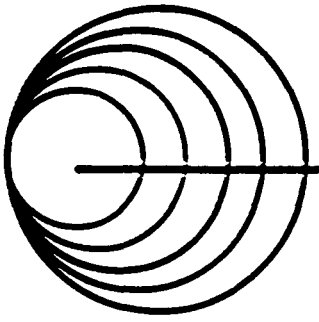
Appendix I



COMPUTER STUDIES

TECHNICAL REPORT

Reliability of Voting in Fault-Tolerant Software Systems for Small Output Spaces

McAllister, David F.
Sun, Chien-En
Vouk, Mladen A.

TR-87-16

North Carolina State University

Raleigh, N. C. 27650

Reliability of Voting in Fault-Tolerant Software Systems for Small Output Spaces

David F. McAllister
Chien-En Sun
Mladen A. Vouk

Department of Computer Science
North Carolina State University
Raleigh, NC 27695-8206

Abstract

Under a voting strategy in a fault-tolerant software system there is a difference between correctness and agreement. An independent N-version programming reliability model is proposed for treating small output spaces which distinguishes between correctness and agreement. System reliability is investigated using analytical relationships and simulation. A consensus majority voting strategy is proposed and its performance is analysed and compared with other voting strategies. Consensus majority strategy automatically adapts the voting to different component reliability and output space cardinality characteristics. It is shown that absolute majority voting strategy provides a lower bound on the reliability provided by the consensus majority, and 2-of-n voting strategy an upper bound. If r is the cardinality of the output space it is proved that $1/r$ is a lower bound on the average reliability of fault-tolerant system components below which the system reliability begins to deteriorate as more versions are added.

*Research supported in part by NASA Grant No. NAG-1-667

I. Introduction

Recent experiments with multiversion software have demonstrated the fact that identical and incorrect answers can occur with perhaps higher frequency than is expected [Sco84, Kni86, Vou85,86] particularly when small output spaces are involved. For example, if the output space is binary, {0,1}, then all incorrect responses must agree. Such phenomena make the fault-tolerant techniques of N-version programming [Avi77, Avi84] and Consensus Recovery Block [Sco83a,b,84,87] more likely to fail during the voting process since the voter may not be able to distinguish between correct and incorrect responses. In the past models of fault-tolerant reliability have equated output agreement with correctness (e.g. [Sco83a,b, Sco87]) which is inadequate for complete modeling of such an environment. In this paper we distinguish between agreement and correctness and develop a reliability model of a voting environment which can be used to determine the number of versions required as a function of the cardinality of the output space.

II. Voting Strategies in N-Version Programming

In an ***m-of-n fault-tolerant software (FTS) system*** the number of functionally equivalent independently developed versions is n, and m is the ***agreement number*** or the number of matching outputs which the voting or adjudication algorithm requires for system success. In the past, because of cost restrictions, n was rarely larger than 3 and m was traditionally chosen as Ceiling[(n+1)/2] which we will call ***absolute majority voting.*** In [Sco87] Scott, Gault and McAllister show that if the output space is large and with true statistical independence of FTS versions, there is no need to choose m > 2 regardless of the size of n although considerable reliability gains occur with larger n. We will use the term ***2-of-n voting*** for this case.

With small output spaces we suggest that a third voting technique be considered, which we will call ***consensus majority voting*** . To motivate this technique consider the following scenario. Suppose we have n = 11 versions and output space cardinality of 3. Let a vector (i,j,k) represent the frequencies of the three possible output states (i + j + k = n) and let the first component, i, represent the frequency of the correct output. In this case, absolute majority is 6, but vectors (5,3,3), (5,4,2), or (5,2,4) may represent likely events which will be declared a system failure under absolute majority voting. Furthermore, the vectors (4,4,3) and (4,3,4) are the only cases in which a correct answer occurs when exactly four versions agree. But if three is chosen as the agreement number, there always exists another output on which more than three versions agree. In such cases an obvious strategy is to choose the output with the largest frequency, if such exists.

When there is more than one choice, as in this example when the output state frequency vector is (5,5,1) or (5,1,5), and if choosing a wrong answer or having no answer has the same impact on the system, then choosing one result with five identical outputs at random is a better strategy (on the average) than declaring system failure. In this example then there is a 50 percent chance that the correct output will be selected when this formal strategy is used.

In consensus majority voting the voter uses the following algorithm to select the "correct" answer:

- If there is an absolute majority agreement ($\geq$ Ceiling[(n+1)/2]) then this answer is chosen as the " correct" answer.
- If there is a unique maximum agreement, but this number of agreeing versions is less than Ceiling[(n+1)/2], then this answer is chosen as the "correct" one.
- If there is a tie in the maximum agreement number then one set is chosen at random and the answer associated with this set is chosen as the "correct" one.

We discuss this strategy further in the following sections and compare it with 2-of-n and absolute majority voting. We will first develop the mathematics for treating the difference between agreement and correctness.

III. The Correctness Factor

We first define a correctness factor c_i which is the probability that an output is correct given that i versions agree. If $Pr_c(i)$ is the probability that i versions are correct and $Pr_a(i)$ is the probability that i versions agree we have

$$\begin{aligned} c_i &= Pr\{\text{output is correct} \mid \text{i versions agree}\} \\ &= Pr\{\text{i versions are correct}\} \,/\, Pr\{\text{i versions agree}\} \\ &= Pr_c(i) \,/\, Pr_a(i) \end{aligned} \qquad (1)$$

Now

$$\begin{aligned} Pr\{\text{i versions agree}\} = \; & Pr\{\text{i versions agree and are correct}\} + \\ & Pr\{\text{i versions agree and are incorrect}\} \end{aligned} \qquad (2)$$

Also, using "correct" and "incorrect" to mean "output is correct" and "output is incorrect" respectively

$$\Pr\{\text{incorrect} \mid i \text{ versions agree}\} + \Pr\{\text{correct} \mid i \text{ versions agree}\} = 1$$

Hence,

$$\Pr\{\text{incorrect} \mid i \text{ versions agree}\} = 1 - c_i$$

Using the data domain approach of Scott et. al. [Sco83a,b], the reliability of an m-of-n system which we denote by $R_{m|n}$ becomes

$$R_{m|n} = \sum_{i=m}^{n} \Pr_c(i) = \sum_{i=m}^{n} c_i \Pr_a(i) \qquad (3)$$

We expect that the sequence

$$C = \{c_i \mid i = 2,\ldots,n\}$$

is nondecreasing, i.e. that the chance of an output being incorrect does not increase as the number of versions which agree increases. In particular, it is clear that

$$R_{m|n} \leq \max\{c_i\} \sum_{i=m}^{n} \Pr_a(i) \leq \max\{c_i\}$$

and hence if, for all i, $c_i < 1$ then $R_{m|n} < 1$.

For tractability we will assume that all software versions have the same reliability or probability of obtaining the correct answer for a given input. Let this reliability be p. Then we have

$$\Pr_c(i) = {}_nC_i\, p^i (1-p)^{n-i}$$

where ${}_nC_i$ denotes the number of combinations of n items taken i at a time. It follows that

$$\Pr_a(i) = \Pr_c(i) + {}_nC_i\, (1-p)^i\, p^{n-i} \qquad (4)$$

In the following section we consider the impact of small output spaces on a voting strategy.

Table 3.1 gives correctness factors as a function of version reliabilities when the output space is

binary. The correctness factors converge to 1 only for p>1/r where r is the size of the output space. (Indeed the sequence is decreasing for p < 1/r) .This is not an accident as we will show in the following section.

