<table>
<thead>
<tr>
<th>Sr. No.</th>
<th>Title</th>
<th>Page No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>Introduction</td>
<td>45</td>
</tr>
<tr>
<td>3.2</td>
<td>System Description</td>
<td>49</td>
</tr>
<tr>
<td>3.3</td>
<td>Combined Allocation of the Queue Lengths</td>
<td>51</td>
</tr>
<tr>
<td>3.4</td>
<td>Stationary Mean Waiting Time</td>
<td>62</td>
</tr>
<tr>
<td>3.5</td>
<td>Approximation and Splitting in Queue</td>
<td>63</td>
</tr>
<tr>
<td>3.6</td>
<td>Case for Merging</td>
<td>73</td>
</tr>
<tr>
<td>3.7</td>
<td>Reduction of Variability in Split Merge System</td>
<td>75</td>
</tr>
</tbody>
</table>
3.1 INTRODUCTION

A novel two-station single-processing tandem queuing model with task jamming, splitting, and feedback is considered. Further two different cases are also discussing in which a limited buffering at every station and in other case mainly the middle finite buffering were considered. After completing processing at every station, a task may leave from the system, and then join in to the next station, consequently move to the subsequent stations or return to its own station. The solid-state joint distribution for the related queue lengths which will go ahead in to the behavior of queue lengths, possibilities to reach at every station, mean coming up time for the system may be hectic and also being proper maintained via models. Their algebraic illustration is provided.

The structure of simple multi-station Markovian tandem queue explains that: Tasks enter in to a buffer (having infinite, finite or a zero capacity; where capacity of zero tasks means that it does not wait or sometimes lost from the system) ahead of station first by means of a Poisson technique. Mainly the tasks are arranging, according to the coming of their arrival, from a single-server capability for exponential division. Then they are proceeding in to a second place in which supply equally. Tandem queues among fixed barriers also studied widely in the earlier period century. They are too much helpful in modeling and also study of some distinct occurrence systems.

If buffer size is finite in tandem queue, there must be chance in which a system becomes uncreative. Analysis which does not including barriers through blockings going on track in 1971 by Meswak and Fradin. Mainly there are three types of blockings are normally studied: first type blocking is after-service blocking, second type blocking is before-service blocking, and third type blocking is repetitive-service blocking. Sometimes if a task goes from a first station to second station and there is no space in second station, then first station is being blocked. Now the service in that station may stop until a space becomes available for the tasks to move in the second station. That’s why it is known as after service blocking. Conversely, before service blocking occurs when there is a first task decides its destination, which is second station, before starting their service and there is no space in second station, their tune-up discontinue till a gap
turn into vacant within second station. In conclusion, cyclic-tune-up occurs if responsibilities finish its tune-up in first location then requires moving to second location however there is no space in second station, hence the first task carry on in their rotation.

Comparisons of these types of blocking are explained in Neural and Peres (1993). Various queuing system containing above three kinds of blockings in practice is also mentioned in literature review. Service effects in queuing models and queuing networking having at the back examine blocking clarify in Kwilt. Geometrical estimation of environment through blocking & continues effort is explained in Góhel-Corasl (2002). Blocking-after-service within system which focuses on main concern carry out invention from estimate depends in the rule of highest entropy, particularly for a finite capacity opening in queuing network techniques is also discussed in Kouvtsos and Awan (2003). It also explains several techniques in network queues having blocking which propose a final explanation to a tandem queue having blocking.

It is possible for a pair of blocking types to be equivalent. In 1994 Perros, for instance, prove that before service blocking while the position before the server may be occupied is equivalent to after service blocking (which is mutually have the same matrix rate underlying) if there exists just three stations. Presentation valuation for an untie queuing system with past-examine-blocking explained in Lee (1985). A multi situation tandem line, having after services jamming comprise an exact solution as well as an exact in active distribution, described in Ahuildiz and Crand (1991).

Generally jamming representations discussed in above all literatures is past-examine blocking. Here during 1976 by Konheim and Reiser the utilization in computer programming is mentioned. In this they reflect on a scheme in which a limited second buffer size is M. Poisson distribution at station 1 (by constraint $\lambda_1$), is understood. At station 2; the arrival constraint is $\lambda_2$ and their respective service made via single-server for every position with constraint $\mu_1$ and $\mu_2$. Required circumstances for the steadiness in this system are specified.

In 1968 Neuts focuses a multi-station tandem queue for fixed intermediate buffer. Here universal examine distribution is given for the initial station; at the same time as in the next station the distribution is not given. He also gives a principle for the case in which time is dependent with
highlighting in the fixed process. He also mentioned about the demanding period. In 1998 Browning reflects about Dependent and independent blocking tandem queues. In 2000 Grassman and Drekić; A multi-station tandem queue having blocking and driving logical solution for a combined probability distribution in symmetry by a universal eigenvalues is considered. Many remaining blocking models considered may also be establish in: in 1990 Tsiotras, in 1991 Cassandra, in 1993 Kook and Serfozo (stopover times), in 2000 Roxes and Perrosand also in Nakade, in 1991 Suk and Cassandras, In 2000 Schmidt and Jackman, and in 2002 Gómez-Corral.

In 1988 Warland states that apart from only in some cases, there does not exist solution for a product-form in tandem queues having blocking. Here the invention-form indicates that every station may be dealt with separately. In 1989; Akyildiz, calculated result approximations for queuing models with many-servers and jamming. For product-form queuing models with blocking; a convolution algorithm is given in 1998 by Balsamo and Clo’. They check the different kinds of blocking explained above. Also in below particular constraints, exists a convolution algorithm in models of multi-form stopped network queues with blocking, to estimate queue size division and usual appearance parameters. Mathematical techniques continue staying for a preferred in order to determining exact solution.

Tandem lines by blocking generally formulated via approximation. In 1980 Neuts; derived the formula for steadiness of a finite multi-server tandem queue containing blocking. Normally the Poisson arrival rate may not beat a significant rate which they depends, in a difficult mode, in the checkup rate. They propose a model for getting main features for a queue. An estimation technique, in which precise assumptions are there, tandem queues having jamming and splitting is offered in Gizmo & Konheim 1981. Especially proposed for multi-situation tandem model, focus for decay their mechanism in roughly free MI/M/1 models then highlight fixed allocation for the queuing model. They are doing by using a rounded grouping of inactive allocation throughout every stage.

Here remark “splitting” is frequently given in the paper for average dissimilar routing (for a departure time and at arrival point). In 2000; Girish and Hu both developed a second order estimate for only one server queue having “merging, splitting and opinion”. They consider a case
which dealing for the merging of departure or arrival technique into many unusual processes. Every one of the above normally is depart from system. They derived stable-situation conditions for divide techniques. Also their thought is related to our normal subsequent to ending of service by a model. Here, a mission is splitting in two different tasks later than end of a service and normal splitting. This is totally dissimilar by a splitting for the arrival or departure techniques.

Not much has been done for splitting in sense contain in the above paper, mainly when situations is random. Some studies be mentioned as : In 1989 Durham, reflects on the splitting feature into a mix explanation. In 1997 ; Montazer-Haghighi and True blood explains a hectic phase for a comparable multi-processor for which they are task-splitting . In 1998 ; Montazer-Haghighi explains a corresponding many-processor structure which are task-splitting and feedback. Refer references also which are listed in these papers.

