On randomized algorithms for the majority problem

Demetres Christofides*

Abstract

In the majority problem, we are given n balls coloured black or white and we are allowed to query whether two balls have the same colour or not. The goal is to find a ball of majority colour in the minimum number of queries. The answer is known to be $n - B(n)$ where $B(n)$ is the number of 1's in the binary representation of n. In this paper we study randomized algorithms for determining majority, which are allowed to err with probability at most ε. We show that any such algorithm must have expected running time at least $\left(\frac{2}{3} - o(1)\right) n$. Moreover, we provide a randomized algorithm which shows that this result is best possible. These extend a result of De Marco and Pelc [6].

1 Introduction

In the 'majority problem', we are given n balls coloured black or white. At any stage we are allowed to select two balls and ask whether they have the same colour or not. Our task is to find a ball of majority colour, or decide that no such ball exists. How many questions do we need to ask, in the worst case? Clearly $n - 1$ questions suffice. For example, we can compare the first ball with all the rest.

The following recursive algorithm does slightly better (at least when n is not a power of 2). If n is odd, it is enough to determine majority in the first $n - 1$ balls. Indeed, a ball which is in majority colour when restricted to the first $n - 1$ balls, is also in majority in the totality of n balls. On the other hand if no majority exists in the first $n - 1$ balls, then the n-th ball is in majority. If n is even, we can pair the balls arbitrarily, make $n/2$ comparisons, throw out all pairs for which the colours were different, and keep one ball from each pair for which the colours were the same. Then, clearly, it is enough to determine majority in the balls left. An easy inductive argument now shows that this algorithm determines majority in at most $n - B(n)$ questions, where $B(n)$ is the number of 1's in the binary representation of n. Saks and Werman [9] showed that in the worst case we do need that many questions. The problem was also solved later, by Alonso, Reingold and Schott [3] and Wiener [10], using different methods. See [1, 2] for surveys on the majority problem and some of its variants.

*Supported by grants from the Engineering and Physical Sciences Research Council and from the Cambridge Commonwealth Trust.

What happens if we allow some randomization in our algorithm for determining majority? To be more precise, at each step we are allowed to pick the two balls to be compared using some probability distribution which is allowed to depend on our current knowledge so far. We allow our algorithm to err with probability at most ε. In other words, given any input, the randomized algorithm must produce a correct answer with probability at least $1-\varepsilon$. To the best of our knowledge, this was first studied by De Marco and Pelc [6]. They showed that randomization does not improve the running time by much, in the sense that there are inputs for which the expected running time of any randomized algorithm is linear. In fact, they showed that if the difference between the number of black and white balls is bounded, then any randomized algorithm which errs with probability at most ε has expected running time $\Omega(n)$ on some input. Although it does not appear explicitly in their work, their proof shows that the expected running time is at least $n/40$. We already know that one can solve the majority problem in n steps (even without randomization) so it is natural to ask what the right constant for the speed of the running time (in the worst case) is. This question is answered by the following two theorems.

Theorem 1. *For every $\delta > 0$, there exists an $\varepsilon = \varepsilon(\delta) > 0$ such that, whenever n is large enough (depending on δ), any randomized algorithm for determining majority on n balls, with expected running time less that $\left(\frac{2}{3} - \delta\right) n$, errs with probability at least ε on some input.*

Theorem 2. *Given $\varepsilon > 0$, if n is large enough (depending on ε), then there is a randomized algorithm for determining majority on n balls which errs with probability at most ε and has expected running time at most $\frac{2}{3}\left(1 - \frac{\varepsilon}{3}\right) n$.*

In the proofs of the theorems we will be rather crude with our estimates, and indeed it can be easily seen that by being more careful one can obtain better estimates for the dependence on ε. Note also that, although we have stated the results for large enough n, one can immediately deduce corresponding results holding for any n, provided that one subtracts or adds a constant term to the expected running times in Theorem 1 and Theorem 2 respectively.

In Section 2 we will prove Theorem 1, and in Section 3 we will prove Theorem 2. In Section 4 we provide a negative answer to a related question of De Marco and Pelc [6], who asked whether the majority problem can be solved with constant error probability in sublinear expected time provided that the difference in the number of black and white balls is $O(\sqrt{n})$.

