How to play the majority game with a liar [1]

Steve Butler [a,*], Jia Mao [b,2], Ron Graham [b,3]

[a] *Dept. of Mathematics, UC San Diego,*
La Jolla, CA 92093-0112, USA

[b] *Dept. of Computer Science and Engineering, UC San Diego,*
La Jolla, CA 92093-0404, USA

Abstract

The *Majority* game is played by a questioner ($\mathbf{Q}$) and an answerer ($\mathbf{A}$). $\mathbf{A}$ holds n elements, each of which can be labeled as 0 or 1. $\mathbf{Q}$ is trying to identify some element $\mathbf{A}$ holds as having the majority label or in the case of a tie claim there is none. To do this $\mathbf{Q}$ asks questions comparing whether two elements have the same or different label. $\mathbf{Q}$'s goal is to ask as few questions as possible while $\mathbf{A}$'s goal is to delay $\mathbf{Q}$ as much as possible. Let q^* denote the minimal number of questions needed for $\mathbf{Q}$ to identify a majority element regardless of $\mathbf{A}$'s answers.

In this paper we investigate upper and lower bounds for q^* in a variation of the Majority game where $\mathbf{A}$ is allowed to *lie* up to t times. We consider two versions of the game, the adaptive (where questions are asked sequentially) and the oblivious (where questions are asked in one batch).

Key words: majority game, liar games, adaptive, oblivious
1991 MSC: 91A05, 91A43

1 Introduction

The *Majority* game consists of two players: a questioner **Q** and an answerer **A**. Initially **A** holds a set of n elements, each of which **A** will assign a binary label (e.g., 0 or 1), and **Q** asks questions as to whether two elements have the same or different labelling. In the game, **Q**'s goal is to identify one element of the majority label (or in the case of a tie, claim that there is none), while **A**'s goal is to delay **Q**. We say **Q** has a *winning strategy* of length q if **Q** can always win the game with at most q questions, regardless of what **A** answers. The goal of this paper is to investigate strategies with minimal q, denoted by q^*.

1.1 History

The earliest variant of the *Majority* problem was proposed by Moore in the context of fault-tolerant system design in 1981 [11]. A number of different variants were subsequently proposed and analyzed. This problem resurfaced after about twenty years in a military application where communication needs to be minimized to locate one sensor that has not been corrupted among a group of sensors [6].

There have been several variations of the majority game studied in past literature. These variations have included examining different k (the number of permissible labels); considering *adaptive* or *oblivious* versions of the game (in an adaptive game, **Q** learns the answer to each question before asking the next question, while in an oblivious game, **Q** asks all questions in one batch before getting any answers from **A**); and whether or not a majority label is known to exist or not. (See [1,3,6,7,9,14,17].)

In the adaptive case with 2 labels, Saks and Werman [14] were the first to prove a tight bound of the minimum length of a winning strategy for **Q** to be $n - \mu_2(n)$, where $\mu_2(n)$ is the number of 1s in n's binary expansion. Different proofs for the same result were subsequently given in [3] and [17]. When k is unknown, a tight bound of $\lceil 3n/2 \rceil - 2$ was given in [9]. The average case of the same setting was analyzed in [4]. Similar bounds were proved for randomized algorithms [10].

In the oblivious case, when k is unknown, the optimal winning strategy for **Q** is much harder to design or analyze. If the existence of a majority label is not known a priori, **Q** needs $\Omega(n^2)$ many questions [15]. However, if a majority label is known to exist, by using a Ramanujan graph there is a constructive strategy for **Q** that uses no more than $(1 + o(1))19.5n$ queries [7].

1.2 Error tolerance

Previously **A** was assumed to be truthful, i.e., answers to **Q**'s questions would always be consistent. However, an *error-tolerant* feature is desired when the answers to the queries in the application may be faulty (i.e., inconsistent) due to communication errors. In this paper we address this issue by putting the *Majority* game in a broader context of fault-tolerant communication, namely, searching games with a bounded number of errors. Generally, these games are more difficult to analyze, but have a much wider range of applications compared to perfect information two person games. One such famous game is the *Rényi-Ulam liar game* [13,16]. For a comprehensive overview of this topic, we refer the reader to a recent survey [12].