The study of QBD techniques by using the geometric technique has been widely spread by Neutsin in 1981. The matrix R plays a main role in universal theory which discuss later in the model. It is being normally designed using the logarithmic decrease model of Latouche and Ramaswami In1993, which contains repeated convergence techniques. Based only on normal explanation, they mention that it has high-quality mathematical constancy individuality and so many iterations are not very large. Though, in the above article, selected homogeneous matrix; go also for a higher order matrix approach in some cases (if necessary) and give models to find division for the combined sharing for the queue lengths which will guide the events appearing in the queue models, average coming up time, possibility of every station and a system being very active and being a proper.

### 3.2 SYSTEM DESCRIPTION

The system contains each time two service stations having a similar single-server in both. The procedure is explained in Figure 3.1. Tasks coming up into an infinite outside resource related a Poisson distribution by parameter $\lambda$. There must be a buffer before every server. When the performance of task is finish after the starting station, it continue into the following way: put down the structure with possibility $q_1$, immediately come back to its individual buffer at last
waiting line through probability $p_1^1$, entering second buffer station through probability $p_1^2$, or go away for come apart through possibility $p_1^{S_1}$. Here it is remarkable that $q_1 + p_1^1 + p_1^2 + p_1^{S_1} = 1$

Once completed a service during second station task must be required to continue in the following routes: put down the structure with possibility $q_2$, immediately come back to its individual buffer at last waiting line through possibility $p_2^2$, come back to first station via first buffer with possibility $p_2^1$, or go away for splitting with possibility $p_2^{S_2}$. At this juncture $q_2 + p_2^1 + p_2^2 + p_2^{S_2} = 1$. At first station, leave out is impossible, i.e. the outer arrival must continue through first station only; straight arrival at second station is not allowable.
Here as the case of splitting, a task splits into two independent subtasks. Thus in first station, if task divides into two, first is attend the line at last in buffer 1 consequently the second may also instantly attend line at end in buffer 1 with possibility $p_{s_1}^1$, may be go out from the system through possibility $q_{s_1}$ or may be go to second station with possibility $p_{s_1}^2$, at this time $p_{s_1}^1 + q_{s_1} + p_{s_1}^2 = 1$. Similarly in second station, if task divides in to two, first determination come back at end of procession inside second buffer, then additional can instantly move towards finish procession in second buffer through possibility $p_{s_2}^2$, might be leave the structure through possibility $q_{s_2}$, or attend first situation through probability $p_{s_2}^1$, here $p_{s_2}^2 + q_{s_2} + p_{s_2}^1 = 1$.

We consider following two cases for a buffer capacities.

(1) Both buffers are finite with capacities $M_1 - 1$ and $M_2 - 1$ for buffers 1 and 2, respectively.

(2) Buffer 1 has infinite capacity while buffer 2 is finite with capacity $M_2 - 1$.

In first case, when a task is finished its service in first station and then it is continue to the second station but second buffer is not empty, first station will be blocked. Then served task must wait until the second station becomes available. We indicate this system by $(m_1, M_2 + 1)$, means that second station is filled with any one task individual served and $m_1$ tasks in first station, from which one has finished its service and is move on to second station immediately when second station is accessible. When second station is available, the served-coming up task will go onward and the condition of the system will become $(m_1 - 1, M_2)$. When first station is blocked but is empty, outer arrivals may be present. Same position happens for second station.

When a supply-task in second station is to carry on to first station and the station is not empty, second station will be blocked and the served-task must wait until the first station becomes
available. We indicate this condition of the system by \((M_1+1, m_2)\), which means that first station is full and \(m_2\) tasks are there in second station from which one has finished its service and is move on to second station immediately when first station is accessible. When first station is available, the served-coming up task will move without delay and the condition of the system will become \((M_1, m_2 - 1)\). It is clear that while second station is blocked, the external arriving tasks are lost.

If first station is jammed and a splitting occurs in second station, then any one subtask must go out from the system or move to first station. In the same way, if second station is jammed and a splitting occurs in first station, then any one subtasks must go out from the system or go to second station.

### 3.3 Combined Allocation of the Queue Lengths

Let the arbitrary variable \(\xi_i (t), i = 1, 2\), denotes the number of tasks in station \(i\), at time \(t\), including the one being processed. Also let the joint probability of \(m_1\) tasks in station 1, including the one being processed, and \(m_2\), \(0 \leq m_2 \leq M_2\), tasks in second station, counting the one being processed, at time \(t\) be denoted by \(P_{m_1, m_2}(t)\), i.e. \(P_{m_1, m_2}(t) = P\{\xi_1(t) = m_1, \xi_2(t) = m_2\}\). Here it is noted that the values of \(m_1\) for Model 1 and Model 2 are \(0 \leq m_1 \leq M_1\), and \(m_1 \geq 0\), respectively.

Further, let \(P_{m_1, m_2} = \lim_{t \to \infty} P_{m_1, m_2}(t)\) be the steady-state possibility of having \(m_1\) tasks in first station and \(m_2\) tasks in second station, i.e. \(m_1 + m_2\) tasks in the system. Our objective is to obtain the steady-state joint probability distribution of \(\xi_i(t), i = 1, 2\), i.e. \(P_{m_1, m_2}\).

#### 3.3.1 Model 1: Both Buffers are Finite

In order to study about structure for steadiness of Model 1, choose routine case when \(3 \leq M_1 \leq \infty\) and \(3 \leq M_2 \leq \infty\). Here following eight cases are there:

\((M_1, M_2) = (1, 1), (1, 3), (1, \geq 3), (2, 1), (2, 2), (2, \geq 3), (\geq 3, 1)\) and \((\geq 3, 2)\).

Above possibilities definitely deal through several changes for the routine case.
System for above all cases may be obtained from (3.3.1), below, by changing it accordingly. On behalf of example, for \( M_1 = M_2 = 1 \) the expression is as under:

\[
\lambda P_{0,0} = q_1 \mu_1 P_{1,0} + q_2 \mu_2 P_{0,1}
\]

\[
(q_1 + p_1^2 + p_1^{S_1} p_2^{S_2}) \mu_1 P_{1,0} = \lambda P_{0,0} + p_1^1 \mu_2 P_{0,1} + q_2 \mu_2 P_{1,1} + q_1 \mu_1 P_{2,1}
\]

\[
[\lambda + (q_2 + p_2^1 + p_2^{S_2} p_1^{S_1}) \mu_2] P_{0,1} = q_1 \mu_1 P_{1,1} + p_1^2 \mu_1 P_{1,0} + q_2 \mu_2 P_{1,2}
\]

\[
[(q_1 + p_1^2) \mu_1 + (q_2 + p_2^1) \mu_2] P_{1,1} = (\lambda + p_2^{S_2} p_1^{S_2} \mu_2) P_{0,1} + p_1^{S_2} p_2^{S_2} \mu_1 P_{1,0}
\]

\[
q_2 \mu_2 P_{1,2} = p_1^2 \mu_1 P_{1,1}
\]

\[
q_1 \mu_1 P_{2,1} = p_2^1 \mu_2 P_{1,1}
\]

\[
\sum_{m_1=0}^{1} \sum_{m_2=0}^{1} P_{m_1 m_2} + P_{2,1} + P_{1,2} = 1
\]

General Case: \( 3 \leq M_1, M_2 < \infty \) for Model 1 (both buffer finite).

System of equations for this possibility is given as under

\[
Q X = 0 \tag{3.3.1}
\]

Where \( Q \) and \( X \) is considering as under:

Let \( X = (X_0; X_1; X_2; \ldots; X_{M_1}; X_{M_1+1}) \) be a column vector, where each of \( X_0, X_1, X_2, \ldots, X_{M_1} \) and \( X_{M_1+1} \) are as under

\[
X_0 = (x_{0,0}, x_{0,1}, x_{0,2}, \ldots, x_{0,M_2+1})
\]

\[
X_1 = (x_{1,0}, x_{1,1}, x_{1,2}, \ldots, x_{1,M_2+1})
\]

\[
X_2 = (x_{2,0}, x_{2,1}, x_{2,2}, \ldots, x_{2,M_2+1})
\]
The number of elements in $X$ is: $q = M_1 M_2 + 2(M_1 + M_2) + 1$.