2 Proof of Theorem 1

Observe that a randomized algorithm is nothing else than a probability distribution on the set of all deterministic algorithms. Thus, it seems reasonable to expect that a good understanding of the behaviour of every deterministic algorithm, will yield a good understanding on the behaviour of randomized algorithms as well. Indeed our approach will be to find a random input such that every deterministic algorithm

which fails with probability at most ε, has large expected running time on that input. Having proved this, a simple averaging argument will yield the required result. (The observation that in order to find lower bounds on the running time of randomized algorithms it is enough to find lower bounds on the running time of each deterministic algorithm is due to Yao [11].)

So we begin by looking at deterministic algorithms. As observed by previous authors [9, 3, 10], our knowledge after each step can be described by a graph G, on vertex set $[n]$, where i is joined to j if and only if we have already compared ball i to ball j. The edges of G are labelled with a YES or a NO, depending on the answer we have obtained. Within each component, we have enough information to determine whether two balls have the same colour or not. Let M_i be the difference, in absolute value, between the number of black and white balls in component i. We will ignore the components where the difference is 0, and order the other components so that $M_1 \geqslant M_2 \geqslant \ldots \geqslant M_C$. So, regarding the majority problem, the vector $(M_1, \ldots, M_C)$ contains all the information that we are interested in.

Fix a deterministic algorithm A for determining majority, where the possibility of error is allowed. We write $T = T(A)$ for the time taken by the algorithm to terminate, $C = C(A)$ for the number of non-zero components left, and $M_1 = M_1(A)$ for the largest difference in the sizes of the colour classes within the components when the algorithm terminates. Colour the balls independently and uniformly at random. Then T, C and M become random variables.

To begin with, we will need the following result from [4]. We provide a proof both for completeness and because our proof is considerably shorter.

Lemma 3. *Let A be as above and suppose that the balls are coloured independently and uniformly at random. Then $\frac{3}{2}\mathbb{E}(T) + \mathbb{E}(C) \geqslant n$.*

Proof. We proceed by induction. For technical convenience, the induction is on the sum, over all possible 2^n colourings, of the number of steps taken by the algorithm in each colouring. If this is 0, then the algorithm does not take any step (whatever the colouring) so $\mathbb{E}(T) = 0$, $\mathbb{E}(C) = n$ and the result follows. Moreover, whenever A takes an extra step (with respect to another algorithm) by comparing balls from components C_i and C_j, then either $M_i \neq M_j$ and the number of non-zero components decreases by 1, or $M_i = M_j$ and the number of non-zero components decreases on average by 3/2. In both cases, $\frac{3}{2}\mathbb{E}(T) + \mathbb{E}(C)$ increases, and so we are done. □

Our next and crucial step in the proof will be to show that if A succeeds with large (but constant) probability when the balls are coloured independently and uniformly at random, then $\mathbb{E}(C)$ is small and thus, by Lemma 3, $\mathbb{E}(T)$ is large.

Theorem 4. *Let A be any deterministic algorithm which errs with probability at most ε when the balls are coloured independently and uniformly at random. Then, provided n is large enough (depending on ε), we have that $\mathbb{E}(T) \geqslant \frac{2}{3}(1-\gamma)n$, where $\gamma = \gamma(\varepsilon) \to 0$ as $\varepsilon \to 0$.*

To prove Theorem 4 we will need some estimates for sums of binomial coefficients. We will use the symmetric case of the de Moivre-Laplace Theorem, i.e. the normal approximation to the binomial distribution in the case $p = 1/2$. Before stating the theorem, let us recall that

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2}\, dt,$$

is the distribution function of the standard normal distribution.

Theorem 5 (De Moivre-Laplace; symmetric case)**.** *Let S_n be the number of successes in n independent Bernoulli trials each with probability of success $1/2$. Then*

$$\Pr\left(n/2 - x_1\sqrt{n} < S_n < n/2 + x_2\sqrt{n}\right) \sim \Phi(2x_2) - \Phi(2x_1).$$

For a proof of this theorem, we refer the reader to the classic book of Feller [8, Chapter VII].