IV. Small Output Spaces

Let the output space have cardinality r and assume as above that all versions have the same reliability p. We further assume, for tractability, that the probability of failure of a version is independent of the failure of any other. Assume a labeling of the outputs 1,2 ..., r such that output 1 is the correct one and occurs n_1 times. Let q_i , i≥2 denote the probability that the incorrect output i will occur n_i times where

$$n_1 + n_2 + \ldots + n_k = n$$

and

$$p + q_2 + q_3 + \ldots + q_k = 1.$$

Then the probability that the correct output 1 will occur n_1 times is

$$P_c(n_1) = \sum_{\sum_{i=2}^{n} n_i = n-n_1} \frac{n!}{n_1!n_2!\ldots n_r!}\, p^{n_1} q_2^{n_2} \ldots q_r^{n_r} \qquad (5)$$

where i >1. The reliability of an m-of-n system becomes

$$R_{min} = \sum_{n_1=m}^{n} \left[\sum_{\sum_{i=2}^{n} n_i = n-n_1} \frac{n!}{n_1!n_2!\ldots n_r!}\, p^{n_1} q_2^{n_2} \ldots q_r^{n_r} \right] \qquad (6)$$

Equation (6) does not allow for multiple incorrect outputs, however. Let M $(i;n_j)$ denote that i replaces n_j in equation (5). That is,

$$M(i\,;n_j) = \sum_{\sum n_k = n-i} \frac{n!}{n_1!\ldots n_{j-1}!\; i!\; \ldots n_r!}\, p^{n_1} q_2^{n_2} \ldots q_{j-1}^{n_{j-1}} q_j^{i} \ldots q_r^{n_r} \qquad (7)$$

It follows that the correctness factor c_i then becomes

$$c_i = \frac{M(i\,;n_1)}{\sum_{j=1}^{r} M(i\,;n_j)} \qquad (8)$$

where i lies between Ceiling[(n+1)/2] and n. The terms $M(i;n_j)$ are the probabilities of exactly i identical incorrect outputs where $2 \le j \le r$. We note that the expression becomes considerably more complicated when i lies between Floor [(n+r-1)/r] and Ceiling [(n+1)/2] since the $M(i\,;n_j)$'s may have terms in common, i.e., it is possible to have more than one output occurring more than i times. For example, in a case in which r=3 and n=11 it is possible that the correct output and one incorrect output can each occur 5 times for the same input. In this case the denominator of equation (8) overestimates the correct result. We assume henceforth that $i \ge$ Ceiling [(n+1)/2].

For tractability we also assume the occurrence of each incorrect output has the same probability q, and (r-1)q = 1-p. (This assumption is also 'best case' in the sense that different probabilities tend to effectively reduce the output space requiring higher version reliability. We will discuss this further in the last section.) Equation (7) then becomes

$$M(i\,;n_j) = \sum_{\sum n_k = n-i} \frac{n!}{n_1!\ldots n_{j-1}!\; i!\; \ldots n_r!}\, p^{n_1} q^{n-n_1} \qquad (9)$$

Then we have

$$M(i;n_2) = M(i;n_3) = \ldots = M(i;n_k)$$

and equation (8) becomes

$$c_i = \frac{M(i\,;n_1)}{M(i\,;n_1) + (r-1)M(i\,;n_2)} \qquad (10)$$

From [Tri82], the marginal probability function of $M(i;n_j)$ is given by

$$M(i;n_j) = {}_nC_i\, q^i (1-q)^{n-i} \tag{11}$$

When $i = n_1$ the equation becomes

$$M(i;n_1) = {}_nC_i\, p^i (1-p)^{n-i} \tag{12}$$

Substituting (11) and (12) into (10) gives

$$c_i = \frac{[{}_nC_i\, p^i (1-p)^{n-i}]}{[{}_nC_i\, p^i (1-p)^{n-i} + (r-1){}_nC_i\, q^i (1-q)^{n-i}]} \tag{13}$$

From our comments above and equation (3), it follows that we cannot have R_{mln} converging to 1 unless the sequence $\{c_i\}$ converges to 1. In the following theorem we show that a necessary and sufficient condition that $\lim\{c_i\} = 1$ is that $p > 1/r$.

Theorem 4.1:

The sequence $\{c_i\}$ is increasing and $\lim\{c_i\} = 1$ (as $n \to \infty$) if and only if $p > 1/r$ ($r \geq 2$).

Proof:

The sequence $\{c_i\}$ shown in equation (13) can be simplified to

$$c_i = \frac{1}{1 + (r-1)(q/p)^i [(1-q)/(1-p)]^{n-i}} \tag{14}$$

In equation (14), let

$$Q_i = (q/p)^i [(1-q)/(1-p)]^{n-i} \tag{15}$$

then equation (14) becomes

$$c_i = 1/[1+(r-1)Q_i] .$$

Substitution of q=(1-p)/(r-1) into equation (15) gives

$$Q_i = \frac{(1-p)^{2i-n}(r-2+p)^{n-i}}{(r-1)^n p^i} \qquad (16)$$

We note that $\{c_i\}$ is an increasing sequence and its limit is one if and only if $\{Q_i\}$ is a decreasing sequence converging to zero. For the sequence $\{Q_i\}$ to be monotone we must have $(Q_{i+1}/Q_i) < 1$. Since,

$$\frac{Q_{i+1}}{Q_i} = \frac{(1-p)^2}{(r-2+p)p} = \frac{1-2p+p^2}{p(r-2)+p^2} < 1$$

or

$$p(r-2) + p^2 > 1 - 2p + p^2$$

or

$$(r-2)p > 1-2p$$

or

$$rp > 1$$

Hence, we must have

$$p > 1/r .$$

This proves the necessity.

Since the boundary reliability p is larger than 1/r, let Δ denote a positive number such that

$$p = 1/r + \Delta$$

Substituting $(1/r + \Delta)$ into equation (16), we have

$$Q_i = \frac{\left(1 - \frac{1}{r} - \Delta\right)^{2i-n}\left(r - 2 + \frac{1}{r} + \Delta\right)^{n-i}}{(r-1)^n\left(\frac{1}{r} + \Delta\right)^i} \qquad (17)$$

Then

$$\frac{Q_{i+1}}{Q_i} = \frac{\left(1 - \frac{1}{r} - \Delta\right)^2}{\left(r - 2 + \frac{1}{r} + \Delta\right)\left(\frac{1}{r} + \Delta\right)}$$

Because r is an integer and larger than or equal to two,

$$\left(\frac{1}{r} + \Delta\right)\left(r - 2 + \frac{1}{r} + \Delta\right) - \left(1 - \frac{1}{r} - \Delta\right)^2 = r + r\Delta - 1 > 1$$

Therefore, $\{Q_i\}$ is a decreasing sequence, and hence $\{c_i\}$ is an increasing sequence. When $i = n$, Q_n becomes

$$Q_n = \left(\frac{(r-1) - r\Delta}{(r-1) + (r-1)r\Delta}\right)^n$$

Because the fraction is less than one, Q_n approaches zero as n approaches infinity. Therefore, the limit of $\{c_i\}$ is one when p is larger than 1/r. **This proves sufficiency** and completes the proof of Theorem 4.1.

The following theorem gives sufficient conditions that

$$\lim_{n \to \infty} R_{mln} = 1$$

<u>Theorem 4.2:</u>

Let the output space have cardinality r and assume all components are independent and have the same reliability p. Further assume unique correct outputs. Then the following are sufficient conditions for

$$\lim_{n \to \infty} R_{mln} = 1$$

(1) $p > 1/r$.

(2) The agreement number m is equal to Floor[(n+r-1)/r]. If Floor[(n+r-1)/r] is zero, m is set to two. (We note that if n becomes arbitrarily large then m must also).

(3) When a version fails, the probability of occurrence of any incorrect output is q, where q=(1-p)/(r-1).

