

**T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

**FAZ DAĞILIMLI KUYRUK SİSTEMLERİNDE HİPOTEZ TESTLERİ İÇİN
ARDIŞIK ÇÖZÜMLEME YÖNTEMİNİN KULLANILMASI**

DOKTORA TEZİ

Müjgan ZOBU

İstatistik Anabilim Dalı

**Kasım 2012
SAMSUN**



**T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**



İSTATİSTİK ANABİLİM DALI

**FAZ DAĞILIMLI KUYRUK SİSTEMLERİNDE HİPOTEZ TESTLERİ İÇİN
ARDIŞIK ÇÖZÜMLEME YÖNTEMİNİN KULLANILMASI**

DOKTORA TEZİ

Müjgan ZOBU

Tezin Savuma Tarihi : 9 Kasım 2012

Tez Danışmanı : Doç. Dr. Vedat SAĞLAM

**Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü
İstatistik Anabilim Dalında
Müjgan Zobu Tarafından Hazırlanan**

**FAZ DAĞILIMLI KUYRUK SİSTEMLERİNDE HİPOTEZ
TESTLERİ İÇİN ARDIŞIK ÇÖZÜMLEME YÖNTEMİNİN
KULLANILMASI**

**başlıklı bu çalışma jürimiz tarafından 09/11/2012 tarihinde yapılan sınav ile
DOKTORA tezi olarak kabul edilmiştir.**

Başkan : Prof.Dr. Hüseyin DEMİR
Ondokuz Mayıs Üniversitesi

Jüri Üyeleri :Doç.Dr. Vedat SAĞLAM
Ondokuz Mayıs Üniversitesi

Doç.Dr. Erol EĞRİOĞLU
Ondokuz Mayıs Üniversitesi

Doç.Dr. Sinan ÇALIK
Fırat Üniversitesi

Doç.Dr. Kamil ALAKUŞ
Ondokuz Mayıs Üniversitesi

....../....../2012

Prof. Dr. Recep TAPRAMAZ

Enstitü Müdürü

Anneannem Şaibe Şenkal'a,

ÖNSÖZ

Doktora ders ve tez aşaması boyunca destek ve ilgilerini esirgemeyen danışman hocam Sayın Doç.Dr. Vedat SAĞLAM'a, ayrıca yönlendirici eleştiri ve katkılarıyla beni destekleyen değerli hocalarım Sayın Doç.Dr. Erol EĞRİOĞLU ve Sayın Prof.Dr. Hüseyin DEMİR'e ve her zaman destek ve teşvikleriyle yanımda bulunan annem ve babama teşekkürlerimi sunarım.

Kasım 2012

Müjgan Zobu
(Uzman)

İÇİNDEKİLER

	<u>Sayfa</u>
ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xii
KISALTMALAR	xv
ÖZET.....	xvii
SUMMARY	xix
1. GİRİŞ	1
2. GENEL BİLGİLER.....	3
2.1 Basit Hipotez	3
2.2 Poisson Süreci ve Özellikleri	4
2.3 Ardışık Çözümleme.....	7
2.3.1 Poisson dağılımı için ardışık çözümleme.....	9
2.4 Markov Zincirleri	10
2.4.1 Gömülü Markov zinciri.....	12
2.5 Kuyruk Sistemleri	13
2.5.1 $M/G/1$ Kuyruk modeli	16
2.5.2 $M/G/1/N$ Kuyruk modeli	17
2.5.3 $G/M/1$ Kuyruk modeli	18
3. KUYRUK SÜRECİ VE ARDIŞIK OLASILIK ORAN TESTİ.....	21
3.1 $M/G/1$ ve $GI/M/c$ Kuyrukları İçin Ardışık Olasılık Oran Testi.....	23
3.1.1 $M/G/1$ Kuyruk modeli	23
3.1.2 $M/E_k/1$ Kuyruk modeli	24
3.1.3 $GI/M/c$ Kuyruk modeli.....	27
3.1.4 $E_k/M/c$ Kuyruk modeli.....	28
3.2 Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayısı	29
4. FAZ DAĞILIMLI KUYRUK MODELLERİ İÇİN ARDIŞIK TEST	33
4.1 Hyperexponential ($M/H_2/1$) Kuyruk Modeli.....	33
4.2 Mixed Erlang ($M/mixE_k/1$) Kuyruk Modeli.....	36
4.3 Hypoexponential ($M/HYPO_k/1$) Kuyruk Modeli	40
4.4 Coxian ($M/C_k/1$) Kuyruk Modeli	44
5. SAYISAL ÖRNEK.....	47
5.1 Erlang ($M/E_k/1$) Kuyruk Modeli İçin Sayısal Örnek	47
5.2 Hyperexponential ($M/H_2/1$) Kuyruk Modeli İçin Sayısal Örnek	52
5.3 Mixed Erlang ($M/mixE_k/1$) Kuyruk Modeli İçin Sayısal Örnek	65
5.4 Hypoexponential ($M/HYPO_k/1$) Kuyruk Modeli İçin Sayısal Örnek	69
5.5 Coxian ($M/C_k/1$) Kuyruk Modeli İçin Sayısal Örnek	71
6. SONUÇ VE ÖNERİLER.....	75
KAYNAKLAR	77
EKLER.....	79
ÖZGEÇMİŞ.....	143

