Majority Voting in the Presence of Random Voters

Danny Dolev[1] Dror G. Feitelson[1] Haim Sompolinsky[2]

[1]Department of Computer Science
[2]The Racah Institute of Physics
The Hebrew University of Jerusalem
91904 Jerusalem, Israel

Abstract

The literature about computation in the presence of faulty elements usually deals with worst-case (Byzantine) faults or with fail-stop faults. In this paper we address an intermediate failure mode, namely random behavior. We limit the discussion to the effect that random behavior can have on a majority-voting process.

Analytic results for single rounds of votes and for iterative votes are presented; the iterative case is a simple extension of a single round of votes because it is a Markov process. The concept of quasi-stable states in a Markov process is introduced to account for the characteristics of the consensus states. Then conditions for reliable voting in the face of random participants are found, using an analogy with a physical system of spins; this eliminates the need for a complex analysis of the Markov matrix. Indeed, an important contribution of this paper is that it provides an example of a physical model that helps solve a computer-science problem.

Index Terms: Consensus, Ising spin model, Majority voting, Markov chain, Quasi-stable state, Random behavior

1 Introduction

Given a multiprocessor system, where each processor has an initial (binary) value, the simplest way in which the system can derive a common (consensus) value which is a function of the initial values is by having each processor send its value to all the others, and then having each one see what the majority wants by inspecting the messages it had received. Note that each processor reaches the decision individually, based on the information it had collected. That is why this does not work if some of the processors are faulty. Two fault models have been considered in the literature [7, 3]:

- *Fail-stop.* A faulty processor stops all its interactions with the other processors and "dies". Majority voting may fail here if all the processors wait for the vote of a dead processor. This may be solved by using timeouts.
- *Byzantine failure.* A faulty processor displays worst-case behavior, and maliciously tries to interfere with the others so as to prevent them from reaching an agreement. It can play dead, send contradicting messages, lie about messages that it had received from others, etc. Faulty processors can also form coalitions so as to cause maximum damage. In this case rather sophisticated protocols are needed in order to guarantee that a consensus is reached, and even these protocols work only if less than a third of the processors are faulty.

In this paper we consider another, intermediate type of fault: that of random behavior. In other words, a faulty processor in this model sends random messages to all the other processors. It is clear that majority voting is not the best strategy in this case: it is simpler to conduct a few rounds of dummy votes, and note which processors change their votes randomly. Then in the real vote these processors are simply ignored. Other, more sophisticated voting schemes have also been investigated, both for large systems of voters [2] and for networks of noisy elements [11, 10]. Our goal is therefore not to find a way to *overcome* random behavior, but rather to *characterize* the situation where random voters participate in a majority voting process.

Our main results concern iterative votes, where the outcome of one round is used as the initial value in the next. It turns out that when the number of random voters exceeds the number of good voters, the state in which all the good voters reach consensus is quasi-stable. This means that the system tends to settle into a consensus state fairly quickly, and then additional votes generally do not matter. However, there is a small probability that the random voters will still confuse the good voters, and then the process starts again and might settle into the opposite consensus state (where everyone agrees on the other value). Majority voting provides a good example of the importance of such quasi-stable states, that are dependent on the initial conditions. Previous research on dynamic processes typically focused on long-term behavior, and therefore emphasized conditions under which quasi-stable states do *not* exist [6, 13].

Another motivation for this study is that our system is analogous to a system of interacting magnetic spins; hence techniques from statistical physics may be used to model a multiprocessor computer system. It is felt that such interactions between computer science and physics hold much promise. Indeed, the interdisciplinary approach of using a methodology developed for physics to solve a computer-science problem has been used successfully a number of times before [8, 9, 4]. With the advent of highly parallel systems, the resemblance between computer systems and physical systems increases, and so does the relevance of this approach [5]. In our case the physical analogy allows us to characterize the system directly in terms of its basic parameters. Specifically, we find the condition that the consensus states are indeed quasi-stable, in terms of the numbers of good and random processors. A full mathematical analysis would be very complex and would probably not lead to such simple results.

The next section details the questions we are about to tackle. Sections 3 and 4 give a mathematical analysis: section 3 deals with one round of voting, while section 4 addresses iterative votes (which turn out to be a Markov process). Section 5 uses the analogy with physics to find a necessary condition for consensus in terms of random vs. correct processors, and section 6 finds the number of iterations that are needed.

2 Statement of the Problem

We study a system of n fully interconnected processors. Each of them has an initial binary opinion; this is its *vote.* It lets the other processors know what its vote is by sending them a message when asked. Each processor notes the votes of the others by adding 1 or -1 to a grand total that it keeps. If after accumulating all the votes the outcome is positive or zero, it *decides* that the majority voted for 1; otherwise it decides that the majority vote is -1.

If all the processors accumulate the votes at once and then decide at the same time, this is called a *concurrent* scheme. If they do it one at a time, it is *serial.* The difference is that with a serial scheme processors late in the line use the *final* values decided on by the processors that were before them, rather than the *initial* values of those processors. Serial schemes are further divided into those in which the order of updating is defined in advance, as opposed to those where it is stochastic.

r of the processors may be faulty, i.e., they produce a random vote (1 or -1 with equal probability). A new random vote is produced each time they send their vote to another processor. Therefore the expected value of the partial sum derived from the random votes only is 0; its variance is $\Delta^2 = r$. Of the $k = n - r$ "good" processors, a certain number k_1 have an initial value 1 and the rest (denoted by k_{-1}) a value of -1. The questions we want to answer are:

1. Characterize the fidelity and consistency of the concurrent scheme. Given, for example, that $k_1 > k_{-1}$, what is the probability p that the random processors will cause at least one "good" processor to think that the outcome of the vote is -1 rather than 1? In this case, consensus will not be reached.

