TECHNISCHE UNIVERSITEIT DELFT
Faculteit Elektrotechniek, Wiskunde en Informatica

IN4390 – Quantitative Evaluation of Embedded Systems

January 24th, 2022, from 09:00 to 12:00

Koen Langendoen, Lydia Chen, George Iosifidis, Marco Zuniga

Question:	1	2	3	4	5	6	7	Total
Points:	13	6	6	20	15	15	15	90
Score:								

- This is a closed book exam
- You may use a **simple** calculator only (i.e. graphical calculators are not permitted)
- Write your answers with a black or blue pen, not with a pencil
- **Always justify your answers, unless stated otherwise**

- The exam covers the following material:
  - (a) the paper "Basic Concepts and Taxonomy of Dependable and Secure Computing" by A. Avizienis ; J.-C. Laprie ; B. Randell ; C. Landwehr
  - (b) chapters 18-20,22-23 (DoE), and 30-33 (Queueing Theory) of the book "The Art of Computer Systems Performance Analysis" by R. Jain
  - (c) the paper "Petri nets: Properties, analysis and applications" by T. Murata
  - (d) chapters 11.2 (DTMC), and 11.3 (CTMC) of the book "Introduction to probability, statistics, and random processes" by H. Pishro-Nik

Operational Laws	
Utilization law	$U = XS$
Little's law	$N = XR$
Forced-flow law	$X_k = V_k X$
Bottleneck law	$U_k = D_k X$
Operational Bounds	
Througput	$X \leq \min\left(\frac{1}{D_{max}}, \frac{N}{D+Z}\right)$
Response time	$R \geq \max\left(D, N \times D_{max} - Z\right)$
Queueing Theory M/M/1	
Utilization	$U = XS = \lambda/\mu = \rho$
Probability of n clients in the system	$P_n = \rho^n(1-\rho)$
Mean #clients in the system	$N = \rho/(1-\rho) = \lambda/(\mu-\lambda)$
Mean #clients in the queue	$N_Q = N - \rho$
Mean response time	$R = N/\lambda = 1/(\mu - \lambda)$
Mean waiting time	$W = R - S = \rho/(\mu-\lambda)$
Basic Math	
Geometric series	$\sum_{k=0}^{\infty} r^k = \frac{1}{1-r}$, for $\lvert r\rvert < 1$

ANOVA Table for One Factor Experiments

Component	Sum of Squares	%Variation	DF	Mean Square	F-Comp.	F-Table
y	SSY=$\sum y_{ij}^2$		ar			
$\bar{y}_{..}$	SS0=$ar\mu^2$		1			
y-$\bar{y}_{..}$	SST=SSY-SS0	100	ar-1			
A	SSA $= r\Sigma\ \alpha_i^2$	$100\left(\frac{\text{SSA}}{\text{SST}}\right)$	a-1	MSA $= \frac{\text{SSA}}{a-1}$	$\frac{\text{MSA}}{\text{MSE}}$	$F_{[1-\alpha; a-1, a(r-1)]}$
e	SSE=SST- SSA	$100\left(\frac{\text{SSE}}{\text{SST}}\right)$	$a(r-1)$	MSE $= \frac{\text{SSE}}{a(r-1)}$		

Washington University in St. Louis CSE567M ©2008 Raj Jain

Question 1 [13 points]

Measuring is at the basis of performance evaluation. Data obtained from experiments allows validating abstract models of the system under study. As not everything is modeled and/or under control, repeated measurements of a single experiment will result in (slightly) different numbers. Averaging these numbers is a crude – but often convenient – way to summarize them.

(a) [2 points] Explain when it is better to use the median than the average (mean).

> **Solution:** The mean is susceptible to outliers, the median is agnostic to this.

(b) [2 points] Mention a downside of using the median.

> **Solution:** You need to remember (and sort) **all** the data (while for the mean you can keep a running sum and count).

(c) [2 points] Mention a data distribution where both the mean and the median do not provide meaningful information. How does the alternative "solution" of reporting the mean and variance do in this case?

> **Solution:** Data with a bi-modal distribution is troublesome, especially when both clusters contain half the data points. In this case the mean will end up in the middle (no mans land), while the median will "randomly" pick a value from either cluster.
>
> When reporting mean (bogus value) and variance (very large) at best one can conclude that the outcome is not to be trusted. So no, this is not a solution. A histogram would be more appropriate for a bi-modal distribution.