ÇİZELGE LİSTESİ

Sayfa

Çizelge 5.1: $N = 4$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.....	47
Çizelge 5.2: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.....	51
Çizelge 5.3: $N = 7$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.....	53
Çizelge 5.4: Test için hesaplanan c_{ij} katsayısı	53
Çizelge 5.5: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.....	55
Çizelge 5.6: $N = 5$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.....	57
Çizelge 5.7: Test için hesaplanan c_{ij} katsayısı	57
Çizelge 5.8: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.....	59
Çizelge 5.9: $N = 4$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.....	61
Çizelge 5.10: Test için hesaplanan c_{ij} katsayısı	61
Çizelge 5.11: $N = 4$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.....	62
Çizelge 5.12: $\alpha = 0.005$, $\beta = 0.05$ için Karakteristik İşlem Eğrisi ve Ortalama örneklem Sayıları.....	63
Çizelge 5.13: $\alpha = 0.05$, $\beta = 0.1$ Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.....	63
Çizelge 5.14: $N = 10$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı	65
Çizelge 5.15: Test için hesaplanan c_{ij} katsayısı	66
Çizelge 5.16: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.....	68
Çizelge 5.17: $N = 7$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.....	70
Çizelge 5.18: Test için hesaplanan c_{ij} katsayısı	70
Çizelge 5.19: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.....	71
Çizelge 5.20: $N = 3$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.....	73
Çizelge 5.21: Test için hesaplanan c_{ij} katsayısı	73
Çizelge 5.22: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.....	74

