n and $i \in [n]_0$, let $\sigma_i \in S_n$ denote the permutation defined by $\sigma_i(j) = (j + i) \pmod n$ for all $j \in [n]_0$. Let G be an n-vertex graph with vertex set $[n]_0$, and let $i \in [n]_0$. Define $\sigma(G, i)$ to be the graph obtained by relabelling each vertex v of G by $\sigma_i(v)$.

▶ **Definition 3** (Tournament, Subgraph, Subtournament). A *tournament* $T = (V, E)$ is a *complete* directed graph where V is the set of vertices, E is the set of directed edges, and for every edge $(u, v) \in E$, $(v, u) \notin E$. A *subgraph* induced by $G = (V, E)$ on $V' \subseteq V$ is a graph $G[V'] = (V', E')$ such that $E' = E \cap \{(u, v) : u \in V' \text{ and } v \in V'\}$. A *subtournament* is analogously defined with respect to tournaments.

We denote $V(G)$ and $E(G)$ as the vertex and edge sets of G, respectively. We often use the notation (u, v) and $u \to v$ interchangeably. For an edge $e = (u, v)$, we say that u is the *source* of e and v is the *target* of e. A vertex $v \in V$ is an *out-neighbor* of $u \in V$ if $(u, v) \in E$, and v is an *in-neighbor* of u if $(v, u) \in E$. We say that a vertex v *dominates* u when $v \to u$. The set of all out-neighbors of a vertex v is denoted by $\Gamma^+(v)$, and the set of all in-neighbors of v is denoted by $\Gamma^-(v)$. Define $d^+(v) = |\Gamma^+(v)|$ and $d^-(v) = |\Gamma^-(v)|$. A (directed) *path* of length n is an ordered sequence of edges $e_1, e_2, \ldots, e_n$ such that the target of e_i is the source of e_{i+1} for all $1 \leq i < n$.

▶ **Definition 4** (King). A *king* in a directed graph is a vertex v such that, for all other vertices $u \neq v$, at least one of the following holds:

- $v \to u$ is an edge in the graph, or
- there exists a vertex w such that $v \to w$ and $w \to u$ are edges in the graph.

In other words, a king is a vertex that can reach every other vertex by a path of at most two edges. It is well known [10] that every tournament has at least one king, and the maximum-out-degree vertex is always a king. A king that dominates all other vertices is called a *source*.

▶ **Lemma 5** ([13, Theorem 7]). *Let $T = (V, E)$ be an arbitrary tournament. T contains exactly one king if and only if that king is a source.*

Define $\delta(u, v)$ as the number of length-2 paths from vertex u to vertex v. A king k is said to be *strong* if, for each vertex v such that $v \to k$, we have $\delta(k, v) > \delta(v, k)$. That is, a king k is said to be strong if, for any vertex v that directly dominates it, there are strictly more length-2 paths from k to v than there are from v to k. Ho and Chang [8, Lemma 1] observed that every tournament has at least one strong king. We include a proof for completeness.

▶ **Lemma 6** ([8, Lemma 1]). *Let $T = (V, E)$ be an arbitrary tournament, and let $u \in V$ be a maximum-out-degree vertex in T. Then, u is a strong king.*

Proof. Towards a contradiction, suppose u is not a strong king. Then, there must exist some vertex v such that $v \to u$ and $\delta(v, u) \geq \delta(u, v)$.

Define $W = \{w \in V \setminus \{u, v\} : u \to w \text{ and } v \to w\}$. Note that $d^+(u) = \delta(u, v) + |W|$ and $d^+(v) = \delta(v, u) + |W| + 1 > d^+(u)$, which is a contradiction since we assumed u to be a maximum-out-degree vertex. ◀

A set of vertices $S \subseteq V$ is a *dominating set* in $T = (V, E)$ if, for every $v \in V$, either $v \in S$ or there exists $w \in S$ such that $w \to v$. A *dominating pair* is a dominating set of exactly two vertices.

2.2 Formal Definitions of Computational Tasks

We represent an n-vertex graph $G = (V = [n]_0, E)$ as a binary string in $\{0,1\}^{\alpha_n}$, where $\alpha_n = \binom{n}{2}$ if G is a tournament, or $\alpha_n = n^2$ otherwise. An element of $[n]_0$ corresponds to the label of a vertex, and there is one variable ($\{i, j\}$) per edge (between vertex i and vertex j) that defines its direction. In a tournament, we assume $\alpha_n = \binom{n}{2}$ because for each pair of vertices, there is exactly one directed edge between them. Define the relation $\mathsf{KING}_n \subseteq \{0,1\}^{\alpha_n} \times [n]_0$ by

$$(G, v) \in \mathsf{KING}_n \text{ if } \forall u \in [n]_0 \setminus \{v\}\,, \text{ either } v \to u \text{ or } \exists w, v \to w \to u,$$