Also $\vec{X}$ be a column vector as $X$ except $X_0$ is without the first element $X_{0,0}$.

The size of $\vec{X}$ is $\bar{q} = M_1 M_2 + 2(M_1 + M_2)$.

Further, let the matrix $Q$ be an $(M_1 + 2) \times (M_2 + 2)$ tri-diagonal block matrix as:

$$Q = \begin{bmatrix}
B_1 & C_1 & & \\
A_1 & B & C & \\
& A & B & C \\
& & \ddots & \ddots \\
& & & A & B & C \\
& & & & A & B_2 & C_2 \\
& & & & & A_2 & B_3
\end{bmatrix}$$

Here the elements is described as:

Consider $A$ as $(M_2 + 2) \times (M_2 + 2)$ associate matrix among components are defined as

$$a(j,j+1) = p_2^1 \mu_2 , \ j = 1,2,\ldots,M_2.$$  

$$a(j,j) = \begin{cases} 
\lambda + p_1^{S_1} p_2^{1,1} \mu_1, & j = 1 \\
\lambda + p_1^{S_1} p_2^{1,1} \mu_1 + p_2^{S_2} p_2^{1,1} \mu_2, & j = 2,3,\ldots,M_2 + 1 \\
\lambda, & j = M_2 + 2 
\end{cases} \tag{3.3.2}$$

And remaining elements are zero.

$A_1$ is an $(M_2 + 2) \times (M_2 + 1)$ associate matrix with elements $a_1(i,j)$ as under:

$$a_1(i,i + 1) = p_2^1 \mu_2 ; \ i = 1,2,\ldots,M_2.$$
\[
\alpha_1(i,i) = \begin{cases} 
\lambda, & i = 1 \\
\lambda + p_2 S_2 p_{S_2}^1 \mu_2, & i = 2,3, \ldots, M_2 + 1
\end{cases}
\]  

(3.3.3)

and remaining elements are zero.

Here \( A_2 \) is describe as \( M_2 \times (M_2 + 2) \) sub matrix with elements \( a_2(i,j) \) as:

\[
a_2(i, i + 1) = p_2^1 \mu_2, \quad i = 1,2, \ldots, M_2.
\]

and remaining elements including entire last column with all elements zero.

\( B \) is an \( (M_2 + 2) \times (M_2 + 2) \) sub matrix with elements \( b(i,j) \) as:

\[
\begin{align*}
E(C, C + 1) &= \langle \lambda, C = 1 \\
E(C + 2, C + 1) &= \langle \lambda + q_2, C = 1,2, \ldots, M_2 \\
E(C, C) &= \langle -K, C = 1 \\
E(C + 1, C) &= \langle - \lambda, C = 1 \\
E(C + 2, C + 1) &= \langle - \lambda + q_2, C = 1,2, \ldots, M_2 \\
E(C + 3, C + 2) &= \langle - \lambda + (q_2 + p_2^1 + p_2^2 p_2 S_2^1) \mu_1, C = 1 \\
E(C + 4, C + 3) &= \langle - \lambda + (q_2 + p_2^1 + p_2^2 p_2 S_2^1) \mu_2, C = 2,3, \ldots, M_2 \\
E(C + 5, C + 4) &= \langle - \lambda + (q_2 + p_2^1 + p_2^2 p_2 S_2^1) \mu_2, C = M_2 + 1 \\
E(C + 6, C + 5) &= \langle - \lambda + (q_2 + p_2^1 + p_2^2 p_2 S_2^1) \mu_2, C = M_2 + 2
\end{align*}
\]

(3.3.4)

and remaining elements are zero.

\( B_1 \) is an \( (M_2 + 1) \times (M_2 + 1) \) sub matrix with elements \( b(i,j) \) as:

\[
\begin{align*}
\alpha_1(i,i) &= \begin{cases} 
\lambda, & i = 1 \\
\lambda + p_2 S_2 p_{S_2}^1 \mu_1, & i = 1 \\
\lambda + p_2 S_2 p_{S_2}^1 \mu_2, & i = 2,3, \ldots, M_2 + 1
\end{cases}
\]  

(3.3.5)
Remaining elements are zero.

$B_2$ is $(M_2 + 2) \times (M_2 + 2)$ sub matrix through elements $b_2(i,j)$ as

$$b_2(i, i + 1) = q_2 \mu_2, \quad i = 1, 2, ..., M_2.$$  

$$b_2(i, i) = \begin{cases} 
-\left( q_1 + p_1^2 + p^S_1 p^S_2 \right) \mu_1, & i = 1 \\
-\left[ \left( q_1 + p_1^2 + p^S_1 p^S_2 \right) \mu_1 + (q_2 + p_2^2 + p^S_2 p^S_2) \mu_2 \right], & i = 2, 3, ..., M_2 \\
-\left( q_1 + p_1^2 \right) \mu_1 + (q_2 + p_2^2) \mu_2, & i = M_2 + 1 \\
-q_2 \mu_2, & i = M_2 + 2
\end{cases}$$  

$$b_2(i + 1, i) = \begin{cases} 
p^S_1 p^S_2 \mu_1, & i = 1, \\
p^S_1 p^S_2 \mu_1 + p^S_2 p^S_2 \mu_2, & i = 2, 3, ..., M_2 \\
p^2_1 \mu_1, & i = M_2 + 1
\end{cases}$$

Remaining components are zero.

$B_3$ is an $M_2 \times M_2$ sub matrix with elements:

$$b_3(i, i) = -q_3 \mu_1, \quad i = 1, 2, ..., M_2.$$  

Along with remaining components is zero.

$C$ is $(M_2 + 2) \times (M_2 + 2)$ combine matrix through constituent $c(i,j)$ as:

$$c(i, i) = q_1 \mu_1, \quad i = 1, 2, ..., M_2 + 1$$

$$c(i + 1, i) = p^2_1 \mu_1 \quad i = 1, 2, ..., M_2$$

$$c(i, i + 1) = q_2 \mu_2 \quad i = 1, 2, ..., M_2.$$  

Along with remaining components is zero.

$C_1$ is an $(M_2 + 1) \times (M_2 + 2)$ sub matrix through elements $c_1(i,j)$ as:

$$c_1(i, i) = q_1 \mu_1, \quad i = 1, 2, ..., M_2 + 1$$
\[ c_1(i, i + 1) = q_2 \mu_2, \quad i = M_2 + 1 \]
\[ c_1(i + 1, i) = p_1^2 \mu_1, \quad i = 1, 2, \ldots, M_2 \]  \hspace{1cm} (3.3.7)

all other elements are zero

\( C_2 \) is an \((M_2 + 2) \times M_2\) sub matrix through elements \( c_2(i, j) \) as:

\[ c_2(i, j) = q_1 \mu_1, \quad i = 1, 2, \ldots, M_2, \]

and remaining element becomes zero.

Standardized formula for current structure is:

\[
\sum_{m_2=0}^{M_2} P_{0,m_2} + \sum_{m_1=1}^{M_1} \sum_{m_2=0}^{M_2+1} P_{m_1,m_2} + \sum_{m_2=1}^{M_2} P_{m_1+1,m_2} = 1
\]

**Solution for Model 1 in which both buffers are finite (general case: \( 3 \leq M_1, M_2 < \infty \))**

System (1) is finite dependent system. It can be effectively solved by matrix method even if precise solution is not obtained, access to latest computers permit for algorithm. Algorithm of six-step discussed here, explain how we obtain a solution using mass matrix techniques.