ŞEKİL LİSTESİ

Sayfa

Şekil 4.1:	Hyperexponential ($M/H_2/1$) kuyruk modeli.....	33
Şekil 4.2:	Mixed Erlang ($M/mixE_k/1$) kuyruk modeli.	36
Şekil 4.3:	Hypoexponential ($M/HYPO_k/1$) kuyruk modeli.	41
Şekil 4.4:	Coxian ($M/C_k/1$) kuyruk modeli.....	44
Şekil 5.1:	$H_0: \rho = 0.1, H_1: \rho = 0.9$ testi için t 'ye karşı gelen $\lambda(t, \rho)$ 'nin grafiği .	50
Şekil 5.2:	$H_0: \rho = 0.1, H_1: \rho = 0.9$ testi için Karakteristik İşlem Eğrisi	51
Şekil 5.3:	$H_0: \rho = 0.1, H_1: \rho = 0.9$ testi için Ortalama Örneklem Sayısı	52
Şekil 5.4:	$H_0: \rho = 1.3, H_1: \rho = 0.7$ testi için t 'ye karşı gelen $\lambda(t, \rho)$ 'nin grafiği.	54
Şekil 5.5:	$H_0: \rho = 1.3, H_1: \rho = 0.7$ testi için Karakteristik İşlem Eğrisi	55
Şekil 5.6:	$H_0: \rho = 1.3, H_1: \rho = 0.7$ testi için Ortalama Örneklem Sayısı	56
Şekil 5.7:	$H_0: \rho = 0.3, H_1: \rho = 1.5$ testi için t 'ye karşı gelen $\lambda(t, \rho)$ 'nin grafiği	58
Şekil 5.8:	$H_0: \rho = 0.3, H_1: \rho = 1.5$ testi için Karakteristik İşlem Eğrisi	59
Şekil 5.9:	$H_0: \rho = 0.3, H_1: \rho = 1.5$ testi için Ortalama Örneklem Sayısı	60
Şekil 5.10:	$H_0: \rho = 0.5, H_1: \rho = 0.8$ testi için t 'ye karşı gelen $\lambda(t, \rho)$ 'nin grafiği.	62
Şekil 5.11:	$H_0: \rho = 0.5, H_1: \rho = 0.8$ testi için Karakteristik İşlem Eğrisi	64
Şekil 5.12:	$H_0: \rho = 0.5, H_1: \rho = 0.8$ testi için Ortalama Örneklem Sayısı.....	64
Şekil 5.13:	$H_0: \rho = 0.1, H_1: \rho = 0.9$ testi için t 'ye karşı gelen $\lambda(t, \rho)$ 'nin grafiği..	67
Şekil 5.14:	$H_0: \rho = 0.1, H_1: \rho = 0.9$ testi için Karakteristik İşlem Eğrisi	68
Şekil 5.15:	$H_0: \rho = 0.1, H_1: \rho = 0.9$ testi için Ortalama Örneklem Sayısı.....	69

KISALTMALAR

FIFO	: First In First Out
LIFO	: Last In First Out
i. i. d.	: Bağımsız ve aynı dağılıma sahip
d. d.	: Diğer durumlarda
o.y.f	: Olasılık yoğunluk fonksiyonu
GI	: Genel Dağılım

FAZ DAĞILIMLI KUYRUK SİSTEMLERİNDE HİPOTEZ TESTLERİ İÇİN ARDIŞIK ÇÖZÜMLEME YÖNTEMİNİN KULLANILMASI

ÖZET

Kuyruk sistemlerindeki çalışmalarda kuyrukların kontrolü dikkat çeken problemlerden biridir ve geliş oranı, servis zamanı, kanal sayısı , kuyruk disiplini gibi kontrol edilebilir parametrelerden etkilenmektedir. Kuyruk sistemlerinin en önemli karakteristik elemanlarından birisi de trafik yoğunluğu ρ 'dur.

Trafik yoğunluğundaki değişiklikleri mümkün olduğu kadar hızlı tespit etmek ve uygun düzeltici eylemleri alarak trafik yoğunluğunu arzu edilen düzeyde tutmak ve bunlara bağlı olarak uygulamalarda ne kadarlık bir örneklem genişliğine ihtiyaç duyulacağını belirlemek amaçlanır. Buna göre Ardışık Olasılık Oran Testi kullanılarak klasik yöntemlere göre örneklem genişliğinde yüzde elliye varan bir tasarruf sağlanır. Bu bağlamda literatürde $M/G/1$ ve $GI/M/c$ tipindeki kuyruk sistemlerinde trafik yoğunluğundaki değişimleri ortaya çıkarmak için Ardışık Olasılık Oran Testine dayalı bir metot geliştirilmiştir.

Bu çalışmada ise dört farklı faz dağılımlı kuyruk modelleri olan Hyperexponential, Mixed Erlang, Hypoexponential ve Coxian kuyruk modelleri için Ardışık Olasılık Oran Testi teorik biçimde incelenerek olabirlik oran katsayıları elde edildi. Ayrıca $H_0: \rho = \rho_0$ ve $H_1: \rho = \rho_1$ hipotezleri için α ve β türü hataların farklı durumlarına göre çok noktalı türev bağıntısı kullanılarak karakteristik işlem eğrileri ve muhtelif optimal örneklem genişlikleri nümerik yöntemler ile elde edildi. Bu modellerin her birisi için Matlab yazılım programı ile sayısal örnekler üretilerek Ardışık Olasılık Oran Testine dayalı farklı uygulamalar yapıldı.