2. If we use the outcome of one voting round as the initial configuration for additional voting rounds, how does the situation change? As we shall see, such *iterative* voting schemes improve the probability for reaching consensus. They are easily described by Markov-chain formalism, based on the probabilities derived in answering question #1.

3. Find conditions for reliable voting; i.e., what is the largest number r of faulty processors that can be tolerated, if we insist on getting the correct result with probability tending to 1 (that is, we require that consensus is reached)? It is clear that r may be more or less quadratic in k; a physical analogy allows for a calculation that provides an exact relation. Remarkably, this relation holds for both the concurrent and the serial voting schemes.

4. Given a system in which reliable voting is ensured by an iterative voting scheme, estimate the time needed to reach consensus. In other words, how many voting rounds are required? It turns out that in most cases a very small number suffices.

3 The Probability of a Wrong Outcome

We start with the concurrent decision scheme. The *state* of a system of processors as described above, assuming concurrent decisions, is characterized by a single variable $d = k_1 - k_{-1}$, i.e. the outcome of the votes of the good processors. The votes of the faulty processors may be considered as a sort of *noise* that is added onto the state to produce the final outcome. This random noise is added in each good processor individually (it is a *local* noise). Thus distinct processors might end up with different opinions as to the result of the vote. If we now count these "personal" opinions just as we counted the initial values, we can characterize the state of the system after a round of voting by a new set of variables $d' = k'_1 - k'_{-1}$.

For any pair of states d and d', $-k \leq d, d' \leq k$, the probability of a transition from d to d' in one round of voting is calculated as follows. In order to enter state d', exactly $\frac{k+d'}{2}$ processors must reach a positive total. As the good processors contribute d to the total, this means they should get a noise term of *more than* $-d$. The noise is actually the sum of r Bernoulli random variables, each with a probability of $\frac{1}{2}$ for being 1 and $\frac{1}{2}$ for being -1. Such a sum is described by a Binomial random variable; in our case, a total of δ will result from $\frac{r+\delta}{2}$ 1's and $\frac{r-\delta}{2}$ -1's, so the probability to get it is

$$\Pr(\delta) = \binom{r}{\frac{r+\delta}{2}} \left(\frac{1}{2}\right)^{\frac{r+\delta}{2}} \left(\frac{1}{2}\right)^{\frac{r-\delta}{2}} = \binom{r}{\frac{r+\delta}{2}} \left(\frac{1}{2}\right)^{r} \tag{1}$$

However, we do not really care what the exact value of the noise is as long as it is above $-d$. Using the Normal approximation for the Binomial, the probability for not exceeding x is given by $\Phi\left(\frac{x}{\sqrt{r}}\right)$, where $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{z^2}{2}} dz$. Because of the symmetry of Φ, this is also the probability that the sum is larger than $-x$, which is what we need. As there are $\binom{k}{\frac{k+d'}{2}}$ ways in which to choose the "lucky" processors that will get a positive total, the final formula for the probability of transition from state d to state d' is

$$\Pr(d \to d') = \binom{k}{\frac{k+d'}{2}} \left[\Phi\left(\frac{d}{\sqrt{r}}\right)\right]^{\frac{k+d'}{2}} \left[\Phi\left(-\frac{d}{\sqrt{r}}\right)\right]^{\frac{k-d'}{2}} \tag{2}$$

The voting round must cause a transition into some state, so $\sum_{d'} \Pr(d \to d') = 1$. The correct consensus state, starting from state d, is $\operatorname{sgn}(d) \cdot k$. Therefore the probability that the system does *not* reach the desired consensus value is

$$\begin{aligned} p &= 1 - \Pr(d \to \operatorname{sgn}(d) \cdot k) \\ &= 1 - \Phi\left(\frac{|d|}{\sqrt{r}}\right)^k \end{aligned} \tag{3}$$

This formula provides an answer to question #1, for the case $r \geq k$ (if applied when $r < k$, it gives wrong results). Regretfully, it is impossible to calculate, and Φ has to be looked up in tables. Of course, known approximations may be used when appropriate. Note that this formula also characterizes the degree of stability of the consensus states. If we start in a consensus state, i.e. $d = \pm k$, equation (3) gives the probability that consensus will be lost.

In the serial voting scheme, the situation is more complicated. We cannot fully describe the system by its state d, but have to specify the exact *configuration* of the system (denoted by $\mathbf{s}$). This is a vector with k elements, each of which may assume the values ± 1; element number i (denoted by s_i) represents the value of processor number i. The formula that describes the transitions from one configuration to another, in analogy with eq. (2), is

$$\Pr(\mathbf{s} \to \mathbf{s}') = \begin{cases} \Phi\left(\operatorname{sgn}(-s_i)\frac{\sum_j s_j}{\sqrt{r}}\right) & \text{if } \mathbf{s} \text{ differs from } \mathbf{s}' \text{ only in } s_i, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

A full voting round potentially involves k such transitions. The equations that describe all the possibilities are very cumbersome, and it does not seem justifiable to follow them up. An alternative approach, using a physical analogy, is described in appendix A. The following section is therefore limited to the concurrent case.

4 Description of Iterative Votes by Markov Chains

An interesting point is that if each processor takes its outcome from the vote as a new initial value, and a new voting round is conducted, equation (2) may be applied again. Iterative voting is thus not sensitive to the exact history, only to the outcome of the last iteration. In other words, it is a Markov process. It is described by a $(k+1) \times (k+1)$ matrix $\mathcal{D}$ for even k, or a $k \times k$ matrix for odd k (note that the possible states are integers in jumps of 2), where $\mathcal{D}_{ij} = \Pr(i \to j)$. Probabilities of transitions by t iterations are therefore given by the elements of $\mathcal{D}^t$ (this answers question #2).