Proof:

The probability that the j^{th} incorrect item within the output space is generated by i versions is $M(i ; n_j)$. From equation (11), the marginal probability of $M(i ; n_j)$ is

$$M(i ; n_j) = {}_nC_i \, q_j^i \, (1 - q_j)^{n-i}$$

where q_j is the probability that the jth incorrect output is generated. When j is not smaller than the majority this incorrect output will be voted as correct in N-version programming. But it may or may not be voted as the correct answer when j is a number between m and Ceiling[(n+1)/2]. Depending on the voting strategy the probability that the j^{th} incorrect item may be chosen as the correct answer under m-of-n voting algorithm is no larger than

$$H_j = \sum_{i=m}^{n} {}_nC_i \, q_j^i \, (1 - q_j)^{n-i}$$

In [Aik55], the above binomial formula is approximated using the following expression

$$H_j = \frac{1}{\sqrt{2\pi}} \int_{k_1}^{k_2} e^{-\frac{i^2}{2}} di + \frac{1 - 2q_j}{6\rho\sqrt{2\pi}} \left[(1 - k^2) e^{-\frac{k^2}{2}} \right]_{k_1}^{k_2} + \omega \qquad (18)$$

where

$$k_1 = \frac{m - nq_j - \frac{1}{2}}{\sqrt{nq_j (1 - q_j)}} \qquad (19)$$

$$k_2 = \frac{m - nq_j + \frac{1}{2}}{\sqrt{nq_j (1 - q_j)}} \qquad (20)$$

$$\rho = \sqrt{nq_j(1-q_j)} \tag{21}$$

and the error term ω satisfies the inequality

$$|\omega| < \frac{.13 + .18|1-2p|}{\rho^2} + e^{-\frac{3\rho}{2}}$$

Let $p = (1/r + \Delta)$, where Δ is a positive number.

Because $q_j = q = (1\text{-}p)/(r\text{-}1)$,

we have $q_j = q = (1/r)\text{-}\partial$, where $\partial = \Delta/(r\text{-}1)$.

Substituting $q_j = (1/r)\text{-}\partial$ into equation (19) it becomes

$$k_1 = \frac{m - n\left(\frac{1}{r} - \partial\right) - \frac{1}{2}}{\sqrt{n\left(\frac{1}{r} - \partial\right)\left(1 - \frac{1}{r} + \partial\right)}}$$

Since $m \geq n/r$, (m=Floor[(n+r-1)/r]), we have

$$k_1 \geq k' = \frac{\partial n - \frac{1}{2}}{\sqrt{n\left(\frac{1}{r} - \partial\right)\left(1 - \frac{1}{r} + \partial\right)}}$$

$$= \frac{\partial n - \frac{1}{2}}{\beta\sqrt{n}}$$

$$= \frac{\partial}{\beta}\sqrt{n} - \frac{1}{2\beta\sqrt{n}}$$

where

$$\beta = \sqrt{\left(\frac{1}{r} - \partial\right)\left(1 - \frac{1}{r} + \partial\right)}$$

Obviously, when n approaches infinity, k' becomes arbitrarily large. Since k' is no larger than k_1 and k_2 ($k_1 < k_2$),

$$\lim_{n \to \infty} k_1 = \infty$$

and

$$\lim_{n \to \infty} k_2 = \infty$$

When n approaches infinity, ρ becomes arbitrarily large. The error term ω in equation (18) approaches zero, and the limit can be written as

$$\lim_{n \to \infty} H_j = \lim_{k_1, k_2 \to \infty} \left[\frac{1}{\sqrt{2\pi}} \int_{k_1}^{k_2} e^{-\frac{i^2}{2}} di + \frac{1 - 2q_j}{6\rho\sqrt{2\pi}} [(1 - k^2)e^{-\frac{k^2}{2}}]_{k_1}^{k_2} \right] \quad (22)$$

The integral in the above equation is the accumulation function of the standard normal distribution. Since k_1 and k_2 approach infinity the limit is zero.

In treating the limit of the second part, let

$$(H_j)_2 = (1 - k)e^{-\frac{k^2}{2}} \quad (23)$$

Applying L'Hopital's rule and differentiating both numerator and denominator with respect to k gives

$$(H_j)_2' = \frac{1 - 2k}{ke^{\frac{k^2}{2}}}$$

$$(H_j)_2'' = \frac{1}{e^{\frac{k^2}{2}} + k^2 e^{\frac{k^2}{2}}}$$

Obviously, the numerator of the above equation approaches zero when k becomes very large. Therefore, we have

$$\lim_{k \to \infty} (H_j)_2 = \lim_{k \to \infty} (H_j)''_2 = 0.$$

The second part of equation (22) consists of two items

$$(1 - k_1^2)\, e^{-\frac{k_1^2}{2}} \text{ and } (1 - k_2^2)\, e^{-\frac{k_2^2}{2}}$$

Since the values of both approach zero as k_1 and k_2 become arbitrarily large, the difference between the two approaches zero. This establishes that as n becomes arbitrarily large the value computed by equation (18) approaches zero.

Since the occurrence of each incorrect output has the same probability of appearing, the unreliability of this m-of-n software system (F_{mln}) = 1 - R_{mln} satisfies

$$F_{mln} < (r - 1) \sum_{i=m}^{n} {}_nC_i\, q^i\, (1 - q)^{n-i} \qquad (24)$$

Therefore, when n approaches infinity the right side of the inequality is zero. Because F_{mln} should be non-negative, it also approaches zero. The reliability of the system approaches one. **This completes the proof** of Theorem 4.2.

V. Examples

In this section we present numerical examples which illustrate the effect on the system reliability of different version reliabilities, different output space cardinality, and different voting strategies.

In Table 5.1 and Figure 5.1 we show results obtained using equation (6) with m=Ceiling[(n+1)/2] and r=2. This is the classical majority voting approach with a binary output space. The boundary version reliability in this case is 1/r = 0.5. The three rows in the middle of Table 5.1 show that the version reliability must be larger than the boundary version reliability in order to improve the performance of the system. Figure 5.1 shows that with a fixed version reliability larger than 0.5

system reliability increases when more versions are added. This, of course, is in agreement with findings of Eckhardt and Lee [Eck85] who also studied the absolute majority voting process.

Again using equation (6) if we assume an output space cardinality of r=3 then Table 5.2 and Figure 5.2 show the effect on system reliability of varying the version reliabilities and the number of versions m for the consensus majority voting strategy. The minimal agreement number is Floor[(n+r-1)/r] = Floor[(n+2)/3]. The average boundary reliability of the versions is 1/r = 1/3. Below this reliability value the sequence of correctness factors $\{c_i\}$ decreases and the system reliability deteriorates as more versions are added. All versions are assumed to have the same reliability, and all failure states (j=2,3) the same probability (1-p)/(1-r)=(1-p)/2 of being excited.

Table 5.3 and Figure 5.3 summarize the effect of the 2-of-n voting strategy for different numbers of components. The reliability was computed using equations (3) and (4). The agreement number is m=2, and it is assumed that the output space cardinality is infinite. As the number of components in the system increases, the reliabilities rapidly decrease, but here this effect is related to the number of components involved in the voting rather than the output space cardinality. Of course, unless $c_2 = c_3 = \ldots = c_n = 1$ this voting strategy can lead to disaster.

The relationship between r and voting strategies is illustrated in Figure 5.4 for n=15. It is important to note that both the absolute majority and the 2-of-n are effectively output space insensitive and can lead to ambiguous or nonunique results. For odd n the former behaves as if r=2 since for absolute majority voting the agreement number is Ceiling[(n+1)/2] which is equivalent to letting r=2 in the agreement number equation for consensus majority voting, Floor[(n+r-1)/2]. For even n Ceiling[(n+1)/2]>Floor[(n+2-1)/2]. Therefore from equations (3) and (4) it follows that given r=2 for even n the reliability of the system using absolute majority voting will be lower than when consensus majority is used, while for odd n it will be equal to it. The 2-of-n voting behaves as if r=infinity since for infinite r the consensus majority agreement number reduces to 2 (see Theorem 4.2). The consensus majority voting is r sensitive and therefore will perform better than absolute majority voting for r>2 since Floor[(n+r-1)/r]≤Ceiling[(n+1)/2]. The absolute majority represents a lower limit of the consensus majority voting with r=2, while the 2-of-n is an upper limit.