and define $K_n \subseteq \{0,1\}^{\alpha_n}$ as the language of all n-vertex graphs G such that $\exists v \in V$, $(G, v) \in \mathsf{KING}_n$. In other words, K is the language of all n-vertex graphs containing at least one king. Let $\mathsf{EXIST\text{-}KING}_n \subseteq \{0,1\}^{\alpha_n} \times \{0,1\}$ be the relation such that $(G, 1) \in \mathsf{EXIST\text{-}KING}_n$ if and only if $G \in K_n$.[2] Finally, for $v \in V$, define the relation $\mathsf{STRONG\text{-}KING}_n \subseteq \{0,1\}^{\alpha_n} \times [n]_0$ by

$$(G, v) \in \mathsf{STRONG\text{-}KING}_n \text{ if } \forall u \in [n]_0 \text{ such that } u \to v, \delta(v, u) > \delta(u, v).$$

2.3 Deterministic Query Complexity

A deterministic decision tree $\mathcal{T}$ on m variables is a binary tree where the internal nodes are labelled by variables, and leaves are labelled with elements of a set $\mathcal{R}$. Each internal node has a left child, corresponding to an edge labelled 0, and a right child, corresponding to an edge labelled 1. On an input $x \in \{0,1\}^m$, $\mathcal{T}$'s computation traverses a path from root to leaf as follows: At an internal node, the variable associated with that node is *queried*: if the value obtained is 0, the computation moves to the left child, otherwise it moves to the right child. The output of $\mathcal{T}$ on input x, denoted by $\mathcal{T}(x)$, is the label of leaf node reached.

We say that a decision tree $\mathcal{T}$ *computes* the relation $f \subseteq \{0,1\}^m \times \mathcal{R}$ if $(x, \mathcal{T}(x)) \in f$ for all $x \in \{0,1\}^m$. The deterministic query complexity of f, is

$$\mathsf{D}(f) := \min_{\mathcal{T}:\mathcal{T} \text{ computes } f} \text{depth}(\mathcal{T}).$$

That is, $\mathsf{D}(f)$ represents the minimum depth of any deterministic decision tree that computes f.

2.4 Randomized Query Complexity

A randomized decision tree $\mathcal{A}$ is a distribution $\mathcal{D}_\mathcal{A}$ over deterministic decision trees. On input $x \in \{0,1\}^m$, the computation of $\mathcal{A}$ proceeds by first sampling a deterministic decision tree $\mathcal{T}$ according to $\mathcal{D}_\mathcal{A}$, and outputting the label of the leaf reached by $\mathcal{T}$ on x. We say $\mathcal{A}$ computes f to error ε if for every input x, $\Pr[(x, \mathcal{A}(x)) \in f] \geq 1 - \varepsilon$. The randomized query complexity of $f \subseteq \{0,1\}^m \times \mathcal{R}$ is defined as follows:

$$\mathsf{R}_\varepsilon(f) = \min_{\substack{\mathcal{A} \text{ computing } f \\ \text{with error} \leq \varepsilon}} \max_{\mathcal{T}:\mathcal{D}_\mathcal{A}(\mathcal{T})>0} \text{depth}(\mathcal{T}).$$

[2] Note that we could also define $\mathsf{EXIST\text{-}KING}_n$ as a function from $\{0,1\}^{\alpha_n} \to \{0,1\}$ where $\mathsf{EXIST\text{-}KING}_n(G) = 1$ iff G contains a king, but we choose to define it as a relation for the sake of consistency with other definitions.

When we drop the subscript ε, we assume $\varepsilon = 1/3$. We use Yao's minimax principle [20], stated below in a form convenient for us.

▶ **Lemma 7** (Yao's Minimax Principle). *For a relation $f \subseteq \{0,1\}^m \times \mathcal{R}$, we have $\mathsf{R}_\varepsilon(f) \geq k$ if and only if there exists a distribution $\mu : \{0,1\}^m \to [0,1]$ such that $\mathsf{D}_\mu(f) \geq k$. Here, $\mathsf{D}_\mu(f)$ is the minimum depth of a deterministic decision tree that computes f to error at most ε when inputs are drawn from the distribution μ.*

3 Balanced Tournaments

A *balanced* tournament is one in which every vertex is a king. Maurer [13] showed that a balanced tournament exists for every positive integer n, except when $n = 2$ or $n = 4$; this result naturally guarantees the existence of a balanced tournament for every odd n.

In this section, we present two distinct constructions of balanced tournaments. In subsequent sections, we analyze key properties of these constructions that help us prove our bounds. We clarify here that both of these constructions have appeared in prior works, but the properties that we require of them have not been analyzed earlier, to the best of our knowledge.