When designing an Embedded System, often multiple performance metrics are to be considered. Having multiple objectives complicates the exploration of design alternatives as the different objectives may conflict, that is, objective 1 may be better in configuration A, while configuration B excels in objective 2.

(d) [2 points] Describe what a Pareto front denotes.

> **Solution:** A point on a Pareto front either dominates all other points (that is it is better in at least in one dimension, and equal or better in all other dimensions) or is incompatible (one dimension is better, another is worse).

(e) [5 points] Give an multi-objective example – from the ES domain – that highlights the issue, and report the corresponding Pareto front.

> **Solution:** See question 7 from the 2020-01-30 exam where different Arduino models are to be compared. Having a design space of three points (two on the Pareto front, one being dominated) is enough to illustrate the concept.

Question 2 [6 points]

A two-factor ANOVA table is obtained through a factorial design.

v	DF	SS	MS	F
A	**	150	37.50	15
B	1	100	**	**
Interaction	**	**	**	**
Error	**	50	2.50	
Total	29	380		

(a) 1 point What is the degree of freedom of the error term?

Solution: DF = SS / MS = 50 / 2.50 = 20

(b) 1 point many replicates of experiments were performed?

Solution: $\text{DF}_A = \text{SS}_A$ / $\text{MS}_A = 150$ / $37.50 = 4$, hence, a=5. $\text{DF}_{Err} = ab(r-1) = 5 \times 2(r-1) = 20$, hence, r-1 = 20/10 = 2. Thus the number of replicates equals 3.

(c) 2 points What are the F statistics for factor B, and factor AB (interaction)?

Solution:

- $F_B = MS_B/SS_E = (SS_B/DF_B)/SS_E = (100/1)/2.50 = 40$
- $SS_{AB} = SS_T - SS_A - SS_B - SS_E = 380 - 150 - 100 - 50 = 80$
  $F_{AB} = MS_{AB}/SS_E = (SS_{AB}/DF_{AB})/SS_E = (80/4)/2.50 = 8$

(d) 2 points Which factors are significant given the following Table of significance level of 5%.

	$F_{1,5}$	$F_{2,5}$	$F_{3,5}$	$F_{4,5}$	$F_{5,5}$	$F_{1,10}$	$F_{2,10}$	$F_{3,10}$
Values	6.61	5.78	5.41	5.19	5.05	4.96	4.10	3.71

	$F_{4,10}$	$F_{5,10}$	$F_{1,20}$	$F_{2,20}$	$F_{3,20}$	$F_{4,20}$	$F_{5,20}$	
Values	3.47	3.32	4.35	3.49	3.09	2.86	2.71	

Solution: All are significant:
$F_A = 15 > F4, 20 = 2.86$, Factor A is significant
$F_B = 40 > F1, 20 = 4.35$, Factor B is significant
$F_{AB} = 8 > F4, 20 = 2.86$, Factor AB is significant

Question 3 [6 points]

An 8-run experiment was performed to figure out the important system parameters to run an image classifier on mobile phones. Based on the following sign table, answer the questions below:

Run	CPU speed [GHZ] (A)	Memory size[GB] (B)	Classier sizes [MB] (C)	Memory size [KB](D)
1	2.1	1	30	10
2	4.2	1	30	50
3	2.1	2	30	50
4	4.2	2	30	10
5	2.1	1	60	50
6	4.2	1	60	10
7	2.1	2	60	10
8	4.2	2	60	50

(a) 1 point What kind of design is this?

Solution: 2^{4-1}, with resolution IV

(b) 2 points What is the generator function?

Solution: I=ABCD

(c) 1 point Which factor is confounded with factor A, B, C, D respectively?

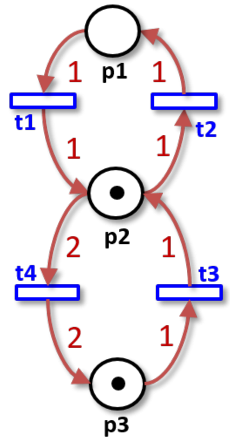
Solution: A is confounded with BCD, B is confounded with ACD, C is confounded with ABD, and D is confounded with ABC.

