The University of Alabama in Huntsville
Electrical and Computer Engineering
Homework #6 Solution
CPE 633 01
Spring 2008

Chapter 5: Problem 5(25 points),
Chapter 6: Problem 1(20 points), 3(15 points), 4(20 points), 7(20 points)

5. In this problem, we will use Bayes's law to provide some indication of whether bugs still remain in the system after a certain amount of testing. Suppose you are given that the probability of uncovering a bug (given that at least one exists) after t seconds of testing is $1 - e^{-\mu t}$. Your belief at the beginning of testing is that the probability of having at least one bug is q. (Equivalently, you think that the probability that the program was completely bug-free is p = 1 - q.) After t seconds of testing, you fail to find any bugs at all. Bayes's law gives us a concrete way in which to use this information to refine your estimate of the chance that the software is bug-free: find the probability that the software is actually bug-free, given that you have observed no bugs at all, despite t seconds of testing.

Let us use the following notation:

- A is the event that the software is actually bug-free.
- B is the event that no bugs were caught despite t seconds of testing.

(a) Show that $\Pr ob\{A \mid B\} = \dfrac{p}{p = qe^{-\mu t}}$

(b) Fix p = 0.1, and plot curves of Prob{A|B} against t for the following parameter values: $\mu = 0.001$, 0.01, 0.1, 1.0, $0 \le t \le 10000$.
(c) Fix $\mu = 0.01$ and plot curves of Prob{A|B} against t for the following parameter values: p = 0,1, 0,2, 0,3, 0,4, 0,5.
(d) What conclusions do you draw from your plots in (b) and (c) above?

(a) Based on the conditional probability formula, we have:

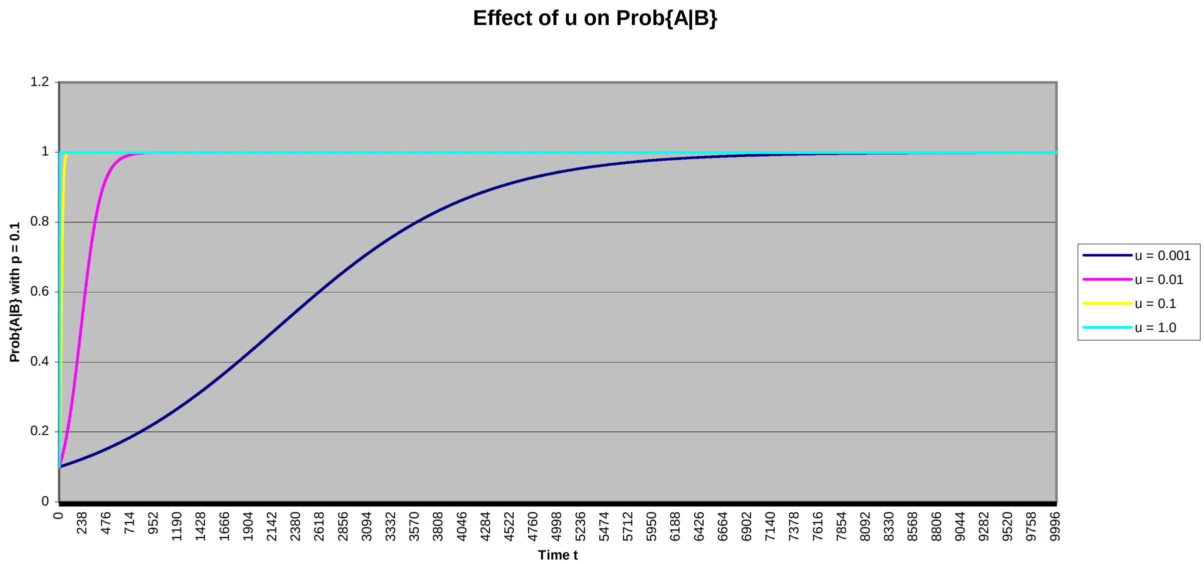
$$\Pr ob\{A \mid B\} = \frac{\Pr ob\{A \cap B\}}{\Pr ob\{B\}} = \frac{\Pr ob\{B \mid A\} \Pr ob\{A\}}{\Pr ob\{B \cap A\} + \Pr ob\{B \cap C\}}$$