Anahtar Kelimeler: Faz Dağılımlı Kuyruk Modelleri; Ardışık Olasılık Oran Testi; Karakteristik İşlem Eğrisi; Ortalama Örneklem Sayısı; Gömülü Markov Zinciri.

USING SEQUENTIAL ANALYSIS FOR HYPOTHESIS TESTS IN THE PHASE-TYPE DISTRIBUTION QUEUEING SYSTEMS

SUMMARY

The control of queues is one of the interesting problems in the study of queueing systems and can be affected through controllable parameters like arrival and service rate, number of service channels, the queue discipline. A performance characteristic of great importance and interest of many queueing applications is the traffic intensity ρ .

The objective of the control technique is to detect changes in the traffic intensity ρ as quickly as possible and then to take appropriate corrective action and to determine how much of needed a sample size in the applications. Thus the sequential probability ratio test provides a saving of up to fifty percent in the sample size according to traditional methods. Accordingly, in the literature, a method depends on the Sequential Probability Ratio Test developed to detect changes in the traffic intensity of $M/G/1$ and $GI/M/c$ queueing systems.

In this thesis, the sequential probability ratio test was theoretically investigated for four different phase-type queueing which consists of Hyperexponential, Mixed Erlang, Hypoexponential and Coxian queueing. Also, for testing $H_0: \rho = \rho_0$ against $H_1: \rho = \rho_1$ Operating Characteristic and Average Sample Number functions obtained with numerical methods using multi-point derivative equations according to different situations of α and β type errors. After, numerical illustrations for each model were provided with Matlab programming.

Keywords: Phase-Type Distribution Queueing Model; Sequential Probability Ratio Test; Operating Characteristic; Average Sample Number; Imbedded Markov Chain.

1. GİRİŞ

Kuyruk sistemleri stokastik süreçler teorisinde oldukça önemli bir konudur ve son yıllarda bu alanda yapılan çalışmalar ile büyük ilerlemeler kaydedilmiştir. Bu gelişmelere rağmen kuyruk teorisinin istatistiksel analizleri ise biraz daha ikinci planda kalmıştır.

Stokastik süreçlerdeki istatistiksel çıkarım ve özellikle kuyruk sistemindeki çıkarımlar son yıllarda araştırmacıların dikkatini çekmektedir. Geliş oranı, servis oranı, trafik yoğunluğu gibi kuyruk sürecinin parametreleriyle ilgili kestirim problemi ve hipotez testi kuyruk sistemlerinin çalışmalarının karşılaştırılması ve kuyruğun dizayn edilmesinde önemlidir.

Kuyruk teorisinde parametre kestirimi üzerine ilk çalışmalar $M/M/1$ kuyruk modelinde geliş oranı ve servis oranının maksimum olabilirlik kestiriminin çıkarımı Clarke (1957) tarafından yapılmıştır. Dave ve Shah (1980) heterojen servisli $M/M/2$ kuyruk modelinde parametrelerin maksimum olabilirlik kestirimini elde ettiler. Benes (1957) sonsuz kanallı $M/M/\infty$ kuyrukları için geliş oranı ve servis oranının maksimum olabilirlik kestirimini tartışırken, Basava ve Prabhu (1981) ise $GI/G/1$ kuyruk modelinin servis zamanı dağılımlarını ve gelişler arası parametrelerinin maksimum olabilirlik moment kestirimlerini ele almıştır. Basava ve ark. (1996) başarılı müşterilerin sistemde bekleme zamanlarına dayalı $GI/G/1$ kuyruk modelinin parametreleri için maksimum olabilirlik kestirimini elde etmiştir.