(d) 2 points If the generator function I=ABD is used, which other factor will be founded with factor A, B, C, and D, respectively? Is this a better solution?

Solution: A is confounded with BD, B is confounded with AD, C is confounded with ABD, and D is confounded with AB. It is a worse design because factor A, B, D are confounded with 2 factor interactions here, instead of 3 factor interactions in the case of I=ABCD.

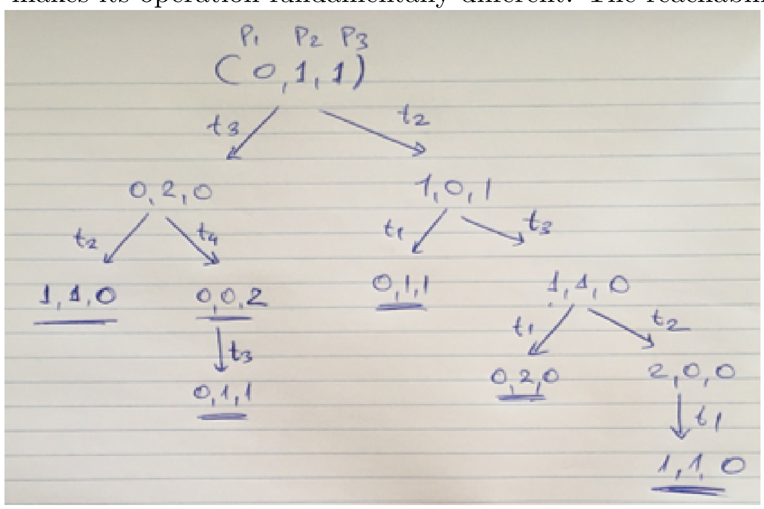
Question 4 [20 points]

(a) Consider the following Petri Net:



i. 4 points Draw the reachability or the coverability graph of this Petri Net.

Solution: This is very similar to the PN we had in the first exam, but with a tweaked number that makes its operation fundamentally different. The reachability graph is:



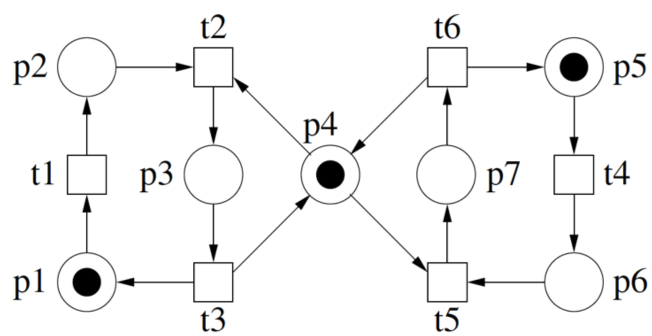
ii. 2 points Is this Petri Net k-bounded for some positive integer k? If yes, what is the minimum value of k? If no, explain your reasoning.

Solution: The petri net is 2-bounded (there is no action producing more tokens than it consumes).

(b) i. $\boxed{\text{4 points}}$ Draw the Petri Net that is associated with the following incidence matrix:

$$A = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & -1 & 1 \\ 0 & 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

Solution:



ii. $\boxed{\text{2 points}}$ If the initial marking m_0 is as follows:

$$m_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

and **t1** (corresponding to first column of A) fires, calculate analytically, i.e., using algebraic equations, what will be the obtained marking m_1. Please include the calculations in your answer.

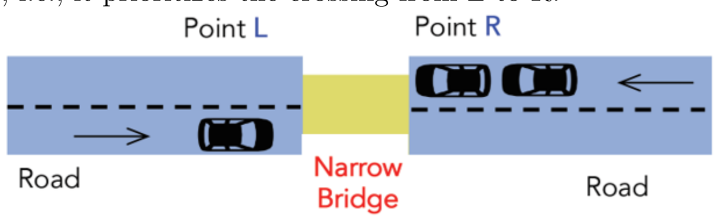
Solution:

