A game theoretic model of the behavioural gaming that takes place at the EMS - ED interface

1 Abstract

Emergency departments (EDs) in hospitals are usually under pressure to achieve a target amount of time that describes the arrival of patients and the time it takes to receive treatment. For example in the UK this is often set as 95% of patients to be treated within 4 hours. There is empirical evidence to suggest that imposing targets in the ED results in gaming at the interface of care between the EMS and ED. If the ED is busy and a patient is stable in the ambulance, there is little incentive for the ED to accept the patient whereby the clock will start ticking on the 4 hour target. This in turn impacts on the ability of the EMS to respond to emergency calls.

This study explores the impact that this effect may have on an ambulance's utilisation and their ability to respond to emergency calls. More specifically multiple scenarios are examined where an ambulance service needs to distribute patients between neighbouring hospitals. The interaction between the hospitals and the ambulance service is defined in a game theoretic framework where the ambulance service has to decide how many patients to distribute to each hospital in order to minimise the occurrence of this effect. The methodology involves the use of a queueing model for each hospital that is used to inform the decision process of the ambulance service so as to create a game for which the Nash Equilibria can be calculated.

Contents

2 Introduction

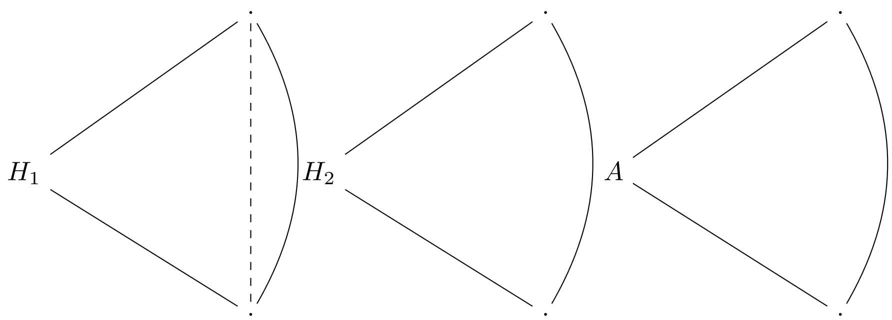


Figure 1: Ambulance Decision Problem

States:

1. A = Ambulance
2. H_i = Hospital i

Notation:

- Λ = total number of patients that need to be hospitalised
- p_i = proportion of patients going to Hospital i ($p_i\Lambda$ = number of patients going to hospital i)
- d_i = distance from Hospital i
- $\hat{c}_i$ = capacity of hospital i
- $W(c, \lambda\, \mu)$ = waiting time in the system function
- μ_i = service rate of hospital i
- λ_i^o = arrival rate of other patients to the hospital (not by ambulance)
- $C_i(p_i) = d_i + W(c = \hat{c}_i,\ \lambda = p_i\Lambda + \lambda_i^o,\ \mu = \mu_i)$

3 Game Theory component:

Players:

- Ambulance
- Hospital A
- Hospital B

Strategies of players:

- Hospital i:
    1. Close doors at $\hat{c}_i = 1$
    2. Close doors at $\hat{c}_i = 2$
    3. ...
    4. Close doors at $\hat{c}_i = C_i$
- Ambulance:
    1. Choose $p_1 \in [0, 1]$

Cost Functions: Waiting times + the distance to each hospital.

4 Quick Methodology

- Fix the parameters Λ, λ_i^o, μ_i and C_i.
- $\forall \ \hat{c_i} \in \{1, 2, \ldots, C_A\}$ and $\forall \ \hat{c_j} \in \{1, 2, \ldots, C_B\}$
- Calculate p_A and $p_B = 1 - p_A$ s.t. $(W_q)_A = (W_q)_B$.
- Calculate the probability $P((W_q)_i \leq 4$ hours
- Fill matrix A with $U^A_{\hat{c_i},\hat{c_j}} = 1 - |0.95 - P((W_q)_A \leq 4)|$ and
- fill matrix B with $U^B_{\hat{c_i},\hat{c_j}} = 1 - |0.95 - P((W_q)_B \leq 4)|$

A =

$U^A_{1,1}$	$U^A_{1,2}$	$\ldots$	U^A_{1,C_B}
$U^A_{2,1}$	$U^A_{2,2}$	$\ldots$	U^A_{2,C_B}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$U^A_{C_A,1}$	$U^A_{C_A,2}$	$\ldots$	$U^A_{C_A,C_B}$

B =

$U^B_{1,1}$	$U^B_{1,2}$	$\ldots$	U^B_{1,C_B}
$U^B_{2,1}$	$U^B_{2,2}$	$\ldots$	U^B_{2,C_B}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$U^B_{C_A,1}$	$U^B_{C_A,2}$	$\ldots$	$U^B_{C_A,C_B}$

- Ambulance decides the proportion of people to distribute to each hospital based on optimal patient distribution.

5 Proper Methodology

The problem is formulated as a normal form game where the players are the two hospitals. Each hospital is given N_A and N_B number of strategies where N_A and N_B are the total capacities of the hospitals. In other words, depending on the capacity of each hospital, they may choose to stop receiving patients from arriving ambulances whenever they reach a certain capacity threshold. The goal of this problem is to satisfy the ED regulations which state that 95% of the patients should see a specialist within 4 hours of their arrival to the hospital. The mean of the random variable W_q is the average waiting time in the queue for hospital i.

$$W_q(\lambda_i, \mu_i, \hat{c_i}) = \frac{1}{\hat{c_i}\mu_i} \frac{(\hat{c_i}\rho_i)^{\hat{c_i}}}{\hat{c_i}!(1-\rho_i)^2} P_0, \quad i \in \{A, B\} \tag{1}$$

Thus, the utilities of the two players should be the proportion of people that fall within the 4 hours target. This is also equivalent to the probability of the waiting time of an individual to be less than or equal to 4 hours.

$$P(W_q(\lambda_i, \mu_i, \hat{c_i}) \leq 4), \quad i \in \{A, B\} \tag{2}$$

Therefore, a sensible goal for each player should be to minimise that probability, but the actual target of the hospitals is to satisfy 95% of those patients within the 4-hour time limit. Therefore, the goal should be to get that probability as close to 0.95 as possible. Thus each player should aim to minimise:

$$|0.95 - P(W_q(\lambda_i, \mu_i, \hat{c_i}) \leq 4)|, \quad i \in \{A, B\} \tag{3}$$

The classic formulation of a normal form game looks into the maximisation of each player's payoff. Consequently the utilities can be altered such that the goal of each player is to maximise:

$$U^A_{\hat{c_A}, \hat{c_B}} = 1 - |0.95 - P(W_q(\lambda_A, \mu_A, \hat{c_A}) \leq 4)| \tag{4}$$

$$U^B_{\hat{c_A}, \hat{c_B}} = 1 - |0.95 - P(W_q(\lambda_B, \mu_B, \hat{c_B}) \leq 4)| \tag{5}$$

Finally, the problem can be expressed as a normal form game with two players where each player/hospital has N_A and N_B strategies respectively. The two $N_A \times N_B$ payoff matrices for the utilities of the two hospitals can be defined as:

$$A = \begin{array}{|c|c|c|c|} \hline U^A_{1,1} & U^A_{1,2} & \dots & U^A_{1,C_2} \\ \hline U^A_{2,1} & U^A_{2,2} & \dots & U^A_{2,C_2} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline U^A_{C_1,1} & U^A_{C_1,2} & \dots & U^A_{C_1,C_2} \\ \hline \end{array} \qquad B = \begin{array}{|c|c|c|c|} \hline U^B_{1,1} & U^B_{1,2} & \dots & U^B_{1,C_2} \\ \hline U^B_{2,1} & U^B_{2,2} & \dots & U^B_{2,C_2} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline U^B_{C_1,1} & U^B_{C_1,2} & \dots & U^B_{C_1,C_2} \\ \hline \end{array}$$

Once the certain strategies of the game have been selected the ambulance service can decide what would be the optimal way to distribute patients. However, the way the ambulance service distributes patients can affect the utilities of the game. So how would one solve this kind of problem?

5.1 Solution

As mentioned before the problem requires the construction of two queuing models that will be needed for the formulation of the normal form game. Based on those utilities the ambulance service will then decide the percentage of patients that will distribute to each hospital.

First and foremost, the model consists of several parameters that are unknown and are assumed to be fixed. The model will be run multiple times for various values of these parameters.

Λ_2	Number of patients that need to be distributed
$\lambda_{1,i}$	Arrival rate of other patients that enter hospital i
μ_i	Service rate of hospital i
N_i	Total capacity of hospital i

Table 1: Fixed Parameters

Having established the fixed parameters of the model, the hospitals' utilities need to be calculated. In order to do so a backwards induction approach will be used. The EMS aims to distribute the patients such that the mean waiting time of patients is minimal. This can be further interpreted as when the mean waiting time of hospital A equals the mean waiting time of hospital B. Thus, the minimal mean waiting time can be found for the values of p_A and p_B that solve the following equation:

$$W_q(\lambda_A, \mu_A, \hat{c_A}) = W_q(\lambda_B, \mu_B, \hat{c_B}) \tag{6}$$

Equation (6) needs to be solved for all values of $c_i \in \{1, 2, \ldots C_A\}$ and $c_j \in \{1, 2, \ldots C_B\}$. Then, for every c_i and c_j the utility equation (4) has to be calculated for both hospitals. In order to solve it though, one must first estimate the probability $P[(W_q)_{\{A,B\}}] \leq 4]$. That is the probability that the waiting time in the queue for one of the hospitals is less than 4 hours. For a multi-server system, the distribution of the waiting time can be given by equation 7. The above expression returns the probability that the waiting time in the queue is less than some time T.

$$P(W_q > T) = \frac{(\frac{\lambda}{\mu})^c P_0}{c!(1 - \frac{\lambda}{c\mu})}(e^{-(c\mu - \lambda)T}) \tag{7}$$

Consequently when incorporating equation (7) into (4) a newer utility equation can be acquired:

$$U^{\{A,B\}}_{\hat{c_i}, \hat{c_j}} = 1 - \left| \left[\frac{(\frac{\lambda}{\mu})^c P_0}{c!(1 - \frac{\lambda}{c\mu})} \left(e^{-(c\mu - \lambda)T}\right) \right] - 0.05 \right| \tag{8}$$

A =

$U^A_{1,1}$	$U^A_{1,2}$	$\ldots$	U^A_{1,C_2}
$U^A_{2,1}$	$U^A_{2,2}$	$\ldots$	U^A_{2,C_2}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$U^A_{C_1,1}$	$U^A_{C_1,2}$	$\ldots$	$U^A_{C_1,C_2}$

B =

$U^B_{1,1}$	$U^B_{1,2}$	$\ldots$	U^B_{1,C_2}
$U^B_{2,1}$	$U^B_{2,2}$	$\ldots$	U^B_{2,C_2}
$\vdots$	$\vdots$	$\ddots$	$\vdots$
$U^B_{C_1,1}$	$U^B_{C_1,2}$	$\ldots$	$U^B_{C_1,C_2}$

6 Markov chain model

The following Markov chain represents the transition between states of a service centre while capturing the interactions between it and a buffer centre. The service centre accepts two types of individuals; Class 1 and Class 2. Class 2 individuals are accepted until a pre-determined threshold T of individuals is reached. When reached, all Class 2 individuals that arrive will remain *"blocked"* in the buffer centre until the number of people in the system is reduced below T. Additionally, if the people in the service centre keep rising, they may exceed the number of servers C available, which will in turn mean that every new person will have to wait for a server to become free. The states of the Markov chain are denoted by (u, v) where:

- u = number of Class 2 individuals blocked
- v = number of Class 1 individuals in the service centre

6.1 Markov-chain state mapping function

The transition matrix of the Markov-chain representation described above can be denoted by a state mapping function. The state space of this function is defined as:

$$\begin{aligned} S(T) =& S_1(T) \cup S_2(T) \text{ where:} \\ S_1(T) =& \left\{(0, v) \in \mathbb{N}_0^2 \mid v < T\right\} \\ S_2(T) =& \{(u, v) \in \mathbb{N}_0^2 \mid v \geq T\} \end{aligned} \tag{9}$$

Therefore, the entries of the transition matrix Q, can be given by $q_{i,j} = q_{(u_i,v_i),(u_j,v_j)}$ which is the transition rate from state $i = (u_i, v_i)$ to state $j = (u_j, v_j)$ for all $(u_i, v_i), (u_j, v_j) \in S$.

$$q_{i,j} = \begin{cases} \Lambda, & \textbf{if } (u_i, v_i) - (u_j, v_j) = (0, -1) \textbf{ and } v_i < \text{t} \\ \lambda_1, & \textbf{if } (u_i, v_i) - (u_j, v_j) = (0, -1) \textbf{ and } v_i \geq \text{t} \\ \lambda_2, & \textbf{if } (u_i, v_i) - (u_j, v_j) = (-1, 0) \\ v_i\mu, & \textbf{if } (u_i, v_i) - (u_j, v_j) = (0, 1) \textbf{ and } v_i \leq C \textbf{ or} \\ & \quad (u_i, v_i) - (u_j, v_j) = (1, 0) \textbf{ and } v_i = T \leq C \\ C\mu, & \textbf{if } (u_i, v_i) - (u_j, v_j) = (0, 1) \textbf{ and } v_i > C \textbf{ or} \\ & \quad (u_i, v_i) - (u_j, v_j) = (1, 0) \textbf{ and } v_i = T > C \\ -\sum_{j=1}^{|Q|} q_{i,j} & \textbf{if } i = j \\ 0, & \textbf{otherwise} \end{cases} \tag{10}$$