1. Remove first row of \( Q \) then remaining matrix say \( Q_1 \).
2. Select first column of \( Q_1 \) and multiply it by \((-1)\) call the matrix as \( Q_3 \).
3. Remove first column of \( Q_1 \), say new matrix as \( Q_2 \).
4. \( \tilde{X} = Q_2^{-1} Q_3 \).
5. Calculate \( S = 1 + \sum_{m_2=1}^{M_2} x_0 m_2 + \sum_{m_1=1}^{M_1} \sum_{m_2=0}^{M_2+1} x_{m_1,m_2} + \sum_{m_2=1}^{M_2} x_{M_1+1,m_2} \)
6. Calculate \( P_{0,0} = \frac{1}{s}, P_{0,m_2} = \frac{x_{0,m_2}}{s}, 1 \leq m_2 \leq M_2, P_{m_1,m_2} = \frac{x_{m_1,m_2}}{s} \)

\[ 1 \leq m_1 \leq M_1, \quad 0 \leq m_2 \leq M_2 + 1 \quad \text{and} \quad P_{M_1+1,m_2} = \frac{x_{M_1+1,m_2}}{s} ; \quad 1 \leq m_2 \leq M_2. \]

**3.3.2 Model 2 : With Buffer 1 Infinite and Buffer 2 Finite :**
The system of balanced equation is written (15 in all) for the general case $M_1 \geq 3$ and $3 \leq M_2 < \infty$. There are special cases that can be handled by modifying the general case. The matrix equation for this case is given below.

$$X = 0$$

(3.3.8)

Where $X$ is define as under.

Let $X = \langle X_0, X_1, X_2, ..., X_i, ... \rangle$ is lineup-vector, wherever every of $X_0, X_1, X_2, ...$ is a line-vector and can be defined as

$$X_0 = \langle x_{0,0}, x_{0,1}, x_{0,2}, ..., x_{0,m_2} \rangle$$

$$X_i = \langle x_{i,0}, x_{i,1}, x_{i,2}, ..., x_{i,M_2}, x_{i,M_2+1} \rangle, \ i = 1, 2, 3, ...$$

The system represented by equation (3.3.8) is a quasi-birth-and-death (QBD) process and thus a Markov process with insignificant canonical matrix generator of $\bar{Q}$ as :

$$\bar{Q} = \begin{bmatrix}
B_1 & C_1 \\
A_1 & B & C \\
A & B & C & \ldots \\
A & B & C & \ldots \\
& & & \\
& & &
\end{bmatrix},$$

Where $A$, $A_1$, $B$, $C$ and $C_1$ are given by equations (3.3.2), (3.3.3), (3.3.4), (3.3.6) and (3.3.7) respectively. The generator matrix $\bar{Q}$ is assumed to be irreducible. The normalizing equation for this system can be given as :

$$\sum_{m_2=0}^{M_2} P_{0,m_2} + \sum_{m_1=1}^{\infty} \sum_{m_2=1}^{M_2+1} P_{m_1,m_2} = 1$$

**Solution for Model 2: (Buffer 1 Infinite and Buffer 2 Finite)**

To solve equations described in equation (3.3.8) the method of Neuts (1980) can be used. Let $\bar{X}$ be a line vector similar to $X$, Then structure is represented as:
B_1 X_0 + C_1 X_1 = 0

A_1 X_0 + B X_1 + C X_2 = 0

AX_1 + BX_{i+1} + CX_{i+2} = 0, \ i = 1,2,3,… (3.3.9)

Using the matrix geometric method of Neuts', here \( X_i, i = 1,2,3,… \) is represented in form of \( X_1 \) as:

\[ X_i = R^{i-1} X_1, \ i = 1,2,3,… \]

Where \( R \) is defined as matrix \((M_2 + 2) \times (M_2 + 2)\), is smallest result of:

\[ A + BR + CR^2 = 0 \quad (3.3.10) \]

Here, \( R \) is finding by means of rewriting the above expression as

\[ R_{k+1} = -B^{k-1} A - B^{-1} C R_k^2; \ \text{where} \ k = 0, 1, 2, …… (3.3.11) \]

Also fixed \( R_0 = 0 \) and associate vectors \( X_0 \) and \( X_1 \) are accessible using following relation

\[ B_1 X_0 + C_1 X_1 = 0 \]

\[ A_1 X_0 + (B+CR)X_1 = 0 \]

Here it is to be noted that

\[ X_1 + X_2 + X_3 + …… = X_1 + RX_1 + R^2 X_1 + … + R^i X_1 + …… = (I + R + R^2 + … + R^i + …) = (I - R)^{-1} X_1 \]

Taking into consideration this background of the system, the algorithm to solve the system defined in (3.3.8), can be given by following steps.

1. Consider \( k = 0 \) as well as \( R_0 = 0 \)
2. Find the matrix \( R \) (using (3.3.11)) recursively. During the each iteration check if
\[
\max_{i,j} \left| (R_{k+1} - R_k)_{ij} \right| < \varepsilon
\]

Where \( \varepsilon \) is very small positive number. If this condition is not satisfied go to the next iteration, and if the condition is satisfied, subsequently decide \( R = R_{k+1} \).

3. Calculate value of \( X_0 \) using following sub steps.

3.1 Find \( P = -(B + C \cdot R)^{-1} \cdot A_1 \), \hspace{1cm} (3.3.12)

Where \( B \), \( C \) and \( A_1 \) are calculating from (3.3.4), (3.3.6) and (3.3.3) respectively.

3.2 Find \( K = B_1 + C_1 \cdot P \),

Where \( B_1 \), \( C_1 \) and \( P \) are defined in (3.3.5), (3.3.7) and (3.3.12) respectively.

3.3 Find the solution of system \( K X_0 = 0 \),

Where \( X_0 = \langle 1, x_{0,1}, x_{0,2}, \ldots, x_{0,M_2} \rangle \), \hspace{1cm} (3.3.13)

Follow the below sub steps

3.3.1 Rewrite (3.3.13) as

\[
X_0 = \langle 1, \bar{X}_0 \rangle
\]

Where \( \bar{X}_0 = \langle x_{0,1}, x_{0,2}, \ldots, x_{0,M_2} \rangle \).

3.3.2 Remove the first row of \( K \) then say the remaining as \( K_1 \).

3.3.3 Select the starting column of \( K_1 \), after multiplying it by (-1) say modified matrix as \( K_3 \).
3.3.4 Remove starting column in $K_1$ say new matrix $K_2$.

3.3.5 Express $\tilde{X}_0 = K_2^{-1} \cdot K_3$.

3.4 Find $X_1 = P \cdot X_0$.

4 Find $P_{i,j}$ using following steps.

4.1 Write $S = 1 + x_{0,1} + x_{0,2} + \cdots + x_{0,M_2} + [(I - R)^{-1}X_1] \cdot e$, Wherever $e$ is $(M_2+X_2)$ vector having each entry is 1.