$$m_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

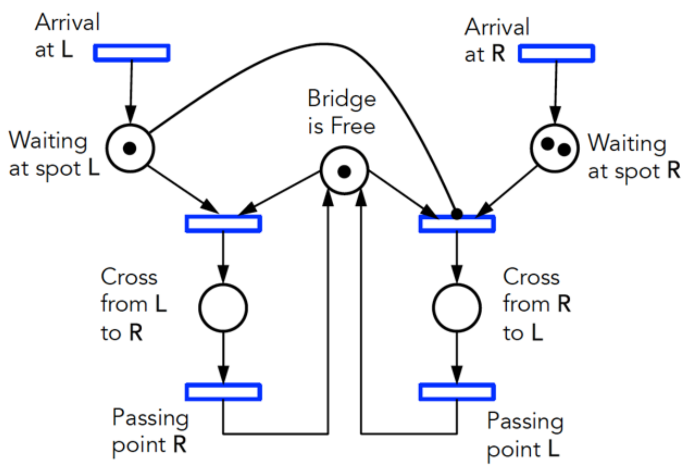
which follows from the matrix multiplication and addition, i.e.,:

$$\begin{bmatrix}1\\0\\0\\1\\1\\0\\0\end{bmatrix} + \begin{bmatrix}-1 & 0 & 1 & 0 & 0 & 0\\ 1 & -1 & 0 & 0 & 0 & 0\\ 0 & 1 & -1 & 0 & 0 & 0\\ 0 & -1 & 1 & 0 & -1 & 1\\ 0 & 0 & 0 & -1 & 0 & 1\\ 0 & 0 & 0 & 1 & -1 & 0\\ 0 & 0 & 0 & 0 & 1 & -1\end{bmatrix} \cdot \begin{bmatrix}1\\0\\0\\0\\0\\0\end{bmatrix} = m_1$$

(c) $\boxed{\text{8 points}}$ Consider the two-way road in the figure below, where the cars need to cross a narrow bridge that has only one lane. This means that, at each time, only one car can cross the bridge between points L and R, and the other cars have to wait at the respective sides (Point L and Point R). Please draw a Petri Net that models this road system and gives priority to the cars waiting at Point L, i.e., it prioritizes the crossing from L to R.



Solution:



Note: there are other possible solutions.

Question 5 [15 points]

A computer card game has been designed and you are asked to prove the correctness of the algorithm used to shuffle a deck of n cards, where $n = 52$. The algorithm works as follows. Starting with any initial ordering of the cards, one of the numbers $(1, 2, ..., 52)$ is chosen at random and with equal probability. If number i is chosen, we move the card from position i in the deck to the top, i.e. to position 1. We repeatedly perform the same operation.

In the sequel below you will show that, in the limit, the deck is perfectly shuffled in the sense that the resultant ordering is equally likely to be any of the $n!$ possible orderings.

(a) $\boxed{\text{1 point}}$ To model this problem as a Discrete-Time Markov Chain, how would you define the random process X_n and the states that X_n can take? How many states do you have?

Solution: Let's encode all possible card orderings as states; we have $n!$ of those. X_n then denotes the ordering (sequence of cards in the deck) after the nth shuffle.

(b) 1 point Do all states have the same number of incoming and outgoing links? Explain your reasoning.

Solution: The picking of a random card is agnostic to the actual ordering of the deck, so in that sense all states are alike.

Every state has n outgoing arcs pointing towards the ordering that results from moving card i to the top of the deck. (Picking card 1 will result in a self loop). Conversely all states have n incoming arcs from the n states that had the top card at place i and matching cards at the other positions before the shuffle. In this case, $n = 52$.

(c) 3 points Is the resulting Markov Chain irreducible and aperiodic? Explain your reasoning.

Solution: It is irreducible as we can use a "sorting" algorithm to go from one state to any other; build the destination sequence by picking the elements from the back to the front.

It is a periodic as every state has a self-loop, which eliminates the presence of periods.

(d) 7 points Prove that the shuffling algorithm is correct by showing that the limiting distribution of the DTMC leads to the uniform distribution (i.e. $1/n!$).
Hint: start your proof using the equation: $\pi P = \pi$ *and show that* $\pi_i = 1/n!$ *for all* i.

Solution: Since the chain is irreducible and aperiodic, there is a *unique* stationary distribution, which is also the limiting distribution. Thus the limiting distribution π that satisfies $\pi P = \pi$ is the only solution to our problem.