In order to acquire an exact solution of the problem a slight adjustment needs to be considered. The problem defined above assumes no upper boundary to the number of individuals that can wait for service or the ones that are blocked in the buffer centre. Therefore, a different state space $\tilde{S}$ needs to be constructed where $\tilde{S} \subseteq S$ and there is a maximum allowed number of people N that can be in the system and a maximum allowed number of people M that can be blocked in the buffer centre:

$$\tilde{S} = \{(u, v) \in S \mid u \leq M, v \leq N\} \tag{11}$$

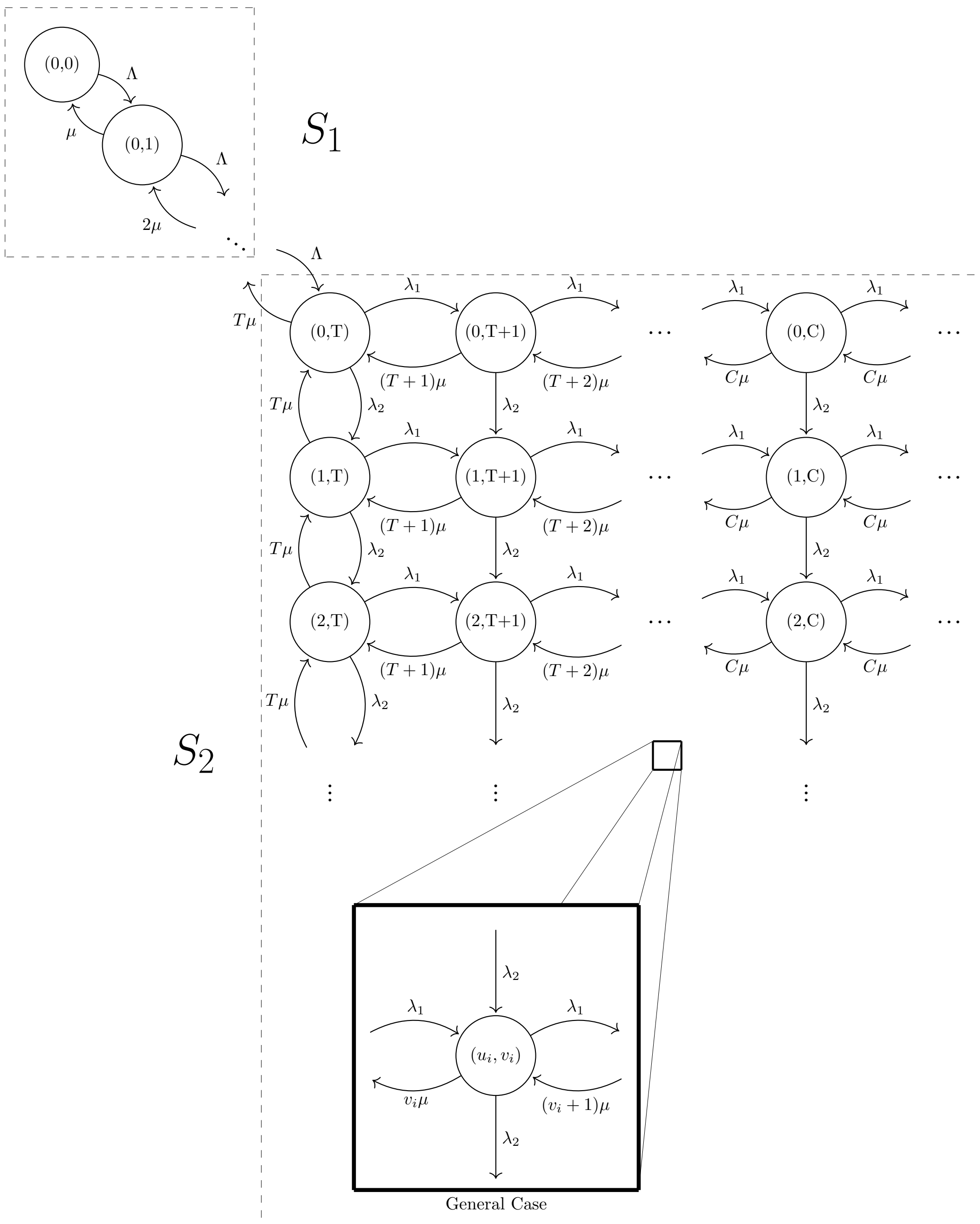


Figure 2: Markov chain

6.2 Steady State

Having calculated the transition matrix Q for a given set of parameters the probability vector π needs to be considered. The vector π is commonly used to study such stochastic systems and it's main purpose is to keep track of the probability of being at any given state of the system. The term *steady state* refers to the instance of the vector π where the probabilities of being at any state become stable over time. Thus, by considering the steady state vector π the relationship between it and Q is given by:

$$\frac{d\pi}{dt} = \pi Q = 0$$

There are numerous methods that can be used to solve problems of such kind. In this paper only numeric and algebraic approaches will be considered.

6.2.1 Numeric integration

The first approach to be considered is to solve the differential equation numerically by observing the behaviour of the model over time. The solution is obtained via python's SciPy library. The functions odeint and solve_ivp have been used in order to find a solution to the problem. Both of these functions can be used to solve any system of first order ODEs.

6.2.2 Linear algebraic approach

Another approach to be considered is the linear algebraic method. The steady state vector can be found algebraically by satisfying the following set of equations:

$$\pi Q = 0$$

$$\sum_i \pi_i = 1$$

These equations can be solved by slightly altering Q such that the final column is replaced by a vector of ones. Thus, the resultant solution occurs from solving the equation $\tilde{Q}^T \pi = b$ where $\tilde{Q}$ and b are defined as:

$$\tilde{q_{i,j}} = \begin{cases} 1, & \textbf{if } j = |Q| \\ q_{i,j}, & \textbf{otherwise} \end{cases}$$

$$b = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{bmatrix}$$

6.2.3 Least Squares approach

Finally, the last approach to be considered is the least squares method. This approach is considered because while the problem becomes more complex (in terms of input parameters) the computational time required to solve it increases exponentially. Thus, one may obtain the steady state vector π by solving the following equation.

$$\pi = \mathrm{argmin}_{x \in \mathbb{R}^d} \|Mx - b\|_2^2$$

6.3 Graph Theoretical Approach

This section aims to investigate the connection between Markov chains and graph theory by exploring alternative ways of calculating the steady state probabilities and considering the problem from different perspectives.

6.3.1 Parameters

The parameters considered as inputs are:

- the number of servers C,
- the threshold T,
- the capacity of the service centre N,
- the capacity of the buffer centre M.

Additional parameters of the model are the class 1 individuals arrival rate, the class 2 individuals arrival rate and the service rate $(\lambda_2, \lambda_1, \mu)$. More specifically, the way these parameters are translated into the model are:

- **Number of servers (C):** Affects the weight of all edges $(v_i, v_j) \in E$ in the Markov chain that correspond to a service rate. These edges have a weight of:

$$w_{(v_i,v_j)} = q_{v_i,v_j}$$

  where $q_{i,j}$ is defined in equation 10. Thus, the coefficients of the service rate have a lower bound of 1 and an upper bound of C.

- **Threshold (T):** Determines the length of the left *arm* of the model. In essence the threshold acts as a breakpoint between states where $u = 0$ and states where $0 \leq u \leq M$. Increasing T results in having more set of states where u can only be 0.
- **Service centre capacity (N):** Is the upper bound of v for all states (u, v).
- **Buffer centre capacity (M):** Is the upper bound of u for all states (u, v) such that $v \geq T$.

6.3.2 Example figure of Markov Model

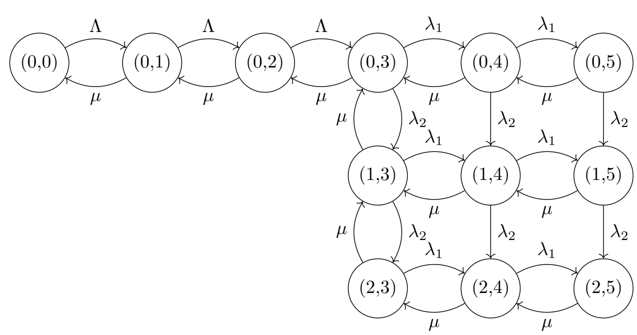


Figure 3: $C = 1, T = 3, N = 5, M = 2$

In figure 3 an example of such a Markov model is shown where $C = 1$, $T = 3$ which means that the *left arm* of the model has a length of 3, $N = 5$ that indicates that the right-most states (u, v) are of the form $(u, 5)$ and $M = 2$ that equivalently shows that the bottom states are of the form $(2, v)$.

6.3.3 A graph theoretic model underling the Markov chain

An additional approach that one may consider to get the state probabilities is the graph theoretical approach for state probabilities. Thus, it can be assumed that a Markov chain model M can be translated as a weighted directed graph $G_M = (V, E)$ where $V = S$ from equation 9 and $(v_i, v_j) \in E$ if and only if $q_{v_i,v_j} > 0$. Furthermore, the weights are given by:

$$w(v_i, v_j) = q_{v_i,v_j}$$

A *directed spanning tree* of a directed graph is defined as a subset of the graph that visits all the vertices of the graph and does not include any cycles. Unlike undirected spanning trees, directed ones also have a root which means that a directed spanning tree that is rooted at a vertex v has to have a path from any other vertex to vertex v. For example, consider the graph shown in figure 4. The graph points out a spanning tree that is rooted at vertex 3.

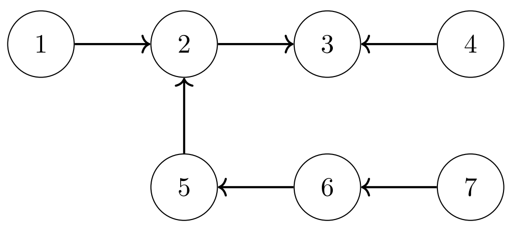


Figure 4: Spanning tree of a graph rooted at vertex 3

Additionally, let us denote the set of all spanning trees of G as $T(G)$ and the subset of $T(G)$ that includes only the spanning trees that are rooted at vertex v as $T_v(G)$. The weight of a spanning tree t can be defined as the product of the weights of the edges it contains:

$$w(t) = \prod_{e \in t} w(e)$$

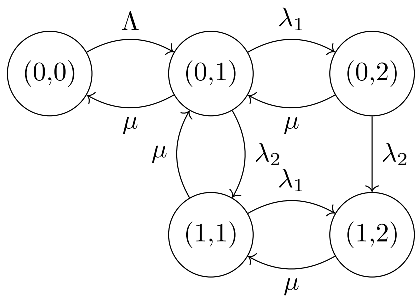
Theorem: Markov chain tree theorem [2]
Let M be an irreducible Markov chain on n states with stationary distribution $\pi_1, \pi_2, \ldots, \pi_n$. Let G_M be the directed graph associated with M. Then the probability of being at state u is given by:

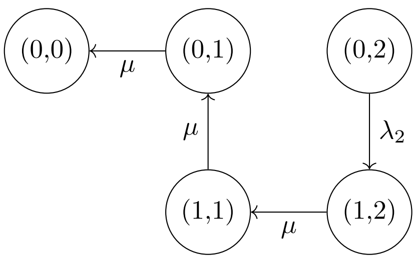
$$\pi_i = \frac{\sum_{t \in T_i(G_M)} w(t)}{\sum_{t \in T(G_M)} w(t)} \tag{12}$$

Equation 12 states that the probability of being at state u can be found by dividing the sum of the weights of all trees in $T_u(G)$ by the sum of the weights of all tress in $T(G)$. Let us ignore the denominator of that fraction for now and focus only on the numerator denoted as $\tilde{\pi}_i = \sum_{t \in T_i(G_M)} w(t)$

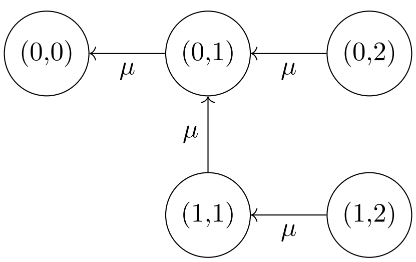
6.3.4 Spanning Trees rooted at $(0,0)$

Let us now consider some examples of spanning trees that are rooted at $(0,0)$. For each of the following examples the complete graph G is shown, then all possible trees of $T_{(0,0)}(G)$ along with the weight associated with each spanning tree. As well as this, the sum of all the weights of the spanning trees denoted by $\tilde{\pi}_{(0,0)}$ is also included.



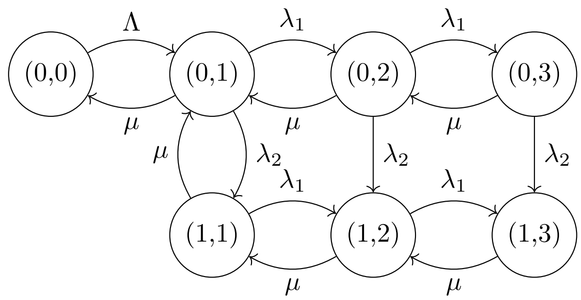


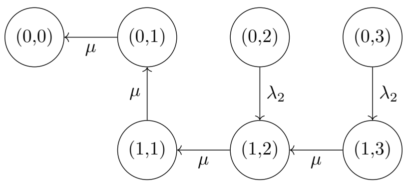
$\longrightarrow \quad \lambda_2\mu^3$



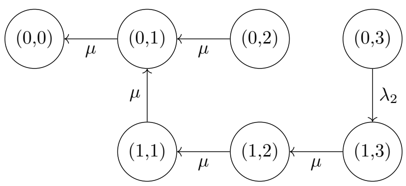
$\longrightarrow \quad \mu^4$

$$\tilde{\pi}_{(0,0)} = \mu^4 + \lambda_2\mu^3$$

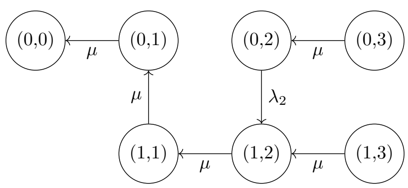




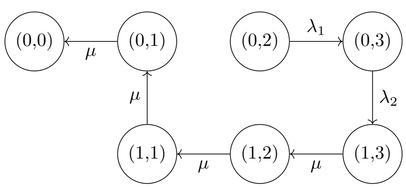
$$\longrightarrow \qquad (\lambda_2)^2\mu^4$$



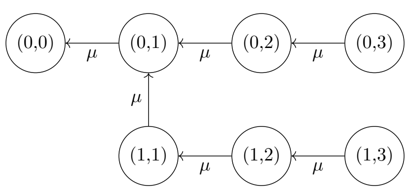
$$\longrightarrow \qquad \lambda_2\mu^5$$



$$\longrightarrow \qquad \lambda_2\mu^5$$

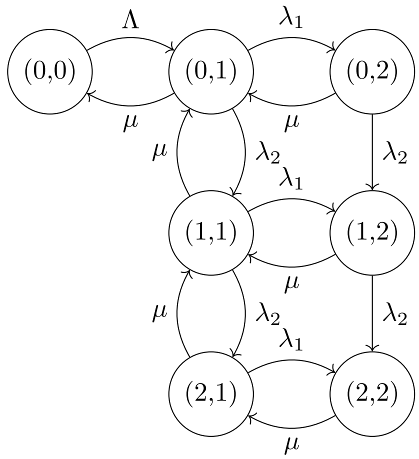


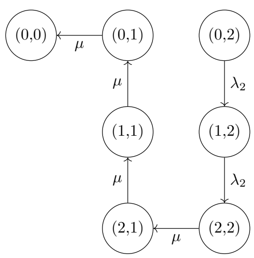
$$\longrightarrow \qquad \lambda_2\lambda_1\mu^4$$



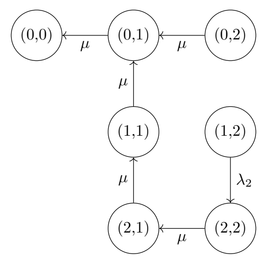
$$\longrightarrow \qquad \mu^6$$

$$\tilde{\pi}_{(0,0)} = (\lambda_2)^2\mu^4 + 2\lambda_2\mu^5 + \lambda_2\lambda_1\mu^4 + \mu^6$$

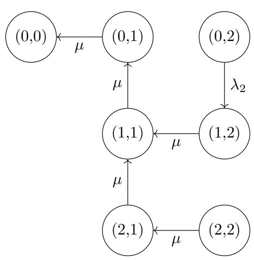




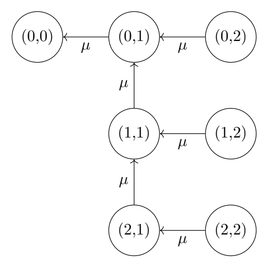
$(\lambda_2)^2\mu^4$



$\lambda_2\mu^5$



$\lambda_2\mu^5$



μ^6

$$\tilde{\pi}_{(0,0)} = (\lambda_2)^2\mu^4 + 2\lambda_2\mu^5 + \mu^6$$

6.3.5 Conjecture of adding rows

Let us consider three Markov models with the same number of servers $C = 1$, the same threshold $T = 1$, the same service centre capacity $N = 2$ but $M \in \{1, 2, 3\}$.