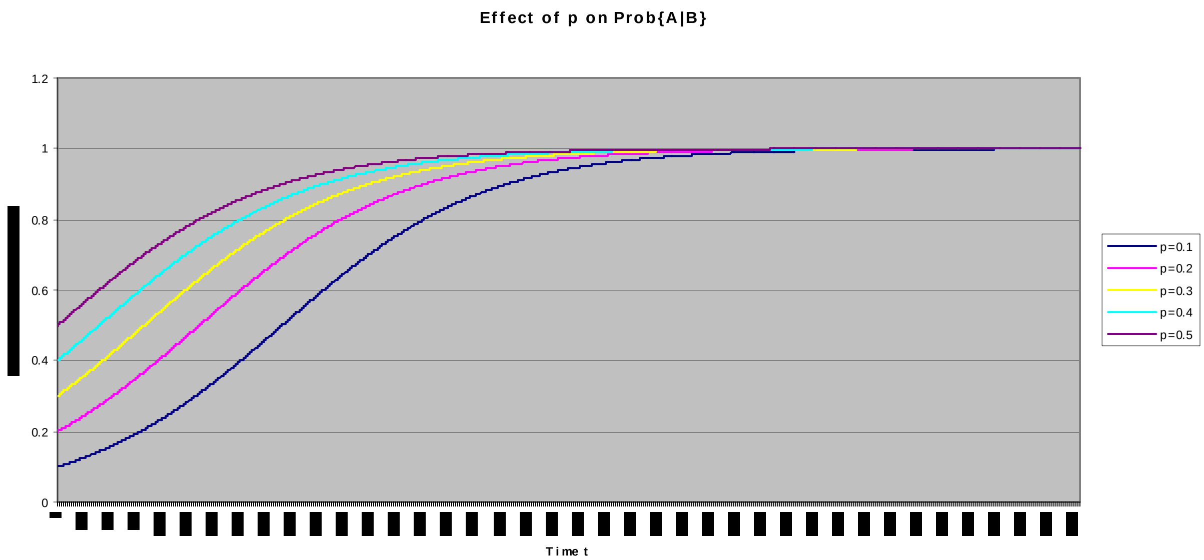
where C is the complement of A, i.e., the event that the software has at least one bug. Let us look at each of the terms in this expression. Clearly, Prob{B|A} = 1. As for Prob{A}, the best estimate we have is that this is equal to p. To calculate Prob{B $\cap$ A} and Prob{B $\cap$ C}, we will need to invoke the conditional probability formula once more:

Prob{B $\cap$ A} = Prob{B|A}Prob{A} = 1*p = p
Prob{B $\cap$ C} = Prob{B|C}Prob{C} = $e^{-\mu t}q$

Once again, we are using the best estimates we have for Prob{A} and Prob{B}. Substituting back into the expression for Prob{A|B}, we have: $\Pr ob\{A \mid B\} = \dfrac{p}{p + qe^{-\mu t}}$