It is tempting to try and use the limiting probabilities of Markov processes to predict the outcome of multiple votes. While formally sound, of course, this approach is of questionable relevance. Two cases may be distinguished:

1. $r < k$, i.e. the number of random voters is small. In this case the states k and $-k$ are absorbing, so the Markov chain is not irreducible; the random voters only have a small influence on the outcome, which typically would be a quick convergence to consensus on the majority value. Therefore this case is simple and does not warrant further discussion.

2. $r \geq k$, i.e. there are more random voters than good voters. This means that there is a positive probability that the random voters cause the good voters to change their consensus value, even after consensus is actually reached! Therefore the Markov chain is irreducible, recurrent, and aperiodic, and the limiting probabilities are independent of the initial state. Thus the limiting probabilities can only tell us the percentage of the time that the system will occupy either of the consensus states in the long run.

Experimental test runs show, however, that in the short range of up to a few hundred voting rounds things are quite different. It is this transient behavior that we seek to characterize. For example, simulations show that with 1000 processors, of which 801 are random and of the others 100 start as 1 and 99 as -1, all 199 good processors tend to end up with a 1 after only 3 to 4 rounds. In a small percentage of the cases, they all end up with -1. In general, when large numbers are involved, the system has an overwhelming tendency to reach consensus (in the right direction) within a small number of rounds. It seems that the states k and $-k$ are *quasi-stable* states, meaning that there is a relatively high probability to enter them and a relatively low probability for transitions from them to other states. This can be made more formal by characterizing the matrix that describes the evolution of the process as

$$\begin{pmatrix} 1-\varepsilon & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 1-\varepsilon & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 1-\varepsilon \\ \vdots & & \vdots & \vdots \\ 0 & \cdots & 0 & 1-\varepsilon \end{pmatrix} \leq \mathcal{D}^t \leq \begin{pmatrix} 1 & \varepsilon & \cdots & \varepsilon \\ \vdots & \vdots & & \vdots \\ 1 & \varepsilon & \cdots & \varepsilon \\ \varepsilon & \cdots & \varepsilon & 1 \\ \vdots & & \vdots & \vdots \\ \varepsilon & \cdots & \varepsilon & 1 \end{pmatrix} \tag{5}$$

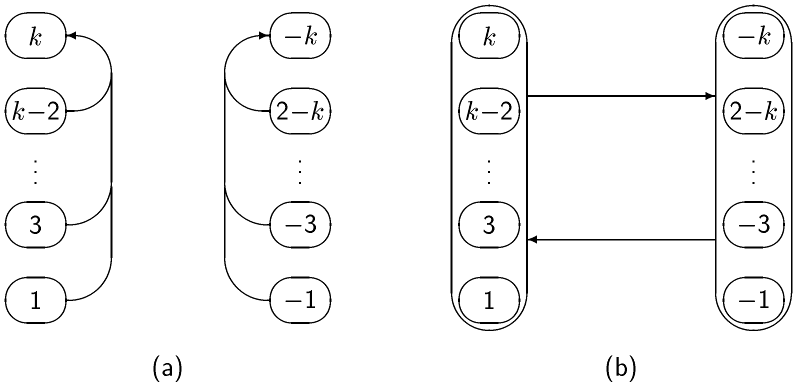

Figure 1: *The behavior of the Markov process (a) on the immediate time scale, and (b) on a time scale of* $1/\varepsilon$*. It is assumed that* k *is odd.*

where the inequalities hold for each element individually. If $\varepsilon = 1$ this characterization is obviously void. The degree to which ε approaches zero after a number of multiplications t testifies to the degree to which the consensus states are indeed quasi-stable.

More formally, the system displays different behaviors at different time scales [12]. In the immediate time scale, only transitions with a probability of $O(1)$ are observed. These are shown in fig. 1 (a). On the time scale of $1/\varepsilon$, transitions with a probability of $O(\varepsilon)$ come into play. At this time scale, the states are effectively grouped into two groups, and the transitions occur between these groups (fig. 1 (b)).

Numerical checks of real examples, using the inequality (5) to define ε as a function of t, show a transient phase in which ε is relatively low (fig. 2). This indicates the time scale at which the consensus states are indeed quasi-stable. At longer time scales, the ε-transitions cause a mixing of the states, and the system settles into the limiting probabilities. The relatively large values of ε in the figure are a result of using a small system in the example, coupled with the fact that eq. (5) is somewhat idealistic. The real matrix does not have such a sharp step between values that are $O(1)$ and those that are $O(\varepsilon)$. The graph actually shows the largest value of those that are not $O(1)$.

What property of the matrix causes it to have the quasi-stable transient states? If $\varepsilon = 0$, the theory of Markov chains provides some clues to an answer. Note that in this case the states in the Markov chain may be classified into three classes: The two extreme ones are each a recurrent class by itself, while all the others constitute a transient class. However, in the real matrix $\mathcal{D}$ (where $\varepsilon > 0$), all the states form one recurrent class; therefore we are actually dealing with an *ergodic* Markov process. Applying this observation, we can restate the question thus: under what conditions will our ergodic Markov process behave

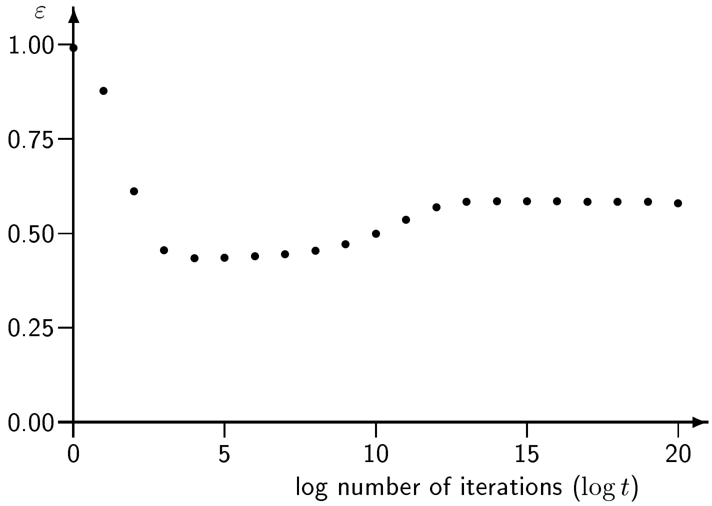

Figure 2: *Characterization of a voting process with 9 good voters and 17 random ones, using the inequality (5).*

similarly to a non-ergodic process like the one of (5) with $\varepsilon = 0$?