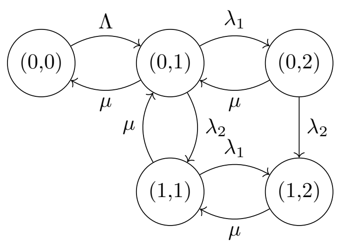


Figure 5: $M = 1$

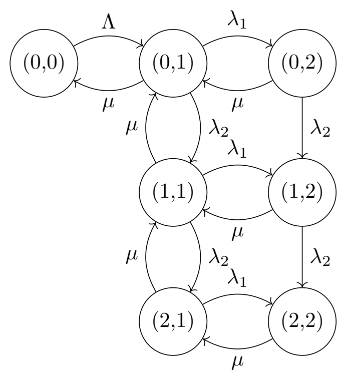


Figure 6: $M = 2$

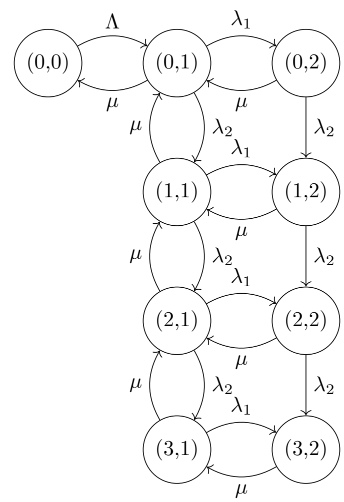


Figure 7: $M = 3$

By increasing the buffer centre capacity of the system it can be observed that $|T_{(0,0)}(G)|$ increases as well since more combinations of paths can be generated using the new edges and vertices. The corresponding values of $\tilde{\pi}_{(0,0)}$ of the three systems are:

$$
\begin{aligned}
M = 1 : \tilde{\pi}_{(0,0)} &= \mu^4 + \mu^3\lambda_2 = \mu^3(\mu + \lambda_2) && (13)\\
M = 2 : \tilde{\pi}_{(0,0)} &= \mu^6 + 2\mu^5\lambda_2 + \mu^4(\lambda_2)^2 = \mu^4(\mu^2 + 2\mu\lambda_2 + (\lambda_2)^2) = \mu^4(\mu + \lambda_2)^2 && (14)\\
M = 3 : \tilde{\pi}_{(0,0)} &= \mu^8 + 3\mu^7\lambda_2 + 3\mu^6(\lambda_2)^2 + \mu^5(\lambda_2)^3 \\
&= \mu^5(\mu^3 + 3\mu^2\lambda_2 + 3\mu(\lambda_2)^2 + (\lambda_2)^3) \\
&= \mu^5(\mu + \lambda_2)^3 && (15)
\end{aligned}
$$

Note that in equations (13),(14) and (15), the following equation holds:

$$\tilde{\pi}_{(0,0)} = \mu^{(N+M)}(\mu + \lambda_2)^M \tag{16}$$

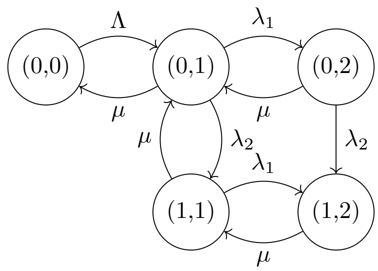
This relationship has been verified experimentally for ... and the data set is archived at ... A generalisation of equation 16, where $N \geq 1$, is given in terms of an unknown function $k(C, T, N)$ as:

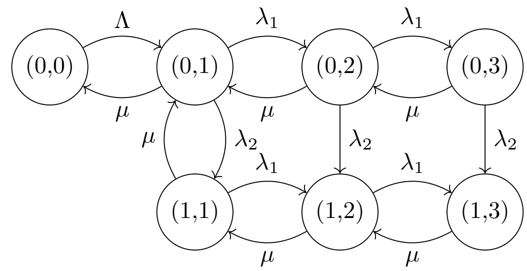
$$\tilde{\pi}_{(0,0)} = \mu^{(N+M)}(k(C, T, N))^M \tag{17}$$

Thus, having investigated the effect of adding rows (increasing M) it remains to investigate the effect of adding columns (increasing N) and finding an expression for $k(C, T, N)$.

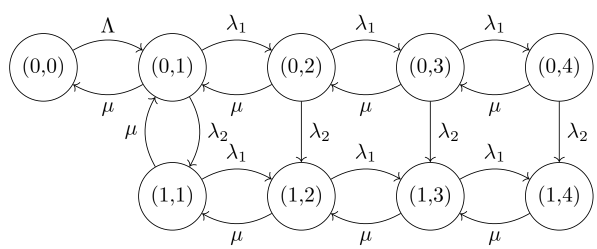
6.3.6 The effect of increasing N (Incomplete section)

In this section we will consider a buffer centre capacity of $M = 1$ and see the effect of modifying other parameters on $k(C, T, N)$.

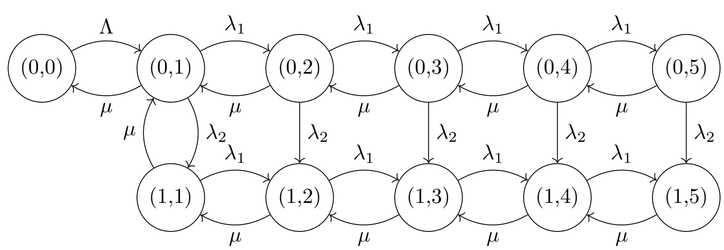




$$\tilde{\pi}_{(0,0)} = \mu^3[\lambda_2 + \mu] \qquad \tilde{\pi}_{(0,0)} = \mu^4[(\lambda_2)^2 + \lambda_2\lambda_1 + 2\lambda_2\mu + \mu^2] \tag{18}$$



$$\tilde{\pi}_{(0,0)} = \mu^5[(\lambda_2)^3 + 2(\lambda_2)^2\lambda_1 + 3(\lambda_2)^2\mu + \lambda_2(\lambda_1)^2 + 2\lambda_2\lambda_1\mu + 3\lambda_2\mu^2 + \mu^3] \tag{19}$$



$$\begin{aligned}\tilde{\pi}_{(0,0)} =& \mu^6[(\lambda_2)^4 + 3(\lambda_2)^3\lambda_1 + 4(\lambda_2)^3\mu + 3(\lambda_2)^2(\lambda_1)^2 + 6(\lambda_2)^2\lambda_1\mu \\ &+ 6(\lambda_2)^2\mu^2 + \lambda_2(\lambda_1)^3 + 2\lambda_2(\lambda_1)^2\mu + 3\lambda_2\lambda_1\mu^2 + 4\lambda_2\mu^3 + \mu^4]\end{aligned} \tag{20}$$

As explained in equation 16 the expressions defined above can boil down to a general form equation of the form $\tilde{\pi}_{(0,0)} = \mu^{(N+M)}(k(C,T,N))^M$. The only thing missing is an expression for $k(C,T,N)$. An initial attempt to get such an expression can be seen below:

$$
\begin{aligned}
k(C,T,N) &= \sum_{p_1=0}^{C-1} \sum_{p_2=0}^{C-p_1-1} \sum_{p_3=C-p_1-p_2-1}^{C-p_1-p_2-1} R(p_1,p_2,p_3)(\lambda_2)^{p_1}(\lambda_1)^{p_2}\mu^{p_3} \\
&= \sum_{p_1=0}^{C-1} \sum_{p_2=0}^{C-p_1-1} R(p_1,p_2,C-p_1-p_2-1)(\lambda_2)^{p_1}(\lambda_1)^{p_2}\mu^{C-p_1-p_2-1}
\end{aligned}
\tag{21}
$$

In equation 21 the coefficient function $R(p_1,p_2,p_3)$ is introduced were takes as arguments the powers of λ_2, λ_1 and μ. Note here that p_3, the power of μ, is defined as $p_3 = C - p_1 - p_2 - 1$ since for all base models they need to satisfy $p_1 + p_2 + p_3 = C - 1$. For the starting coefficients of the model the function $R(p_1,p_2,p_3)$ gives the values of the coefficients and is defined as:

$$
R(p_1,p_2,p_3) = \begin{cases}
0 & \text{if } p_1 = 0 \text{ and } p_2 > 0 \\
1 & \text{if } p_1, p_2 = 0 \text{ and } p_3 > 0 \\
\binom{p_1+p_3}{p_3} & \text{if } p_2 = 0 \text{ and } p_1 > 0 \\
\binom{p_1+p_2-1}{p_2} & \text{if } p_3 = 0 \text{ and } p_1, p_2 > 0 \\
p_3 + 1 & \text{if } p_1 = 1 \\
\binom{p_1+p_3+1}{p_1} + p_3\binom{p_1+p_3}{p_3+1} - \binom{p_1+p_3}{p_3} & \text{if } p_2 = 1 \text{ and } p_1 > 1 \\
\binom{p_1+p_2+1}{p_1} - \binom{p_1+p_2-1}{p_2-1} + \sum_{i=p_2}^{p_1+p_2-2} i\binom{i-1}{p_2-1} & \text{if } p_3 = 1 \text{ and } p_1, p_2 > 1 \\
U_{p_1,p_2,p_3} & \text{otherwise}
\end{cases}
\tag{22}
$$

Note here that the final value U_{p_1,p_2,p_3} corresponds to coefficients that are unknown and are currently investigated. The function R takes as arguments a possible combination of numbers of λ_2, λ_1 and μ for a given system and outputs the coefficient of that term which in turn represents how many spanning trees exist in the graph with that specific combination. For instance consider the coefficients (p_1,p_2,p_3) of some of the terms from the equations above:

- (18) $\Rightarrow (\lambda_2)^2$: $R(2,0,0) = \binom{2+0}{0} = 1$
- (18) $\Rightarrow \lambda_2\lambda_1$: $R(1,1,0) = \binom{1+1-1}{1} = 1$
- (18) $\Rightarrow 2\lambda_2\mu$: $R(1,0,1) = \binom{1+1}{1} = 2$
- (18) $\Rightarrow \mu^2$: $R(0,0,2) = 1$
- (19) $\Rightarrow 2(\lambda_2)^2\lambda_1$: $R(2,1,0) = \binom{2+1-1}{1} = 2$
- (19) $\Rightarrow 3(\lambda_2)^2\mu$: $R(2,0,1) = \binom{2+1}{1} = 3$
- (19) $\Rightarrow 3\lambda_2\mu^2$: $R(1,0,2) = \binom{1+2}{2} = 3$
- (20) $\Rightarrow 3(\lambda_2)^3\lambda_1$: $R(3,1,0) = \binom{3+1-1}{1} = 3$
- (20) $\Rightarrow 3(\lambda_2)^2(\lambda_1)^2$: $R(2,2,0) = \binom{3}{2} = 3$
- (20) $\Rightarrow 6(\lambda_2)^2\mu^2$: $R(2,0,2) = \binom{2+2}{2} = 6$
- (20) $\Rightarrow 6(\lambda_2)^2\lambda_1\mu$: $R(2,1,1) = \binom{2+1+1}{2} + 1\binom{2+1}{1+1} - \binom{2+1}{1} = 6 + 3 - 3 = 6$
- (e.g) $\Rightarrow (\lambda_2)^2(\lambda_1)^2\mu$: $R(2,2,1) = \binom{2+2+1}{2} - \binom{2+2-1}{2-1} + \sum_{i=2}^{2+2-2} i\binom{i-1}{2-1} = 10 - 3 + (2 \times 1) = 9$

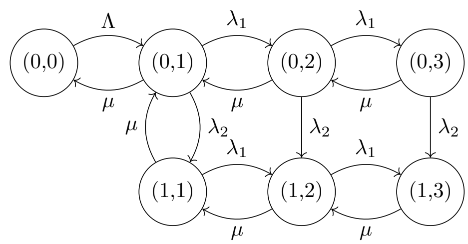
6.3.7 Unknown terms

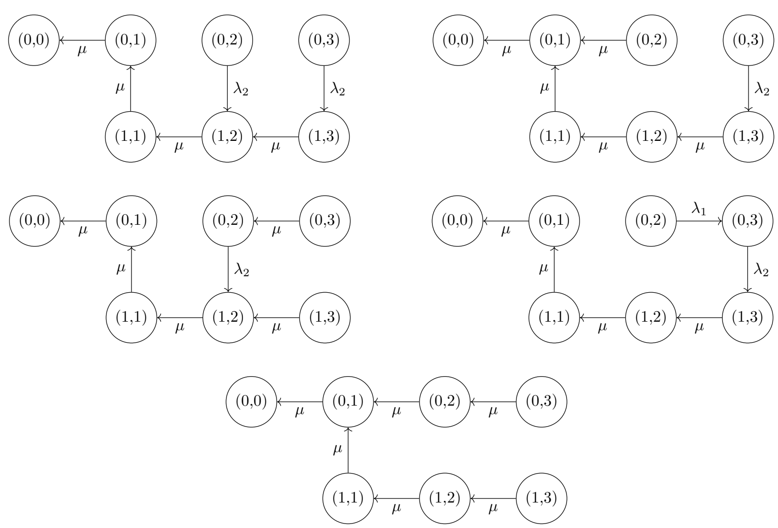
The terms that remain unknown are the terms where $p_1, p_2, p_3 \geq 2$. Here are some of these values with the corresponding values of the $R(p_1, p_2, p_3)$ function.