The next step is to show that, assuming $\pi_i = 1/n!$, the incoming and outgoing flow of the next step are equal (i.e. stationary). Every row and column of matrix P contains n non-zero entries with value $1/n$ as picking the next card to go on top is random. Assuming that all states have equal probability, then $\pi_i = 1/n!$ as there are $n!$ states. The mass across each arc is then $1/n \times 1/n!$. Thus, the mass flowing into state i equals $n \times (1/n \times 1/n!) = 1/n! = \pi_i$, hence, we are at a fixed point (steady state distribution) with respect to matrix P.

(e) 3 points Considering a deck of 3 cards, with cards A,B,C, and assuming that X_0 starts with the ordering (A,B,C), how many steps are required to have a non-zero probability of reaching all other possible orderings (i.e. states)? Explain your reasoning.

Solution:

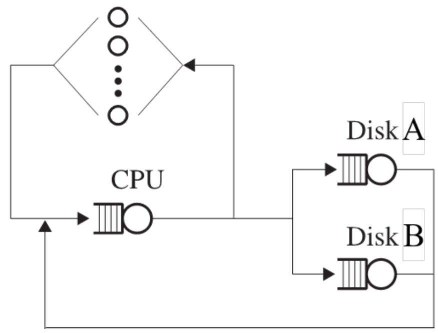
With 3 cards we have $3! = 6$ states. The transition probability matrix is:

	ABC	ACB	BAC	BCA	CAB	CBA
ABC	1/3		1/3		1/3	
ACB		1/3	1/3		1/3	
BAC	1/3		1/3			1/3
BCA	1/3			1/3		1/3
CAB		1/3		1/3	1/3	
CBA		1/3		1/3		1/3

Starting with ABC, one can cover all states with a non-zero probability in just 2 steps: $[1, 0, 0, 0, 0, 0]P^2$

Question 6 [15 points]

Consider a timesharing system with two disks (A and B) configured as follows:



Through an hour-long measurement, the probabilities for jobs completing the service at the CPU were found to be 0.75 to disk A, 0.15 to disk B, and 0.1 to the terminals. The user think time was clocked to be 5 seconds, the disk service times are 30 milliseconds and 25 milliseconds, while the average service time per visit to the CPU was 40 milliseconds.

(a) 4 points For each job, what are the visit ratios for the CPU, disk A, and disk B?
Hint: Argue that $V_{CPU} = 1 + V_A + V_B$.

Solution: The number of visits to the CPU is one (upon dispatch) plus the number of returns through disks A and B, hence, $V_{CPU} = 1 + V_A + V_B$. Factoring in that a job can only once exit the subsystem and has a 10% chance of doing so, we find $V_{CPU} = 10$. The 9 remaining leaves from the CPU are distributed in a 0.75:0.15 = 5:1 ratio over disks A and B, so $V_A = 7.5$ and $V_B = 1.5$

(b) 3 points For each device, what is the (total) service demand?

Solution:

- $D_{CPU} = 10 * 40\, ms = 400\, ms$
- $D_A = 7.5 * 30\, ms = 225\, ms$
- $D_B = 1.5 * 25\, ms = 37.5\, ms$

(c) 3 points If disk A's utilization is 50%, what is the utilization of the CPU and disk B?

Solution:

- $U_A = XD_A$, hence, $X = U_A/D_A = 0.5/0.225 = 20/9$
- $U_{CPU} = XD_{CPU} = 20/9 \times 0.400 = 8/9$
- $U_B = XD_B = 20/9 \times 0.0375 = 1/12$

(d) $\boxed{\text{2 points}}$ What is the average response time when there are 20 users on the system?

Solution: $R = N/X - Z = 20/(20/9) - 5 = 9 - 5 = 4$ seconds

(e) $\boxed{\text{3 points}}$ What is the best possible response time if one may upgrade one component (CPU or disk)?