Dependence of the system reliability on the output space cardinality is further illustrated in Figure 5.5 for consensus majority with n=5 and n=15. Failure state probabilities are the same for all j=2..r incorrect outputs. We note that the asymptotic system reliability (r=∞) corresponds to 2-of-n voting approach, while the r=2 point corresponds to the absolute majority voting. Equations (3) to

(6) with consensus voting and simulation were used to compute the data points shown in Figures 5.4 and 5.5.

VI. Simulation

The relationships given in section IV were derived by assuming that the probabilities of all failure states are equally likely. In practice this may not be true. In fact, it is quite possible that one of the failure states j, $2 \leq j \leq r$ is preferentially excited because of very high visibility (under given input conditions) of the fault(s)/errors mapping into it. In the extreme, even in a large output space, this leads to the behaviour of the fault-tolerant system as if the output space cardinality is small, i.e. highly visible errors force a partitioning on the output space into equivalence classes which effectively reduces the output space cardinality.

Another approximation that was made is the assumption that all the components have the same reliability. In practice a range of reliabilities around a mean value, p, would be expected. To study the influence of the scatter of individual component reliabilities and of the scatter in the probabilities of incorrect outputs, and to check on analytical solutions we used simulation.

The model we have used is illustrated in Figure 6.1. A single component i is assumed to exhibit a probability $q_i = (1-p_i)$ of failing. We do not separately model different errors contributing to this average failure rate, and we assume that all the components exhibit mutual independence with respect to the probability of failure. For each simulated input the component state (failed, not_failed) is chosen randomly with the probability, q_i, assigned to that component. If the final component state is a failure state the actual output state j, one of the (r-1) incorrect outputs, is selected randomly with the conditional probability P(j|i-failed) associated with that output. The process is repeated for all n components. The final states of the components are then voted using the absolute majority, consensus majority and 2-of-n strategies.

Simulation of systems described by the equations given in section IV yielded results that coincided with the computations obtained using analytic solutions to within the confidence interval of the simulation runs.

The influence of the scatter in component reliabilities is illustrated in Figures 6.2 and 6.3. The

standard deviation of component reliability, the square root of $\sigma_p^2=\Sigma[(p_i-p)^2/(n-1)]$, where $p = \Sigma p_i/n$, for i=1..n, is used to measure the dispersion of the component reliability values. In Figure 6.2 we plot system reliability against the standard deviation of the component reliability using n=5 with r=4 and p=0.95. It was assumed that each incorrect output state j has an equal probability (1-p)/(r-1) of being selected. In Figure 6.3 we have n=5, r=4 and p=0.623. Each pair of points (majority-absolute) shown in Figures 6.2 and 6.3 was obtained from a separate 100,000 case simulation run.

From the figures we see that the larger the standard deviation of the component reliabilities the more reliable the system. This confirms that from the point of view of component reliabilities the equations discussed in section IV provide a conservative estimate of the system behaviour since they use the same component reliability value for all components and imply a standard deviation of zero. Of course, if scatter is large enough then it may happen that one of the components is in fact more reliable than the system as a whole under some or all of the discussed voting strategies.

For example, given that n=5, r=4, and p=0.623 with $\sigma_p=0.186$ we can have $p_1=0.759$, $p_2=0.522$, $p_3=0.357$, $p_4=0.819$, $p_5=0.658$. These component reliability values give system reliability of 0.735 for absolute majority voting, and 0.851 for consensus majority voting. Clearly, component i=4 on its own is more reliabile than the 5-version system under absolute majority vote, and is marginally worse than the system under consensus majority voting. As another example consider a system composed of more reliable components. Let n=5, r=4, and p=0.95000 with $\sigma_p=0.05333$ which we can produce with $p_1=0.96456$, $p_2=0.91313$, $p_3=0.87732$, $p_4=0.99999$, $p_5=0.99500$. The resulting system reliability under absolute majority voting is 0.99953, and under consensus majority voting is 0.99979. Component i=4 is more reliable than the system under either of the strategies. Values in first example were computed using 2,000,000 case simulations giving a 95% confidence range about obtained system reliabilities of about $\pm$ 0.00003. In the second example we used a 10,000,000 case simulation giving 95% confidence limits of about $\pm$ 0.000013 about the reported values.

Therefore, if all the components are nearly equally reliable, i.e. scatter is small, then using equations given in section IV to predict system reliability will provide a conservative estimate of this reliability. But if the scatter of the reliabilities is large it means that at least one of the

components is much more reliable than the average over all the components, and it may happen that there is at least one component which is more reliable than the N-version system. In such a situation, one should either reduce the system by discarding the most unreliable component(s), or perhaps employ modified voting strategies. To illustrate this consider the following.

Returning to the second example above and discarding component p_3=0.87732 and keeping the other four, results in system reliability of 0.99635 under absolute majority voting and 0.99937 under consensus majority voting. By discarding p_3 and p_2 we obtain 0.99979 for both absolute and consensus majority voting. It is only when we reduce the system down to two components that the sytem reliability becomes larger than that of component 4. Simulations ran for 10,000,000 cases giving 95% confidence limits on system reliability of about $\pm$ 0.00001. Clearly, when working with high reliability components, it is very important to have a well balanced set of components of nearly equal reliability in order to achieve best results. A possible adaptive voting strategy would be given information about the individual component reliabilities and may, for example, attach more importance to answers from sets containing the most reliable component(s). This is a subject for future research.

Figure 6.4 illustrates the effect of variation in the conditional probabilities of selecting incorrect output states. If $q_{ij}=q_i P(j|i\text{-failed})$ represents the probability of selecting j^{th} incorrect output for i^{th} component, where P(j|i-failed) is the conditional probability that j^{th} state will be chosen, then $q_i=\Sigma q_{ij}=q_i \Sigma P(j|i\text{-failed})=1-p_i$, j=2..r. Let $P_i=\Sigma P(j|i\text{-failed})/(r-1)=1/(r-1)$ be the average conditional probability. Then the standard deviation shown in Figure 6.4 is the square root of $\sigma_p^2 = \Sigma [(P(j|i\text{-failed})-P_i)^2/(r-2)]$, where the sum is over j = 2..r output states. Simulations were performed assuming that for individual component reliabilities $p_1=p_2=...=p_n=p$. We see that the larger the scatter of the conditional failure probabilities, assuming the same p for all components, the more the system behaviour tends towards that associated with the lower r values, i.e. toward that exhibited when absolute majority voting is employed. The curve for the latter is, of course, level since absolute majority voting is r insensitive (effective output space becomes binary, r=2). Simulation for each pair of points ran for 100,000 test cases.

As an example, let n=5, p=0.95 (all components), and r=4 with equal conditional failure probabilities for all incorrect outputs (j=2,3,4; P(j|i-failed)=1/3). Then absolute majority voting has

system reliability of 0.99884 and consensus majority voting has reliability 0.99994. On the other hand, if we let r=11 but P(2|i-failed)=0.99910 while for j=3..11 P(j|i-failed)=0.00010, then consensus majority reliability drops to 0.99884 which is equal to that obtained by absolute majority voting, and is equivalent to an effective reduction of the output space cardinality to r=2. Example values were obtained by simulation, and their standard deviation is 0.000025.

The above discussion leads to the conclusion that for conservative estimates we should use r=2 and an average p value. However, in practical applications use of consensus majority voting is recommended since it provides automatic adaptation of the voting strategy to the component reliability and output space characteristics. In the lower limit the reliability provided by consensus majority is never worse than the absolute majority voting, while in the upper limit it is equivalent to the 2-of-n voting strategy.

VII. Conclusions

We have analyzed fault-tolerant software systems using N-Version Programming and different voting algorithms assuming output spaces with small cardinality and version failure independence. We have proposed an alternative voting strategy which we call consensus majority voting to treat cases when there may be agreement among incorrect outputs, a case which can occur with small output spaces. Consensus majority voting provides automatic adaptation of the voting strategy to varying component reliability and output space characteristics. We show that if r is the cardinality of the output space then 1/r is a lower bound on the average reliability of fault-tolerant system versions below which system reliability begins to deteriorate as more versions are added.