For the proof of Theorem 4 we will also need a result of Erdős [7] on the Littlewood-Offord problem. The problem is the following. Given real numbers $x_1, x_2, \ldots, x_n$ all of modulus at least one, at most how many of the sums $\sum_{i=1}^{n} \varepsilon_i x_i$, where $\varepsilon_i \in \{-1, 1\}$, can lie in the interval $[-r, r]$? (In fact, the problem is more general, but the solution to this version of the problem will do for our purposes.) If all the x_i's have modulus exactly one, then all sums in which exactly k of the ε_i's are equal to 1, where $k \in [\lceil (n-r)/2 \rceil, \lfloor (n+r)/2 \rfloor]$, lie in $[-r, r]$. Erdős gave an elegant combinatorial proof that this example is in fact best possible.

Theorem 6 (Erdős [7])**.** *Let $x_1, \ldots, x_n$ be real numbers of modulus at least 1. Then the number of sums $\sum_{i=1}^{n} \varepsilon_i x_i$, where $\varepsilon_i \in \{-1, 1\}$, lying in the interval $[-r, r]$ is at most*

$$\sum_{k=\lceil (n-r)/2 \rceil}^{\lfloor (n+r)/2 \rfloor} \binom{n}{k}.$$

Proof of Theorem 4. Suppose that when A announces a ball of majority colour, there are $C \geqslant \varepsilon^{1/2} n$ components left, of sizes $M = M_1 \geqslant M_2 \geqslant \ldots \geqslant M_C \geqslant 1$, with $M \leqslant \alpha C^{1/2}$, for some α to be determined later. The probability that the announced ball is not in the majority is at least

$$\Pr(M < \varepsilon_2 M_2 + \ldots + \varepsilon_C M_C) = \frac{1}{2} \Pr(\varepsilon_2 M_2 + \ldots + \varepsilon_C M_C \notin [-M, M]),$$

where the ε_i take the values ± 1 uniformly and independently at random. By Theorem 6, this is at least

$$\frac{1}{2} - \frac{1}{2^{C-1}} \sum_{k=\lceil (C-1-M)/2 \rceil}^{\lfloor (C-1+M)/2 \rfloor} \binom{C-1}{k}.$$

But if n is large enough, then by the de Moivre-Laplace Theorem, this is at least $\Phi(-2\alpha) - \varepsilon^{1/2}$. In particular, if $\alpha \leqslant -\frac{1}{2}\Phi^{-1}\left(2\varepsilon^{1/2}\right)$, then this probability is at least $\varepsilon^{1/2}$.

Since A fails with probability at most ε, it follows that with probability at least $1-\varepsilon^{1/2}$, either $C \leqslant \varepsilon^{1/2}n$, or $M \geqslant -\frac{1}{2}\Phi^{-1}\left(2\varepsilon^{1/2}\right)C^{1/2}$. Therefore

$$\mathbb{E}(C) \leqslant 2\varepsilon^{1/2}n + \left(\frac{2}{\Phi^{-1}(2\varepsilon^{1/2})}\right)^2 \mathbb{E}(M^2).$$

We claim that $\mathbb{E}(M^2)$ cannot be too large. Indeed, it is easily seen that $\mathbb{E}(M^2)$ increases after every step, and so

$$\begin{aligned}\mathbb{E}(M^2) &\leqslant \frac{1}{2^n}\sum_{k=0}^{n}(n-2k)^2\binom{n}{k} \\ &= \frac{1}{2^n}\left[n^2\sum_{k=0}^{n}\binom{n}{k} - 4(n-1)\sum_{k=0}^{n}k\binom{n}{k} + 4\sum_{k=0}^{n}k(k-1)\binom{n}{k}\right] \\ &= \frac{1}{2^n}\left(n^2 2^n - 4(n-1)n2^{n-1} + 4n(n-1)2^{n-2}\right) = n.\end{aligned}$$

We deduce that $\mathbb{E}(C) \leqslant \gamma n$ and so by Lemma 3, $\mathbb{E}(T) \geqslant \frac{2}{3}(1-\gamma)n$ where

$$\gamma = \gamma(\varepsilon) = 2\varepsilon^{1/2} + \left(\frac{2}{\Phi^{-1}(2\varepsilon^{1/2})}\right)^2. \qquad \square$$

We can now complete the proof of Theorem 1.