Harishchandra ve Rao (1984), $M/G/1$ kuyruk modeli için gömülü Markov zinciri analizini kullanarak $M/E_k/1$ kuyruk modelinde trafik yoğunluğu parametresi ρ 'nun maksimum olabilirlik kestirimini elde etti. Muddapur (1972), Clarke (1957) tarafından verilen aynı olabilirlik fonksiyonunu kullanarak $M/M/1$ ve $M/M/\infty$ kuyruk modellerinin parametrelerinin kestirimi için Bayes metodunu çalıştılar.

Kuyruk teorisinde hipotez testi birkaç yazar tarafından araştırılmıştır (Rao ve ark.,1987; Harishchandra ve Rao, 1984; Choudhury ve Borthakur, 2008). Kuyruk modelleme yolları ve kuyruklarda istatistiksel çıkarımlar üzerine kapsamlı bir araştırma Bhat (2008) tarafından yapılmıştır.

Kuyruk sisteminde hipotez testi hala keşfedilmemiş bir alan olarak kalmıştır. Geliş ve servis zamanı dağılımlarının parametreleri hakkında ya da trafik yoğunluğu gibi ölçümler hakkında sonuç çıkarmamız gerektiğinde hipotez testine başvururuz. Dağılımın yapısı örneklemeyle elde edilen veriye dayanmaktadır. Elde edilen sonuçlar kontrol yöntemlerine kadar uzanmaktadır.

Kuyruk teorisinde hipotez testinin kullanılabildiği çoğu durumlar yöntemin parametre değerleri hakkında bazı ön bilgilere sahip olunduğunda ya da gelişler arası zaman ya da servis zamanı dağılımının uyum iyiliği ile tespit edilebildiği durumlarda gerektiğinde meydana çıkar. Bütün bu durumlarda, sürecin örneklemeden gerekli bilgi elde edilebilseydi istatistik teorisindeki standart teknikler kullanılabilirdi. Fakat tam bilginin olmadığı çoğu durumlarda bu geçerli değildir. Thiagaragan ve Harris (1979), servis zamanı dağılımı herhangi bir dağılım olan $M/G/1$ kuyruk modelinin hipotez testi için bir yöntem geliştirdiler. Harishchandra ve Rao (1988), $M/M/1$ kuyruk modelinde, servis zamanı esnasında gelen müşterilerin sayısını kullanarak ρ trafik yoğunluğu için olasılık oran testini geliştirdi. Kuyruk teorisinde kullanılabilen testin diğer şekli ardışık olasılık oran testidir.

Bu tez çalışmasında bu bölümde literatür bilgileri verildi. İkinci bölümde konunun temel bilgileri olan Basit Hipotez Testi, Poisson Süreci ve Özellikleri, Ardışık Çözümleme Yöntemi, Markov Zincirleri ve Kuyruk Sistemleri hakkında genel bilgiler verildi. Üçüncü bölümde Rao ve ark. (1984) tarafından incelenen $M/G/1$ ve $GI/M/c$ kuyrukları için Ardışık Olasılık Oran Testine ait yöntem verildi. Dördüncü bölümde ise Ardışık Olasılık Oran Testine ait katsayılar dört farklı faz dağılımlı kuyruk modelleri için bulundu. Beşinci bölümde, üçüncü bölümde verilen kuyruk modellerinden $M/E_2/1$ kuyruk modelinin bir uygulaması yapıldı ve dördüncü bölümde verilen kuyruk modellerinin Matlab programıyla uygulamaları yapılarak farklı örnekler için değişik sonuçlar bulundu. Altıncı bölümde ise bulunan sonuçların değerlendirilmesi yapıldı.

2. GENEL BİLGİLER

Bu bölümde kuyruk sistemlerinde hipotez testlerinin temelini oluşturan basit hipotez, Poisson süreci ve özellikleri, ardışık çözümleme yöntemi, gömülü Markov zinciri, kuyruk teorisi ve bazı kuyruk modelleri hakkında bilgiler verilecektir.