In this paper we consider bounded error tolerance for the *Majority* game, where **A** is allowed to *lie up to a fixed number t times*. More precisely, this means that after **Q** specifies an element as being in the majority or state there is none, **A** has the freedom to flip up to t answers of the previously asked queries and reveal a labelling that is consistent with this modified set of answers. Now that **A** can lie how much will this handicap **Q**? What is the new minimal q^* and what strategy should **Q** adopt to achieve this bound? How should **A** force **Q** to ask at least q^* questions?

In this paper we will begin to answer some of these questions for the *Majority* game with binary labels. We will give upper and lower bounds for q^* by producing strategies for **Q** and **A** in various versions of the game. We summarize our results in the table below, where t is the number of lies and n is the number of objects.

Game	n is odd	n is even
Adaptive $t = 1$	$n \leq q^* \leq n + 1$	$n + 1 \leq q^* \leq n + 2$
Adaptive $t > 1$	$\lceil \frac{t+3}{4}n - \frac{t+1}{4} \rceil \leq q^* \leq (\frac{t+1}{2} + o(1))n$	$\frac{t+1}{2}n \leq q^* \leq (\frac{t+1}{2} + o(1))n$
Oblivious $t \geq 1$	$q^* = \lceil (t + \frac{1}{2})n \rceil$	

The upper bound in the adaptive case for $t > 1$ holds when $t = o(n^{1/2})$. A more precise statement of the upper bound is given by Theorem 4.

Also note the difference between the case when n is odd and n is even. This seems to be due to the fact that when n is odd there *must* be a majority element which gives additional information **Q** can use in forming a strategy.

3

2   Adaptive Setting

We can adapt the known strategy for the game with no lies to give a simple upper bound for q^*. Recall that for the majority game of binary labels with no lies the optimal winning strategy takes $(n - \mu_2(n))$ queries.

Theorem 1 *In the adaptive Majority game on n elements with binary labels and at most t lies,*

$$q^* \leq (t+1)(n - \mu_2(n)) + t$$

where $\mu_2(n)$ is the number of 1s in n's binary expansion.

PROOF. Let **Q** ask the same queries as in the best strategy to play the *Majority* game with no lies, only that each query is repeated until $(t+1)$ answers agree (at which point the relationship between the two elements is known) before going to the next query. Because **A** is not allowed to lie more than t times, the total number of queries **Q** needs to ask is $(t+1)(n - \mu_2(n))$ plus at most t.   □

This simple bound for q^* can be improved by **Q** using error detection and correction from the answers of **A**.

2.1   Preliminaries

To keep track of the game as it progresses, we use an auxiliary (multi-)graph. The n objects are the vertices and edges correspond to comparisons between elements. (We will allow queries to be repeated and any multiple queries are represented by multi-edges.) As the game progresses **Q** will give **A** an edge and ask **A** to color it blue if the two elements have the same label and red if the two elements have different labels.

One of the most important tools used in the formation of a strategy is to use the auxiliary graph to detect lies. For instance, note that if **A** were truthful then every cycle would always have an even number of red edges. But when **A** can lie this no longer needs to be the case and we have the following observation.

Observation 1 *A cycle can have an odd number of red edges if and only if an odd number of edges in the cycles correspond to lies.*

This follows by noting that the number of red edges corresponds to the number of times along the cycle that we switch labels. Initially if there were no lies we would have to switch an even number of times (i.e., we must return to the same label we started with). Now for every lie we either increase or decrease by 1 the number of switches made.

From the observation any cycle which contains an odd number of red edges must contain a lie, we will refer to such cycles as invalid, otherwise we say that the cycle is valid. It is important to notice though that just because a cycle is valid it does not imply that there are no lies, only that there are an even number of them.