- $R(2,2,2) = 18$
- $R(3,2,2) = 60$
- $R(2,3,2) = 24$
- $R(2,2,3) = 30$
- $R(4,2,2) = 150$
- $R(3,3,2) = 100$
- $R(3,2,3) = 120$
- $R(2,4,2) = 30$
- $R(2,3,3) = 40$
- $R(2,2,4) = 45$

6.3.8 DRL arrays

In this section a new combinatorial object is defined: DRL arrays. It will be shown that there is a bijection between DRL arrays and the spanning trees in G_M. DRL arrays will then be enumerated which in turn enumerates the trees of $T_{(0,0)}(G_M)$. Consider the following Markov model and the spanning trees rooted at state $(0,0)$ that are associated with it.





Looking at these spanning trees from a different perspective it can be observed that all spanning trees of the specific model have some edges in common.

- $(0,1) \to (0,0)$
- $(1,1) \to (0,1)$
- $(1,2) \to (1,1)$
- $(1,3) \to (1,2)$

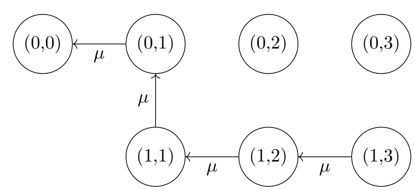
These edges are the ones on the bottom row of the model, on the *threshold column* and on the *arm* of the model. In general the set of edges that are present on all spanning trees can be denoted by:

$$\begin{aligned} S &= S_1 \cup S_2 \cup S_3 \\ S_2 &= \{(M, v) \to (M, v-1) \mid T < v \leq N\} \\ S_1 &= \{(u, T) \to (u-1, T) \mid 0 < u \leq M\} \\ S_3 &= \{(0, v) \to (0, v-1) \mid 0 < v \leq T\} \end{aligned} \tag{23}$$

In addition, these edges that are common to every spanning tree (for a threshold of $T = 1$) have the same weight of μ. In this specified model there are four of these edges, each with a weight of μ. Thus, since these edges exist on all spanning trees, the weight of every spanning tree must have include a term μ^4. Consider the expression of $\tilde{\pi}_{(0,0)}$ associated with this Markov model:

$$\tilde{\pi}_{(0,0)} = \mu^4[(\lambda_2)^2 + \lambda_2\lambda_1 + 2\lambda_2\mu + \mu^2] \tag{24}$$

It can be seen that there is a μ^4 term that is a common factor of all the terms. This term can be more generally calculated as μ^{M+N} and by not worrying about all these edges that belong in S the problem can be slightly simplified.



The specific problem has now been reduced to finding all possible combinations of two edges where one starts from $(0,2)$ and the other from $(0,3)$. The possible edges that can be utilised here may have a direction of either left, right, or down. Thus, the objective of the problem can be transformed into finding all possible permutations of an array of size 2 where elements can be L, R or D and obey certain rules so that the permutation corresponds to a valid spanning tree. These rules are:

1. Permutations ending with an R are not valid.
2. Permutations that have an R followed by an L are not valid.

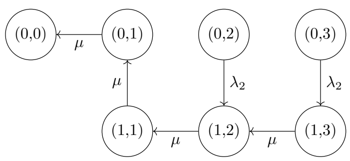
If any of these two rules does not hold then the permutation is immediately invalid. Rule 1 points to the cases where the final state has an edge pointing to the right of it, which cannot occur since that state is the right-most state of the first row. Rule 2 makes sure that there are no neighbour states that point to each other since that would create a cycle and would not generate a valid spanning tree.

For instance, consider the model above. Shown below, are all possible permutations of the array along with the excluded cases. The valid permutations (on the left) are shown in the same order with their corresponding spanning trees from the above figure and the invalid permutations (on the right) are followed by the rule that determines them invalid.

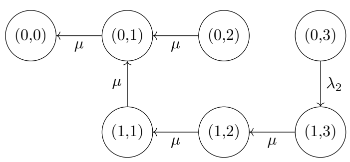
- $[D, D]$
- $[L, D]$
- $[D, L]$
- $[R, D]$
- $[L, L]$

- $\cancel{[R, R]} \rightarrow$ Rule 1
- $\xcancel{[L, R]} \rightarrow$ Rule 1
- $\xcancel{[R, L]} \rightarrow$ Rule 2
- $\cancel{[D, R]} \rightarrow$ Rule 1

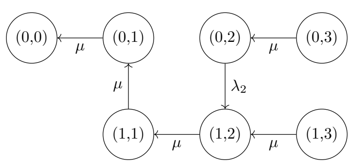
6.3.9 Examples of mappings of directed spanning trees to permutation arrays



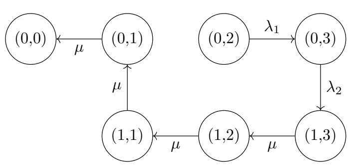
$\longrightarrow$ $[D, D]$



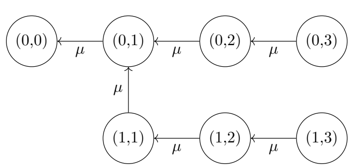
$\longrightarrow$ $[L, D]$



$\longrightarrow$ $[D, L]$



$\longrightarrow$ $[R, D]$



$\longrightarrow$ $[L, L]$

6.3.10 Closed-form formula

A general formula for finding all such permutations can be found, where the inputs are p_1, p_2 and p_3 that correspond to the number of D, R and L respectively and the output would be the coefficient of the term $(\lambda_2)^{p_1}(\lambda_1)^{p_2}\mu^{p_3}$. For instance, by applying such a formula to the example in equation 24, the desired output should be:

- $(\lambda_2)^2 \quad \rightarrow \quad p_1 = 2, p_2 = 0, p_3 = 0 \quad \rightarrow \quad$ coefficient $= 1$
- $\lambda_2\lambda_1 \quad \rightarrow \quad p_1 = 1, p_2 = 1, p_3 = 0 \quad \rightarrow \quad$ coefficient $= 1$
- $2\lambda_2\mu \quad \rightarrow \quad p_1 = 1, p_2 = 0, p_3 = 1 \quad \rightarrow \quad$ coefficient $= 2$
- $\mu^2 \quad \rightarrow \quad p_1 = 0, p_2 = 0, p_3 = 2 \quad \rightarrow \quad$ coefficient $= 1$

Thus, given all possible and valid combinations of powers among λ_2, λ_1 and μ (i.e. p_1, p_2, p_3) generated by equation 21, an alternative and improved form of the value of $R(p_1, p_2, p_3)$ described in equation 22 is given by:

$$R(p_1,p_2,p_3) = T(p_1,p_2,p_3) - E_R(p_1,p_2,p_3) - E_D(p_1,p_2,p_3) - E_L(p_1,p_2,p_3) - E_{RL}(p_1,p_2,p_3) \quad (25)$$

Consider the above undefined equations. The term $T(p_1, p_2, p_3)$ denotes the number of all permutations where neither rule is applied, i.e. all possible ways one can arrange the elements of the array. The term $E_R(p_1, p_2, p_3)$ denotes the number of permutations that end in R, which needs to be removed from the total of all permutations so that rule 1 is satisfied. Having excluded all permutations that end in R it only remains to leave out permutations that have an R followed by an L (rule 2). Although, removing all permutations ending in R was relatively straight forward, removing all permutations that follow rule 2 is not that easy. This is because equation $E_R(p_1, p_2, p_3)$ already considers some cases where there is an R followed by an L. Therefore, in order to consider only new cases, permutations of rule 2 are split into three new terms; E_D, E_L and E_{RL}. These terms denote the permutations that have an R followed by an L AND do not end in R. The term E_D considers all permutations that end in D while E_L the ones that end in L. Finally, the last term (E_{RL}) denotes all permutations that end in R, L where there is no other R followed by an L in any other position apart from the last two. This term is used because in the E_L term, such cases (where R and L are in the last two positions) are only considered when there is another R followed by an L somewhere. Thus, the term E_{RL} is a particular set of permutations that the formula of E_L fails to include by itself.

$$T(p_1,p_2,p_3) = \frac{(p_1+p_2+p_3)!}{p_1! \times p_2! \times p_3!} \tag{26}$$

$$E_R(p_1,p_2,p_3) = \frac{(p_1+p_2+p_3-1)!}{p_1! \times (p_2-1)! \times p_3!} \tag{27}$$

$$E_D(p_1,p_2,p_3) = \sum_{i=1}^{\min(R,L)} (-1)^{i+1} \frac{(p_1+p_2+p_3-i-1)!}{(p_1-1)! \times (p_2-i)! \times (p_3-i)! \times (i)!} \tag{28}$$

$$E_L(p_1,p_2,p_3) = \sum_{i=1}^{\min(R,L-1)} (-1)^{i+1} \frac{(p_1+p_2+p_3-i-1)!}{p_1! \times (p_2-i)! \times (p_3-i-1)! \times (i)!} \tag{29}$$

$$E_{RL}(p_1,p_2,p_3) = \sum_{i=1}^{\min(R,L)} (-1)^{i+1} \frac{(p_1+p_2+p_3-i-1)!}{p_1! \times (p_2-i)! \times (p_3-i)! \times (i-1)!} \tag{30}$$

$$R(p_1,p_2,p_3) = T(p_1,p_2,p_3) - E_R(p_1,p_2,p_3) - E_D(p_1,p_2,p_3) - E_L(p_1,p_2,p_3) - E_{RL}(p_1,p_2,p_3)$$

6.3.11 Example of the permutation algorithm

Consider the term $(\lambda_2)(\lambda_1)\mu^2$ and the above expressions. In order to get the coefficient of that term the permutation algorithm needs to be applied with an input of $p_1 = 1, p_2 = 1, p_3 = 2$, i.e. 1 D, 1 R and 2 Ls in the array. The permutations that correspond to each expression can be seen below:

$$T(p_1,p_2,p_3) = \frac{(1+1+2)!}{1!\ 1!\ 2!} = 12$$

$$\begin{array}{cccccc} [D,R,L,L] & [R,D,L,L] & [D,L,R,L] & [R,L,D,L] & [D,L,L,R] & [R,L,L,D] \\ [L,D,R,L] & [L,R,D,L] & [L,D,L,R] & [L,R,L,D] & [L,L,D,R] & [L,L,R,D] \end{array}$$

$$E_R(p_1,p_2,p_3) = \frac{(1+1+2-1)!}{1!\ (1-1)!\ 2!} = 3$$

$$[D,L,L,|R] \quad [L,D,L,|R] \quad [L,L,D,|R]$$

$$E_D(p_1,p_2,p_3) = \sum_{i=1}^{1} (-1)^{i+1} \frac{(1+1+2-i-1)!}{0!\ (1-i)!\ (2-i)!\ (i)!} = 1 \times \frac{2}{0!\ 0!\ 1!\ 1!} = 2$$

$$[R,L,L,|D] \quad [L,R,L,|D]$$

$$E_L(p_1, p_2, p_3) = \sum_{i=1}^{1} (-1)^{i+1} \frac{(1+1+2-i-1)!}{1!\,(1-i)!\,(2-i-1)!\,(i)!} = 1 \times \frac{2}{1!\,0!\,0!\,1!} = 2$$

$$[D, R, L, |L] \quad [R, L, D, |L]$$

$$E_{RL}(p_1, p_2, p_3) = \sum_{i=1}^{1} (-1)^{i+1} \frac{(1+1+2-i-1)!}{1!\,(1-i)!\,(2-i)!\,(i_1)!} = 1 \times \frac{2}{1!\,0!\,1!\,0!} = 2$$

$$[D, L, |R, L] \quad [L, D, |R, L]$$

6.3.12 Possibly useful theorem: Matrix-tree theorem for directed graphs (Kirchhoff's theorem) [3]:

The number of directed spanning trees rooted at a state i can be found by calculating the determinant of the Laplacian matrix Q of the directed graph and removing row i and column i.

6.4 Expressions derived from π:

One may easily derive the average number of individuals that are at any given state using *pi*. The average number of individuals in state i can be calculated by multiplying the number of individuals that are present in state i with the probability of being at that particular state (i.e $\pi_i(u_i + v_i)$). Using this logic it is possible to calculate any performance measures that are related to the mean number of individuals in the system.

Average number of people in the system:

$$L = \sum_{i=1}^{|\pi|} \pi_i(u_i + v_i) \tag{31}$$

Average number of people in the service centre:

$$L_H = \sum_{i=1}^{|\pi|} \pi_i v_i \tag{32}$$

Average number of people in the buffer centre:

$$L_A = \sum_{i=1}^{|\pi|} \pi_i u_i \tag{33}$$

Consequently getting the performance measures that are related to the duration of time is not as straightforward. Such performance measures are the mean waiting time in the system and the mean time blocked in the system. Under the scope of this study three approaches have been considered to calculate these performance measures; a direct approach, a recursive algorithm and consequently a closed-form formula.

The research question that needs to be answered here is: "When a class 1/2 individuals enters the system, what is the expected time that they will have to wait?". In order to formulate the answer to that question one needs to consider all possible scenarios of what state the system can be in when an individual arrives. Furthermore, different formulas arises for class 1 individuals and a different one for class 2 individuals.

6.5 Mean waiting time

6.5.1 Recursive formula for mean waiting time of class 1 individuals

To calculate the mean waiting time of class 1 individuals one must first identify the set of states (u, v) that will imply that a wait will occur. For this particular Markov chain, this points to all states that satisfy $v > C$ i.e. all states where the number of individuals in the service centre exceed the number of servers. The set of such states is defined as *waiting states* and can be denoted as a subset of all the states, where:

$$S_w = \{(u, v) \in S \mid v > C\} \tag{34}$$

Additionally, there are certain states in the model where arrivals cannot occur. A class 1 individual cannot arrive whenever the model is at any state (u, N) for all u where N is the system capacity. Therefore the set of all such states that an arrival may occur are defined as *accepting states* and are denoted as:

$$S_A^{(1)} = \{(u, v) \in S \mid v < N\} \tag{35}$$

Moreover, another element that needs to be considered is the expected waiting time in each state $c(u, v)$, otherwise known as sojourn time [6]. In order to do so a variation of the Markov model has to be considered where when the individual arrives at any of the states of the model no more arrivals can occur after that.