VIII. References

[Aik55] H.H. Aiken et. al, "Tables of the Cumulative Binomial Probability Distribution", Harvard University Press, Mass., 1955.

[Avi77] A. Avizienis and L. Chen, "On the Implementation of N-version Programming for Software Fault-Tolerance During Program Execution", Proc. COMPSAC 77, 149-155, 1977.

[Avi84] A. Avizienis and P.A. Kelly, "Fault-Tolerance by Design Diversity: Concepts and Experiments", Computer, Vol. 17, pp. 67-80, 1984.

[Eck85] D.E. Eckhardt, Jr. and L.D. Lee, "A Theoretical Basis for the Analysis of Multiversion Software Subject to Coincident Errors", IEEE Trans. Soft. Eng., Vol. SE-11(12), 1511-1517, 1985.

[Kni86] J.C. Knight and N.G. Leveson, "An Experimental Evaluation of the assumption of Independence in Multiversion Programming IEEE Trans. Soft. Eng., Vol. SE-12(1), 96-109,

1986.

[Sco83a] R.K. Scott, "Data Domain Modeling of Fault Tolerant Software Reliability", Ph.D. Dissertation, North Carolina State University, Raleigh, North Carolina, 1983

[Sco83b] R.K. Scott, J.W. Gault, D.F. McAllister and J. Wiggs, "Experimental Validation of Six Fault-Tolerant Software Reliability Models", Proc. IEEE 14th Fault-Tolerant Computing Symposium, pp. 102-107, 1983

[Sco84] R.K. Scott, J.W. Gault, D.F. McAllister and J. Wiggs, "Investigating Version Dependence in Fault-Tolerant Software", AGARD 361, pp. 21.1-21.10, 1984

[Sco87] R.K Scott, J. W. Gault and D. F. McAllister, "Fault-Tolerant Reliability Modeling", IEEE Trans. Soft. Eng. Vol. SE-13, No. 5, pp. 582-592, 1987

[Tri82] K.S. Trivedi, "Probability and Statistics with Reliability, Queueing, and Computer Science Applications, Prentice-Hall, New Jersey, 1982.

[Vou85] M.A. Vouk, D.F. McAllister, K.C. Tai, "Identification of correlated failures of fault-tolerant software systems", in Proc. COMPSAC 85, 437-444, 1985.

[Vou86] M.A. Vouk, D.F. McAllister, and K.C. Tai, "An Experimental Evaluation of the Effectiveness of Random Testing of Fault-tolerant Software", Proc. Workshop on Software Testing, Banff, Canada, IEEE CS Press, July 1986.

Table 3.1 Correctness factors as a function of version reliability under the assumption of version failure independence for 15 functionally equivalent program versions of equal reliability, p. The output space cardinality is r=2, the boundary reliability is $1/r = 1/2 = 0.5$.

c_i

i	p=0.49	p=0.50	p=0.51	p=0.80
2	0.6083	0.5000	0.3917	0.0000
3	0.5891	0.5000	0.4110	0.0000
4	0.5696	0.5000	0.4304	0.0001
5	0.5498	0.5000	0.4502	0.0010
6	0.5300	0.5000	0.4700	0.1154
7	0.5100	0.5000	0.4900	0.2000
8	0.4900	0.5000	0.5100	0.8000
9	0.4700	0.5000	0.5300	0.8846
10	0.4502	0.5000	0.5498	0.9990
11	0.4304	0.5000	0.5696	0.9999
12	0.4110	0.5000	0.5891	1.0000
13	0.3917	0.5000	0.6083	1.0000
14	0.3728	0.5000	0.6272	1.0000
15	0.3543	0.5000	0.6457	1.0000

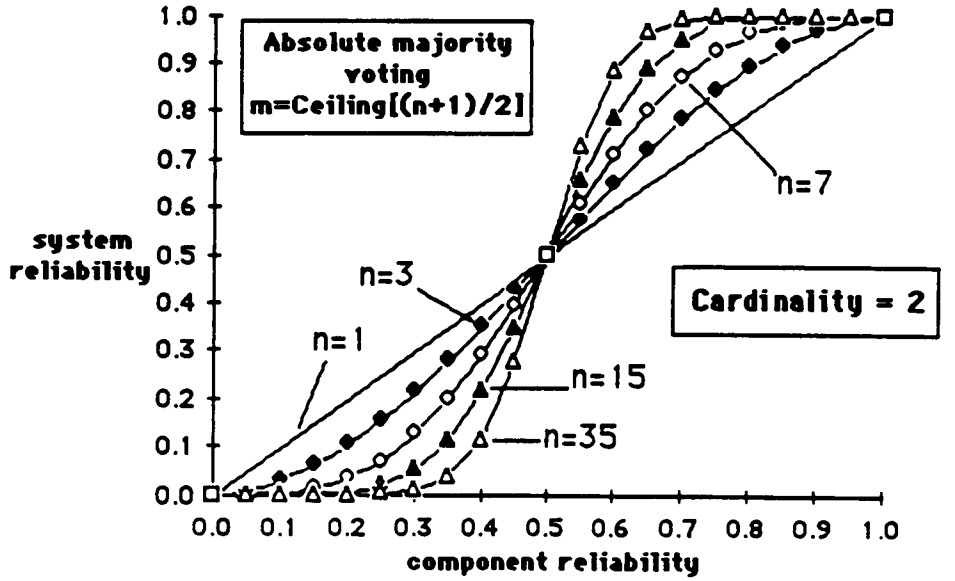


Figure 5.1 System reliability vs. component reliability for absolute majority voting strategy. Number of components used for voting is "n", the agreement number is m, r=2 , and boundary version reliability is 1/r=0.5.

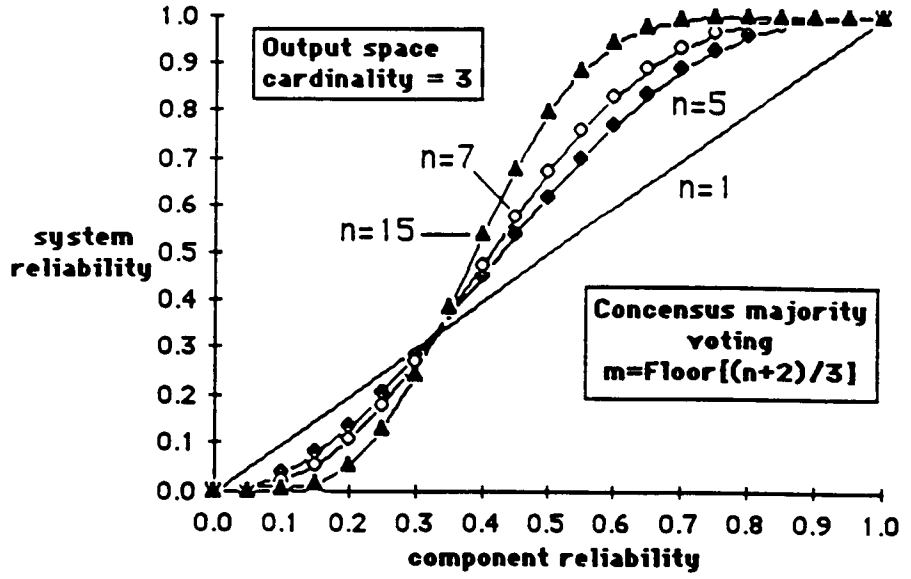


Figure 5.2 System reliability vs. component reliability for the consensus majority voting strategy. The number of voting components is "n", the agreement number is m, r=3, and the boundary version reliability is 1/r = 0.3333.