▶ **Construction 8** ([13, Lemma 7]). Let n be an odd positive integer. Define Δ_n as a tournament on n vertices, labelled by elements of $[n]_0$, recursively constructed as follows:

- If $n = 1$, Δ_n consists of a single vertex labelled 0.
- If $n > 1$, build Δ_{n-2} as a subtournament induced by $[n-2]_0$. Then direct the edges such that:
  1. for each vertex v in $[n-2]_0$, $(n-1) \to v$ and $v \to (n-2)$, and
  2. $(n-2) \to (n-1)$.

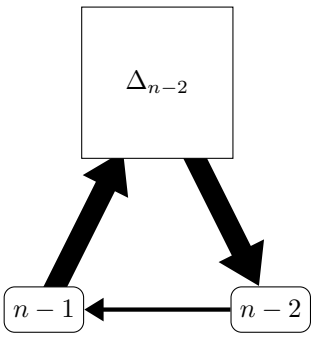

Figure 1 Illustration of Construction 8.

Figure 1 illustrates the construction:[3] vertex $n-1$ is dominated by vertex $n-2$ and dominates all vertices in Δ_{n-2}, and vertex $n-2$ dominates vertex $n-1$ and is dominated by all vertices in Δ_{n-2}. For the rest of this paper, we use Δ_n to denote the n-vertex tournament as defined in Construction 8. While the following was already shown by Maurer [13, Lemma 7], we include a proof for completeness.

▶ **Lemma 9.** *Let n be an odd positive integer. The tournament Δ_n is balanced.*

[3] The shape of the figure may clarify the choice of "Δ".

Proof. Let us proceed by induction on the number of vertices n.

When $n = 1$, Δ_n is clearly a balanced tournament, as its single vertex trivially satisfies the condition of being a king. Indeed, since there are no other vertices, it vacuously reaches all other vertices by a path of length 1 or 2.

Now, suppose the lemma holds for some n, we show that augmenting the tournament by two vertices following the instructions of Construction 8 does not violate the invariant; that is, Δ_{n+2} still remains balanced. By the induction hypothesis, every vertex in $[n]_0$ reaches all other vertices within that set via a path of length 1 or 2. When two new vertices, $n+1$ and n, are added:

- vertex $n+1$ dominates all vertices in $[n]_0$;
- all vertices in $[n]_0$ dominate n; and
- vertex n dominates $n+1$.

Clearly, none of the vertices in $[n]_0$ loses its status as a king, as each vertex in $[n]_0$ reaches n directly and reaches $n+1$ via n. Furthermore, the newly added vertices also satisfy the king condition:

- vertex $n+1$ reaches n via one of the vertices in $[n]_0$, and n reaches $n+1$ directly; and
- both $n+1$ and n reach all vertices in $[n]_0$ (either directly or via n).

Thus, the property is preserved and Δ_{n+2} remains balanced. ◀

The following construction has appeared in the literature as an example of a *rotational tournament* [6, Section 2].

▶ **Construction 10.** Let n be an odd positive integer. Define U_n as a tournament on n vertices, labelled by elements of $[n]_0$. For each pair of vertices i, j with $i < j$, direct the edge as follows:

- If $i + j$ is odd, direct the edge from i to j ($i \to j$).
- Otherwise, direct the edge from j to i ($j \to i$).

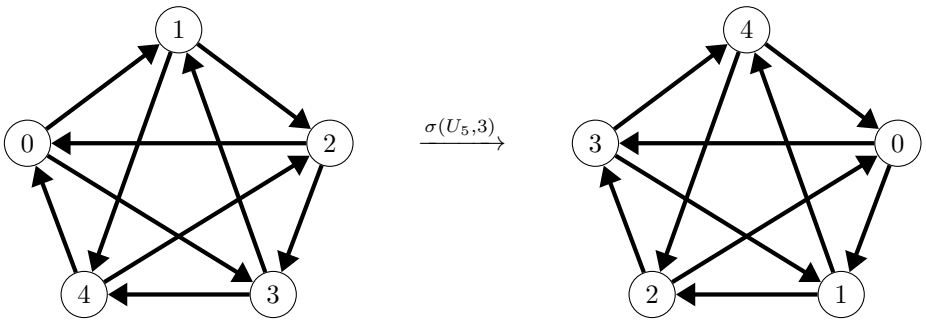

Figure 2 The tournament U_5 and the tournament $\sigma(U_5, 3) \in \mathrm{Aut}(U_5)$ obtained by rotating the labels cyclically clockwise by 3 positions.

An example is depicted in Figure 2 when $n = 5$. While this implicitly follows from the work of [6], we now show from first principles that U_n is balanced for any odd positive integer n. Again, for the rest of this paper, we use U_n to denote the n-vertex tournament as defined in Construction 10.