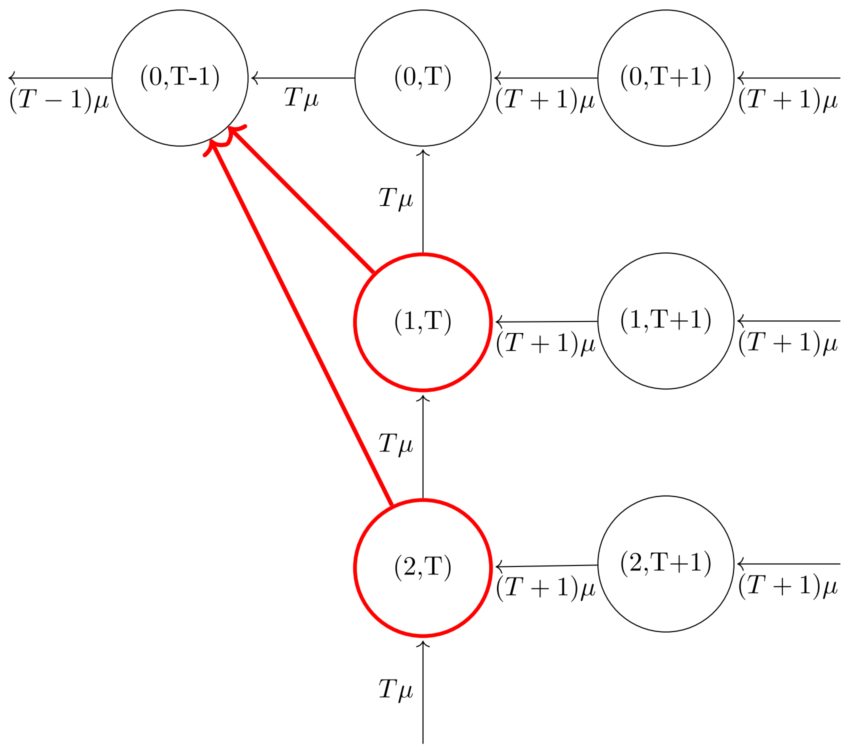


Figure 8: Markov chain - ignoring any arrivals

As illustrated in figure 8 a class 1 individual, when in the threshold column, only visits one of the nodes since they are not affected by class 2 individuals. Thus, one may acquire the desired time by calculating the inverse of the sum of the out-flow rate of that state. Since we are ignoring arrivals though the only way to exit the state will only be via a service. In essence this notion can be expressed as:

$$c^{(1)}(u,v) = \begin{cases} 0, & \textbf{if } u > 0 \textbf{ and } v = T \\ \frac{1}{\min(v,C)\mu}, & \textbf{otherwise} \end{cases} \tag{36}$$

Note that whenever any class 1 individual is at a state (u,v) where $u > 0$ and $v = T$ (i.e. all states $(1,T),(2,T)\dots,(M,T)$) the sojourn time is set to 0. This is done to capture the trip thorough the Markov chain from the perspective of class 1 individuals. Meaning that they will visit all states of the threshold column but only the one in the first row will return a non-zero sojourn time.

Now, using the above equations, and considering all sates that belong in S_w the following recursive formula can be used to get the mean waiting time spent in each state in the Markov model. For class 1 individuals, whenever the model is at state (u,v), any incoming individual will proceed to arrive at state $(u,v+1)$. individuals will then proceed to visit all other states until they reach one which has less than C servers occupied (i.e. until a server becomes available). The formula goes through all states from right to left recursively and adds the sojourn times of all these states together until it reaches a state that is not in the set of waiting states. Thus, the expected waiting time of a class 1 individual when they arrive at state (u,v) can be given by:

$$w^{(1)}(u,v) = \begin{cases} 0, & \textbf{if } (u,v) \notin S_w \\ c^{(1)}(u,v) + w^{(1)}(u-1,v), & \textbf{if } u > 0 \textbf{ and } v = T \\ c^{(1)}(u,v) + w^{(1)}(u,v-1), & \textbf{otherwise} \end{cases} \tag{37}$$

Finally, the mean waiting time can be calculated by summing over all expected waiting times of accepting states multiplied by the probability of being at that state and dividing by the probability of being in any accepting state.

$$W^{(1)} = \frac{\sum_{(u,v)\in S_A^{(1)}} w^{(1)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(1)}} \pi_{(u,v)}} \tag{38}$$

6.5.2 Recursive formula for mean waiting time of class 2 individuals

Equivalently the mean waiting time for class 2 individuals can be calculated in a similar manner. The set of waiting states is the same as before but there is a slight modification in the set of accepting states.

$$S_w = \{(u,v) \in S \mid v > C\}$$

$$S_A^{(2)} = \begin{cases} \{(u,v) \in S \mid u < M\} & \textbf{if } T \leq N \\ \{(u,v) \in S \mid v < N\} & \textbf{otherwise} \end{cases} \tag{39}$$

The set of accepting states is modified in such a way such that a class 2 individual cannot arrive in the model when the model is at any state (M,v) for all $v \geq T$ where M is the buffer centre capacity and T is the threshold. An odd situation here is when the threshold is set to a very high number that is more than the actual system capacity. In such cases the set of accepting states is defined in the same way as the class 1 individuals case (35). That is because whenever $T > N$ no class 2 individual will ever be blocked in the model (since that threshold can never be reached) and thus the last accepting state of the model will be state $(0, N-1)$.

Now just like in the class 1 individuals case the sojourn time is needed. For class 2 individuals whenever individuals are at any row apart from the first one they automatically get a wait time of 0 since they are essentially blocked.

$$c^{(2)}(u,v) = \begin{cases} 0, & \textbf{if } u > 0 \\ \frac{1}{\min(v,C)\mu}, & \textbf{otherwise} \end{cases} \tag{40}$$

Finally, the recursive formula and the mean waiting time equation are identical to the ones described above with the exception that they now use $c^{(2)}(u,v)$ instead of $c^{(1)}(u,v)$.

$$w^{(2)}(u,v) = \begin{cases} 0, & \textbf{if } (u,v) \notin S_w \\ c^{(2)}(u,v) + w^{(2)}(u-1,v), & \textbf{if } u > 0 \textbf{ and } v = T \\ c^{(2)}(u,v) + w^{(2)}(u,v-1), & \textbf{otherwise} \end{cases} \tag{41}$$

$$W^{(2)} = \frac{\sum_{(u,v)\in S_A^{(2)}} w^{(2)}(u,v)\pi_{(u,v)}}{\sum_{(u,v)\in S_A^{(2)}} \pi_{(u,v)}} \tag{42}$$

6.5.3 Direct waiting time formula

An alternative approach to getting the waiting time formula from the Markov chain would be to simply solve the linear system of equations from equations 37 and 41 for type 1 and type 2 individuals respectively. The set of equations that need to be solved for individuals of type i are all $w^{(i)}(u,v)$ for all possible states $(u,v) \in S$.

$$S_w = \{(u,v) \in S \mid v > C\}$$

$$S_A^{(1)} = \{(u,v) \in S \mid v < N\}, \qquad S_A^{(2)} = \begin{cases} \{(u,v) \in S \mid u < M\} & \textbf{if } T \leq N \\ \{(u,v) \in S \mid v < N\} & \textbf{otherwise} \end{cases}$$

$$c^{(1)}(u,v) = \begin{cases} 0, & \textbf{if } u > 0 \textbf{ and } v = T \\ \frac{1}{\min(v,C)\mu}, & \textbf{otherwise} \end{cases} \qquad c^{(2)}(u,v) = \begin{cases} 0, & \textbf{if } u > 0 \\ \frac{1}{\min(v,C)\mu}, & \textbf{otherwise} \end{cases}$$

$$w^{(i)}(u,v) = \begin{cases} 0, & \textbf{if } (u,v) \notin S_b \\ c^{(i)}(u,v) + w^{(i)}(u-1,v), & \textbf{if } u > 0 \textbf{ and } v = T \\ c^{(i)}(u,v) + w^{(i)}(u,v-1), & \textbf{otherwise} \end{cases} \tag{43}$$

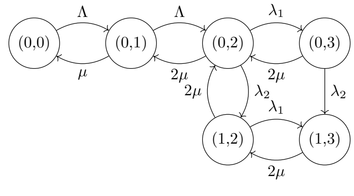


Figure 9: Example of Markov chain

$$w^{(i)}(1,2) = c(1,2) + w^{(i)}(0,2) \tag{44}$$

$$w^{(i)}(1,3) = c(1,3) + w^{(i)}(1,2) \tag{45}$$

$$w^{(i)}(2,2) = c(2,2) + w^{(i)}(1,2) \tag{46}$$

$$w^{(i)}(2,3) = c(2,3) + w^{(i)}(2,2) \tag{47}$$

Thus similar to equations 38 and 42 the mean waiting time for type i individuals is given by:

$$W^{(i)} = \frac{\sum_{(u,v) \in S_A^{(i)}} \pi_{(u,v)} w^{(i)}(\mathcal{A}_i(u,v))}{\sum_{(u,v) \in S_A^{(i)}} \pi_{(u,v)}} \tag{48}$$

Here $S_A^{(2)}$ is the set of accepting states of type 2 individuals and $\mathcal{A}_i(u,v)$ for $i \in \{1,2\}$ is the state that the system would go to when the system is at state (u,v) and an individual of type i arrives.

$$\mathcal{A}_1(u,v) = (u, v+1) \tag{49}$$

$$\mathcal{A}_2(u,v) = \begin{cases} (u, v+1), & \text{if } v < T \\ (u+1, v), & \text{if } v \geq T \end{cases} \tag{50}$$

6.5.4 Mean Waiting Time - Closed-form

Upon closer inspection of the recursive formula a more compact formula can arise. The equivalent closed-form formula eliminates the need for recursion and thus makes the computation of waiting times much more efficient. Just like in the recursive part there are two formulas; one for *class 1* and one for class 2 individuals. The formulas are given by:

$$W^{(1)} = \frac{\sum_{\substack{(u,v) \in S_A^{(1)} \\ v \geq C}} \frac{1}{C\mu} \times (v - C + 1) \times \pi(u,v)}{\sum_{(u,v) \in S_A^{(1)}} \pi(u,v)} \tag{51}$$

$$W^{(2)} = \frac{\sum_{\substack{(u,v) \in S_A^{(2)} \\ min(v,T) \geq C}} \frac{1}{C\mu} \times (\min(v+1, T) - C) \times \pi(u,v)}{\sum_{(u,v) \in S_A^{(2)}} \pi(u,v)} \tag{52}$$

Note here that the summation, in both equations 51 and 52, goes through all states in the set of accepting states of either class 1 or class 2 individuals respectively, where a wait incurs. In equation 51 that includes all states (u,v) in the set of accepting states of class 1 individuals such that $v \geq C$; i.e. whenever an arrival occurs and the system is at a state where the number of individuals in the system is more than or equal to C. Consequently, for the states that are included in the summation the expression $v - C + 1$ indicates the amount of people in service one would have to wait for upon arrival at the hospital.

Additionally, the minimisation function in equation 52 ensures that when a class 2 individual arrives at any state that is greater than the predetermined threshold, the wait that the individual will have to endure remains the same. In essence, the expression $\min(v+1, T) - C$ returns the number of people in line in front of a particular individual upon arrival.

6.5.5 Overall Waiting Time

Consequently, the overall waiting time should can be estimated by a linear combination of the waiting times of class 1 and class 2 individuals. The overall waiting time can be then given by the following equation where c_1 and c_2 are the coefficients of each individual's type waiting time:

$$W = c_1 W^{(1)} + c_2 W^{(2)} \tag{53}$$

The two coefficients represent the proportion of individuals of each type that traversed through the model. Theoretically, getting these percentages should be as simple as looking at the arrival rates of each type but in practise if the service centre or the buffer centre is full, some individuals may be lost to the system. Thus, one should account for the probability that an individual is lost to the system. This probability can be easily calculated by using the two sets of accepting states $S_A^{(2)}$ and $S_A^{(1)}$ defined earlier in equations 35 and 39. Let us define here the probability, for either class type, that an individual is not lost in the system by:

$$P(L_1') = \sum_{(u,v)\, \in S_A^{(1)}} \pi(u,v) \qquad\qquad P(L_2') = \sum_{(u,v)\, \in S_A^{(2)}} \pi(u,v)$$

Having defined these probabilities one may combine them with the arrival rates of each class type in such a way to get the expected proportions of class 1 and class 2 individuals in the model. Thus, by using these values as the coefficient of equation 53 the resultant equation can be used to get the overall waiting time. Note here that the equation below gets the overall waiting time for both the recursive and the closed-form formula.

$$W = \frac{\lambda_1 P(L_1')}{\lambda_2 P(L_2') + \lambda_1 P(L_1')} W^{(1)} + \frac{\lambda_2 P(L_2')}{\lambda_2 P(L_2') + \lambda_1 P(L_1')} W^{(2)} \tag{54}$$

6.6 Mean blocking time

6.6.1 Direct approach for mean blocking time

Unlike the waiting time, the blocking time is only calculated for class 2 individuals. That is because class 1 individuals cannot be blocked. Thus, one only needs to consider the pathway of class 2 individuals to get the mean blocking time of the system. Blocking occurs at states (u, v) where $u > 0$. Thus, the set of blocking states can be defined as:

$$S_b = \{(u, v) \in S \mid u > 0\}$$

In order to not consider individuals that will be lost to the system, the set of accepting states needs to be taken into account. As defined in section 6.5.2, the set of accepting states is given by (39):

$$S_A^{(2)} = \begin{cases} \{(u,v) \in S \mid u < M\} & \textbf{if } T \leq N \\ \{(u,v) \in S \mid v < N\} & \textbf{otherwise} \end{cases}$$

For the waiting time formula in sections 6.5.1 and 6.5.2 the mean sojourn time for each state was considered, ignoring any arrivals. Here, the same approach is used but ignoring only class 2 arrivals. That is because for the waiting time formula, once an individual enters the service centre (i.e. starts waiting) any individual arriving after them will not affect their pathway. That is not the case for blocking time. When a class 2 individual is blocked, any class 1 individual that arrives will cause the blocked individual to remain blocked for more time. Therefore, class 1 arrivals are considered here:

$$c(u,v) = \begin{cases} \frac{1}{\min(v,C)\mu}, & \text{if } v = C \\ \frac{1}{\min(v,C)\mu + \lambda_1}, & \text{otherwise} \end{cases} \tag{55}$$

In equation 55, both service completions and class 1 arrivals are considered. Thus, from a blocked individual's perspective whenever the system moves from one state (u, v) to another state it can either:

- be because of a service being completed: we will denote the probability of this happening by $p_s(u, v)$.
- be because of an arrival of an individual of class 1: denoting such probability by $p_o(u, v)$.