Finally, we will say a component is even or odd depending on whether there are an even or odd number of vertices.

2.2 Majority Game with at most $t = 1$ lie

Theorem 1 gives an upper bound of $(2(n-\mu_2(n))+1)$ for **Q**'s adaptive strategy when **A** is allowed to lie once. This bound can be improved by using *validity checking* of the auxiliary graph. (For the case involving one lie we have that a cycle is valid if and only if it has no lies, in general this will not hold.)

Theorem 2 *In the adaptive Majority game on n elements with binary labels and at most 1 lie*

$$
q^* \leq \begin{cases} n+1 & \text{if } n \text{ is odd,} \\ n+2 & \text{if } n \text{ is even.} \end{cases}
$$

PROOF. We give a two-stage strategy for **Q** satisfying the bounds.

Stage 1: In the first stage **Q** starts by growing long blue paths by the following rule: connect the ends of two blue paths which have an equal number of vertices (we consider isolated vertices as paths on one vertex). This continues until either there are no two paths with the same number of vertices or we get a red edge.

In the first case **Q** takes the longest blue path and closes it up by a single question, if the cycle is blue then **Q** identifies any element on the cycle as a majority element (i.e., in such a case it is easy to see that the cycle contains more than half of the elements and all of them must have the same label), otherwise if the edge is red (i.e., the cycle is invalid so contains the lie) **Q** goes to the second stage.

In the second case there is a path with one single red edge in the middle. **Q** then asks a single question to close it up to form a cycle. If the new edge is red, then the coloring is valid and so the cycle has an equal number of each label, **Q** then removes this component from the graph and continues as before. If the new edge is blue (i.e., the cycle is invalid so contains the lie) **Q** goes to the second stage.

Stage 2: Starting stage 2 we already know that **A** has used their lie and so all subsequent answers must be true. We initially have one cycle with a single red edge (denoted by $\{u, v\}$) and possibly several blue paths. The first step in this stage is to connect one vertex from each blue path to u. The graph now consists of a cycle with a tree attached to u. At this point **Q** has asked exactly n questions.

Because the lie is in the cycle, all edges involved in the tree reflect truthful responses. In particular, **Q** can determine how many of the vertices of the cycle must have the same label as u in order for u to be the majority element, denote this number by k. Starting at u and going in the opposite direction of v, **Q** counts out k vertices. Denote the kth vertex by w. (If the cycle contains fewer than k vertices then u is in the minority and it will be easy to identify a majority element in the tree. Similarly, if $k \leq 0$ then u has a majority label.)

Q now queries the edge $\{u, w\}$. If the edge is blue then all the vertices between u and w have the needed label and we can conclude that u is a majority element. On the other hand if the edge is red then there is a lie somewhere between u and w and so u cannot be a majority element. In the case when n is even we then only need to compare u with the vertex that precedes w to test if there is a tie. If the edge is blue then there is a tie, while if the edge is red then w is a majority element.

The result now follows by counting the number of queries used. □

To find a lower bound for q^* we need to give a strategy for **A**. Since **A** can adopt the same strategy as in the game with no lies we have that $q^* \geq n - \mu_2(n)$. However, **A** can do better as shown in the next theorem.

Theorem 3 *In the adaptive Majority game on n elements with binary labels and at most 1 lie*

$$
q^* \geq \begin{cases} n & \text{if } n \text{ is odd,} \\ n+1 & \text{if } n \text{ is even.} \end{cases}
$$

PROOF. The case for n odd will follow from Theorem 5 with $t = 1$. So here we consider the case n even.

We need to give a strategy for **A** satisfying the bounds. One such strategy is as follows: answer the first $n - 1$ questions consistently in such a way that the difference between the number of vertices with the two labels for an even component is 0 and for an odd component is 1 (that this is always possible is an easy exercise left to the reader). For the nth question, if the resulting query would result in the graph consisting of disjoint even cycles then answer to form an invalid cycle, otherwise answer as before. After the nth question, fix an admissible labelling consistent with the given answers and be truthful afterwards.