▶ **Lemma 11.** *Let n be an odd positive integer. The tournament U_n is balanced.*

One way to prove Lemma 11 is to show that each vertex in U_n (for an odd positive n) dominates exactly $\frac{n-1}{2}$ vertices. Thus, given that a maximum-out-degree vertex in a tournament is a king [10], it follows that all vertices are kings. For completeness, we include a proof from first principles below.

Proof of Lemma 11. Observe that each vertex i dominates all lesser vertices of similar parity and all greater vertices of opposite parity. Define the parity function $p(x) = x \pmod 2$; formally, we have $\Gamma^+(i) = \{j \in [n] : j < i \text{ and } p(j) = p(i)\} \cup \{j \in [n] : j > i \text{ and } p(i) \neq p(j)\}$.

Consider any vertex $i < n-1$. Since $p(i+1) \neq p(i)$, vertex i dominates $i+1$. Thus, $i+1$ in turn dominates all vertices less than i of opposite parity and all greater vertices of the same parity; formally, $\Gamma^-(i) = \Gamma^+(i+1)$. Hence, i reaches all vertices by a path of at most two edges either through direct domination, or via vertex $i+1$.

For $i = n-1$, note that $n-1$ is even, so it dominates all even vertices, including vertex 0. Since vertex 0 dominates all odd vertices, and $n-1$ dominates 0, it follows that $n-1$ is also a king. Therefore, every vertex in U_n is a king, and the tournament is balanced. ◀

An interesting property of U_n is its symmetry. Indeed, observe that rotating the tournament (i.e., shifting its labels) clockwise or anti-clockwise results in the same labelled tournament. This is useful as it allows a property of one vertex to be extended to all vertices in U_n.

▶ **Lemma 12** ([6]). *Let n be an odd positive integer. For $i \in \mathbb{Z}$, we have $\sigma(U_n, i) \in \mathrm{Aut}(U_n)$.*

4 Lower Bounds

In this section, we prove our two main results: determining the existence of a king in an arbitrary n-vertex digraph (Subsection 4.1) and finding a strong king in a tournament (Subsection 4.2) both admit a (randomized) query complexity of $\Theta(n^2)$. To do so, we make use of the constructions from Section 3.

4.1 Kings in Directed Graphs

Below is a property we require of Construction 8. Recall that a 2-element-set of vertices $\{v, w\}$ is said to be a *dominating pair* in $T = (V, E)$ if, for every $u \in V \setminus \{v, w\}$, there exists $x \in \{v, w\}$ such that $x \to u$.

▶ **Lemma 13.** *Let $n > 3$ be an odd positive integer, and let $i, j \in [n]_0$ with $i < j$. The set of vertices $\{i, j\}$ is a dominating pair in Δ_n if and only if $j = n-1$.*

Proof. We show both directions below.

- Let us first show the implication ($\Rightarrow$): that is, any pair $\{i, j\}$ where $j = n-1$ is dominating. According to Construction 8, and as depicted in Figure 1, vertex $n-1$ dominates all vertices except $n-2$. However, it is easy to see that all other vertices are either $n-2$ or dominate $n-2$. Therefore, pairing $n-1$ with any of those vertices produces a dominating pair.
- To show the implication in the other direction ($\Leftarrow$), assume, for the sake of contradiction, that $j \neq n-1$ and $\{i, j\}$ is a dominating pair. Then, $i \in [n-2]_0$ and either $j \in [n-2]_0$, or $j = n-2$ (recall that $i < j$).
  - If $j \in [n-2]_0$, then $\{i, j\}$ does not dominate $n-1$, a contradiction.
  - If $j = n-2$, then $i \in [n-2]_0$ must dominate all other vertices in $[n-2]_0$, and thus must be a source in $\Delta_n[[n-2]_0]$. However, since all (of the at least 3) vertices in $\Delta_n[[n-2]_0]$ are kings by Lemma 9, this contradicts Lemma 5. Thus, the pair $\{i, j\}$ is not a dominating pair.

This concludes the proof. ◀

Simply, the only dominating pairs in Δ_n are $\{0, n-1\}, \{1, n-1\}, \ldots, \{n-2, n-1\}$. Using the above property, we show below that the deterministic and randomized query complexity of finding a king in an arbitrary n-vertex digraph is $\Theta(n^2)$.