(b) Some numerical results are shown

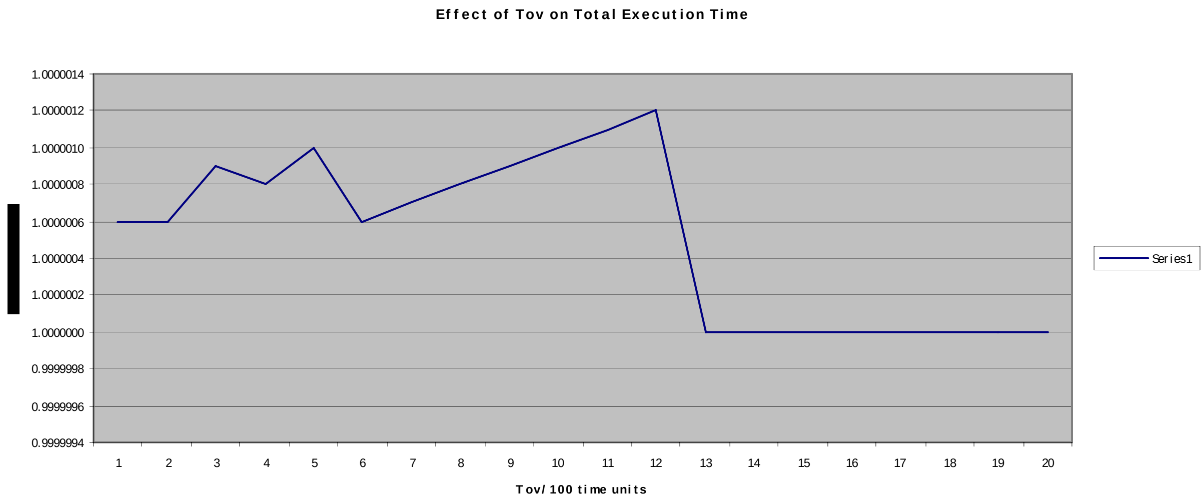




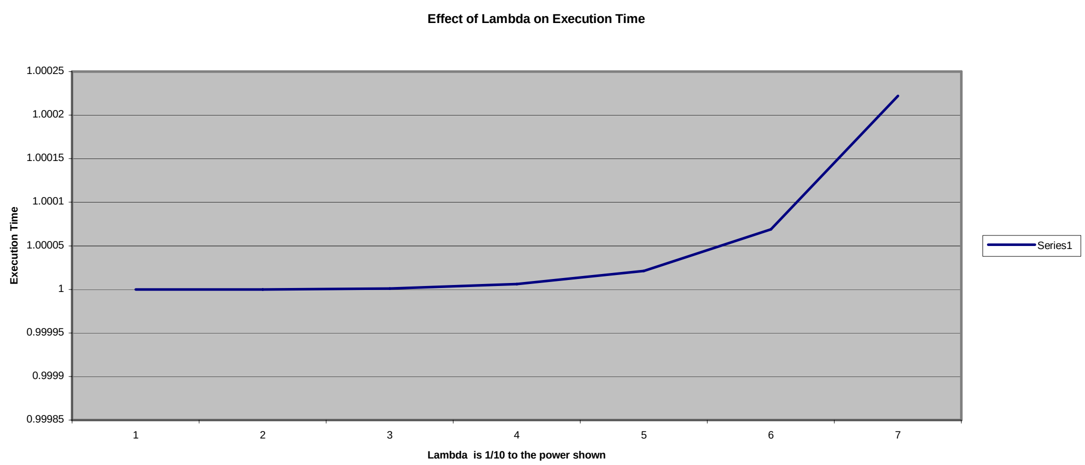
(d) The conclusions one draws from this are the following. μ is the rate at which a faulty piece of software generates errors under testing. So, if μ is small, it takes longer to uncover faults, which is why the probability of being actually bug-free increases more slowly than if μ is large. p is the prior belief in the goodness of the software. If p is large, a small amount of testing without discovering any faults serves to reinforce our belief that the software is bug-free. On the other hand, if p is small, it takes a long testing duration (without any faults being uncovered) for us to reach the same level of confidence that the software is bug-free.

1. In Section 6.3.1, we derived an approximation for the expected time between checkpoints as a function of the checkpoint parameters.

(a) Calculate the optimum number of checkpoints and plot the approximate total expected execution time as a function of T_{ov}. Assume that $T = 1$, $T_{lt} = T_{ov}$ and $\lambda = 10^{-5}$. Vary T_{ov} from 0.01×10^{-5} to 0.2×10^{-5}.