Table 5.1 The reliability of the N-version programming system using majority voting (r=2). Reliability of the system is p, the number of components participating in a vote is n.

p	n=3	n=7	n=15	n=35
0.0500	0.00725	0.00039	0.00000	0.00000
0.1000	0.02800	0.00273	0.00031	0.00000
0.1500	0.06075	0.01210	0.00361	0.00000
0.2000	0.10400	0.03334	0.00424	0.00003
0.2500	0.15625	0.07056	0.01730	0.00070
0.3000	0.21600	0.12604	0.05001	0.00642
0.3500	0.25175	0.19985	0.11323	0.03363
0.4000	0.35200	0.28979	0.21310	0.11431
0.4500	0.42525	0.39171	0.34650	0.27514
0.4900	*0.48500*	*0.47813*	*0.46861*	*0.45257*
0.5000	*0.50000*	*0.50000*	*0.50000*	*0.50000*
0.5100	*0.51500*	*0.52187*	*0.53139*	*0.54743*
0.5500	0.57475	0.60829	0.65350	0.72486
0.6000	0.64800	0.71021	0.78690	0.88569
0.6500	0.71825	0.80015	0.88677	0.96637
0.7000	0.78400	0.87396	0.94999	0.99358
0.7500	0.84375	0.92944	0.89270	0.99930
0.8000	0.89600	0.96667	0.99579	0.99997
0.8500	0.93925	0.98790	0.99639	1.00000
0.9000	0.97200	0.99727	0.99969	1.00000
0.9500	0.99275	0.99961	1.00000	1.00000
0.9900	0.99990	1.00000	1.00000	1.00000
0.9990	1.00000	1.00000	1.00000	1.00000

Table 5.2 The reliability of the N-version programming system using consensus majority voting (r=3). Reliability is p, the number of components participating in a vote is n.

p	n=3	n=7	n=11	n=15
0.0500	0.00247	0.00131	0.00049	0.00008
0.1000	0.03590	0.01708	0.00646	0.00243
0.1500	0.07843	0.05064	0.02888	0.01584
0.2000	0.13472	0.10502	0.07584	0.05420
0.2500	0.20239	0.17870	0.15164	0.12884
0.3000	0.27884	0.26785	0.25339	0.24154
0.3300	*0.32779*	*0.32662*	*0.32511*	*0.29537*
0.3330	*0.33266*	*0.33258*	*0.33251*	*0.33237*
0.3333	*0.33333*	*0.33333*	*0.33333*	*0.33333*
0.3340	*0.33468*	*0.33484*	*0.33499*	*0.33527*
0.3400	*0.34448*	*0.34683*	*0.34994*	*0.35280*
0.3500	0.36727	0.37141	0.37519	0.38248
0.4000	0.44704	0.47123	0.50430	0.53410
0.4500	0.53321	0.57412	0.62970	0.67710
0.5000	0.61719	0.67090	0.74145	0.79646
0.5500	0.69650	0.75753	0.83292	0.88482
0.6000	0.76896	0.83117	0.90141	0.94252
0.6500	0.83276	0.89030	0.94790	0.97534
0.7000	0.88653	0.93474	0.97604	0.99125
0.7500	0.92944	0.96549	0.99804	0.99758
0.8000	0.96128	0.98458	0.99730	0.99953
0.8500	0.98253	0.99470	0.99947	0.99995
0.9000	0.99448	0.99887	0.99995	0.99999
0.9500	0.99926	0.99992	1.00000	1.00000
0.9900	0.99999	1.00000	1.00000	1.00000
0.9990	1.00000	1.00000	1.00000	1.00000

Table 5.3 The reliability of the N-version programming system using 2-of-n voting strategy ($r=\infty$). System reliability is p, the number of components participating in a vote is n.

p	n=3	n=7	n=11	n=15
0.0500	0.00725	0.00019	0.00001	0.00000
0.1000	0.02800	0.14969	0.30264	0.45096
0.1500	0.06075	0.28342	0.50781	0.68141
0.2000	0.10400	0.42328	0.67788	0.83287
0.2500	0.15625	0.55505	0.80290	0.91982
0.3000	0.21600	0.60758	0.88701	0.96473
0.3500	0.28175	0.76620	0.93492	0.98582
0.4000	0.35200	0.84137	0.96977	0.99783
0.4500	0.42525	0.89758	0.98601	0.99831
0.5000	0.50000	0.93750	0.99414	0.99951
0.5500	0.57475	0.96429	0.99779	0.99989
0.6000	0.64800	0.98116	0.99927	0.99997
0.6500	0.71825	0.99099	0.99980	1.00000
0.7000	0.78400	0.99621	0.99995	1.00000
0.7500	0.84375	0.99865	0.99999	1.00000
0.8000	0.89600	0.99963	1.00000	1.00000
0.8500	0.93925	0.99993	1.00000	1.00000
0.9000	0.97200	0.99999	1.00000	1.00000
0.9500	0.99275	1.00000	1.00000	1.00000
0.9900	0.99970	1.00000	1.00000	1.00000
0.9990	1.00000	1.00000	1.00000	1.00000

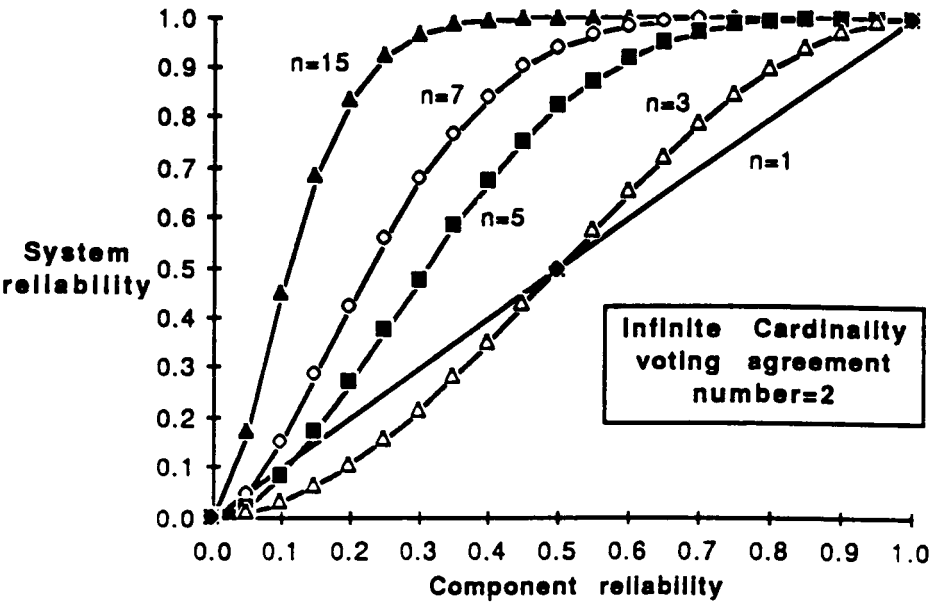


Figure 5.3 System reliability vs. component reliability assuming infinite cardinality of the output space under 2-of-n voting strategy.

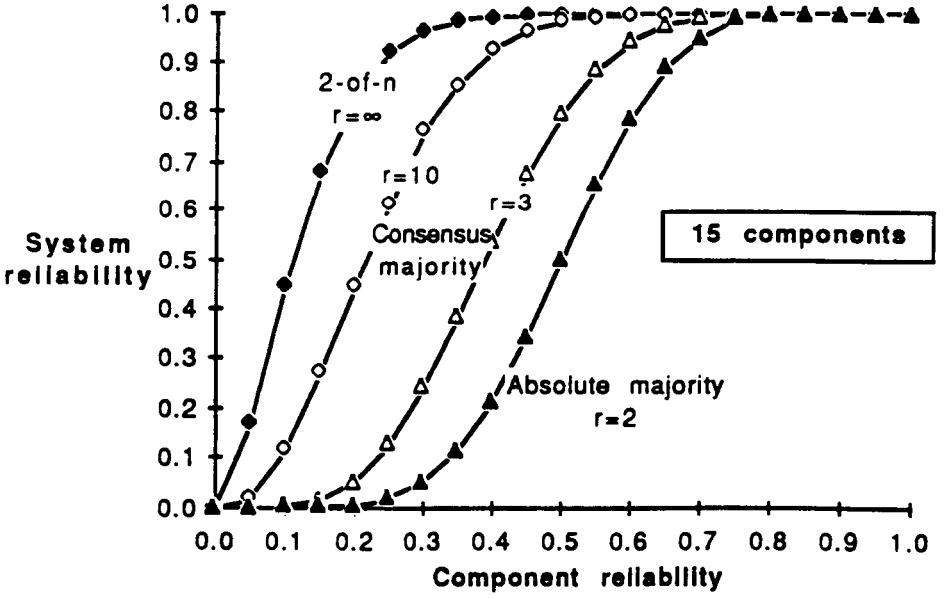


Figure 5.4 System reliability vs. component reliability for n=15 in the range r=2 to r=∞, under appropriate voting strategies. Probability of each j=2..r failure state is (1-p)/(r-1). Simulation was used to compute the r=10 curve.