We now need to show that after asking the nth question **Q** cannot distinguish between the existence of a majority and a tie. We note that by this strategy **A** can always produce an admissible labelling which has a tie. So we only need show that we can also have a labelling which has a majority element.

If the graph after the nth question did not solely consist of cycles then there is a vertex of degree ≤ 1 for which **A** can reverse the label (since **A** has a lie available) and thus produce a labelling with a majority element. Similarly, if the graph after the nth question solely consists of cycles, some of them odd, then by reversing the labelling on an odd cycle **A** can produce a majority element.

Suppose now that **A** answered to form an invalid cycle at the nth step, i.e., the graph after the nth step consists of even cycles only. **Q** can now not distinguish between **A** lying at the nth step, in which case the labelling would be a tie, or lying at a step corresponding to an edge adjacent with the nth step, in which case there would be a majority element. □

2.3 *Majority Game with at most $t \geq 2$ lies*

We now consider the game where **A** is allowed to lie up to $t \geq 2$ times. We first start by establishing an upper bound.

Theorem 4 *In the adaptive Majority game on n elements with binary labels and at most $t \geq 2$ lies,*

$$q^* \leq \frac{t+1}{2}n + 6t^2 + 2t + 3\log n.$$

PROOF. We give a strategy for **Q** which will use two rounds. The first round will be "oblivious" where **Q** will always ask the same set of questions (this round will use $(t + 1)n/2$ questions). In the second round **Q** then uses the answers from the first round to find and correct all lies.

We first consider the case $n > 2t$ with n even.

In the first round $\mathbf{Q}$ forms an n-cycle with the n vertices and asks $\lfloor t/2 \rfloor$ questions on each edge of the cycle. $\mathbf{Q}$ then makes

$$
t + 1 - 2 \left\lfloor \frac{t}{2} \right\rfloor =
\begin{cases}
1 & \text{if } t \text{ is even,} \\
2 & \text{if } t \text{ is odd,}
\end{cases}
$$

queries between opposite vertices of the cycles (we will refer to these queries as spokes). An example is shown in Figure 1.

Fig. 1. "Oblivious" first round queries.   Fig. 2. Looking for lies in the spokes.

This strategy has the following useful property.

Claim: If $\mathbf{A}$ has lied we can find an invalid cycle.

To establish this claim we consider possible locations for lies. Namely they can occur in spokes, or somewhere on the cycle. In the latter case we will distinguish the special case of when lies saturate opposite intervals (i.e., in Figure 2 such a set of intervals would be the edges between u_1 and u_2 along with the edges between u_1' and u_2').

- *There is a lie in the spokes.* Since $n > 2t$ then not all of the spokes can be lies. In particular, visiting the spokes in turn there will be two consecutive spokes (say spokes $u_1 u_1'$ and $u_2 u_2'$) so that the first is not a lie and the second is a lie, then continuing we will find another set of spokes (say spokes $v_1 v_1'$ and $v_2 v_2'$) so that the first is a lie and the second is not a lie. (See Figure 2, note it might happen that $u_1 u_1'$ ($u_2 u_2'$) and $v_2 v_2'$ ($v_1 v_1'$) are the same spoke.)

  In order for all cycles of the form $u_1 u_2 u_2' u_1'$ to be valid then either all queries between u_1 and u_2 were lies or all queries between u_1' and u_2' were lies. Similarly in order for all cycles of the form $v_1 v_2 v_2' v_1'$ to be valid then either all queries between v_1 and v_2 were lies or all queries between v_1' and v_2' were lies.

  In the case that t is even we would need at least $2(t/2) + 1$ lies to validate all of the cycles which is impossible. In the case that t is odd we would need at least $2 \lfloor t/2 \rfloor + 1 = t$ lies, in particular there could be at most one lie used in the spokes, however recall that for t odd each spoke is asked twice and

8

in this case there must be a two cycle of the form u_2u_2' which is invalid and so we can still find an invalid cycle.