The most important clue we have derives from a theorem which states that ergodicity is broken if the eigenvalue $\lambda = 1$ of the Markovian matrix is degenerate (there always exists at least one eigenvector with this eigenvalue — it is the vector of limiting probabilities. The other distinct eigenvalues are smaller than 1). The problem is that when the elements of the matrix are defined by an equation such as (2), it is completely unpractical to expect an analytic expression for the eigenvalues. The best one can do is to check the trace of $\mathcal{D}^t$ for increasing t's. As the trace is equal to the sum of the eigenvalues raised to the tth power, a degenerate $\lambda = 1$ (or another eigenvalue which is extremely close to 1) will be indicated by the trace converging to 2. Numerical checks we made on example matrices show that indeed they have a second eigenvalue of magnitude close to 1. An example is given in fig. 3, where the trace is seen to converge first to 2 (in the same range of iterations for which ε is minimal), and then to 1 as the limiting probabilities are reached. The relative pointwise distance from the limiting probabilities, defined by Jerrum and Sinclair [6], is shown for comparison; it also shows a plateau in the rounds in which ε is minimal, indicating that the progress towards the limiting probabilities is stopped while the system is in the quasi-stable state. This is contrary to other situations in which a rapid convergence to the limiting probabilities is observed [6, 13].

The statistical physics analysis of the following section implicitly depends on the above theorem. All the approximations are based on the assumption that the system under treatment is large. When this is the case the eigenvalues become

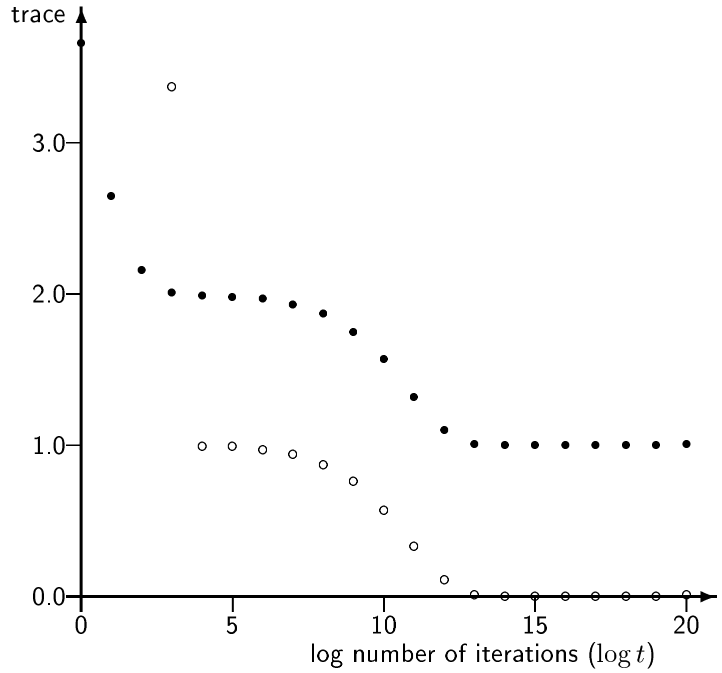

Figure 3: *The trace of a Markov matrix for a voting process with 9 good voters and 17 random ones, indicating the existence of an eigenvalue 1 and another eigenvalue very close to 1. The relative pointwise distance from the limiting probabilities is also shown (circles).*

dense in $[0, 1]$, and with probability tending to 1 there will be an eigenvalue that is arbitrarily close to 1. It also depends upon the fact that we are interested in the quasi-stable state that the system settles into after a short time, and not in the limiting probabilities for being in the various states after a long (possibly exponential) period. Thus it uses the notion of a Markov process, but does not use the properties of limiting probabilities. The advantage of the physical analysis is that it does not require any complex mathematics, such as the evaluation of eigenvalues. In fact, it does not even require an exact expression for the matrix itself.

5 The Condition for Consensus

A different approach to the question of consensus by majority voting in the presence of random voters is afforded by an analogy to a model of a physical system of magnetic spins (the "Ising spin model"). Each spin is a miniature magnet that can point either up or down (1 or -1); it represents a "good" voter in a majority voting process. Each spin exerts a force on the others (sends a message); this force tries to align all the spins in the same direction (consensus). The forces from all the other spins add up, just like the votes from the other processors. The random voters are modeled by a random noise term that is added too. In a real spin system, the forces act continuously and depend on the distance. In the model, the effect on one spin on another is discretized: the spins either update their states concurrently or serially. In addition, we may decide that all spins have the same effect, regardless of distance. In this case the analogy between the spin system and the voting process is complete, as their dynamics are governed by exactly the same rules. There are no approximations in passing from one to the other.

The statistical-physics analysis needed to obtain results about this spin system is somewhat tedious; interested readers may find a detailed sketch of it in the appendix. The main result of this analysis is contained in the following equation:

$$m = \operatorname{erf}\left(\frac{\mathcal{J}_0}{\sqrt{2}\Delta}\, m\right) \tag{6}$$

where m is the average value over all the spins (it is equivalent to $\frac{d}{k}$ in the notation of previous sections),

$\mathcal{J}_0$ is a calibration factor that denotes the strength with which the spins affect each other,

Δ^2 is the variance of the contribution by the random voters,

and the "error function" [1], $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-z^2} dz$, is used instead of Φ as in the preceding sections (the equation can of course be written in terms of Φ, using $\operatorname{erf}(x) = 2\Phi\left(\frac{x}{\sqrt{2}}\right) - 1$, but the expression with erf is simpler).