4.2 Let $P_0 = \frac{1}{s}X_0, P_1 = \frac{1}{s}X_1$, Wherever $P_{0,0} = \frac{1}{s}x_{0,0}, P_{0,1} = \frac{1}{s}x_{0,1}, \ldots, P_{0,M_2} = \frac{1}{s}x_{0,M_2}$ and

$$P_{1,0} = \frac{1}{s}x_{1,0}, P_{1,1} = \frac{1}{s}x_{1,1}, \ldots, P_{1,M_2} = \frac{1}{s}x_{1,M_2}$$

$$P_{1,M_2+1} = \frac{1}{s}x_{1,M_2+1}.$$ 

4.3 Let $X_i = R_i^{-1} \cdot X_1$, $i = 2,3,\ldots$ and $P_i = \frac{1}{s}X_i$, Wherever $P_{2,0} = \frac{1}{s}x_{2,0}, P_{2,1} = \frac{1}{s}x_{2,1}, \ldots, P_{2,M_2} = \frac{1}{s}x_{2,M_2}$ and

$$P_{2,M_2+1} = \frac{1}{s}x_{2,M_2+1}.$$ 

**Stability for Model 2 (Buffer 1 Infinite and Buffer 2 Finite)**

Define condition under which a unique solution for (3.3.9) or (3.3.8) exists.

The process $\tilde{Q}$ is positive recurrent iff $R$, the solution to the equation (3.3.10) contains every Eigen values inside the unit disk. At this juncture $G = A + B + C$ is not decreasable, then
SP_{R} < 1 \text{ iff } e \cdot C_{\pi_k} > e \cdot A_{\pi_k}, \quad (3.3.14)

Where \ \pi = (\pi_0, \pi_1, \ldots, \pi_{N+1}) \quad (3.3.15)

And \ G_{\pi} = 0, \ e \cdot \pi = 1 \quad (3.3.16)

Also \ e \ \text{is a [1 x (M_2 + 2)] vector having every entry is 1}.

It is clear from equation (3.3.14) that

\[
\lambda < (p_2^1 + p_2^2 p_5^1) \mu_2 \pi_0 + [(q_1 + p_1^2 - p_1^5 p_5^1) \mu_1 - (p_2^1 + p_2^5 p_5^1) \mu_2] (1 - \pi_{M_z+1})
\]

\[
(3.3.17)
\]

This can be considered as the stability condition (Prove by Neuts (1980)).

### 3.4 STATIONARY MEAN WAITING TIME

Here very easy to compute above time for model. At the first two stations i = 1,2, assume their expected presence rate is \lambda_i, \ the traffic intensity is \rho_i \ = \lambda_i/\mu_i, \ the expected queue length is L_i and the sojourn time (waiting time + service time) is W_i. Here it is assumed that all the parameters \lambda_i, \rho_i \text{ and } W_i \text{ are finite. Here strength of structure be indicated by } \rho_{sys} \text{ which is calculated from}

\[
\rho_{sys} = \frac{\sum_{i=1}^{2} \lambda_i}{\sum_{i=1}^{2} \mu_i}
\]

Thus here following two equations to indicate the system with two parameters \lambda_1 \text{ and } \lambda_2, \ both are unknown.

\[
\lambda_1 = \lambda + \left[p_1^1 + p_1^5(1 + p_5^1)\right] \lambda_1 + (p_2^1 + p_2^5 p_5^1) \lambda_2
\]

\[ \lambda_2 = (p_1^2 + p_1^S p_2^2) \lambda_1 + \left[ p_2^2 + p_2^S (1 + p_2^S) \right] \lambda_2 . \]  

(3.3.18)

The above expressions can be written as

\[ \lambda_1 = \frac{-\lambda [p_2^2 - 1 + p_2^S (1 + p_2^S)]}{D} \quad \text{and} \quad \lambda_2 = \frac{\lambda (p_1^2 + p_1^S p_2^2)}{D} \]

Where

\[ D = [p_1^1 - 1 + p_1^S (1 + p_1^S)] [p_2^2 - 1 + p_2^S (1 + p_2^S)] - (p_1^1 + p_2^S p_2^1) (p_1^2 + p_1^S p_2^2). \]

In order to find a mean waiting time of a task at every station, small formula is applicable.

\[ W_i = \frac{L_i}{\lambda_i} , \quad i = 1,2 \]

If we obtain the addition of every \( W_i \), provides the mean sojourn time of a task in the structure with inactive techniques.

### 3.5 APPROXIMATIONS AND SPLITTING IN QUEUE

Many real life queuing problems are handled without formal analysis. Practicing engineers and managers on the basis of perception, experience and familiarity with the problem often make reasonable decisions. However, there is no guarantee that these decisions are the best and sometimes they may even lead to costly errors. Alternatively, a mathematical modeling of the system is done through a system of equations and probability distributions so that different alternative management strategies can be evaluated and the optimal strategy that minimizes the total cost can be found.

There are a multitude of different queuing models that have been so far industrialized or could be developed. The choice of models depends on the characteristics of the process in question although the basic components are the same for all models. Selecting the appropriate mathematical model to describe the process is the most crucial and usually the most difficult part.
of any queuing analysis. Once a model has been constructed, analysis of the model can be performed either by (1) Analytical method or by (2) Simulation.

Waiting line problems are problems which engage waiting for service. Queuing problems encircle us from the time we rise in the morning until we retire at night. We are now in a situation to indicate the several types of interruptions occur in business world like a facilities break down and require repair, power failure occur, workers or the needed material do not show up where and when expected. Allocation of facilities considering such interruptions is done and bothers to solving a queuing problem. Sometimes competitive problems may also arise when two or more people are competing for a certain resource which may range from an opponent’s king in a game of chess to a larger share of the market in business world. Many times competitive problems involve command for a contract to perform a service. For thorough information that is required to make a certain decisions, search problems are used. The problem concerning exploring for valuable natural resources like oil or some other minerals is an example of a search problem. Similarly searching the ocean for enemy ships is another example of a search problem. In case of search problems the objective happens to be minimizing the costs associated with collecting and analyzing data, to reduce decision errors and also to minimize the costs associated with the decision errors themselves.

Since its beginning approximation has been applied to a wide variety of problems most of which have been tactical problems rather that strategic ones. Characteristic which differentiate the tactical problems from strategic ones are: (i) One problem is more tactical than another if the effect of its solution has a shorter duration. Thus a problem involving what to produce tomorrow is more tactical than that off where to build an additional plant. (ii) One problem is the more strategic, the larger portion of the organization that is directly affected by its solution. (iii) One problem is the extra strategic, the additional it involves the determination of ends, goals or objectives. Thus one cannot say that a exacting problem is either tactical or strategic in absolute sense but we can simply say that a particular problem is more or less intentional or strategic than another with respect to the said characteristics. Queuing theory is generally concerned with dealings that are intentional rather than strategic in nature.
3.5.1 Approximations

Here we explain higher order approximation methods for the MI/M/1 queue having nonfinite waiting room. From these calculations we derive different formula for approximations, collecting and reviewing in queuing analysis, these formulas can also apply to find large order calculation for queuing analysis depends on the length of the queue. They can be useful for the analysis of review case.

For approximation a single channel queuing problem results from random interarrival time and random service time at a single service station. The approximation time can be described mathematically by a probability distribution. The most common distribution found in queuing problems is Poisson probability distribution. This is used in single channel queuing problems for random arrivals where the service time is exponentially distributed. The sections ahead give the information for insight into the accurate nature of operations research. The difficulties of developing OR models, the need for logical approximations and the utilization of higher mathematics.

Generally approximations do not occur at fixed regular intervals of times but tend to be clustered or scattered in some fashion. A Poisson distribution is a discrete probability distribution which predicts the number of arrivals in a given time and its involves the probability of occurrence of an arrival. It assumes that arrivals are random and independent of all other operating conditions. The mean arrival rate $\lambda$ is assumed to be constant over time and is independent of the number of units already serviced, queue length or any other random property of the queue.