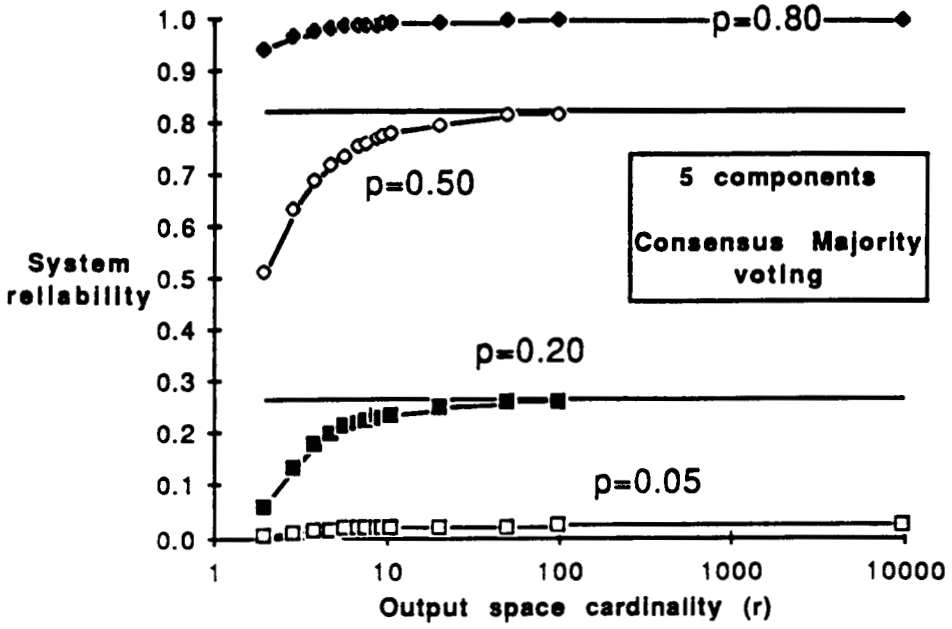


Figure 5.5a System reliability vs. Output space cardinality for n=5 using consensus majority voting. All components have the same reliability, p. Probability of each j=2..r failure state is (1-p)/(r-1). Majority of the data points were computed by simulation.

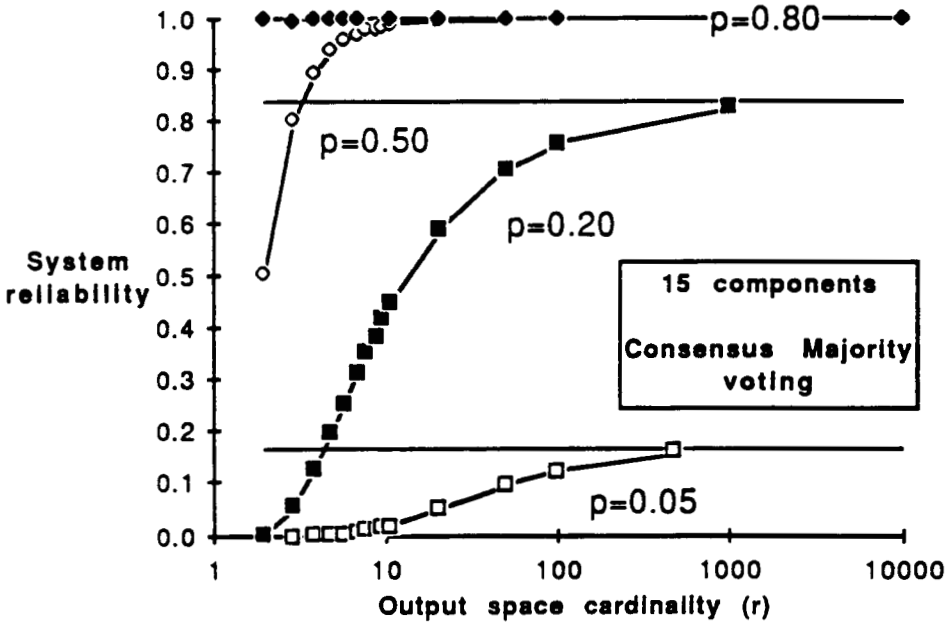


Figure 5.5b System reliability vs. output space cardinality for n=15 using consensus majority voting. All components have the same reliability, p. Probability of each j=2..r failure state is (1-p)/(r-1). Majority of the data points were computed by simulation.

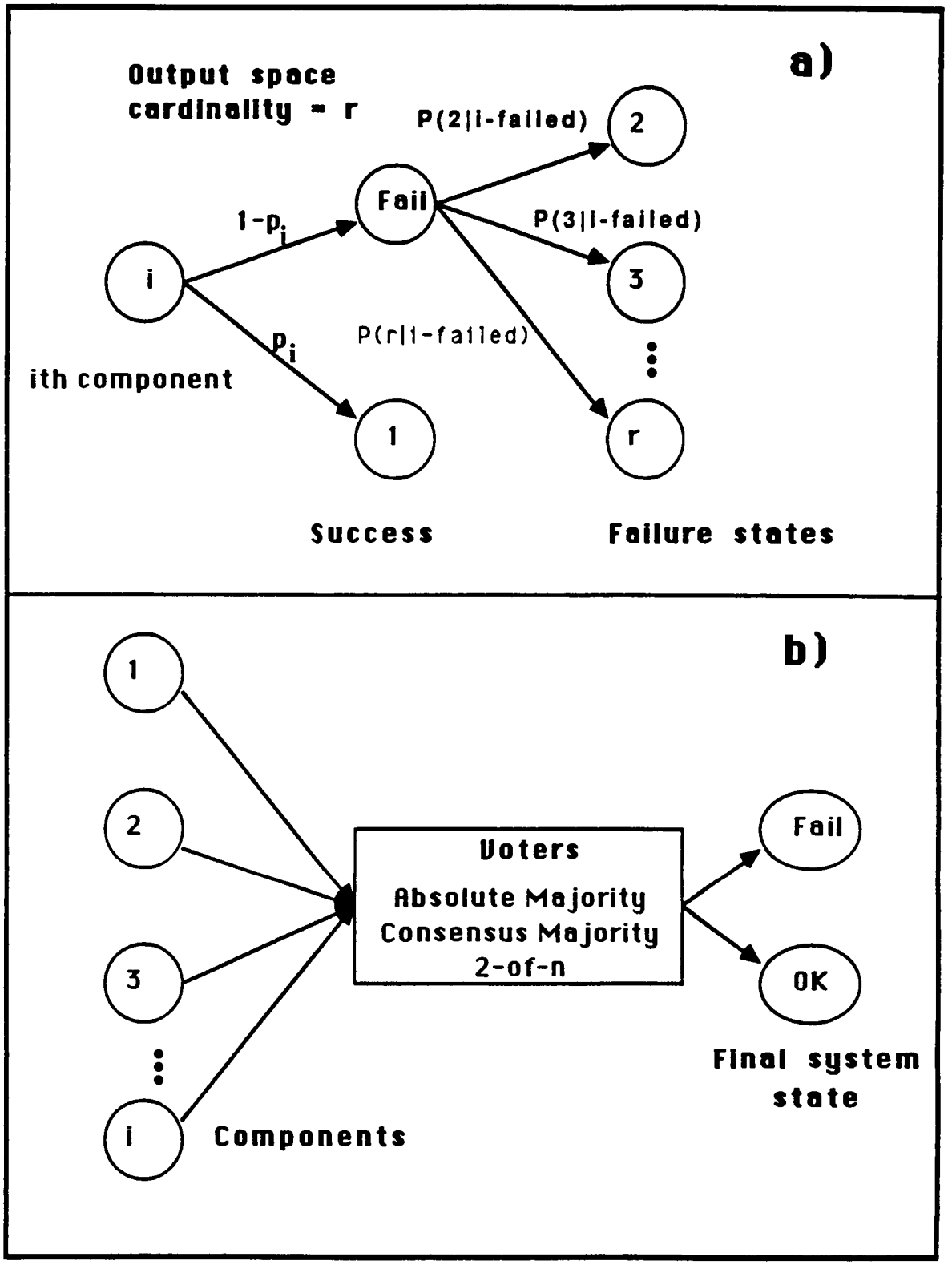


Figure 6.1 Schematic representation of the simulation states for a single component (a), and the voting process (b). Individual component reliability is represented by p_i, and the conditional probability of failing with state j = 2 .. r by P(jli-failed).