▶ **Theorem 14.** *For all positive integers n, $\mathsf{R}(\textsf{EXIST-KING}_n) = \Theta(n^2)$ for general digraphs.*

Proof. The upper bound is trivial. Assume that n is odd. If n is even, an argument similar to that given in the last paragraph of the proof of Theorem 18 can be used. Let $A = ([n]_0, E_A)$ be an arbitrary balanced tournament on n vertices. Consider the n-vertex tournament Δ_n from Construction 8 and relabel all vertex labels by adding n to them. Let the resultant graph be called A'. Construct a tournament C on $2n$ vertices labelled in $[2n]_0$ as follows:

1. Consider the disjoint union of A and A', and
2. add edges from every vertex in A to vertex $(2n-1)$ in A'.

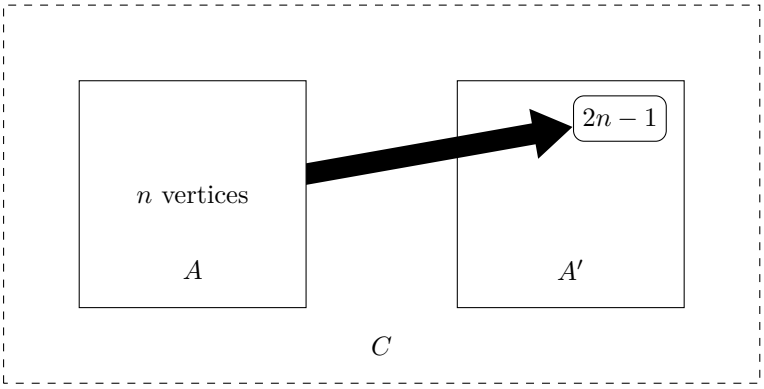

Figure 3 Construction of C from the proof of Theorem 14. $C[A]$ is a balanced tournament, and $C[A']$ is a relabelling of Δ_n from Construction 8.

Figure 3 illustrates the construction of C. Clearly, C does not have a king: indeed, no vertex in A' reaches any vertex in A, and every vertex in A dominates $(2n-1)$, which does not dominate $(2n-2)$ (by construction, see Construction 8).

For $i \in [n]_0$ and $j \in [n-1]_0$, define $C_{i,j}$ as the tournament obtained by adding the edge $i \to (n+j)$ in C. Observe that i is a king in $C_{i,j}$, since:

- i 2-step dominates all vertices in A (as $C_{i,j}[A]$ is a balanced tournament), and
- i dominates $(2n-1)$ and $n+j$, which form a dominating pair in $C_{i,j}[A']$ by Lemma 13.

Define the distribution μ on $2n$-vertex tournaments as follows:

$$\mu(T) = \begin{cases} 1/2 & T = C, \\ \frac{1}{2n(n-1)} & T = C_{i,j}, \quad \text{for all } i \in [n]_0, j \in [n-1]_0. \end{cases}$$

Equivalently, μ can be defined by the following sampling procedure: Start with the tournament C; then, with probability $1/2$, do nothing, or, with probability $1/2$, add a uniformly random (over $i \in [n]_0$ and $j \in [n-1]_0$) edge $i \to n+j$. Towards a contradiction using Lemma 7, let $\mathcal{A}$ be a deterministic algorithm with cost less than $n^2/100$ that decides whether a king exists in digraphs w.r.t. μ of error $\leq 1/3$.

Consider the computation path taken by $\mathcal{A}$ that answers all edges consistent with C, and answers 0 (i.e., "no edge present") to all queries of the form $i \to j$ for $i \in [n]$, $j \in \{n, n+1, \ldots, 2n-1\}$. We consider two cases.

1. Suppose the output at this leaf of $\mathcal{A}$ is 1 (i.e., "a king exists"). The algorithm errs on the input C, which has a μ-mass of 1/2. This is not possible since we assumed $\mathcal{A}$ to make an error of at most 1/3 w.r.t. μ.
2. Suppose the output at this leaf of $\mathcal{A}$ is 0 (i.e., "no king exists"). Since the number of queries on this path was assumed to be at most $n^2/100$, there must be at least $99n^2/100-n$ edges of the form $i \to (n+j)$ that have not been queried, where $i \in [n]_0$ and $j \in [n-1]_0$. For each such i, j, the graph $C_{i,j}$ also follows this computation path. However, the vertex i is a king in $C_{i,j}$. Hence $\mathcal{A}$ errs on $C_{i,j}$. Thus, the total error made by $\mathcal{A}$ is at least $(\frac{99n^2}{100} - n) \cdot \frac{1}{2n(n-1)} > 1/3$ for sufficiently large n. This again contradicts our assumption that $\mathcal{A}$ made error at most 1/3 w.r.t. μ, and hence the theorem follows.

This concludes the proof. ◀

As an easy corollary, we obtain the same deterministic lower bound for determining the existence of a king in a digraph.