The probabilities are given by:

$$p_s(u,v) = \frac{\min(v,C)\mu}{\lambda_1 + \min(v,C)\mu}, \qquad p_o(u,v) = \frac{\lambda_1}{\lambda_1 + \min(v,C)\mu}$$

Having defined $c(u, v)$ and S_b a formula for the blocking time that is expected to occur at each state can be given by:

$$b(u,v) = \begin{cases} 0, & \textbf{if } (u,v) \notin S_b \\ c(u,v) + b(u-1,v), & \textbf{if } v = N = T \\ c(u,v) + b(u,v-1), & \textbf{if } v = N \neq T \\ c(u,v) + p_s(u,v)b(u-1,v) + p_o(u,v)b(u,v+1), & \textbf{if } u > 0 \textbf{ and } v = T \\ c(u,v) + p_s(u,v)b(u,v-1) + p_o(u,v)b(u,v+1), & \textbf{otherwise} \end{cases} \tag{56}$$

Unlike equations (37) and (41), equation (56) will not be solved recursively. A direct approach will be used to solve this equation here. By enumerating all equations of (56) for all states (u, v) that belong in S_b a system of linear equations arises where the unknown variables are all the $b(u, v)$ terms. For instance, let us consider a Markov model where $C = 2, T = 3, N = 6, M = 2$. The Markov model is shown in Figure 10 and the equivalent equations are (57)-(62). The equations considered here are only the ones that correspond to the blocking states.

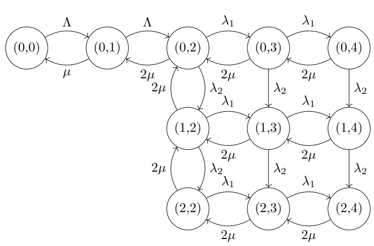


Figure 10: Example of Markov chain

$$b(1,2) = c(1,2) + p_o b(1,3) \tag{57}$$
$$b(1,3) = c(1,3) + p_s b(1,2) + p_o b(1,4) \tag{58}$$
$$b(1,4) = c(1,4) + b(1,3) \tag{59}$$
$$b(2,2) = c(2,2) + p_s b(1,2) + p_o b(2,3) \tag{60}$$
$$b(2,3) = c(2,3) + p_s b(2,2) + p_o b(1,4) \tag{61}$$
$$b(2,4) = c(2,4) + b(2,3) \tag{62}$$

Additionally, the above equations can be transformed into a linear system of the form $Zx = y$ where:

$$Z = \begin{pmatrix} -1 & p_o & 0 & 0 & 0 & 0 \\ p_s & -1 & p_o & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ p_s & 0 & 0 & -1 & p_o & 0 \\ 0 & 0 & 0 & p_s & -1 & p_o \\ 0 & 0 & 0 & 0 & 1 & -1 \end{pmatrix}, x = \begin{pmatrix} b(1,2) \\ b(1,3) \\ b(1,4) \\ b(2,2) \\ b(2,3) \\ b(2,4) \end{pmatrix}, y = \begin{pmatrix} -c(1,2) \\ -c(1,3) \\ -c(1,4) \\ -c(2,2) \\ -c(2,3) \\ -c(2,4) \end{pmatrix} \tag{63}$$

A more generalised form of the equations in (63) can thus be given for any value of C, T, N, M by:

$$b(1,T) = c(1,T) + p_o b(1,T+1) \tag{64}$$
$$b(1,T+1) = c(1,T+1) + p_s(1,T) + p_o b(1,T+1) \tag{65}$$
$$b(1,T+2) = c(1,T+2) + p_s(1,T+1) + p_o b(1,T+3) \tag{66}$$
$$\vdots$$
$$b(1,N) = c(1,N) + b(1,N-1) \tag{67}$$
$$b(2,T) = c(2,T) + p_s b(1,T) + p_o b(2,T+1) \tag{68}$$
$$b(2,T+1) = c(2,T+1) + p_s b(2,T) + p_o b(2,T+2) \tag{69}$$
$$\vdots$$
$$b(M,T) = c(M,T) + b(M,T-1) \tag{70}$$

The equivalent matrix form of the linear system of equations (64) - (70) is given by $Zx = y$, where:

$$Z = \begin{pmatrix} -1 & p_o & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ p_s & -1 & p_o & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & p_s & -1 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & -1 & 0 & 0 & 0 & \dots & 0 & 0 \\ p_s & 0 & 0 & \dots & 0 & 0 & -1 & p_o & 0 & \dots & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & p_s & -1 & p_o & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & \dots & 1 & -1 \end{pmatrix}, x = \begin{pmatrix} b(1,T) \\ b(1,T+1) \\ b(1,T+2) \\ \vdots \\ b(1,N) \\ b(2,T) \\ b(2,T+1) \\ \vdots \\ b(M,T) \end{pmatrix}, y = \begin{pmatrix} -c(1,T) \\ -c(1,T+1) \\ -c(1,T+2) \\ \vdots \\ -c(1,N) \\ -c(2,T) \\ -c(2,T+1) \\ \vdots \\ -c(M,T) \end{pmatrix} \tag{71}$$

Thus, having calculated the mean blocking time for all blocking states $b(u, v)$, it only remains to put them together in a formula just like in equations 38 and 42. The resultant blocking time formula is given by:

$$B = \frac{\sum_{(u,v)\in S_A} \pi_{(u,v)}\ b(u,v)}{\sum_{(u,v)\in S_A} \pi_{(u,v)}} \tag{72}$$

6.7 Mean proportion of arrivals within time target

Another performance measure that needs to be taken into consideration is the proportion of individuals whose waiting and service times lie within a specified time target. In order to consider such measure though one would need to obtain the distribution of time in the system for all individuals. The complexity of such a task lies on the fact that different individuals arrive at different states of the Markov model. Let us first consider the case when an arrival occurs when the model is at a specific state.

6.7.1 Distribution of time at a specific state (with 1 server)

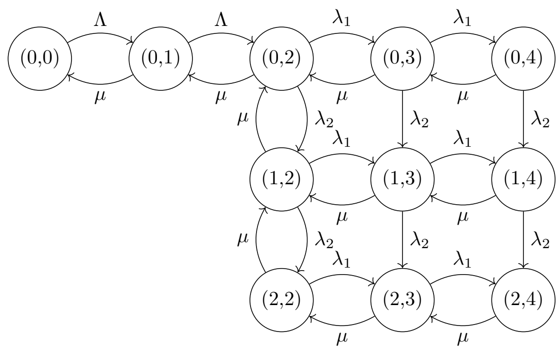


Figure 11: Example Markov model $C = 1, T = 2, N = 4, M = 2$

Consider the Markov model of figure 11 with one server and a threshold of two individuals. Let us now assume that a class 1 individual arrives when the model is at state $(0, 3)$, thus forcing the model to move to state $(0, 4)$. The distribution of the time needed for the specified individual to exit the system from state $(0, 4)$ is given by the sum of exponentially distributed random variables with the same parameter μ. The sum of such random variables form the Erlang distribution which is defined by the number of random variables that are added and their exponential parameter. Note here that these random variables represent the individual's pathway from the perspective of the individual. Thus, X_i represents the random variable of the time that it takes for an individual to move from the i^{th} position of the queue to the $(i-1)^{\text{th}}$ position (i.e. for someone in front of them to finish their service) and X_0 is the time it takes that individual to move from having a service to exiting the system.

$$\begin{aligned}
(0,4) \Rightarrow &\quad X_3 \sim Exp(\mu) \\
(0,3) \Rightarrow &\quad X_2 \sim Exp(\mu) \\
(0,2) \Rightarrow &\quad X_1 \sim Exp(\mu) \\
(0,1) \Rightarrow &\quad X_0 \sim Exp(\mu) \\
S = X_3 + X_2 + X_1 + X_0 &= Erlang(4, \mu) \qquad (73)
\end{aligned}$$

Thus, the waiting and service time of an individual in the model of figure 11 can be captured by an erlang distributed random variable. The general CDF of the erlang distribution $Erlang(k, \mu)$ is given by:

$$P(S < t) = 1 - \sum_{i=0}^{k-1} \frac{1}{i!} e^{-\mu t} (\mu t)^i \tag{74}$$

Unfortunately, the erlang distribution can only be used for the sum of identically distributed random variables from the exponential distribution. Therefore, this approach cannot be used when one of the random variables has a different parameter than the others. In fact the only case where we can use it is only when the number of servers are $C = 1$, just like in the explored example, or when an individual arrives and goes straight to service (i.e. when there is no other individual waiting and there is an empty server).

6.7.2 Distribution of time at a specific state (with multiple servers)

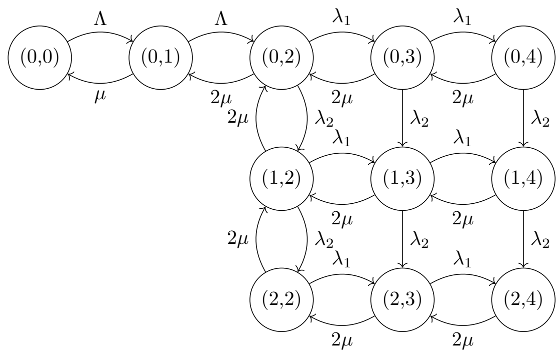


Figure 12: Example Markov model $C = 2, T = 2, N = 4, M = 2$

Figure 12 represents the same Markov model as figure 11 with the only exception that there are 2 servers here. By applying the same logic, assuming that an individual arrives at state $(0, 4)$, the sum of the following random variables arises.

$$\begin{aligned} (0,4) \Rightarrow \quad & X_2 \sim Exp(2\mu) \\ (0,3) \Rightarrow \quad & X_1 \sim Exp(2\mu) \\ (0,2) \Rightarrow \quad & X_0 \sim Exp(\mu) \end{aligned} \tag{75}$$

Since these exponentially distributed random variables do not share the same parameter, an erlang distribution cannot be used. In fact, the problem can now be viewed either as the sum of exponentially distributed random variables with different parameters or as the sum of erlang distributed random variables. The sum of erlang distributed random variables is said to follow the hypoexponential distribution. The hypoexponential distribution is defined with two vectors of size equal to the number of Erlang random variables that are added together [1], [7]. For this particular example:

$$\left.\begin{array}{r}\left.\begin{array}{r}X_2 \sim Exp(2\mu)\\ X_1 \sim Exp(2\mu)\end{array}\right\} X_1 + X_2 = S_1 \sim Erlang(2, 2\mu)\\ X_0 \sim Exp(\mu) \Rightarrow \qquad\qquad X_0 = S_2 \sim Erlang(1, \mu)\end{array}\right\} S_1 + S_2 = H \sim Hypo((2,1),(2\mu,\mu)) \tag{76}$$

The random variable H from equation 76 follows a hypoexponential distribution with two vector parameters. The CDF of this distribution can be therefore used to get the probability of the time in spent in the system being less than a given target. The CDF of the general hypoexponential distribution $Hypo(\vec{r}, \vec{\lambda})$, is given by the following expression, where vector $\vec{r}$ contains all k-values of the erlang distributions and $\vec{\lambda}$ is a vector of the distinct parameters [4].

$$\begin{aligned}
&P(H < t) = 1 - \left(\prod_{j=1}^{|\vec{r}|} \lambda_j^{r_j}\right) \sum_{k=1}^{|\vec{r}|} \sum_{l=1}^{r_k} \frac{\Psi_{k,l}(-\lambda_k) t^{r_k - l} e^{-\lambda_k t}}{(r_k - l)!(l-1)!}\\
&\textbf{where} \qquad \Psi_{k,l}(t) = -\frac{\partial^{l-1}}{\partial t^{l-1}} \left(\prod_{j=0, j\neq k}^{|\vec{r}|} (\lambda_j + t)^{-r_j}\right)\\
&\textbf{and} \qquad \lambda_0 = 0, r_0 = 1
\end{aligned} \tag{77}$$

The computation of the derivative makes equation 77 computationally expensive. In [5] an alternative linear version of that CDF is explored via matrix analysis, and is given by the following formula:

$$\begin{aligned}
F(x) =& 1 - \sum_{k=1}^{n} \sum_{l=0}^{k-1} (-1)^{k-1} \binom{n}{k} \binom{k-1}{l} \sum_{j=1}^{n} \sum_{s=1}^{j-1} e^{-x\lambda_s} \prod_{l=1}^{s-1} \left(\frac{\lambda_l}{\lambda_l - \lambda_s}\right)^{k_s}\\
&\times \sum_{s < a_1 < \cdots < a_{l-1} < j} \left(\frac{\lambda_s}{\lambda_s - \lambda_{a_1}}\right)^{k_s} \prod_{m=s+1}^{a_1 - 1} \left(\frac{\lambda_m}{\lambda_m - \lambda_{a_1}}\right)^{k_m} \prod_{n=a_1}^{a_2-1} \left(\frac{\lambda_n}{\lambda_n - \lambda_{a_2}}\right)^{k_n}\\
&\cdots \prod_{r=a_l - 1}^{j-1} \left(\frac{\lambda_r}{\lambda_r - \lambda_{a_j}}\right)^{k_r} \sum_{q=0}^{k_s - 1} \frac{((\lambda_s - \lambda_{a_1})x)^q}{q!},\\
&\text{for } \geq 0
\end{aligned} \tag{78}$$

6.7.3 Specific CDF of hypoexponential distribution

Equations 77 and 78 refers to the general CDF of the hypoexponential distribution where the size of the vector parameters can be of any size [4]. In the Markov chain models described in figures 11 and 12 the parameter vectors of the hypoexponential distribution are of size two, and in fact, for any possible version of the investigated Markov chain model the vectors can only be of size two. This is true since for any dimensions of this Markov chain model there will always be at most two distinct exponential parameters; the parameter for finishing a service (μ) and the parameter for moving forward in the queue ($C\mu$). For the special case of $C = 1$ the hypoexponential distribution will not be used as this is equivalent to an erlang distribution. Therefore, by fixing the sizes of $\vec{r}$

and $\vec{\lambda}$ to 2, the following specific expression for the CDF of the hypoexponential distribution arises, where the derivative is removed:

$$
\begin{aligned}
&P(H < t) = 1 - \left(\prod_{j=1}^{|\vec{r}|} \lambda_j^{r_j} \right) \sum_{k=1}^{|\vec{r}|} \sum_{l=1}^{r_k} \frac{\Psi_{k,l}(-\lambda_k) t^{r_k - l} e^{-\lambda_k t}}{(r_k - l)!(l-1)!} \\
&\textbf{where} \qquad \Psi_{k,l}(t) = \begin{cases} \frac{(-1)^l (l-1)!}{\lambda_2} \left[\frac{1}{t^l} - \frac{1}{(t+\lambda_2)^l} \right], & k = 1 \\ -\frac{1}{t(t+\lambda_1)^{r_1}}, & k = 2 \end{cases} \\
&\textbf{and} \qquad \lambda_0 = 0, r_0 = 1
\end{aligned} \tag{79}
$$

Note here that the only difference between equation 77 and 79 is the Ψ function. The next section proves why the following expression is true:

$$
-\frac{\partial^{l-1}}{\partial t^{l-1}} \left(\prod_{j=0, j \neq k}^{|\vec{r}|} (\lambda_j + t)^{-r_j} \right) = \begin{cases} \frac{(-1)^l (l-1)!}{\lambda_2} \left[\frac{1}{t^l} - \frac{1}{(t+\lambda_2)^l} \right], & k = 1 \\ -\frac{1}{t(t+\lambda_1)^{r_1}}, & k = 2 \end{cases} \tag{80}
$$

6.7.4 Proof of specific hypoexponential distribution 80

This section aims to show that there exists a simplified expression of equation 77 that is specific to the proposed Markov model. Function Ψ is defined using the parameter t and the variables k and l. Given the Markov model, the range of values that k and l can take can be bounded. First of all, from the range of the double summation in equation 77, it can be seen that $k = 1, 2, \dots, |\vec{r}|$. Now, $|\vec{r}|$ represents the size of the parameter vectors that, for the Markov model, will always be 2. That is because, for all the exponentially distributed random variables that are added together to form the new distribution, there only two distinct parameters, thus forming two erlang distributions. Therefore:

$$k = 1, 2$$

By observing equation 77 once more, the range of values that l takes are $l = 1, 2, \dots, r_k$, where r_1 is subject to the individual's position in the queue and $r_2 = 1$. In essence, the hypoexponential distribution will be used with these bounds:

$$
\begin{aligned}
k = 1 \qquad &\Rightarrow \qquad l = 1, 2, \dots, r_1 \\
k = 2 \qquad &\Rightarrow \qquad l = 1
\end{aligned} \tag{81}
$$

Thus the left hand side of equation 80 needs only to be defined for these bounds. The specific hypoexponential distribution investigated here is of the form $Hypo((r_1, 1)(\lambda_1, \lambda_2))$. Note the initial conditions $\lambda_0 = 0, r_0 = 1$ defined in equation 77 also hold here. Thus the proof is split into two parts, for $k = 1$ and $k = 2$.