2.1 Basit Hipotez

X rasgele değişkeninin olasılık yoğunluk fonksiyonu $f(x; \theta)$ olarak alınsın ve $H_0: \theta = \theta_0$ hipotezine karşı $H_1: \theta = \theta_1$ basit hipotezi test edilmek istenilsin. Her iki hipotez olasılık yoğunluk fonksiyonunu tam olarak belirler. Dolayısıyla α ve β hata olasılıkları tam olarak belirlenebilir. Kritik bölge $C = \{(X_1, X_2, \dots, X_n) | \bar{X} \geq k\}$ olup, $C = (\bar{X} \geq k)$ olarak alalım. $(X_1, X_2, \dots, X_n)$ örneğinin çekilmesi ile elde edilen $(x_1, x_2, \dots, x_n)$ değerler dizisi için $\bar{X} \geq k$ olursa H_0 hipotezi ret, aksi halde kabul edilir. 1. tip hata olasılığı α olan tüm bölgeler arasında 2. tip hata olasılığı β değerinin minimum olanını, yani $\pi(\theta) = 1 - \beta(\theta)$ güç fonksiyonu maksimum olanını seçeceğiz. Bu koşulu sağlayan kritik bölgeye en iyi kritik bölge ve teste de en iyi test ya da en güçlü test adı verilir. Buna göre en iyi kritik bölge aşağıdaki gibi tanımlanabilir.

Tanım 2.1.1: H_0 hipotezine karşılık H_1 hipotezini test etmek istediğimizi düşünelim.

$$H_0: \theta = \theta_0 ; H_1: \theta = \theta_1$$

α , önceden seçilen anlamlılık düzeyi olmak üzere aşağıdaki iki koşulu sağlaması halinde C ye α düzeyinde en iyi kritik bölge adı verilir.

a) $P(C|H_0) = \alpha$

b) C 'den farklı herhangi başka bir kritik bölge A ve bunlara ait 2. tip hata olasılıkları sırasıyla β ve β' olsun. Başka bir deyişle

$$P(C|H_1) \geq P(A|H_1) \tag{2.1}$$

Son koşul $\beta \leq \beta'$ eşitsizliğini ifade eder. $\beta = 1 - P(C|H_1)$ ve $\beta' = 1 - P(A|H_1)$ 'den istenilen eşitsizlik bulunur.

En iyi kritik bölge Neyman-Pearson teoremi ile tespit edilebilir.

Teorem 2.1.1 (Neyman-Pearson): Yoğunluk fonksiyonu $f(x; \theta)$ olan bir kitleden çekilen n boyutlu bir örnek $(x_1, x_2, \dots, x_n)$ olsun. Örneğin olabilirlik fonksiyonu

$$L(\theta) = L(x_1, x_2, \dots, x_n; \theta) = \prod_{i=1}^n f(x_i; \theta) \quad (2.2)$$

yazılır. $H_0 = \theta = \theta_0$ hipotezine karşı $H_1 = \theta = \theta_1$ hipotezini test etmek istediğimizi düşünelim. Burada $\theta_0 < \theta_1$ dir. $k > 0$ bir reel sayı ve α 'da önceden seçilen anlamlılık düzeyi olmak üzere aşağıdaki 3 koşulu sağlayan bir C bölgesine (C' , C 'nin tümleyeni), eğer varsa, testin α düzeyindeki en iyi kritik bölgesi, teste de en güçlü test denir (Bener, 2002).

$$\begin{aligned} \text{a) } & \alpha = P(C|H_0) \\ \text{b) } & \frac{L(\theta_0)}{L(\theta_1)} \leq k, \quad \forall (x_1, x_2, \dots, x_n) \in C \\ \text{c) } & \frac{L(\theta_0)}{L(\theta_1)} \geq k, \quad \forall (x_1, x_2, \dots, x_n) \in C' \end{aligned} \quad (2.3)$$

2.2 Poisson Süreci ve Özellikleri

a) Tanımı

$\{X_t, t \geq 0\}$ stokastik süreci aşağıdaki 3 özelliği sağlarsa bu sürece Poisson süreci denir (Sağlam, 1997).