▶ **Corollary 15.** *For all positive integers n, $\mathsf{D}(\text{EXIST-KING}_n) = \Theta(n^2)$ for general digraphs.*

4.2 Strong Kings in Tournaments

Recall that $\delta(v, u)$ denotes the number of length-2 paths from v to u in an underlying digraph/tournament. Recall that a strong king in a tournament is a king v such that, for every vertex u that dominates v (i.e., $u \to v$), we have $\delta(v, u) > \delta(u, v)$. Below, we show that in the construction U_n from Construction 10, for every vertex $v \in V(U_n)$ and an in-neighbor u of it, we have $\delta(v, u) - \delta(u, v) = 1$. In other words, there is exactly one more length-2 path from v to u than from u to v. This observation is crucial to argue that flipping the direction of a single edge used by v to reach u via a length-2 path will result in v no longer being a strong king.

▶ **Lemma 16.** *Let n be an odd positive integer. For each $j \to i$ in U_n, we have $\delta(i, j) - \delta(j, i) = 1$.*

Proof. Define the parity function $p(x) = x \pmod 2$. From the proof of Lemma 11, each vertex i in U_n dominates all lesser vertices of the same parity as i, and all greater vertices of opposite parity.

- Assume that $j > i$. Since $j \to i$, Construction 10 implies $p(j) = p(i)$. The vertex j dominates all vertices $v < j$ with $p(v) = p(i)$ and all vertices $w > j$ with $p(w) \neq p(i)$.
  To count length-2 paths from i to j, we first show that if $i \to v \to j$, then $i < v < j$.
  - If $v < i$, then $p(v) = p(i)$ (since $i \to v$), but also $p(v) \neq p(j)$ (since $v \to j$ and $v < j$), contradicting $p(i) = p(j)$.
  - If $v > j$, then $p(v) \neq p(i)$ (since $i \to v$), but also $p(v) = p(j)$ (since $v \to j$ and $v > j$); again, a contradiction.

  Thus, v must satisfy $i < v < j$, and since $i \to v$, it follows that $p(v) \neq p(i)$. This means $\delta(i, j) = |\{v : i < v < j \text{ and } p(v) \neq p(i)\}| = \frac{j-i}{2}$.
  Using nearly the same argument, one can show that $\delta(j, i) = |\{v : i < v < j \text{ and } p(v) = p(i)\}| = \frac{j-i}{2} - 1$.
- The case where $j < i$ follows from the case where $j > i$ by symmetry (Lemma 12). Indeed, let i' and j' be the new labels of i and j in $\sigma(U_n, n-i)$, respectively. Then, $i' = 0 < j'$, and given that $\sigma(U_n, n-i) \in \mathrm{Aut}(U_n)$, the lemma follows.

Therefore, we have $\delta(i, j) - \delta(j, i) = \frac{j-i}{2} - (\frac{j-i}{2} - 1) = 1$. ◀

For the tournament U_n and a vertex v of it, define $D(v)$ be the set of edges such that, if the direction of exactly one edge in $D(v)$ were flipped, v would no longer be a strong king.

Now, we show that $|D(v)| = \Theta(n^2)$.

▶ **Lemma 17.** *Let n be an odd positive integer. For the tournament U_n and $D(\cdot)$ as defined above, we have $|D(v)| = (n^2-1)/4$ for all $v \in [n]$.*

Proof. We show below that $|D(0)| = (n^2-1)/4$. The result for general v immediately follows by symmetry (Lemma 12), as $|D(0)| = |D(1)| = \cdots = |D(n-1)|$.

We first consider the effect of flipping an edge connected to 0.

- Let $j \in [n]_0$ be odd. The construction of U_n ensures the direction of the edge between 0 and j to be $0 \to j$. Flipping $(0, j)$ decreases $\delta(0, n-1)$ by 1 as the path $0 \to j \to n-1$ is gone, and does not affect $\delta(n-1, 0)$.
- Let $k \in [n]_0 \setminus \{0\}$ be even. The construction of U_n ensures the direction of the edge between 0 and k to be $k \to 0$. Flipping the $(k, 0)$ edge cannot increase $\delta(j, 0)$ for any j, and cannot decrease $\delta(0, j')$ for any j'. Since 0 was a strong king in U_n, this means 0 is a strong king in the new tournament with $(k, 0)$ flipped as well.

The first case above contributes $(n-1)/2$ elements to $D(0)$. Next, we consider the effect of flipping an edge between j and k where $j \neq 0$ and $k \neq 0$.