- *No spoke is a lie, and some set of opposite intervals on the cycle are not completely filled with lies.* In this case we can find some cycle similar to $u_1u_2u_2'u_1'$ which is invalid.
- *There is some set of opposite intervals of the cycle which are completely filled with lies, and no other lies are present.* We first point out that it must be the case that only one pair of opposite intervals are fully filled of lies (i.e., not enough lies to do this with more than one pair). In this case we can find a cycle of length $n/2$ which is invalid, namely start with any spoke and form an $n/2$ cycle by going over the spoke and then use edges along the outer cycle to close our new cycle.

This concludes the proof of the claim.

We can quickly find and remove all errors from invalid 2-cycles and 4-cycles. Namely, for each such cycle we make $2t$ queries on each of 3 edges and take the majority answer on each edge, if we have not yet found the lie from this then the remaining edge is a lie and we can correct. Thus we would need at most $6t^2$ queries to correct these lies.

If after correcting these queries there is still an invalid $n/2$ cycle then it must be the case that there are two opposite intervals "saturated" with lies (i.e., all queries between u_1u_2 and $u_1'u_2'$ in Figure 2 are lies). In particular, there can only be at most one lie left. We now lift up the $n/2$ cycle (which will have only one lie) and locate the lie by using a splitting technique. Namely we join two opposite pairs of vertices with three edges and take the majority answer and use this to split the cycle into two smaller cycles, one of which will be invalid (and which contains the lie of the $n/2$ cycle). We then continue this process of cutting in half each time until we have located the lie. In particular, this technique will locate the lie in $3 \log n$ steps.

Q is now finished because they can detect and correct lies given by **A** and relate all elements together. In particular, **Q** has used at most $(t+1)n/2+6t^2+3\log n$ queries to accomplish this.

For the case n odd **Q** sets aside a single element and runs the procedure and then at the end connects the element back into the graph by making at most $2t + 1$ queries relating the odd element out with some arbitrary element.

Finally for the case $n \leq 2t$ we can simply build a tree where we keep asking questions on each edge until we get $t + 1$ responses which agree. In this case we would need at most $2t^2 + t$ queries in such a case.

Putting it all together gives the desired result. □

To establish the lower bound, we will make use of the following general observation.

Observation 2 *If the coloring is valid (i.e., no lies are detected), then $\mathbf{Q}$ will not be able to determine the correct relationship for an element which is involved in no more than t queries.*

This observation follows by noting that since the coloring is valid and $\mathbf{A}$ is allowed to change the coloring on up to t edges, then $\mathbf{A}$ can change all the queries involved with a vertex of low degree (i.e., no more than t) and still produce an admissible labelling.

Theorem 5 *In the adaptive majority game on n elements with binary labels and at most $t \geq 1$ lies,*

$$
q^* \geq \begin{cases} \dfrac{t+1}{2}n & \textit{if } n \textit{ is even,} \\[2mm] \left\lceil \dfrac{t+3}{4}n - \dfrac{t+1}{4} \right\rceil & \textit{if } n \textit{ is odd.} \end{cases}
$$

PROOF. For the case n even, $\mathbf{A}$ can use the following strategy: initially assign half of the elements with label 0 and the other half with label 1 and answer all of the questions truthfully. If $\mathbf{Q}$ makes fewer than $(\frac{t+1}{2})n$ queries, then by degree considerations there is a vertex with degree at most t. Based on the above observation, $\mathbf{Q}$ will be unable to determine the correct relationship of that vertex with the remaining elements and in particular will not be able to distinguish between a tie and the existence of a majority element.

For n odd, $\mathbf{A}$ will try to employ a similar strategy. But since n cannot be evenly split the strategy becomes more involved to try to keep the labelling in balance and still forcing $\mathbf{Q}$ to make many queries. Now as $\mathbf{A}$ answers the questions he will keep track of two graphs, the first is the auxiliary graph which we have been using so far. The second is an underlying "skeleton" subgraph. The two graphs will always have the same vertices and edge coloring, but in the skeleton graph we only include an edge if it connects two previously disconnected components.