$$1) \quad P(X_0 = 0) = 1 \quad (2.4)$$

$$2) \quad X_t \text{ bağımsız artımlı süreçtir,}$$

$$3) \quad \text{Her bir } t \geq 0 \text{ ve } t' > t \text{ için,}$$

$$P(X_{t'} - X_t = k) = \frac{e^{-\lambda(t'-t)} \lambda^{(t'-t)}}{k!}, \quad k = 0, 1, \dots (\lambda > 0) \quad (2.5)$$

b) Özellikleri

1) X_t 'nin ortalaması, varyansı ve kovaryansı sırasıyla

$$\text{i) } EX_t = \sum_{k=0}^{\infty} kP(X_t = k) \quad (2.6)$$

$$= \lambda t$$

$$\text{ii) } VX_t = EX_t^2 - (EX_t)^2$$

$$EX_t^2 = \sum_{k=0}^{\infty} k^2 P(X_t = k) \\ = (\lambda t)^2 + (\lambda t)$$

EX_t ve EX_t^2 değerlerini varyans formülünde yerine yazarsak,

$$VX_t = (\lambda t)^2 + (\lambda t) - (\lambda t)^2 \\ = (\lambda t) \quad (2.7)$$

$$\text{iii) } Cov(X_t, X_u) = \lambda \min(t, u), \quad \forall t, u \text{ için}$$

$$Cov(X_t, X_u) = V(X_t), \quad t < u \text{ için} \quad (2.8)$$

2) $t \rightarrow 0$ iken,

$$\text{i) } P(X_t = 1) = \lambda t + o(t)$$

$$\text{ii) } P(X_t \geq 1) = o(t) \quad (2.9)$$

$$\text{iii) } P(X_t = 0) = 1 - \lambda t + o(t)$$

3) Poisson sürecinin geliş anları

$0 \leq t_1 < t_2 < \dots < t_n$ 'ler Poisson sürecinin geliş anları ya da sıçrayış anlarıdır.

Bu dizi bağımsız artımlıdır.

$$T_1 = t_1 - t_0, T_2 = t_2 - t_1, \dots, T_n = t_n - t_{n-1}$$

i) T_i 'ler bağımsızdır,

ii) T_i 'ler λ parametrelili üstel dağılıma sahiptir.

$$P(T_i < x) = \begin{cases} 1 - e^{-\lambda x} & x \geq 0 \\ 0 & d.d. \end{cases} \quad (2.10)$$

- 4) $t_1 < t_2 < \dots < t_n$ geliş anlarının ortak olasılık yoğunluk fonksiyonları (2.11)'deki formülle ifade edilir.

$$f(x_1, x_2, \dots, x_n) = \lambda^n e^{-\lambda x_n} \quad (x_1 < x_2 < \dots < x_n) \quad (2.11)$$

- 5) λ parametrelili poisson sürecinde n . geliş anı n ve λ parametrelili Erlang dağılımına sahiptir. Yani t_n 'nin olasılık yoğunluk fonksiyonu

$$G(n; \lambda) = f_n(x) = \frac{\lambda(\lambda x)^{n-1}}{(n-1)!} e^{-\lambda x} \quad x \geq 0, \quad \lambda \geq 0 \quad (2.12)$$

$n = 1$ için $f_1(x) = \lambda e^{-\lambda x}$, $x \geq 0$ üstel dağılıma dönüşür.

- 6) $X(t)$, $t \geq 0$ Poisson sürecidir. $t_1 < t_2 < \dots < t_\infty$ 'ler ise bu sürecin geliş anlarıdır. Her bir geliş anı p olasılığı ile sağlanıyor ya da $1-p$ olasılığı ile sağlanmıyor. N_t ile $[0, t]$ aralığında yerleşen ve sağlanan geliş anlarının sayısını ifade edelim. Bu durumda, N_t , λp parametrelili poisson sürecidir. $[0, t]$ aralığında gelen müşteri sayısı X_t , λ parametrelili Poisson dağılımına sahiptir.

N_t 'nin poisson süreci olması için;

$$i) N(t) = 0 = X(0),$$

$$ii) N(t) \text{ bağımsız artımlı süreç,}$$

$$iii) P(N_t = k) = \frac{(\lambda p t)^k e^{-\lambda p t}}{k!} \quad (2.13)$$

özelliklerini sağlamalıdır.