For approximation the mean arrival rate is constant over time, it follows that the probability of an arrival between $t$ & $t+dt$ is $\lambda \cdot dt$

Thus approximation probability of an arrival in time $t$ is $\lambda \cdot dt$

Also the probability of $n$ arrivals in time $t = \frac{(\lambda t)^n e^{-\lambda t}}{n!}$

In approximation; Service time is the time that required for completion of a service i.e. it is the time interval between beginning of a service and its completion. The indicate overhaul pace be
number of customers served for each component of instance (where assuming that the service to be continuous throughout the entire time unit), while the average service time is the time required to serve one customer. The mainly general type of sharing used for repair times is exponential sharing. It involves the probability of completion of a service. It should be noted that this distribution cannot be applied to servicing because of the possibility of the service facility remaining idle for some time. This distribution also assumed fixed time interval of continuous servicing which cannot be assured in all services.

Let $P$ refers the approximate time variables, $Q$ refers the exact time variables and let $R$ refers the expected time then we have the following formulas

$$
\alpha_k = \frac{R^{(k)}(0^+)}{k!} ;
$$

$$
\beta_k = \frac{E(Q^k)}{k!} ;
$$

$$
\gamma_k = \frac{E(P^k)}{k!} .
$$

### 3.5.1.1 Light Traffic Approximation

For calculation of the starting $n$ events of the approximation time $V$ and the expected time $W$ there is a $[P/Q]$ non integer approximation where $P + Q > n + 1$, the below characters are required.

1. $\beta_1, \beta_2, ..., \beta_{P+Q}$
2. $\alpha_1, \alpha_2, ..., \alpha_{P+Q-k-1}$
3. $\gamma_1, \gamma_2, ..., \gamma_{P+Q}$
The first \((n + 1)\) calculations of time can be calculated is depends on the first \(n\) events of the approximation time. Now we define the \(r^{th}\) term for approximation time, the exact time and the expected time respectively as:

\[
\frac{E[W^k]}{k!} = \sum_{m=0}^{\infty} w_{km} \theta^m, \quad (3.5.1)
\]

\[
\frac{E(T^k)}{k!} = \sum_{m=0}^{\infty} t_{km} \theta^m, \quad (3.5.2)
\]

\[
\frac{E(D^k)}{k!} = \sum_{m=0}^{\infty} d_{km} \theta^m, \quad (3.5.3)
\]

Where, \(w_{km}\), \(t_{km}\) and \(d_{km}\) are calculated as follows

\[
t_{km} = \begin{cases} 
\beta_k, & m = k \\
\sum_{i=1}^{k} \beta_{k-i} w_{i(m-k+i)}, & m > k \\
0, & m < k
\end{cases} \quad (3.5.4)
\]

\[
w_{km} = \begin{cases} 
\sum_{i=0}^{m-k-1} \alpha_i t_{(k+1+i)m}, & m > k \\
0, & m \leq k
\end{cases} \quad (3.5.5)
\]

\[
d_{km} = \sum_{j=\max(1,k-m)}^{k} (-1)^j \beta_{k-j} \left( \sum_{i=1}^{j} (-1)^{i-j} \gamma_{j-i} t_{(m-k+i)} - w_{j(m-k+j)} \right) + \begin{cases} 
\beta_m \gamma_{k-m}, & m \leq k \\
0, & m > k
\end{cases} \quad (3.5.6)
\]

### 3.5.1.2 Interpolation Approximation

The light traffic queuing and heavy traffic queuing both are merged in the requirement of a large order interpolation in alone server queue. In order to calculating the starting \(k\) moments for a straight forward waiting limit \(W\) and server limit for \([P/Q]\) multipoint Pade approximation, below mentioned variables are required

1. \(\beta_1, \beta_2, ..., \beta_{P+Q+k}\)
2. \(\alpha_1, \alpha_2, ..., \alpha_{P+Q-1}\)
3. $\gamma_1, \gamma_2, \ldots, \gamma_{n_0+k}$

The $n^{th}$ term for waiting limit can be calculated from the below mention formulas:

\[
\frac{(1-\rho \theta)^n E[W_n(\theta)]}{\theta^{n+n_0}} \approx \frac{\det \begin{bmatrix}
\varepsilon_2^{(n)} & \varepsilon_{2M-1}^{(n)} & \cdots & \varepsilon_M^{(n)} \\
\vdots & \vdots & & \vdots \\
\varepsilon_{M+1}^{(n)} & \varepsilon_M^{(n)} & \cdots & \varepsilon_1^{(n)} \\
\varepsilon_M^{(n)} & \varepsilon_{M-1}^{(n)} & \cdots & \varepsilon_0^{(n)}
\end{bmatrix}}{\det \begin{bmatrix}
\varepsilon_2^{(n)} & \varepsilon_{2M-1}^{(n)} & \cdots & \varepsilon_M^{(n)} \\
\vdots & \vdots & & \vdots \\
\varepsilon_{M+1}^{(n)} & \varepsilon_M^{(n)} & \cdots & \varepsilon_1^{(n)} \\
\varepsilon_M^{(n)} & \varepsilon_{M-1}^{(n)} & \cdots & \varepsilon_0^{(n)}
\end{bmatrix}},
\]

(3.5.7)

Where

\[
u_i^{(n)} = u_2^{(n)} \theta_i^{l_2-1} - \sum_{k=0}^{m_1-1} u_{i_1(m_1-1-k)}^{(n)} \theta_{i_1+1}^{k+1},
\]

\[
u_2^{(n)} = u_2^{(n)} \theta_2^{l_2-q-1} - \sum_{k=0}^{m_2-1} u_{i_2(m_2-1-k)}^{(n)} \theta_{i_2+1}^{k+1},
\]

\[
u_0^{(n)} = n! \rho^{n+n_0} \left( \frac{(C_0 + C_2 \gamma_1)}{2} \right)^n,
\]

\[
u_1^{(n)} = n! \sum_{i=0}^{n} (-\rho)^i \left( \frac{n}{i} \right) W_{n(m+n-i)}
\]

And $\theta_2 = 1/\rho$, $\rho = \beta_1/\gamma_1$, $n^* = 1$

Further $C_0^2$ and $C_2^2$ respectively represents the starting variables and ending variables, for a inter arrival. The heavy traffic may also require the parameters $\beta_1, \beta_2, \gamma_1, \gamma_2$.

3.5.1.3 Matching factors and correlations

In starting requires the formulas of matching the first three non-central factors for a joining of two Erlang distributions for normal order. Consider $M_i$ is the $i^{th}$ non central term for the required
allocation. On the way toward comparing initial three instants of joining for repeated array (n),
the parameters to be determined are n, p, λ₁ and λ₂.

Assume n* be smallest integer which satisfies the following inequalities

\[ n^* \geq \frac{M_1^2}{M_2 - M_1^2}, \]  \hspace{1cm} (3.5.8)

\[ n^* \geq \frac{2(M_2 - M_1^2)^2 + M_1^2M_2 - M_1M_3}{M_1M_3 - (M_2 - M_1^2)(M_2 - 2M_1^2)}, \]  \hspace{1cm} (3.5.9)

Choose any n ≥ n*. The left over variables for mixed Erlang division be determined after then
from the equations as under

\[ \lambda_1 = \frac{2D_1}{D_2 + \sqrt{D_2^2 - 4D_1D_3}}, \]  \hspace{1cm} (3.5.10)

\[ \lambda_2 = \frac{2D_1}{D_2 - \sqrt{D_2^2 - 4D_1D_3}}, \]  \hspace{1cm} (3.5.11)

\[ p = \lambda_1 \frac{1-\lambda_2 A}{\lambda_1-\lambda_2}, \]  \hspace{1cm} (3.5.12)

Where

\[ A = \frac{M_1}{n}, \]

\[ B = \frac{M_2}{n(n+1)}, \]

\[ C = \frac{M_3}{n(n+1)(n+2)}, \]

\[ D_1 = A^2 - B, \]

\[ D_2 = AB - C, \]

\[ D_3 = B^2 - AC. \]
Now we give the formula for calculating the first two non-central events as well as lag – 1 normal correlation for Markov – transformed method. Here the transition matrix is given by

\[
A = \begin{bmatrix} a & 1 - a \\ 1 - a & a \end{bmatrix}
\]

The required probability is obtained through represents \( x_{11}, x_{12}, x_{21}, x_{22} \).