Given W, a connected subset of the vertices of the skeleton graph, let W_0 and W_1 denote (respectively) the vertices of W labelled 0 and 1. Then let $\delta(W) = ||W_0| - |W_1||$ while $D(W_0)$ and $D(W_1)$ denote (respectively) the sum of the degrees of vertices of W labelled 0 and 1.

The strategy we give for $\mathbf{A}$ is based on the following three rules (where initially all vertices are labelled arbitrarily):

- Answer so that all even components in the skeleton graph have $\delta = 0$ and all odd components in the skeleton graph have $\delta = 1$. For any query involving only one component answer consistently with the already given answers.
- For a query connecting an even component with an odd component, both answers are allowable (i.e., can be made consistent by perhaps a switching of the labelling on the even component), in this case answer so that if W is the newly formed component then $D(W_1) \geq D(W_0)$.
- For a query connecting two odd components or two even components, answer so that $\delta = 0$ for the newly formed component, then if $D(W_1) < D(W_0)$ reverse the labelling on the newly formed component.

The key element of this strategy is that at each stage and for each connected component W we have that $D(W_1) \geq D(W_0)$.

To verify that the second portion of **A**'s strategy is always possible we note that if the query involves at least one vertex with a label of 1 then it easily holds by **A** answering truthfully. So now suppose both vertices are labelled 0. If X is the even component and $D(X_0) = D(X_1)$ then reverse the labelling on the even component and answer truthfully. On the other hand if $D(X_1) \geq D(X_0) + 2$ then it is easy to check that **A** can answer truthfully without needing to switch the labelling. This covers all possible conditions since $D(X_1) \neq D(X_0)+1$ (i.e., the sum of degrees would be odd which is impossible).

If there were three components for which $\delta = 1$ then it is impossible for **Q** to identify a majority element (i.e., win the game). So if **Q** is in a position to win there must be exactly one odd component. We now will compute the minimum number of queries that **Q** needs to ask in order to be in a winning position.

If there is any vertex of degree at most t in an even component then it is easy to verify that **Q** cannot win (i.e., **A** can, if needed, switch the labelling of the even component containing the vertex of degree at most t and/or change all of the queries involving that vertex to lies). Therefore if there are m vertices in the odd component, the sum of the degrees in the even components will be at least $(n - m)(t + 1)$.

Now let us consider the odd component (denoted by W). By the strategy employed by **A** we know that $D(W_1) \geq D(W_0)$. On the other hand $D(W_1) + D(W_0) = 2m - 2$. Combining these statements we have that $D(W_1) \geq m - 1$. Now let us consider the vertices labelled 0, it is easy to verify that if there are two vertices labelled 0 with degree at most t then **Q** cannot win. Therefore the sum of the degrees in the odd component of the auxiliary graph must be at least $(t + 1)(m - 1)/2 + 1 + (m - 1)$.

Thus the minimum number of questions that $\mathbf{Q}$ must ask (given that $n - m$ of the vertices are in even components) is:

$$\frac{1}{2}[(n-m)(t+1) + (t+1)\frac{m-1}{2} + 1 + (m-1)] = \frac{1}{2}[(t+1)n - \frac{t+1}{2} + m(\frac{1-t}{2})].$$

Since this is minimized when $m = n$ (i.e., no even components) it follows that the fewest possible number of questions that $\mathbf{Q}$ needs is at least $(t+3)n/4 - (t+1)/4$ giving the result. $\square$

3 Oblivious Setting

In the oblivious setting, $\mathbf{Q}$ has to specify all the edges in the auxiliary graph G before $\mathbf{A}$ colors any of them. This implies that $\mathbf{Q}$ has to accomplish *detection* and *location* of lies simultaneously. We have another important observation.