- $k = 2, l = 1$

$$\begin{aligned}
LHS &= -\frac{\partial^{1-1}}{\partial t^{1-1}} \left(\prod_{j=0, j\neq 2}^{2} (\lambda_j + t)^{-r_j} \right) \\
&= -\left((\lambda_0 + t)^{-r_0} \times (\lambda_1 + t)^{-r_1} \right) \\
&= -\left(t^{-1} \times (\lambda_1 + t)^{-r_1} \right) \\
&= -\frac{1}{t(t+\lambda_1)^{r_1}}
\end{aligned}$$

□

- $k = 1, l = 1, \ldots, r_1$

$$\begin{aligned}
LHS &= -\frac{\partial^{l-1}}{\partial t^{l-1}} \left(\prod_{j=0, j\neq 1}^{2} (\lambda_j + t)^{-r_j} \right) \\
&= -\frac{\partial^{l-1}}{\partial t^{l-1}} \left((\lambda_o + t)^{-r_0} \times (\lambda_2 + t)^{-r_2} \right) \\
&= -\frac{\partial^{l-1}}{\partial t^{l-1}} \left(\frac{1}{t(t+\lambda_2)} \right)
\end{aligned}$$

In essence, it only remains to show that $-\frac{\partial^{l-1}}{\partial t^{l-1}} \left(\frac{1}{t(t+\lambda_2)} \right) = \frac{(-1)^l (l-1)!}{\lambda_2} \left[\frac{1}{t^l} - \frac{1}{(t+\lambda_2)^l} \right]$.

Proof by Induction:

1. Base case ($l = 1$):

$$\begin{aligned}
LHS &= -\frac{\partial^{1-1}}{\partial t^{1-1}} \left(\frac{1}{t(t+\lambda_2)} \right) = -\frac{1}{t(t+\lambda_2)} \\
RHS &= \frac{(-1)^1 (1-1)!}{\lambda_2} \left[\frac{1}{t^1} - \frac{1}{(t+\lambda_2)^1} \right] \\
&= -\frac{t+\lambda_2 - t}{\lambda_2 t(t+\lambda_2)} \\
&= -\frac{1}{t(t+\lambda_2)} \\
LHS &= RHS
\end{aligned}$$

2. Assume true for $l = x$:

$$-\frac{\partial^{x-1}}{\partial t^{x-1}} \left(\frac{1}{t(t+\lambda_2)} \right) = \frac{(-1)^x (x-1)!}{\lambda_2} \left[\frac{1}{t^x} - \frac{1}{(t+\lambda_2)^x} \right]$$

3. Prove true for $l = x + 1$: $\left(\frac{\partial^x}{\partial t^x} \left(\frac{-1}{t(t+\lambda_2)} \right) = \frac{(-1)^{x+1}(x)!}{\lambda_2} \left[\frac{1}{t^{x+1}} - \frac{1}{(t+\lambda_2)^{x+1}} \right] \right)$

$$\begin{aligned}
LHS &= \frac{\partial}{\partial t} \left[\frac{\partial^{x-1}}{\partial t^{x-1}} \left(\frac{-1}{t(t+\lambda_2)} \right) \right] \\
&= \frac{\partial}{\partial t} \left[\frac{(-1)^x (x-1)!}{\lambda_2} \left(\frac{1}{t^x} - \frac{1}{(t+\lambda_2)^x} \right) \right] \\
&= \frac{(-1)^x (x-1)!}{\lambda_2} \left(\frac{(-x)}{t^{x+1}} - \frac{(-x)}{(t+\lambda_2)^x} \right) \\
&= \frac{(-1)^x (x-1)!(-x)}{\lambda_2} \left(\frac{1}{t^{x+1}} - \frac{1}{(t+\lambda_2)^x} \right) \\
&= \frac{(-1)^{x+1} (x)!}{\lambda_2} \left(\frac{1}{t^{x+1}} - \frac{1}{(t+\lambda_2)^x} \right) \\
&= RHS
\end{aligned}$$

□

6.7.5 Proportion within target for class 1 and class 2 individuals

Given the two CDFs of the Erlang and Hypoexponential distributions a new function has to be defined to decide which one to use among the two. Based on the state of the model, there can be three scenarios when an individual arrives.

1. There is a free server and the individual does not have to wait

$$X_{(u,v)} \sim Erlang(1, \mu)$$

2. The individual arrives at a queue at the n^{th} position and the model has $C > 1$ servers

$$X_{(u,v)} \sim Hypo((n, 1), (C\mu, \mu))$$

3. The individual arrives at a queue at the n^{th} position and the model has $C = 1$ servers

$$X_{(u,v)} \sim Erlang(n + 1, \mu)$$

Note here that for the first case $Erlang(1, \mu)$ is equivalent to $Exp(\mu)$. Let us define $X^{(1)}_{(u,v)}$ the distribution of class 1 individuals and $X^{(2)}_{(u,v)}$ the distribution of class 2 individuals, when arriving at state (u, v) of the model.

$$X^{(1)}_{(u,v)} \sim \begin{cases} \textbf{Erlang}(v, \mu), & \textbf{if } C = 1 \textbf{ and } v > 1 \\ \textbf{Hypo}\left(\vec{r} = (v - C, 1), \vec{\lambda} = (C\mu, \mu)\right), & \textbf{if } C > 1 \textbf{ and } v > C \\ \textbf{Erlang}(1, \mu), & \textbf{if } v \leq C \end{cases} \tag{82}$$

$$X^{(2)}_{(u,v)} \sim \begin{cases} \textbf{Erlang}(\min(v, T), \mu), & \textbf{if } C = 1 \textbf{ and } v, T > 1 \\ \textbf{Hypo}\left(\vec{r} = (\min(v, T) - C, 1), \vec{\lambda} = (C\mu, \mu)\right), & \textbf{if } C > 1 \textbf{ and } v, T > C \\ \textbf{Erlang}(1, \mu), & \textbf{if } v \leq C \textbf{ or } T \leq C \end{cases} \tag{83}$$

Equations 74 and 79 can now be used. Therefore, the probability that an individual arriving at a specific state is within a given time target t is given by the following formulas:

$$P(X^{(1)}_{(u,v)} < t) = \begin{cases} 1 - \sum_{i=0}^{v-1} \frac{1}{i!} e^{-\mu t} (\mu t)^i, & \textbf{if } C = 1 \textbf{ and } v > 1 \\ \\ 1 - (\mu C)^{v-C} \mu \sum_{k=1}^{|\vec{r}|} \sum_{l=1}^{r_k} \frac{\Psi_{k,l}(-\lambda_k) t^{r_k - l} e^{-\lambda_k t}}{(r_k - l)!(l-1)!}, & \textbf{if } C > 1 \textbf{ and } v > C \\ \textbf{where } \vec{r} = (v - C, 1) \textbf{ and } \vec{\lambda} = (C\mu, \mu) & \\ \\ 1 - e^{-\mu t}, & \textbf{if } v \leq C \end{cases} \tag{84}$$

$$P(X^{(2)}_{(u,v)} < t) = \begin{cases} 1 - \sum_{i=0}^{\min(v,T)-1} \frac{1}{i!} e^{-\mu t} (\mu t)^i, & \textbf{if } C = 1 \textbf{ and } v, T > 1 \\ \\ 1 - (C\mu)^{\min(v,T)-C} \mu \sum_{k=1}^{|\vec{r}|} \sum_{l=1}^{r_k} \frac{\Psi_{k,l}(-\lambda_k) t^{r_k - l} e^{-\lambda_k t}}{(r_k - l)!(l-1)!}, & \textbf{if } C > 1 \textbf{ and } v, T > C \\ \textbf{where } \vec{r} = (\min(v,T) - C, 1) \textbf{ and } \vec{\lambda} = (C\mu, \mu) & \\ \\ 1 - e^{-\mu t}, & \textbf{if } v \leq C \textbf{ or } T \leq C \end{cases} \tag{85}$$

In addition the set of accepting states for class 1 ($S_A^{(1)}$) and class 2 ($S_A^{(2)}$) individuals defined in (35) and (39) are also needed here. Note here that, S denotes the set of all states of the Markov chain model.

$$S_A^{(1)} = \{(u,v) \in S \mid v < N\}$$
$$S_A^{(2)} = \begin{cases} \{(u,v) \in S \mid u < M\}, & \textbf{if } T \leq N \\ \{(u,v) \in S \mid v < N\}, & \textbf{otherwise} \end{cases}$$

Having defined everything, it only remains to use a formula similar to the ones of equations 38, 42 and 71. The following formula uses the state probability vector π to get the weighted average of the probability below target of all states in the Markov model.

$$P(X^{(1)} < t) = \frac{\sum_{(u,v) \in S_A^{(1)}} P(X^{(1)}_{u,v} < t) \pi_{u,v}}{\sum_{(u,v) \in S_A^{(1)}} \pi_{u,v}} \tag{86}$$

$$P(X^{(2)} < t) = \frac{\sum_{(u,v) \in S_A^{(2)}} P(X^{(2)}_{u,v} < t) \pi_{u,v}}{\sum_{(u,v) \in S_A^{(2)}} \pi_{u,v}} \tag{87}$$

6.7.6 Overall proportion within target

The overall proportion of individuals for both class 1 and class 2 individuals is given by the equivalent formula of equations (53) and (54). The following formula uses the probability of lost individuals from both classes to get the weighted sum of the two already existing probabilities.

$$P(L'_1) = \sum_{(u,v)\in S_A^{(1)}} \pi(u,v), \qquad P(L'_2) = \sum_{(u,v)\in S_A^{(2)}} \pi(u,v)$$

$$P(X < t) = \frac{\lambda_1 P(L'_1)}{\lambda_2 P(L'_2) + \lambda_1 P(L'_1)} P(X^{(1)} < t) + \frac{\lambda_2 P(L'_2)}{\lambda_2 P(L'_2) + \lambda_1 P(L'_1)} P(X^{(2)} < t) \tag{88}$$

7 Markov chain VS Simulation

7.1 Example model

Consider the Markov chain paradigm in figure 13. The illustrated model represents the unrealistically small system with a system capacity of five and a buffer capacity of three. The hospital in this particular example also has four servers and a threshold of three; meaning that every ambulance that arrives in a time that there are three or more individuals in the hospital, will proceed to the buffer centre.

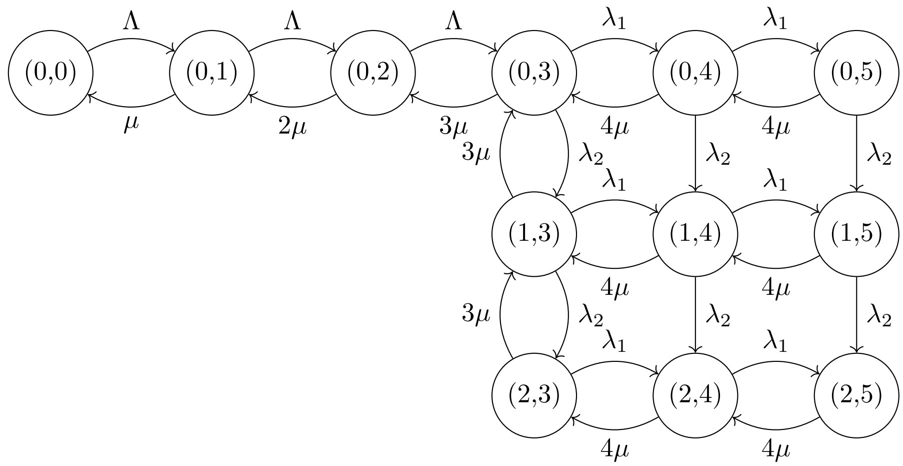


Figure 13: Markov chains: number of servers=4

In addition to the Markov chain model a simulation model has also been built based on the same parameters. Comparing the results of the Markov model and the equivalent simulation model the resultant plots arose.

The heatmaps in figure 14 represent the state probabilities for the Markov chain model, the simulation model and the difference between the two. Each pixel of the heatmap corresponds to the equivalent state of figure 13 and represents the probability of being at that state in any particular moment of time.

It can be observed that both Markov chain and simulation models' state probabilities vary from 5% to 25% and that states $(0, 1)$ and $(0, 2)$ are the most visited ones. Looking at the differences' heatmap, one may identify that the differences between the two are minimal.