- If j and k are both odd, then neither of them is an in-neighbor of 0, and hence 0 remains a strong king.
- If j and k are both even and $j \to k$ is flipped, the construction of U_n ensures that $j > k$, and both j, k are in-neighbors of 0. We have $\delta(k, 0)$ increase by 1 due to the new path $k \to j \to 0$, and $\delta(0, k)$ is unchanged as $j \to 0$. By Lemma 16, the new tournament now has $\delta(k, 0) = \delta(0, k)$, and hence 0 is no longer a strong king.
- If j is even and $k > j$ is odd, the direction of the edge between j and k is $j \to k$. The vertex j is an in-neighbor of 0, and k is not. Flipping the edge increases $\delta(0, j)$ by 1 due to the new path $0 \to k \to j$, and $\delta(j, 0)$ is unaffected. No other values of $\delta(0, \cdot)$ or $\delta(\cdot, 0)$ are affected for in-neighbors of 0. Thus, 0 remains a strong king in this case.
- The final case is when j is odd and $k > j$ is even. Recall that k is an in-neighbor of 0. We have $\delta(0, k)$ *reduce* by 1 due to the removal of the path $0 \to j \to k$ (as the $j \to k$ edge is now flipped), and $\delta(k, 0)$ is unaffected. By Lemma 16, we have $\delta(0, k) = \delta(k, 0)$ in the new tournament, and hence 0 is no longer a strong king.

The number of edges from the second case above equals the number of ways of choosing two even numbers, none of which is 0: this is $\binom{(n-1)/2}{2}$. The number of edges from the fourth case is the number of choices of an odd vertex and a higher even vertex, which equals $(n-1)/2 + (n-3)/2 + \cdots + 1 = (n^2-1)/8$. Combining the above, we have

$$\begin{aligned} |D(0)| &= \frac{n-1}{2} + \frac{(n-1)(n-3)}{8} + \frac{n^2-1}{8} = \frac{n-1}{2}\left[1 + \frac{n-3}{4} + \frac{n+1}{4}\right] \\ &= \left(\frac{n-1}{2}\right)\left(\frac{n+1}{2}\right) = \frac{n^2-1}{4}. \end{aligned}$$

This concludes the proof. ◀

We now analyze the randomized query complexity of $\mathsf{STRONG\text{-}KING}_n$.

▶ **Theorem 18.** *For all positive integers n, $\mathsf{R}(\mathsf{STRONG\text{-}KING}_n) = \Theta(n^2)$ for tournaments.*

Proof. The upper bound is trivial. Assume that n is odd. For $i, j \in [n]_0$, let $U_n^{i,j}$ denote the tournament that is obtained from U_n by flipping the direction of the edge between i and j. Consider μ to be the following distribution on n-vertex tournaments:

$$\mu(T) = \begin{cases} 1/2 & T = U_n, \\ \frac{1}{\binom{n}{2}} & T = U_n^{i,j}, \end{cases} \qquad \text{for all } i,j \in [n]_0.$$

Towards a contradiction using Lemma 7, let $\mathcal{A}$ be a deterministic algorithm with cost less than $c \cdot n(n-1)$ (for some constant $c \in (0,1)$ to be fixed later in the proof, it will turn out that any $c < 1/12$ works) computing $\mathsf{STRONG\text{-}KING}_n$ to error less than $1/3$ when inputs are drawn from the distribution μ. Consider the computation path of $\mathcal{A}$ that outputs the answers of all queries consistently with the input U_n. Say the output at this leaf of $\mathcal{A}$ is vertex v. The algorithm errs on all inputs of the form $U_n^{i,j}$ where $(i,j) \in D(v)$ (recall that $D(v)$ consists of precisely those edges that, on being flipped in U_n, cause v to no longer be a strong king). There are at least $|D(v)| - cn(n-1)$ such edges (i,j) which are unqueried (and thus $U_n^{i,j}$ reaches this leaf). The error probability of $\mathcal{A}$ is at least its error probability on this leaf, which by Lemma 17 is at least $\frac{1}{\binom{n}{2}}\left(\frac{n^2-1}{4} - cn(n-1)\right) > \frac{2}{n(n-1)}\left(\frac{n(n-1)}{4} - cn(n-1)\right) = \frac{1-4c}{2} > 1/3$ for $c < 1/12$. Lemma 7 yields the required lower bound.

The proof can be easily made to work for the case where n is even by assuming that one vertex is a sink, and applying the argument above on the subtournament induced on the remaining vertices (this sink vertex does not affect the proof above in any way). ◀

A deterministic lower bound follows as an immediate corollary.

▶ **Corollary 19.** *For all positive integers n, $\mathsf{D}(\mathsf{STRONG\text{-}KING}_n) = \Theta(n^2)$ for tournaments.*

5 Conclusion

In this paper, we have shown the deterministic and randomized query complexities of deciding whether a king exists in an arbitrary n-vertex digraph to be $\Theta(n^2)$. We also showed that deterministic and randomized query complexities of finding a strong king in a tournament is $\Theta(n^2)$, answering a question of Biswas et al. [2] in the negative. Our proofs relied on showing some key properties of known constructions of balanced tournaments (tournaments where every vertex is a king). The most interesting question that remains open is to determine the deterministic query complexity of finding a king in an n-vertex tournament. The best-known upper and lower bounds are $O(n^{3/2})$ and $\Omega(n^{4/3})$, respectively, and are from over 20 years ago [19]. Perhaps our analysis of special balanced tournaments in this paper can provide insight towards approaching this general problem.