Now considering the square of coefficient of variation as

\[
C^2 = \left( \frac{M_2}{M_1^2} \right) - 1
\]

The unknown parameters can now be determined as follows

\[
x_{11} = x_{12} = M_1 \left( 1 + \sqrt{\frac{c^2 - 1}{2}} \right), \quad (3.5.13)
\]

\[
x_{21} = x_{22} = M_1 \left( 1 - \sqrt{\frac{c^2 - 1}{2}} \right), \quad (3.5.14)
\]

\[
p = \frac{2X + (c^2 - 3)M_1^2}{2(c^2 - 1)M_1^2}. \quad (3.5.15)
\]

These formulas can be applied with certain conditions.

3.5.2 The Splitting Feature

The outer arrival events can be merged into a fix number of different lines then its probability is considered. It is assumed that there are no reactions. Assume \( D \) is the inter departure time random variable and there are \( n \) sources that originate from this node having possibilities \( p_1, p_2, \ldots, p_n \). The random variable for \( i^{th} \) source is represented by \( Q(i) \). Subsequently instant producing purpose of \( Q(i) \) is given by :

\[
M_{Q(i)}(S) = \frac{p_i M_D(S)}{1 - (1 - p_i) M_D(S)}.
\]
Where $M_D(S)$ is the instant creating function of $D$. Here it is stress-free to evaluate the moments of $Q(i)$ using the above mentioned formula. Another formula prepared by Johnson,(1988) can compare for the only initially 6 moments therefore we find the initially six non crucial instant which are as under

$$E[Q(i)] = \frac{E[D]}{p_i},$$

$$E[Q^2(i)] = \frac{E[D^2]}{p_i} + \frac{2(1-p_i)}{p_i^2} (E[D])^2,$$

$$E[Q^3(i)] = \frac{E[D^3]}{p_i} + \frac{6(1-p_i)}{p_i^2} E[D]E[D^2] + \frac{6(1-p_i)^2}{p_i^3} (E[D])^3,$$

$$E[Q^4(i)] = \frac{E[D^4]}{p_i} + \frac{(1-p_i)}{p_i^2} \{8E(D)E(D^3) + 6(E[D^2])^2\} + \frac{36(1-p_i)^2}{p_i^3} (E[D])^2E[D^2]$$

$$+ \frac{24(1-p_i)^3}{p_i^4} (E[D])^4,$$

$$E[Q^5(i)] = \frac{E[D^5]}{p_i} + \frac{10(1-p_i)}{p_i^2} \{E[D]E[D^4] + 2E[D^2]E[D^3]\}$$

$$+ \frac{30(1-p_i)^2}{p_i^3} \{2(E[D])^2E[D^3] + 3E[D](E[D^2])^2\}$$

$$+ \frac{240(1-p_i)^3}{p_i^4} (E[D])^3E[D^2] + \frac{120(1-p_i)^4}{p_i^5} (E[D])^5,$$

$$E[Q^6(i)] = \frac{E[D^6]}{p_i} + \frac{2(1-p_i)}{p_i^2} \{6E[D]E[D^5] + 15E[D^2]E[D^4] + 10(E[D^3])^2\}$$

$$+ \frac{90(1-p_i)^2}{p_i^3} \{(E[D])^2E[D^4] + 4E[D]E[D^2]E[D^3] + (E[D^2])^3\}$$

$$+ \frac{120(1-p_i)^3}{p_i^4} \{4(E[D])^3E[D^3] + 9(E[D])^2(E[D^2])^2\}.$$
Here dealing is only with lag 1 auto connection, as there is too much important and our study reflects only the first lag for the auto connection. Though, it is cleared that this concept can be stretched to the approximation of the as possible as large lag also. To decide lag 1 auto connection in single action, the subsequent discussion is useful.

Decide any single divided group which is divided with probability \( p \) into the main withdrawals group \( D \). Now consider \( D_n \) as an initial withdrawal time having two clients \( n \) and \( (n+1) \). Take \( Q_n(i) \) is a initial acceptance instance for the clients \( n \) and \( (n+1) \) which consider in divided group for the interest \( i \). Here it is expected that this method is static. Let \( k \) is the number of withdrawals earlier the initial entering of the divided group after the ending entering then \( Q_1(i) \) is the parallel inter coming time. Correspondingly, \( j \) is the number of departures before the next one and \( Q_2(i) \) be the corresponding inter arrival time. Here it is easy to verify that \( k \) and \( j \) is numerically divided as well as both independent from each other. Hence the following relations are there:

\[
E[Q_1(i)Q_2(i)] = \sum_{k} \sum_{j} (1 - p)^{j-1} p (1 - p)^{k-1} p E \left[ \left( \sum_{l=1}^{k} D_l \right) \left( \sum_{l=k+1}^{k+j} D_l \right) \right]
\]

\[
= \sum_{k} \sum_{j} p^2 (1 - p)^{k+j-2} E \left[ \left( \sum_{l=1}^{k} D_l \right) \left( \sum_{l=k+1}^{k+j} D_l \right) \right] \tag{3.5.17}
\]

Now the procedure is divided in the exterior coming procedure which is a restoration process of earlier, then the principle mentioned as above leads to

\[
E [Q_1(i)Q_2(i)] = \left( \frac{E[D]}{p} \right)^2
\]

Which indicates that the lag 1 parallel coefficient is zero. This occurs since the next divided process is obviously a repeated process.
Now divided process is the withdrawal process coming inside a MI/M/1 queue, then we define the auto connection as a relation for MI/M/1 model queue constraints since

\[ D_n = T_{n+1} - T_n + Q_n; \]

Where \( Q_n \) represents time of interarrival amongst clients \( n \) and \( (n+1) \) & \( T_n \) represents whole structure instance of client. Here we conclude that

\[
E \left[ \left( \sum_{i=1}^{k} D_i \right) \left( \sum_{i=k+1}^{k+j} D_i \right) \right] = k j (E[Q])^2 - E[T^2] + E[T_1 T_{k+1}] + E[T_j T_{j+1}] - E[T_1 T_{K+j+1}]
\]

\[ + E[T_{k+j+1} \sum_{i=1}^{k} Q_i] - E[T_{k+1} \sum_{i=1}^{k} Q_i] \]

(3.5.18)

However in fixed state

\[ E[T_{k+1} T_{k+j+1}] = E[T_1 T_{j+1}] , \]

\[ E[T_{k+1}^2] = E[T^2] , \]

\[ E[T_1 \sum_{i=k+1}^{k+j} Q_i] = E[T_{k+1} \sum_{i=k+1}^{k+j} Q_i] = jE[T] E[Q] , \]

\[ E \left[ \left( \sum_{i=1}^{k} Q_i \right) \left( \sum_{i=k+1}^{k+j} Q_i \right) \right] = k j (E[Q])^2 \]