The significance of the above formula lies in the fact that it describes the state that the system settles into. In other words, the physical analogy indicates that in a large enough system an overwhelming percentage of the time will be spent in one specific state: the probability to leave the quasi-stable state is so low that the system is stuck in it. Solving for m reveals what the average spin value will be in this (quasi-)stable state.

The equation always has the trivial solution $m = 0$. If the argument is too small (specifically, if $\frac{\mathcal{J}_0}{\sqrt{2}\Delta}\frac{2}{\sqrt{\pi}} < 1$), this is the only solution. In the terms of voting schemes, it means that if there are too many random voters (so Δ is large), the good voters will not reach consensus: half will decide on 1 while the other half decide on -1. Thus the outcome is completely random.

If, on the other hand, $\frac{\mathcal{J}_0}{\sqrt{2}\Delta}\frac{2}{\sqrt{\pi}} > 1$, there are two additional solutions (which we shall denote $\pm m_0$). These solutions approach ± 1 as $\frac{\mathcal{J}_0}{\Delta}$ goes to infinity, i.e. as the noise decreases. In other words, the degree of agreement increases as the number of random voters goes down.

An interesting question is the specification of the conditions under which we can say that consensus will be reached with probability tending to 1. A sufficient condition is simply that $m_0 > 1 - \frac{1}{k}$, because if the average is so near to 1, all the spins must be in the "1" state. Using an approximation of erf valid for large arguments, we write

$$m = 1 - \frac{\mathcal{J}_0}{\Delta}\sqrt{\frac{2}{\pi}}\, e^{-\frac{\mathcal{J}_0^2}{2\Delta^2}} > 1 - \frac{1}{k} \tag{7}$$

which develops into

$$\begin{aligned} &\frac{\mathcal{J}_0}{\Delta}\sqrt{\frac{2}{\pi}}\, e^{-\frac{\mathcal{J}_0^2}{2\Delta^2}} < \frac{1}{k} \\ &\frac{\mathcal{J}_0^2}{2\Delta^2} > \ln k - \ln\frac{\mathcal{J}_0}{\Delta} + \ln C \end{aligned} \tag{8}$$

As $\ln\frac{\mathcal{J}_0}{\Delta}$ is insignificant relative to $\frac{\mathcal{J}_0^2}{\Delta^2}$, and $\ln C$ is just a constant, we may conclude that

$$\Delta < \frac{\mathcal{J}_0}{\sqrt{2\ln k}}. \tag{9}$$

Recall that Δ^2 is the variance of the noise term (so Δ may represent the sum of the votes of the random voters). We want to calibrate this, so that each of the random voters has the same weight as a good voter. Therefore we write

$$\Delta = \frac{\mathcal{J}_0}{k}\sqrt{r}, \tag{10}$$

and eq. (9) reduces to

$$r < \frac{k^2}{2\ln k}. \tag{11}$$

This is an upper bound on the tolerable number of random voters: they may be nearly quadratic in the good voters (it answers our question #3). If the total number of random voters is smaller, consensus is guaranteed. If, however, the total number of random voters exceeds this limit, then in the average case consensus will not be reached. In this case, m_0 gives the expected outcome of the voting process. These results are summarized in table 1.

The physical analysis that lead to the above results required a number of approximations and simplifying assumptions. This raises the question of the validity of the results. A review of the derivation reveals that the only significant assumption that was made is that k is big; therefore the validity of the results will increase for large systems. Specifically, the importance of the quadratic relation between k and r should be apparent in system of a few hundred elements. The logarithmic factor in the denominator is only expected to come into play with systems of a few thousand elements.

$r : k$ *ratio*	*outcome*
$\frac{2k^2}{\pi} < r$	random (no consensus): half of the good voters end with 1 and half with -1
$\frac{k^2}{2 \ln k} < r < \frac{2k^2}{\pi}$	no consensus: the expected outcome is $k_1 - k_{-1} = m_0 k$
$r < \frac{k^2}{2 \ln k}$	consensus with probability tending to 1

Table 1: *Summary of the results.*

6 The Number of Iterations Required for Consensus

An important aspect of the suitability of iterative majority voting for the consensus problem is the number of iterations that are required. It is clear that when no random voters are present, one round of voting is sufficient. In the preceding section we showed that even when there are very many random voters, consensus may be reached (provided that (11) is satisfied). However, this result relys on the use of iterative rounds of votes, and up to now we have not made any statement as to the number of rounds that are needed. In this section we show that a small number is usually enough (thus answering question #4).

The dynamics of the system may be described by the following equation:

$$\begin{aligned} m(t+1) &= \operatorname{erf}\left(\frac{\mathcal{J}_0}{\sqrt{2}\Delta}\, m(t) \right) \\ &\approx 1 - e^{-\frac{\mathcal{J}_0^2}{2\Delta^2} m(t)^2} \end{aligned} \tag{12}$$

(see equations (31) and (35) in the appendix). Going back to eq. (11), we define a parameter θ that characterizes the system to be

$$\theta = \frac{k^2}{2r \ln k}. \tag{13}$$

Note that $\theta > 1$. Using eq. (10), we get

$$\theta \ln k = \frac{k^2}{2r} = \frac{\mathcal{J}_0^2}{2\Delta^2} \tag{14}$$

Therefore

$$
\begin{aligned}
m(t+1) &\approx 1-\left(e^{-\frac{\mathcal{J}_0^2}{2\Delta^2}}\right)^{m(t)^2} \\
&\approx 1-\left(\frac{1}{k}\right)^{\theta\, m(t)^2}
\end{aligned}
\tag{15}
$$

In order to reach consensus in *one step*, the average spin value after only one iteration, $m(1)$, must be very close to 1:

$$
1-m(1) \approx \left(\frac{1}{k}\right)^{\theta\, m(0)^2} < \frac{1}{k} \tag{16}
$$

Thus if $m(0)$ is above a certain critical value, namely if

$$
m(0) > \frac{1}{\sqrt{\theta}}, \tag{17}
$$

we may expect to reach consensus in one step even in the presence of r random voters. Note that the larger θ is, i.e. the better the ratio of good to random processors, the nearer $m(0)$ may be to zero. But if θ is close to 1, i.e. if the ratio is close to the allowed limit, then $m(0)$ must be near 1 to ensure consensus in one step.