(b) Plot the approximate total expected execution time as a function of λ. Fix $T = 1$, $T_{ov} = 0.1$, and vary λ from 10^{-7} to 10^{-1}.



3. You have a task with execution time, T. You take N checkpoints, equally spaced through the lifetime of that task. The overhead for each checkpoint is T_{ov} and $T_{lt} = T_{ov}$. Given that during execution, the task is affected by a total of k point failures (i.e., failures from which the processor recovers in negligible time), answer the following questions.

(a) What is the maximum execution time of the task?

(b) Find N such that this maximum execution time is minimized. It is fine to get a non-integer answer (say x): in practice, this will mean that you will pick the better of $\lceil x \rceil$ and $\lfloor x \rfloor$.

$T_{wc} = T + NT_{ov} + k(T_{ex} + T_{ov})$

$$T_{wc} = T + NT_{ov} + k\left(\frac{T}{N+1} + T_{ov}\right)$$

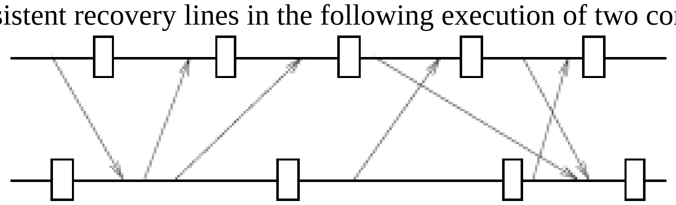
$$\frac{dT_{wc}}{dN} = 0 + T_{ov} + \frac{-kT}{(N+1)^2} \qquad 0 = 0 + T_{ov} + \frac{-kT}{(N+1)^2} \qquad T_{ov} = \frac{kT}{(N+1)^2}$$

$$(N+1)^2 = \frac{kT}{T_{ov}} \qquad N + 1 = \sqrt{\frac{kT}{T_{ov}}} \qquad N = \sqrt{\frac{kT}{T_{ov}}} - 1$$

4 .Solve Equation 6.10 numerically and compare the calculated T^{opt}_{ex} to the value obtained in Equation 6.8 for the simpler model. Assume $T_r = 0$ and $T_{lt} = T_{ov} = 0.1$. Vary λ from 10^{-7} to 10^{-2}.

Lambda	First Order Approximation	Numerical Solution
10-7	1414.214	1414.00002
10-6	447.2136	447.14693
10-5	141.4214	141.35469
10-4	44.72147	44.65472
10-3	14.14249	14.07555
10-2	4.473254	4.40572

7. Identify all the consistent recovery lines in the following execution of two concurrent processes:



Denote the checkpoints of the top process by CP1, CP2, CP3, CP4, CP5 and the the checkpoints of the bottom process by CQ1, CQ2, CQ3, CQ4. Also, label the messages in order by time sent, m0, m1, m2, m3, m4, m5, m6. The consistent recovery lines are: {CP1, CQ1}, {CP1, CQ2}, {CP1, CQ3}, {CP2, CQ2}, {CP2, CQ3}, {CP3, CQ2}, {CP3, CQ3}, {CP4, CQ3} and {CP5, CQ4}. For the checkpoint pairs that do not form a consistent recovery line, the orphaned messages are listed: {CP1, CQ4} – {m4, m6}, {CP2, CQ1} – {m1}, {CP2, CQ4} – {m4, m6}, {CP3, CQ1} – {m1, m2}, {CP3, CQ4} – {m4, m6}, {CP4, CQ1} – {m1, m2, m3}, {CP4, CQ2} – {m3}, {CP4, CQ4} – {m6}, {CP5, CQ1} – {m1, m2, m3, m5}, {CP5, CQ2} – {m3, m5}, {CP5, CQ3} – {m5}.