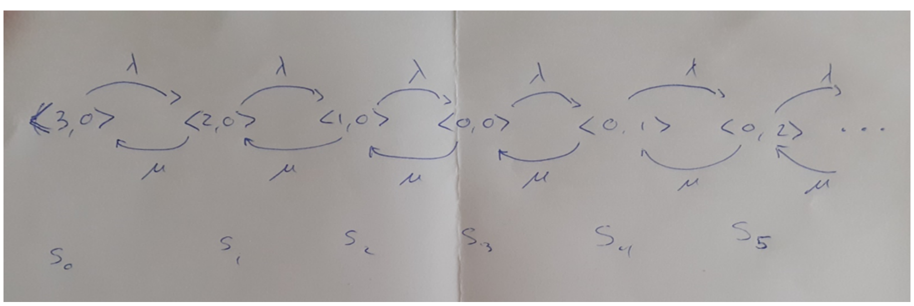
Solution: The response time is bounded: $R \geq \max(D, N \times D_{max} - Z)$. The CPU is the bottleneck and after upgrading it (setting its processing time to zero) disk A (with a utilization of 50%) will become the bottleneck. Then $D_{max} = 225\,\text{ms}$, and the total demand $D = D_A + D_B = 225 + 37.5 = 262.5\,\text{ms}$. This leads to $R > max(262.5, 20 \times 225 - 5000 = 4500 - 5000 = -500) = 262.5\,\text{ms}$.

Question 7 [15 points]

At a large hotel, taxi cabs arrive at a rate of 15 per hour, and passengers arrive at the rate of 12 per hour. Whenever taxi cabs are waiting, passengers are served immediately upon arrival (one per taxi). Whenever passengers are waiting, taxi cabs are loaded immediately upon arrival. A maximum of three cabs can wait at a time (other cabs must go elsewhere).

(a) $\boxed{\text{5 points}}$ Find an appropriate way to model this queueing system as an M/M/1 queue; clearly state what information is encoded in a state, and draw the corresponding state transition diagram.

Solution: Let's start by modeling the system state as a tuple capturing the <#cabs, #passengers>. For convenience let passengers arrive with rate λ (12/hr) and cabs arrive with rate μ (15/hr), leading to a load $\rho = 12/15 = 4/5$. The sate transition diagram then looks as:



Now we need to (re)number to a normal M/M/1 queue. State S_n with $n = 3 - \#cabs + \#passengers$ denotes the number of clients in the queue+server.

(b) $\boxed{\text{3 points}}$ Calculate (i) the expected number of cabs waiting and (ii) the expected number of passengers waiting.

Solution:

- The number of cabs equals $3P_0 + 2P_1 + 1P_2 = (3 + 2\rho + \rho^2)(1-\rho) = (3 + 8/5 + 16/25)(1/5) = (75 + 40 + 16)/125 = 131/125 = 1.048$.
- The number of clients in an M/M/1 queue equals $N = \rho/(1-\rho) = 4$. Using the state equation we find that $N = E[n] = E[3 - \#cabs + \#passengers] = E[3] -$

$E[\#cabs] + E[\#passengers]$, which yields $E[\#passengers] = N - E[3] + E[\#cabs] = 4 - 3 + 131/125 = 256/125 = 2.048$.

(c) $\boxed{\text{4 points}}$ Calculate the average waiting times of (i) cabs and (ii) passengers.

Solution: This is a little tricky. Little's law states that $R = N/X$. Response time (R) includes waiting in the queue (W) + processing by the server (S). In this case, however, the server is virtual: the taxi at the head of the queue has to wait for a passenger (server) showing up. Thus an arriving taxi has first to wait for any taxis in front (W) and then for a passenger to show up (S), so the total time spent waiting equals W+S = R. A similar reasoning applies for passengers.

- For the taxis the throughput X is the rate at which *passengers* arrive, so their waiting time (R_{taxi}) equals $N_{taxi}/\lambda = (131/125)/12 = 131/1500 = 0.087333\ hr = 5.24\ min$.
- For the passengers the throughput X is the rate at which *taxis* arrive, so their waiting time ($R_{passenger}$) equals $N_{passenger}/\mu = (256/125)/15 = 256/1875 = 0.1365333\ hr = 8.192\ min$.

(d) $\boxed{\text{3 points}}$ What would be the impact of allowing four cabs to wait at a time? Compute the average number of taxis and passengers waiting in this case.

Solution: The state encoding becomes S_n with $n = 4 - \#cabs + \#passengers$

- The number of cabs equals $4P_0 + 3P_1 + 2P_2 + P_3 = (4 + 3\rho + 2\rho^2 + \rho^3)(1 - \rho) = (4 + 12/5 + 32/25 + 64/125)(1/5) = (500 + 300 + 160 + 64)/625 = 1024/625 = 1.6384$.
- The number of passengers equals $N - E[4] + E[\#cabs] = 4 - 4 + 1024/625 = 1024/625 = 1.6384$.