Let us now find the condition for consensus in *two* steps. The same argument is used: consensus will be reached after two steps if the average spin value after two iterations, $m(2)$, is close enough to 1. This is expressed by

$$
1-m(2) \approx \left(\frac{1}{k}\right)^{\theta\, m(1)^2} < \frac{1}{k} \tag{18}
$$

Using eq. (15) to express $m(1)$, this turns into

$$
\left(\frac{1}{k}\right)^{\theta\left[1-\left(\frac{1}{k}\right)^{\theta m(0)^2}\right]^2} < \frac{1}{k} \tag{19}
$$

Solving this for $m(0)$ involves straightforward algebra; the result is the following condition:

$$
m(0)^2 > \frac{\ln\frac{\sqrt{\theta}}{\sqrt{\theta}-1}}{\theta\,\ln k} \tag{20}
$$

It may be more revealing to express this as

$$
m(0) > \frac{1}{\sqrt{\theta}} \cdot \sqrt{1-\frac{\ln(k-\sqrt{2r\ln k})}{\ln k}} \tag{21}
$$

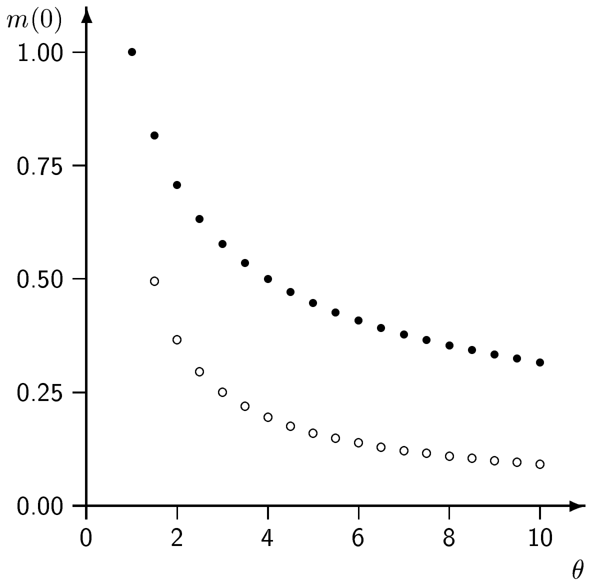

Figure 4: *Values of $m(0)$ required to ensure consensus after only one round (bullets) or after two rounds (circles, with $k = 100$).*

which shows the improvement over the condition for consensus in one round as expressed in eq. (17). Again we find that as the ratio of good to random processors improves, $m(0)$ may be nearer to zero, but if θ is close to 1 then $m(0)$ too must be large (there is no solution for $\theta = 1$, but that is taken care of in one round). Note that the absolute number of good processors, k, is also important in this case.

Expressions for values of $m(0)$ that ensure consensus after three or more rounds can also be calculated, but they become very cumbersome and are omitted from this discussion. The general behavior can be seen from fig. 4, which plots the required $m(0)$ for consensus in one or two rounds.

7 Conclusions

This paper is innovative in both its topic and its method. First, we study random behavior as a possible failure mode that lies between the extremes of fail-stop behavior and Byzantine behavior. We characterize random behavior in how it affects majority-voting schemes; this is but a first step towards a better understanding of random behavior in general.

As a by-product of this characterization, we demonstrate the importance of quasi-stable states in a Markov process. This is in sharp contrast with the prevailing line of research, which emphasizes the importance of the limiting probabilities and regards quasi-stable states as a nuisance [12, 6, 13]. But when the probability to stay in the quasi-stable states is near enough to unity, these states cannot be ignored. A system with quasi-stable states will typically be observed in these states, and if we want to model such systems we must include the quasi-stable states in the model. It should be emphasized that this indeed adds more power to the model, which is lost if only the limiting probabilities are investigated, because the quasi-stable states are *not* independent of the initial conditions.

The second innovative aspect of this paper is the use of statistical-physics techniques to solve a computer science problem. As the number of processors in a computer system increases, the similarity between it and a typical physical system increases [5]. Thus we may hope that in the future it will be possible to tap the vast amounts of knowledge and intuition accumulated by physicists to the advantage of computer science.

Acknowledgements

Fruitful discussions with Hanoch Gutfreund, Michael Ben-Or, Barak Perelmutter, Mike Werman, and Aviad Cohen are gratefully acknowledged.

References

[1] M. Abramowitz and I. A. Stegun (eds.), *Handbook of Mathematical Functions*, chap. 7. United States Department of Commerce, 1964.

[2] M. Ben-Or and N. Linial, "*Collective coin flipping, robust voting schemes, and minima of Banzhaf values*". In 26th *Symp. Foundations of Computer Science*, pp. 408–416, Oct 1985.

[3] M. J. Fischer, "*The consensus problem in unreliable distributed systems (a brief survey)*". In *Proc. Intl. Conf. Foundations of Computation Theory*, M. Karpinski (ed.), pp. 127–140, Springer-Verlag, 1983. Lecture Notes in Computer Science Vol. 158.