Observation 3 *In the Majority game of binary labels with at most t lies, if an edge e is part of $2t$ cycles that pairwise edge-intersect only at e (though they might share many vertices in common), then a lie is located at e if and only if at least $(t+1)$ of these cycles are invalid.*

This observation follows by noting that if an edge corresponds to a truthful answer then there can be at most t of the $2t$ cycles intersecting at e which can be invalid. On the other hand if the edge corresponded to a lie then at most $t-1$ of the $2t$ cycles intersecting at e can be valid, or equivalently, at least $t+1$ invalid cycles.

Theorem 6 *In the oblivious Majority game on n elements with binary labels and at most $t \geq 1$ lies,*

$$q^* = \left\lceil (t+\frac{1}{2})n \right\rceil.$$

PROOF. We first establish the upper bound. The observation above implies that if we can construct an auxiliary graph for $\mathbf{Q}$ such that for any particular edge we can find $2t$ cycles that are pairwise edge-joint only at that edge, we can locate and thus correct all possible lies with no more queries needed.

We handle the base cases first. For $n = 2$, we use $(2t+1)$ edges for the same query. For $n = 3$, the query graph is a triangle with one query asked t times and the other two queries each asked $(t+1)$ times.

12

For even $n \geq 4$, we construct a multigraph as shown in Figure 3 where all edges in the outer cycle are multi-edges (repeated t times) and single edges (or spokes) connect each pair of opposite vertices. The total number of edges is therefore $(t + \frac{1}{2})n$. For odd $n \geq 3$, first construct a graph as in the $n + 1$ case and then contract a set of edges on the outer cycle, an example is shown in Figure 4. In this case it can be checked that there are $\lceil (t + \frac{1}{2})n \rceil$ edges in the graph.
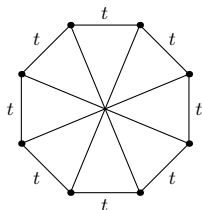

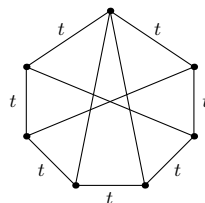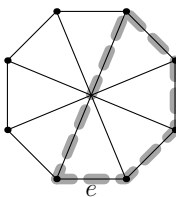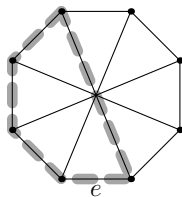Fig. 3. Oblivious graph, n even.   Fig. 4. Oblivious graph, n odd.

Fig. 5. The remaining two cycles for a side edge e.

For each spoke, we can find t cycles using one half of the outer cycle and another t using the other half. For each outer edge e, first we can find $(t - 1)$ small cycles by joining it with the other $(t - 1)$ multiedges with the same endpoints, then we use edges in the outer cycle to obtain another $(t-1)$ cycles. We need two more cycles and these are constructed using the spokes and the remaining unused edges of the outer cycle as shown in Figure 5. Because each edge lies in at least $2t$ cycles pairwise edge-joint only at that edge, all lies can be located and hence corrected. Establishing the upper bound.

For the lower bound, we can again use a degree argument. If $\mathbf{Q}$ asks fewer than $\lceil (t + \frac{1}{2})n \rceil$ then since $\mathbf{Q}$'s strategy is oblivious, $\mathbf{A}$ can examine the entire auxiliary graph and find a vertex v with degree at most $2t$. $\mathbf{A}$ then splits the remaining vertices into two sets U and V as equally as possible (i.e., $||U| - |V|| \leq 1$) with label 0 and 1 respectively. For queries not involving v, $\mathbf{A}$ answers truthfully. For queries involving v, $\mathbf{A}$ answers half of the queries as if v is labelled 0 and the other half as if v is labelled 1. This is possible because $\mathbf{A}$ is allowed to lie up to t times. Now $\mathbf{Q}$ cannot distinguish which half are lies and hence cannot determine the label for v which is essential because the other vertices are in an (almost) exact balance. This establishes the lower bound and concludes the proof. □

4   Conclusion and Remarks

Motivated by the practical need of an *error-tolerant* feature, we have concentrated on optimizing **Q**'s strategy in the presence of lies (or errors) for binary labels in the *Majority* game. We point out that when the number of lies is upper bounded by a constant t, **Q** can still win the game with a linear (in n) number of questions. Upper and lower bounds on the length of **Q**'s optimal strategy were derived in both the adaptive setting and the oblivious setting.