Proof of Theorem 1. Consider a randomized algorithm which errs with probability at most ε, when the balls are coloured independently and uniformly at random. Viewing the algorithm as a probability distribution on deterministic algorithms, it must be the case that with probability at least $1-\varepsilon^{1/2}$, the deterministic algorithm used, errs with probability at most $\varepsilon^{1/2}$ and so, by Theorem 4, it has expected running time at least $\frac{2}{3}(1-\gamma(\varepsilon^{1/2}))n$. Thus, the randomized algorithm has expected running time at least $\frac{2}{3}(1-\varepsilon^{1/2})(1-\gamma(\varepsilon^{1/2}))n$. Since $\gamma(\varepsilon^{1/2}) \to 0$ as $\varepsilon \to 0$, this is exactly what we wanted to prove. $\square$

3 Proof of Theorem 2

De Marco and Pelc [6] showed that if the difference in the sizes of the colour classes grows at least as fast as linear, then we can find a ball of majority colour (with probability at least $1-\varepsilon$) in a constant number of steps. For completeness, we repeat their argument here.

Lemma 7. *Suppose that the difference between the number of black and white balls is at least $2\alpha n$. Then there is a randomized algorithm for determining majority which errs with probability at most $\varepsilon/3$ and has expected running time at most $\frac{1}{2\alpha^2}\log\left(\frac{3}{\varepsilon}\right)$.*

Proof. Pick $k = \lceil \frac{1}{2\alpha^2} \log\left(\frac{3}{\varepsilon}\right) \rceil$ balls independently and uniformly at random, with replacement. (So a ball is allowed to be picked several times.) With $k-1$ questions we can determine a ball of majority colour (according to multiplicity) from those. By the Chernoff bound [5], the probability that this ball is not in majority, is at most $e^{-2k\alpha^2} \leqslant \varepsilon/3$, as required. □

On the other hand, our next lemma says that if we know that the difference in the colour classes is not large, then we can find a ball in majority colour with no error, and expected running time not much more than $2n/3$.

Lemma 8. *Suppose that the difference, in absolute value, between the number of white and black balls is at most $d = 2\alpha n$. Then there is a randomized algorithm which determines majority (with no error) whose expected running time is at most $\frac{2n+d}{3}$.*

Proof. Let π be a random permutation of $[n]$, and compare the balls $\pi(2i-1)$ with $\pi(2i)$ for each $1 \leqslant i \leqslant n/2$. (In particular, if n is odd, we do not compare ball $\pi(n)$ with any other ball.) Pick one ball from each pair for which the answer was YES, and notice that it is enough to determine majority in these balls. The expected number of balls left is at most

$$\left(\left(\frac{1}{2}-\alpha\right)^2 + \left(\frac{1}{2}+\alpha\right)^2\right)\frac{n}{2} \leqslant \left(\frac{1}{4}+\frac{\alpha}{2}\right)n.$$

Also, the expected difference in the number of black and white balls left is at most

$$\left(\left(\frac{1}{2}+\alpha\right)^2 - \left(\frac{1}{2}-\alpha\right)^2\right)\frac{n}{2} = \alpha n.$$

Repeating the algorithm on the balls left, we deduce by the induction hypothesis, that the expected running time is at most

$$\frac{n}{2} + \frac{2}{3}\left(\frac{1}{4}+\frac{\alpha}{2}\right)n + \frac{1}{3}\alpha n = \frac{2}{3}n + \frac{2\alpha n}{3},$$

as required. (Notice that the lemma is indeed true for $n \leqslant 3$.) □

We are almost done now, since, with very few questions, we can determine with high probability whether the difference between the number of black and white balls is small or large.

Proof of Theorem 2. Let us write d for the absolute value of the difference in the number of black and white balls. With probability $2\varepsilon/3$ we will pick a ball uniformly at random, and declare it in majority. The probability of failure is at most $\varepsilon/3$. With probability $1-2\varepsilon/3$, we proceed as follows. Firstly, we make $\varepsilon n/9$ comparisons uniformly and independently at random. Note that the expected number of NO's is $\left(\frac{1}{2} - \frac{d^2}{2n^2}\right)\frac{\varepsilon n}{9}$. Consider whether the number of NO's we obtain is less than $\left(\frac{1}{2} - \frac{\varepsilon^2}{72}\right)\frac{\varepsilon n}{9}$