[4] L. A. Glasser and C. A. Zukowski, "*Continuous models for communication density constraints on multiprocessor performance*". *IEEE Trans. Comput.* **C-37(6)**, pp. 652–656, Jun 1988.

[5] W. D. Hillis, "*New computer architectures and their relationship to physics, or why computer science is no good*". *Intl. J. Theoret. Phys.* **21(3/4)**, pp. 255–262, Apr 1982.

[6] M. Jerrum and A. Sinclair, "*Conductance and the rapid mixing property for Markov chains: the approximation of the permanent resolved*". In 20th *Ann. Symp. Theory of Computing*, pp. 235–244, May 1988.

[7] L. Lamport, R. Shostak, and M. Pease, "*The Byzantine generals problem*". *ACM Trans. Prog. Lang. Syst.* **4(3)**, pp. 382–401, Jul 1982.

[8] N. Linial, L. Lovász, and A. Wigderson, "*A physical interpretation of graph connectivity, and its algorithmic applications*". In 27th *Symp. Foundations of Computer Science*, pp. 39–48, Oct 1986.

[9] G. Maturana, "*Static allocation in massively parallel computers*". *Computer Physics Communications* **45(1–3)**, pp. 319–322, Aug 1987.

[10] N. Pippenger, "*Analysis of error correction by majorith voting*". In *Randomness and Computation*, S. Micali (ed.), JAI Press, 1988.

[11] N. Pippenger, "*Reliable computation in the presence of noise*". In *Proc. Inter. Cong. Math.*, pp. 1469–1477, 1986.

[12] J. R. Rohlicek and A. S. Willsky, "*The reduction of perturbed Markov generators: an algorithm exposing the role of transient states*". *J. ACM* **35(3)**, pp. 675–696, Jul 1988.

[13] A. Sinclair and M. Jerrum, "*Approximate counting, uniform generation and rapidly mixing Markov chains*". *Info. & Comput.* **82(1)**, pp. 93–133, Jan 1989.

Appendixes:

A Consensus in the Serial Decision Scheme

The results presented in section 5 are based on equation (6). This appendix shows how the equation is derived in the case of serial dynamics. As most readers are expected to be unfamiliar with this analysis, a detailed derivation is given.

A system of k spins is described by a k-vector $\mathbf{s}$, called the *configuration* of the system, whose elements s_i assume the values 1 or -1. In the general case, the *local field* on spin i is

$$h_i(t) = \sum_{j \neq i} \mathcal{J}_{ij}\, s_j(t) + \eta_i(t) \tag{22}$$

where $\mathcal{J}_{ij}$ indicates the strength of the influence of spin j on spin i,
η_i is the local random noise. This is assumed to have a Gaussian distribution, with width Δ.

The system is assumed to follow a serial dynamics, i.e. at each moment *one* spin (chosen at random or by a predefined order) aligns itself in parallel to its local field.

As in section 4, we begin by observing that the evolution of the system is a Markov process. However, we now regard each configuration of the system as a distinct state, and denote the transition matrix by $\mathcal{W}$. The serial dynamics assumption implies that when the system moves from one configuration to another, only one spin flips. Therefore the probability for a transition is just the probability that the local field on that particular spin is anti-parallel to its current direction; this is given by

$$\mathcal{W}(\mathbf{s} \to \mathbf{s}') = \frac{1}{2}\left(1 - s_i \operatorname{erf}\left(\frac{\sum_j \mathcal{J}_{ij}\, s_j}{\sqrt{2}\Delta}\right)\right) \tag{23}$$

where the "error function", $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-z^2} dz$ (see [1]), is used instead of Φ as in section 3, and s_i denotes the spin in which $\mathbf{s}$ differs from $\mathbf{s}'$. This equation is equivalent to eq. (4).

If we denote by $\mathrm{P}(\mathbf{s}, t)$ the probability that the system will be in configuration $\mathbf{s}$ at time t, the *change* of $\mathrm{P}(\mathbf{s}, t)$ with time may be expressed by

$$\dot{\mathrm{P}}(\mathbf{s}, t) = \frac{\partial \mathrm{P}(\mathbf{s}, t)}{\partial t} = \sum_{\mathbf{s}'} \mathcal{W}(\mathbf{s}' \to \mathbf{s})\mathrm{P}(\mathbf{s}', t) - \sum_{\mathbf{s}'} \mathcal{W}(\mathbf{s} \to \mathbf{s}')\mathrm{P}(\mathbf{s}, t) \tag{24}$$

We will use this expression shortly.

We start the analysis of the system by defining m_i to be the *average* state of s_i, taken over all possible configurations:

$$m_i(t) = \langle s_i(t) \rangle = \sum_{\mathbf{s}} s_i\, \mathrm{P}(\mathbf{s}, t) \tag{25}$$

(the triangular brackets $\langle \cdots \rangle$ denote averages). The change of m_i with time is given by

$$\dot{m}_i(t) = \frac{\partial}{\partial t}\left[\sum_{\mathbf{s}} s_i\, \mathrm{P}(\mathbf{s}, t)\right] = \sum_{\mathbf{s}} s_i\, \dot{\mathrm{P}}(\mathbf{s}, t) \tag{26}$$