Consideration of fault-tolerance may also be useful for the many other variants of the *Majority* game, such as when the number of different labels is more than two. A natural generalization of the Majority game is the *Plurality* game where **Q** wants to identify one element of the *plurality* label (most frequently occurring), still using only pairwise equal/unequal label comparisons of elements. Much attention has been given to designing adaptive strategies (deterministic or randomized) for fixed or unknown k (see [2,5,8,10,15]). We remark here that the same reasoning of Theorem 1 applies to existing bounds for all variants of the *Majority* game (including the *Plurality* game) if the maximum number t of lies allowed is fixed. The new upper bounds will only be at most worse by a multiplicative term $(t + 1)$ and an additive term t.

In this paper, we gave a complete picture for the *oblivious* setting in the *Majority* game with a constant bounded number of lies. In the *adaptive* setting, however, there are still various gaps between the upper and lower bounds obtained. Closing these gaps would be interesting to pursue. In the meantime, other types of error-tolerance may also be considered, such as bounded error fraction or random errors.

Acknowledgements

References

[1]   M. Aigner, "Variants of the majority problem", *Applied Discrete Mathematics* **137** (2004), 3–25.

[2] M. Aigner, G. De Marco, M. Montangero, "The plurality problem with three colors and more", *Theoretical Computer Science* **337** (2005), 319–330.

[3] L. Alonso, E. Reingold, R. Schott, "Determining the majority", *Information Processing Letters* **47** (1993), 253–255.

[4] L. Alonso, E. Reingold, R. Schott, "Average-case complexity of determining the majority", *SIAM J. Computing* **26** (1997), 1–14.

[5] L. Alonso, E. Reingold, "Determining plurality", *manuscript*, 2006.

[6] F. R. K. Chung, R. L. Graham, J. Mao, and A. C. Yao, "Finding favorites", *Electronic Colloquium on Computational Complexity (ECCC)* (2003) 078, 15pp.

[7] F. R. K. Chung, R. L. Graham, J. Mao, and A. C. Yao, "Oblivious and adaptive strategies for the majority and plurality problems", *COCOON 2005*, 329–338; *Algorithmica*, to appear.

[8] Z. Dvořák, V. Jelínek, D. Král, J. Kynčl, and M. Saks, "Three optimal algorithms for balls of three colors", *STACS 2005*, Lecture Notes in Comp. Sci. **3404**, Springer, Berlin, 2005, 206–217.

[9] M. J. Fischer and S. L. Salzberg, "Finding a majority among n votes", *J. Algorithms* **3** (1982), 375–379.

[10] D. Král, J. Sgall, and T. Tichý, "Randomized strategies for the plurality problem", *manuscript*, 2005.

[11] J. Moore, "Proposed problem 81-5", *J. Algorithms* **2** (1981), 208–210.

[12] A. Pelc, "Searching games with errors — fifty years of coping with liars", *Theoret. Comput. Sci.* **270** (2002), 71–109.

[13] A. Rényi, "On a problem in information theory", *Magyar Tud. Akad. Mat. Kutató Int. Közl.* **6** (1961), 505–516.

[14] M. Saks and M. Werman, "On computing majority by comparisons", *Combinatorica* **11** (1991), 383–387.

[15] N. Srivastava, and A. D. Taylor, "Tight bounds on plurality", *Information Processing Letters* **96** (2005), 93–95.

[16] S. M. Ulam, **Adventures of a mathematician**, Charles Scribner's Sons, New York, 1976, xi+317pp.

[17] G. Wiener, "Search for a majority element", *J. Statistical Planning and Inference* **100** (2002), 313–318.