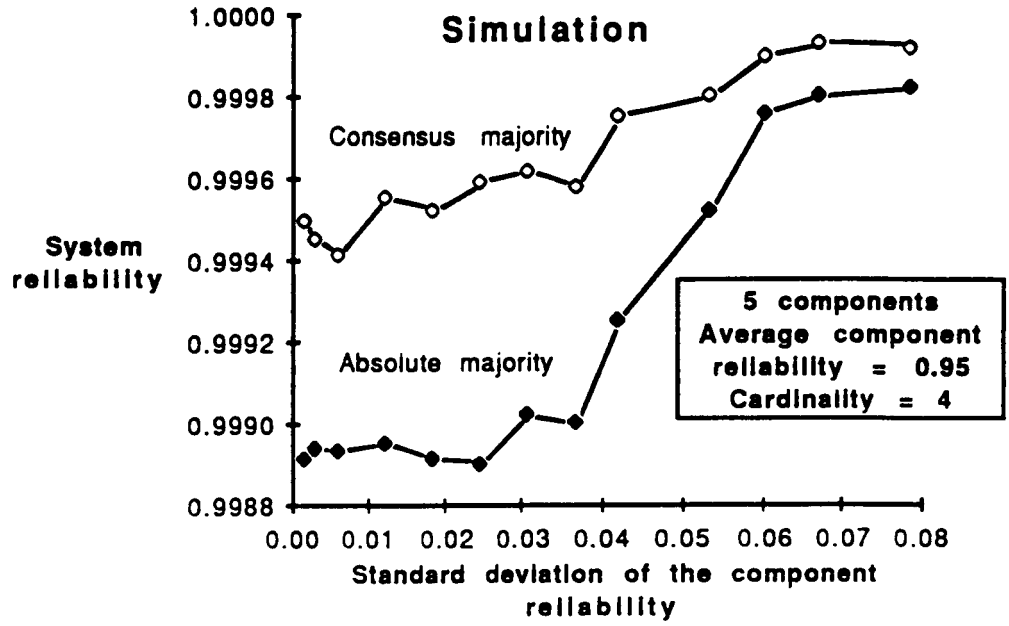


Figure 6.2 System reliability vs. the standard deviation of the component reliability. Probability of each failure state is (1-p)/(r-1) = 0.05/3.

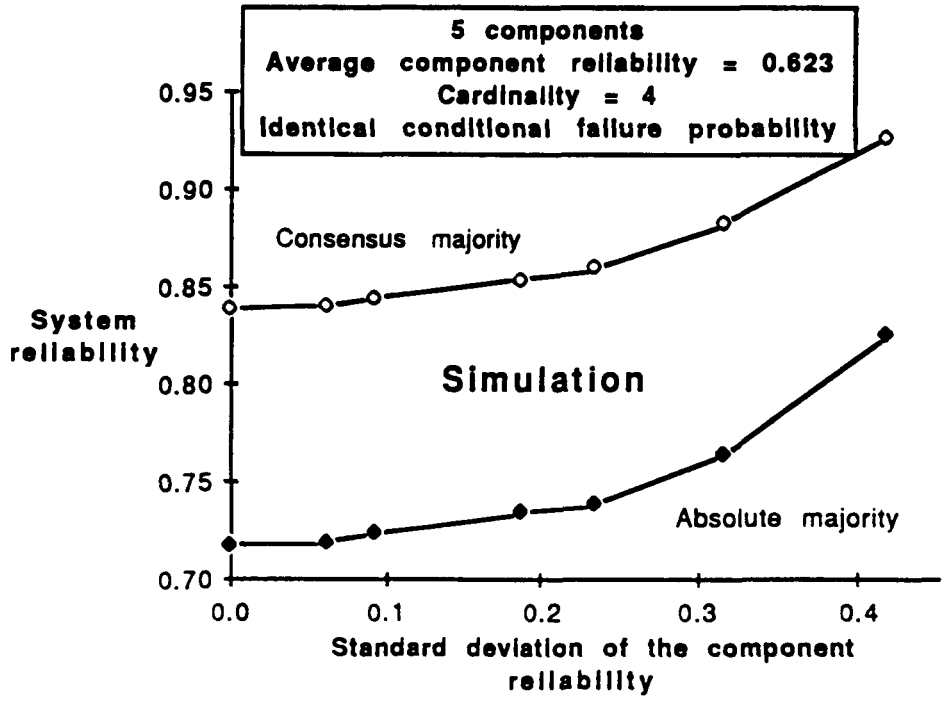


Figure 6.3 System reliability vs. the standard deviation of the component reliability. Probability of each failure state is (1-p)/(r-1) = 0.377/3.

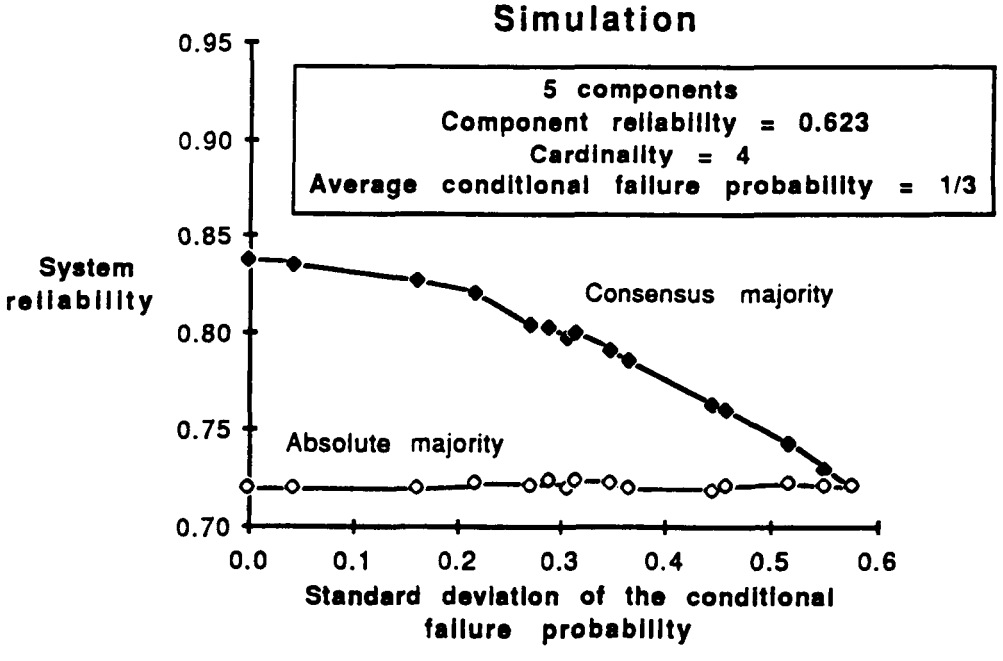


Figure 6.4 System reliability vs. the standard deviation of the conditional failure state probability. All components have identical reliability p = 0.623.