References

1 Amir Abboud, Tomer Grossman, Moni Naor, and Tomer Solomon. From donkeys to kings in tournaments. In *32nd Annual European Symposium on Algorithms, ESA*, volume 308 of *LIPIcs*, pages 3:1–3:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.ESA.2024.3.

2 Arindam Biswas, Varunkumar Jayapaul, Venkatesh Raman, and Srinivasa Rao Satti. Finding kings in tournaments. *Discret. Appl. Math.*, 322:240–252, 2022. doi:10.1016/j.dam.2022.08.014.

3 Amit Chakrabarti and Subhash Khot. Improved lower bounds on the randomized complexity of graph properties. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pages 285–296. Springer, 2001. doi:10.1007/3-540-48224-5_24.

4 An-Hang Chen, Jou-Ming Chang, Yuwen Cheng, and Yue-Li Wang. The existence and uniqueness of strong kings in tournaments. *Discrete mathematics*, 308(12):2629–2633, 2008. doi:10.1016/j.disc.2007.05.008.

5 Palash Dey. Query complexity of tournament solutions. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, pages 2992–2998. AAAI Press, 2017. doi:10.1609/aaai.v31i1.10702.

6 David C Fisher, J Richard Lundgren, Sarah K Merz, and K Brooks Reid. The domination and competition graphs of a tournament. *Journal of Graph Theory*, 29(2):103–110, 1998. doi:10.1002/(SICI)1097-0118(199810)29:2\%3C103::AID-JGT6\%3E3.0.CO;2-V.

7 Péter Hajnal. An $\Omega(n^{4/3})$ lower bound on the randomized complexity of graph properties. *Combinatorica*, 11:131–143, 1991. doi:10.1007/BF01206357.

8 Ting-Yem Ho and Jou-Ming Chang. Sorting a sequence of strong kings in a tournament. *Information processing letters*, 87(6):317–320, 2003. doi:10.1016/S0020-0190(03)00346-6.

9 Oded Lachish, Felix Reidl, and Chhaya Trehan. When you come at the king you best not miss. In *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022*, volume 250 of *LIPIcs*, pages 25:1–25:12. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.FSTTCS.2022.25.

10 HG Landau. On dominance relations and the structure of animal societies: III the condition for a score structure. *The bulletin of mathematical biophysics*, 15:143–148, 1953.

11 Nikhil S. Mande, Manaswi Paraashar, Swagato Sanyal, and Nitin Saurabh. On the communication complexity of finding a king in a tournament. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, volume 317 of *LIPIcs*, pages 64:1–64:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.APPROX/RANDOM.2024.64.

12 Nikhil S. Mande, Manaswi Paraashar, and Nitin Saurabh. Randomized and quantum query complexities of finding a king in a tournament. In *43rd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, volume 284 of *LIPIcs*, pages 30:1–30:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.FSTTCS.2023.30.

13 Stephen B Maurer. The king chicken theorems. *Mathematics Magazine*, 53(2):67–80, 1980.

14 John W Moon. *Topics on tournaments in graph theory*. Courier Dover Publications, 2015.

15 Nathan Muyinda, Bernard De Baets, and Shodhan Rao. Non-king elimination, intransitive triad interactions, and species coexistence in ecological competition networks. *Theoretical Ecology*, 13(3):385–397, 2020.

16 K.B. Reid. Every vertex a king. *Discrete Mathematics*, 38(1):93–98, 1982. doi:10.1016/0012-365X(82)90173-X.

17 Ronald L Rivest and Jean Vuillemin. On recognizing graph properties from adjacency matrices. *Theoretical Computer Science*, 3(3):371–384, 1976. doi:10.1016/0304-3975(76)90053-0.

18 Arnold L Rosenberg. On the time required to recognize properties of graphs: A problem. *ACM SIGACT News*, 5(4):15–16, 1973. doi:10.1145/1008299.1008302.

19 Jian Shen, Li Sheng, and Jie Wu. Searching for sorted sequences of kings in tournaments. *SIAM J. Comput.*, 32(5):1201–1209, 2003. doi:10.1137/S0097539702410053.

20 Andrew Chi-Chih Yao. Probabilistic computations: Toward a unified measure of complexity. In *18th Annual Symposium on Foundations of Computer Science (SFCS 1977)*, pages 222–227. IEEE Computer Society, 1977.

21 Andrew Chi-Chih Yao. Lower bounds to randomized algorithms for graph properties. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 393–400. IEEE, 1987.