Here \( Q \) represents a general interarrival period while \( T \) represents basic system period for the MI/M/1 model. The basic requirements of equation (3.5.18) are also being calculated from the equations of Hu (1992). The relationships can receive here also be used in finding the distribution for the remaining process, which is initial processes for the remaining queues. If the relationships will not be considered, then corresponding moments can also be relate with respective Erlang distribution. Here the starting delay of correlation goes to significant, then its results are computed by using non-renewal techniques.
3.6 CASE FOR MERGING

In this section, in short we focus over some special techniques for describing the collections of more than two arrival processes first. A detailed discussion of Markov chains is already given in Bitran & Dasu (1989) which are known as, SE sequences. They normally are preferred as a purpose of very good location for stage restoration techniques. Their concept is fundamentally for higher regulate calculation, during which starting both instants compared with each other. Compared the starting moment is minor. Respective the other moment, their calculations can be taken for both the fixed and asymptotic characteristic. Now here calculated experiments indicate that their SE sequences are completely suitable simply while their starting difference is not below their ending variance. In this case, the coefficient for different deviation for every process being combined, but this concept does not give good result. Consequently, if coefficients of variation for the collecting activities are smaller than one, this concept works very well. Here, it is noted that our goal which will be explained in the succeeding small sections is a large order approximation, while the technique which explained in Bitran and Dasu (1989) is only for second order.

Comparison of Super-Erlang chains is complete overview for the great position examples given in Balsamo & Marco (2004) represents second order approximations. In which they also recommended three concepts for comparing the second order moment. In the first method variance is a set for the fixed time variance. Finally third method concept is for mix approximation which mixes the variance to a rounded arrangement for fixed and asymptotic variances. Here the remaining section is dedicated for the expansion of new techniques for higher order approximation. The better position of any repeated processes was explained by S. G. Browning (1998). He claimed that ending procedure is clearly a Markov repeated procedure. Here we use Browning results for the computation of the starting moments for the superposed development. Also these moments can further be used to decide the sharing of the combined process.

Let \( \{X_i; i \geq 1\} \) and \( \{Y_i; i \geq 1\} \) represents the stochastic techniques corresponds for the given two reversible processes which are asymptotic. Also \( f(a) \), \( F(a) \) and \( \bar{F}(a) \) represents the respectively probability, increasing and balancing density functions for the first process. Let \( g(b) \), \( G(b) \) and
\( \tilde{G}(b) \) represents the similar tasks designed for another procedure. The tremendous events is represented through their respective couple, \( \{ A_n, B_n; n \geq 1 \} \) where \( A_n = 1 \) when \( (n-1)^{th} \) result of the combined process is created by the starting process also \( A_n = 2 \) otherwise and \( B_n \) is the time elapsed because of the ending occurrence method which may not be create the \( (n-1)^{th} \) event for the combined process.

The stochastic techniques \( \{A_n, B_n, V_n, n \geq 0\} \) is Markov restoration depends on \( (\{1,2\} \times R^+, (2^{\{1,2\}} \times R^+)) \). However, the moving possibility matrix for corresponding Markov reversible techniques generally be indicated by \( P(Z, A/A_n) \) where we terminate \( (n+1) \) for \( Z \) & \( A \) for calculations straight forwardness.

Above discussions fundamentally generalizes our beginning in the prior smaller sections to the absorption in the Markov chain performance faster and there are so many conditions, where we convey the moments in blocked form for particularly this case as well.

### 3.7 REDUCTION OF VARIABILITY IN SPLIT MERGE SYSTEM

Performance analysis has acquired increased importance due to the growing complexity of automated systems. Presentation modeling facilitates a sympathetic of the connections among scheme workload, organize constraints and key metrics such as client reply time, system consumption and buffer occupancy. For systems that involve the flow and processing of customers and resources, queuing models are an appropriate formalism. Optimization of control parameters allows to minimize, for example, mean response time within given constraints.

It has been observed in a recent paper related to the scheduling of Map Reduce jobs in clusters that delayed scheduling of jobs can counter intuitively lead to greater fairness and a higher level of data locality. Delay scheduling was applied in the context of quality of service in networks. Specifically, adding delays to input packets results in shaping the traffic such that packet inters arrival times must be dependent to exponential distribution. This creation allows the analysis of the network with mathematically tractable Markovian models.

In this section we show that adding judiciously chosen deterministic delays to subtask processing in split and merge systems can result in a reduction of variability in terms of time difference.
between the completion of the first and last subtasks in a job. At the same time, corresponding beneficial effects on output buffer occupancy are observed. A major application area of our approach is automated warehouse systems, where partially completed subtasks need to be held in a physical buffer space. Another application field of this technique is parallel computing where it is sometimes desirable to minimize mean synchronization time between tasks. In healthcare systems, we can minimize the time patients wait for results following treatment. Lastly, in project scheduling we can reduce mean slack time.

![Figure 3.7.1 Split Merge Queuing Model](image)

When all subtask servers are idle and the task queue is not empty, a task is taken from the head of the task queue. This task splits into N subtasks at the split point. Each subtask server then processes its allocated subtask. Outgoing subtasks join the merge buffer. When all subtasks belonging a task are present in the merge buffer, the task exits the system via the merge point.

### 3.7.1 Variability in Split Merge System

Let us define the variability of a split merge system as the mean difference in time between the arrival of the first and last subtasks belonging to each task in the merge buffer. Our challenge is to control this variability via the introduction of a vector of delays which are:

\[
d = (d_1, d_2, \ldots, d_i, \ldots, d_{i-1}, d_n) \quad (3.7.1)
\]
Here element \(d_i\) of the vector represents the deterministic delay that will be applied before a subtask is sent to server \(i\) for processing. Let further define the constant function of a split merge system for a given delay vector as

\[
C(d) = E(X) - E(Y) \tag{3.7.2}
\]

Where \(X\) is the random variable which mention the highest completion time across all subsets arising from a particular task and \(Y\) is the random variable mention the lowest completion time across all subsets. Assuming that subtasks at server \(I\) are served independently with service time sampled from a distribution function \(F_i(t)\) then, taking into account the delay that is applied before each subtask begins processing, \(X\) will have cumulative distribution function :

\[
F_X(t) \sim \prod_{i=1}^{n} F_i(t - d_i) \tag{3.7.3}
\]

Here understood that, for every \(i\) ; \(F_i(t - d_i) = 0\) for all \(t < d_i\). Similarly, \(Y\) has cumulative distribution function:

\[
F_Y(t) \sim \prod_{i=1}^{n} F_i(t - d_i) \tag{3.7.4}
\]

For a given split merge system, our challenge is to find that vector \(D\) which minimizes \(C(d)\). To constrain the solution space while avoiding unnecessary delays to overall mean task processing time, we set \(d_i = 0\) for the subtask server with the largest mean service time. We will denote the resulting vector of optimal delays as

\[
\bar{d} = (\bar{d}_1, \bar{d}_2, ..., \bar{d}_{l-1}, 0, \bar{d}_{l+1}, ..., \bar{d}_{n-1}, \bar{d}_n) \tag{3.7.5}
\]

We note that minimizing \(C\) results in minimum merge buffer utilization in the split merge system. This property is particularly relevant in physical systems which are often constrained in terms of the amount of physical output buffer space available. Although it is our ultimate goal to establish an efficient analytical procedure for determining \(\bar{d}\), in the present paper we apply a
simple simulation based on extensions to the JINQS queueing network simulation package to explore the shape of the cost function landscape and hence to find near optimal solution for $\tilde{d}$. 