or not. In the former case, by the Chernoff bound, $d \geqslant \varepsilon n/12$, with probability at least $1 - \varepsilon/3$ (provided n is large enough), and we use the algorithm of Lemma 7 to determine majority in expected time at most $\frac{288}{\varepsilon^2} \log 3/\varepsilon$ and probability of error at most $\varepsilon/3$. In the latter case, again by the Chernoff bound, $d \leqslant \varepsilon n/3$, with probability at least $1 - \varepsilon/3$ and we use the algorithm of Lemma 8 to determine majority in expected time at most $2n/3 + \varepsilon n/9$. So in total, our algorithm errs with probability at most ε and has expected running time at most

$$\left(1 - \frac{2\varepsilon}{3}\right)\left(\frac{\varepsilon n}{9} + \frac{2n}{3} + \frac{\varepsilon n}{9}\right) \leqslant \frac{2}{3}\left(1 - \frac{\varepsilon}{3}\right) n$$

as required. □

Note that the only reason our randomized algorithm declares a ball (picked uniformly at random) in majority with probability $2\varepsilon/3$ is to reduce the expected running time below $2n/3$.

4 Large differences in colour classes

Recall that if we know that the difference in the number of black and white balls is linear in n, then we can solve the majority problem in a constant number of steps (with high probability). In fact, by Lemma 7, if we know that the difference is $\omega(\sqrt{n})$, then we can solve the majority problem in sublinear expected time. (Indeed, although not explicitly stated, the α in Lemma 7 can be taken to depend on n). De Marco and Pelc [6] asked whether the majority problem can be solved with constant error probability in sublinear expected time if we know that the difference is $O(\sqrt{n})$. It is not too difficult to see that this is not the case.

Theorem 9. *Let $\varepsilon > 0$ be small enough and suppose the difference in the number of black and white balls is at most $\sqrt{n}$. Then any randomized algorithm for finding the majority which errs with probability at most ε, has linear expected time.*

Proof. Suppose our adversary colours the balls uniformly and independently at random. The adversary is kind enough such that if the difference turns out to be larger than $\sqrt{n}$, then he lets us win. He then lets us choose any $n/2$ balls we want and he reveals the full partition of those balls into the colour classes. Note that in order to obtain more informartion we would need to make more than $n/4$ comparisons. Suppose we pick a ball of majority colour in this set and declare it as being in majority in the full set. (It is easy to check that if we declare any other ball as being in majority, then the probability of error increases.) Write W_1 for the number of white balls in this set of $n/2$ balls and W_2 for the number of white balls in the set of $n/2$ balls in which we have no information at all. If $W_1 \in (n/2, n/2 + \sqrt{n}/2)$ and $W_2 \in (n/2 - \sqrt{n}, n/2 - \sqrt{n}/2)$ then we definetely err. The probability of this happening tends to $(\Phi(1) - \Phi(0)(\Phi(2) - \Phi(1)) = 0.046...$ as n tends to infinity. □

Acknowledgements

The author would like to thank Simon Griffiths for several discussions on the results of this paper and Imre Leader for commenting on earlier drafts of the paper.

References

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

[2] M. Aigner, Two colors and more, in Entropy, Search, Complexity, 9–26, Springer, Berlin 2007.

[3] L. Alonso, E. M. Reingold and R. Schott, Determining the majority, Inform. Process. Lett. **47** (1993), 253–255.

[4] L. Alonso, E. M. Reingold and R. Schott, The average-case complexity of determining the majority, SIAM J. Comput. **26** (1997), 1–14.

[5] H. Chernoff, A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations, Ann. Math. Statistics **23** (1952), 493–507.

[6] G. De Marco and A. Pelc, Randomized algorithms for determining the majority on graphs, Combin. Probab. Comput. **15** (2006), 823–834.

[7] P. Erdős, On a lemma of Littlewood and Offord, Bull. Amer. Math. Soc. **51** (1945), 898–902.

[8] W. Feller, *An introduction to probability theory and its applications. Vol. I*, Third edition, Wiley, New York, 1968.

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

[10] G. Wiener, Search for a majority element, J. Statist. Plann. Inference **100** (2002), no. 2, 313–318.

[11] A. C. C. Yao, Probabilistic computations: toward a unified measure of complexity (extended abstract), in *18th Annual Symposium on Foundations of Computer Science (Providence, R.I., 1977)*, 222–227, IEEE Comput. Sci., Long Beach, Calif.

Demetres Christofides

School of Mathematics
The Watson Building
university of Birmingham
Edgbaston, Birmingham B15 2TT
United Kingdom
christod@maths.bham.ac.uk