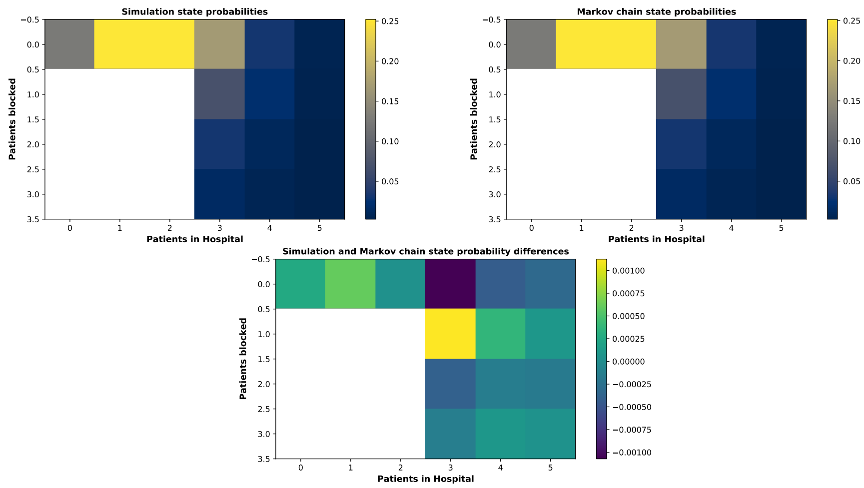
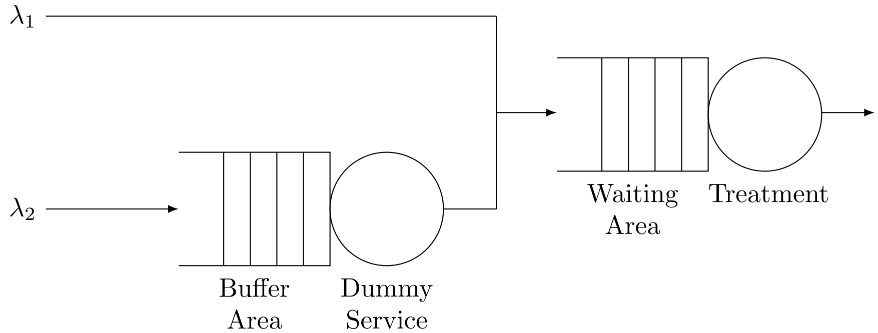
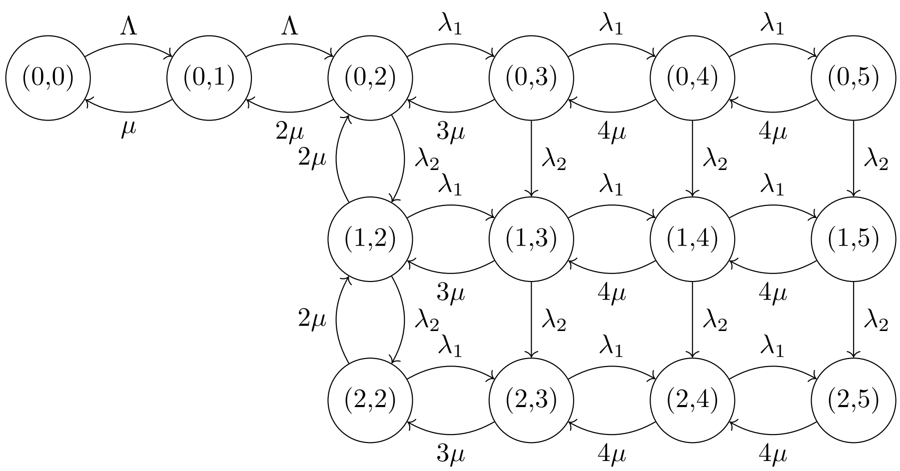


Figure 14: Heatmaps of Simulation, Markov chains and differences of the two

8 Figures that might be useful





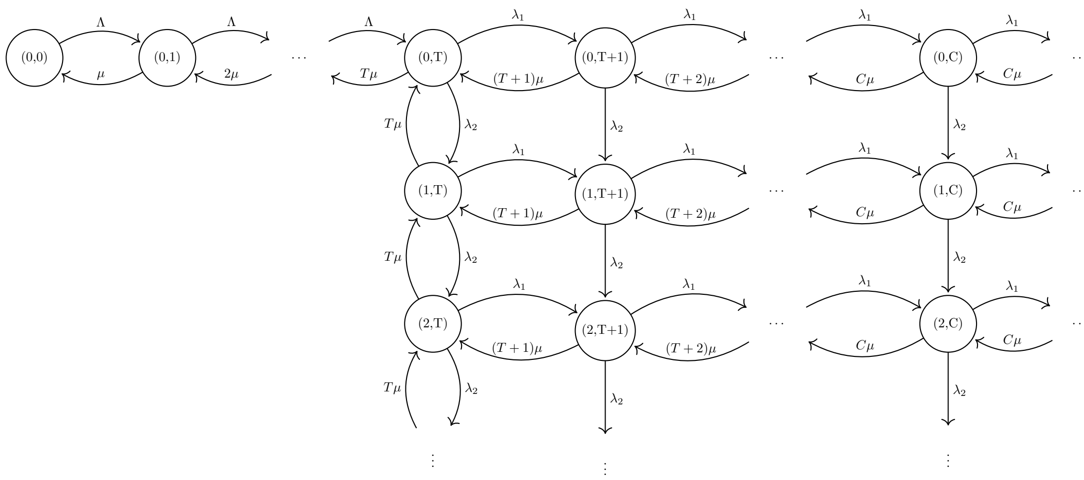


Figure 15: Markov chains

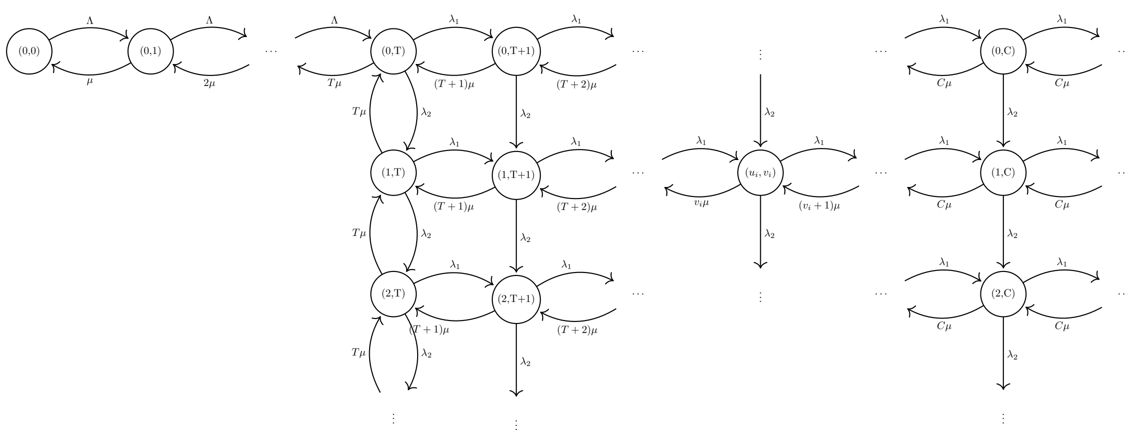


Figure 16: Markov chains

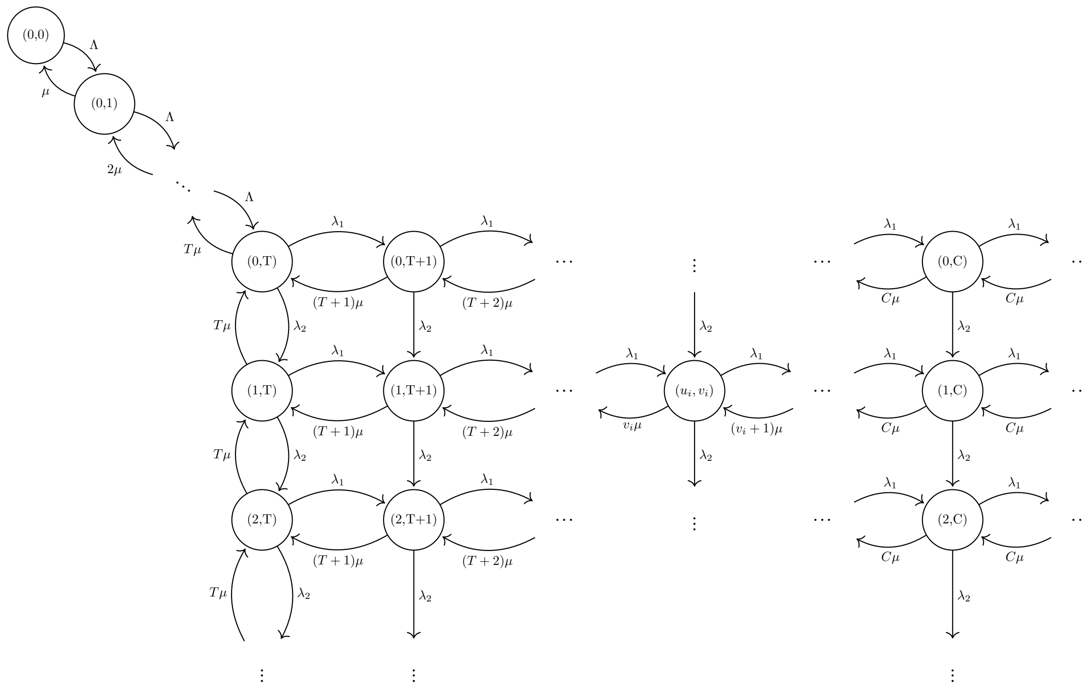


Figure 17: Markov chains

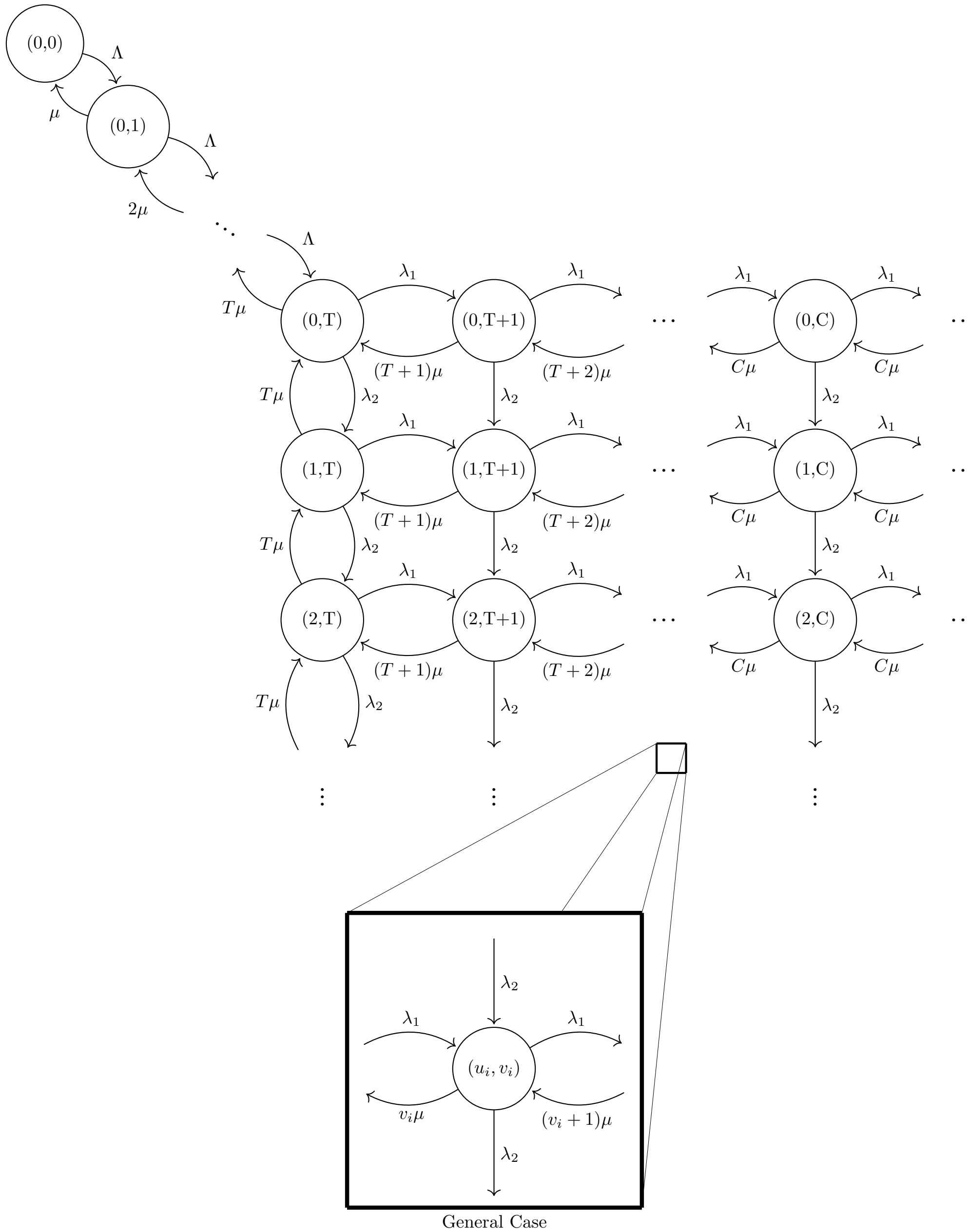


Figure 18: Markov chain

9 Formulas

$$\hat{c}_i \in \{1, 2, \ldots, C_i\}$$

$$\rho_i = \frac{p_i \Lambda + \lambda_i^o}{\hat{c}_i \mu_i}$$

$$(W_q)_i = \frac{1}{\hat{c}_i \mu_i} \frac{(\hat{c}_i \rho_i)^{\hat{c}_i}}{\hat{c}_i!(1-\rho_i)^2} (P_0)_i$$

$$(P_0)_i = \frac{1}{\sum_{n=0}^{\hat{c}_i - 1} \left[\frac{(\hat{c}_i \rho_i)^n}{n!} \right] + \frac{(\hat{c}_i \rho_i)^{\hat{c}_i}}{\hat{c}_i!(1-\rho_i)}}$$

$$P(W_q > T) = \frac{(\frac{\lambda}{\mu})^c P_0}{c!(1 - \frac{\lambda}{c\mu})} (e^{-(c\mu - \lambda)T})$$

References

[1] Mohamed Akkouchi. "On the convolution of exponential distributions". In: *J. Chungcheong Math. Soc* 21.4 (2008), pp. 501–510.

[2] Andrei Z Broder. "Generating random spanning trees". In: *FOCS*. Vol. 89. Citeseer. 1989, pp. 442–447.

[3] Seth Chaiken and Daniel J Kleitman. "Matrix tree theorems". In: *Journal of combinatorial theory, Series A* 24.3 (1978), pp. 377–381.

[4] Stefano Favaro and Stephen G Walker. "On the distribution of sums of independent exponential random variables via Wilks' integral representation". In: *Acta applicandae mathematicae* 109.3 (2010), pp. 1035–1042.

[5] Benjamin Legros and Oualid Jouini. "A linear algebraic approach for the computation of sums of Erlang random variables". In: *Applied Mathematical Modelling* 39.16 (2015), pp. 4971–4977.

[6] K Raghunandanan. "A note on the distribution of sojourn time in a Markov chain". In: *Trabajos de estadistica y de investigacion operativa* 20.2-3 (1969), pp. 199–210.

[7] Khaled Smaili, Therrar Kadri, and Seifedine Kadry. "Hypoexponential distribution with different parameters". In: (2013).