Using equations (24) and (23), we obtain

$$
\begin{aligned}
\dot{m}_i(t) &= \sum_{\mathbf{s}} s_i \left(\sum_{\mathbf{s}'} \mathcal{W}(\mathbf{s}' \to \mathbf{s}) \mathrm{P}(\mathbf{s}', t) - \sum_{\mathbf{s}'} \mathcal{W}(\mathbf{s} \to \mathbf{s}') \mathrm{P}(\mathbf{s}, t) \right) \\
&= \sum_{\mathbf{s},\mathbf{s}'} s_i \, \mathcal{W}(\mathbf{s}' \to \mathbf{s}) \mathrm{P}(\mathbf{s}', t) - \sum_{\mathbf{s},\mathbf{s}'} s_i \, \mathcal{W}(\mathbf{s} \to \mathbf{s}') \mathrm{P}(\mathbf{s}, t) \\
&= \sum_{\mathbf{s},\mathbf{s}'} s_i \, \mathcal{W}(\mathbf{s}' \to \mathbf{s}) \mathrm{P}(\mathbf{s}', t) - \sum_{\mathbf{s},\mathbf{s}'} s_i' \, \mathcal{W}(\mathbf{s}' \to \mathbf{s}) \mathrm{P}(\mathbf{s}', t) \\
&= \sum_{\mathbf{s},\mathbf{s}'} (s_i - s_i') \, \mathcal{W}(\mathbf{s}' \to \mathbf{s}) \mathrm{P}(\mathbf{s}', t) \\
&= \sum_{\mathbf{s}'} -2 s_i' \, \frac{1}{2} \left(1 - s_i' \operatorname{erf} \left(\frac{\sum_j \mathcal{J}_{ij} \, s_j(t)}{\sqrt{2}\Delta} \right) \right) \, \mathrm{P}(\mathbf{s}', t) \\
&= -m_i(t) + \left\langle \operatorname{erf} \left(\frac{\sum_j \mathcal{J}_{ij} \, s_j(t)}{\sqrt{2}\Delta} \right) \right\rangle
\end{aligned}
\tag{27}
$$

The problem with this equation is that it is nonlinear, so taking the average of the erf involves considerations regarding high order statistics between the different spins. However, for our needs a simple special case suffices: we take all the interactions to be of equal strength, and denote it by $\mathcal{J}_{ij} = \frac{\mathcal{J}_0}{k}$. Therefore

$$
\sum_{j \neq i} \mathcal{J}_{ij} \, s_j \simeq \frac{\mathcal{J}_0}{k} \sum_j s_j \tag{28}
$$

(the $\simeq$ is used rather than $=$ because the sum on the right includes the i^{th} term, which the sum on the left does not. For large k's this is negligible).

The next step is to define *global* averages

$$
\begin{aligned}
s(t) &= \tfrac{1}{k} \textstyle\sum_i s_i(t) \\
m(t) &= \tfrac{1}{k} \textstyle\sum_i m_i(t)
\end{aligned}
\tag{29}
$$

In the thermodynamic limit, $k \to \infty$, the correlation between the spins becomes negligible. The average in the final line of eq. (27) may then be moved into the argument of erf, and we get

$$
\left\langle \operatorname{erf} \left(\frac{\mathcal{J}_0}{\sqrt{2}\Delta} \, s(t) \right) \right\rangle \; = \; \operatorname{erf} \left(\frac{\mathcal{J}_0}{\sqrt{2}\Delta} \, m(t) \right) . \tag{30}
$$

This is called the "mean field approximation". In the case of fully connected systems, like our case, it becomes exact when $k \to \infty$. Returning to eq. (27), we see that what we end up with is

$$\dot{m}(t) = -m(t) + \mathrm{erf}\left(\frac{\mathcal{J}_0}{\sqrt{2}\Delta}\, m(t)\right). \tag{31}$$

Note that this equation does not contain any dependency on i.

As we are interested in the steady state that the system settles into, we have $\dot{m}(t) = 0$. The equation then becomes

$$m = \mathrm{erf}\left(\frac{\mathcal{J}_0}{\sqrt{2}\Delta}\, m\right) \tag{32}$$

which is eq. (6).

B Consensus in the Concurrent Decision Scheme

This appendix shows how eq. (6) is derived for the concurrent case. Once the notation is established, it appears that the concurrent case is actually simpler than the serial one. Since all the spins update their positions at the same time, they are all affected by the same average value $m(t)$. the only difference between them is in the local noise. Therefore

$$s_i(t+1) = \mathrm{sgn}\left(\mathcal{J}_0\, m(t) + \eta_i(t)\right) \tag{33}$$

where $m(0)$ reflects the initial bias toward 1 or -1.

To see how the system evolves, we simply note that

$$\begin{aligned} m(t+1) = &\ \frac{1}{k}\sum_i s_i(t+1) \\ = &\ \frac{1}{k}\sum_i \mathrm{sgn}\left(\mathcal{J}_0\, m(t) + \eta_i(t)\right) \end{aligned} \tag{34}$$

The only difference between the terms in the sum is in $\eta_i(t)$; for large systems, therefore, this sum is equivalent to an average on η. As η represents a Gaussian noise, we have

$$
\begin{aligned}
m(t+1) = & \ \frac{1}{\sqrt{2\pi}\Delta}\int_{-\infty}^{\infty}\operatorname{sgn}\left(\mathcal{J}_0\, m(t)+\eta\right)\ e^{-\frac{\eta^2}{2\Delta^2}}\ d\eta \\
= & \ \frac{1}{\sqrt{2\pi}\Delta}\int_{-\mathcal{J}_0 m}^{\infty} e^{-\frac{\eta^2}{2\Delta^2}}\ d\eta - \frac{1}{\sqrt{2\pi}\Delta}\int_{-\infty}^{-\mathcal{J}_0 m} e^{-\frac{\eta^2}{2\Delta^2}}\ d\eta \\
= & \ \frac{2}{\sqrt{2\pi}\Delta}\int_{0}^{\mathcal{J}_0 m} e^{-\frac{\eta^2}{2\Delta^2}}\ d\eta \\
= & \ \frac{2}{\sqrt{\pi}}\int_{0}^{\frac{\mathcal{J}_0 m}{\sqrt{2}\Delta}} e^{-\eta'^2}\ d\eta' \\
= & \ \operatorname{erf}\left(\frac{\mathcal{J}_0\, m(t)}{\sqrt{2}\Delta}\right)
\end{aligned}
\tag{35}
$$

Which leads to eq. (6) again for a steady state.