- 7) Farz edelim X_t ve Y_t sırasıyla λ_1 ve λ_2 parametrelili bağımsız Poisson sürecidir. O halde $X_t + Y_t$ toplamı $\lambda_1 + \lambda_2$ parametrelili poisson süreci olacaktır.

2.3 Ardışık Çözümleme

Günümüzde yapılan araştırmaların çoğunda, örneklem büyüklüğü, önceden belirlenen sabit bir değerdir. Çeşitli örnekleme yöntemleriyle, belirlenen örneklem büyüklüğünde bir örneklem seçilir. Seçilen örnekleme, araştırma konusuna uygun deneme ya da anket uygulanarak veriler elde edilir. Elde edilen verilere de istatistiksel yöntemler uygulanıp kitle hakkında sonuçlara ulaşılabilir. Bu çözümleme yöntemi, çoğu kez, zaman ve para bakımından maliyetin yüksek olmadığı alanlarda kullanılmaktadır.

Ardışık çözümleme yönteminde ise denemenin (araştırmanın) sonuna kadar beklemek gerekmez. Bunun yerine, deneme sürerken veriler çözümlemeye alınabilir. Bazı araştırmaların yapısı gereği ardışık çözümleme yönteminin uygulanması daha uygundur. Örneğin hastaneye başvuran hastalar, bir fabrikada üretilen ürünler, vb. gibi. Ardışık çözümleme yöntemiyle, bir raslantı değişkeni olan örneklem büyüklüğünün beklenen değeri elde edilebilir. Ortalama örneklem sayısı adı verilen bu değer, örneklem büyüklüğü bilinmediği halde, kitle parametre değerine göre, en az ve en çok ne kadarlık bir örneklem büyüklüğüne gerek duyulacağı hakkında ön bilgi verir (Çıngı, 2005).

Kaynaklarda ardışık çözümleme ardışık olasılık oran testi (sequential probability ratio test), ardışık hipotez testi (sequential hypothesis test), ardışık test (sequential test), ardışık çözümleme (sequential analysis) başlıkları altında geçmektedir.

X raslantı değişkeni, ardı ardına gelen $x_1, \dots, x_m$ değerlerini alsın. $f_\theta(x)$ olasılık yoğunluk fonksiyonunun parametresine ilişkin,

$$H_0: \theta = \theta_0, H_1: \theta = \theta_1, (\theta_0 < \theta_1)$$

basit hipotezleri kurulabilir. Burada H_1 , H_0 'a karşıt hipotezdir. Ardışık çözümleme yöntemiyle H_0, H_1 hipotezleri test edilebilir. H_0 hipotezi doğru iken o.y.f. $f_{\theta_0}(x)$, H_1 hipotezi doğru iken $f_{\theta_1}(x)$ ile gösterilir. X raslantı değişkeninin aldığı değerler birbirlerinden bağımsız iseler,

$$f_{\theta_0} = f_{\theta_0}(x_1, \dots, x_m) = f_{\theta_0}(x_1) \dots f_{\theta_0}(x_m), \quad H_0 \text{ doğru iken}$$

$$f_{\theta_1} = f_{\theta_1}(x_1, \dots, x_m) = f_{\theta_1}(x_1) \dots f_{\theta_1}(x_m), \quad H_1 \text{ doğru iken,}$$

yazılabilir. H_0 basit hipotezini H_1 karşıt hipotezine göre, ardışık olasılık oran testi ile incelemek için, o.y.f'ları oranı olan olabilirlik oranı λ_m 'den yararlanılır. İncelenecek hipotez basit olduğunda en güçlü test λ_m olabilirlik oranına dayanarak yapılan testtir (Mood,1974). Olabilirlik oranı,

$$\lambda_m = \prod_{i=1}^m \frac{f_{\theta_1}(x_i)}{f_{\theta_0}(x_i)} \quad (2.14)$$

şeklinde tanımlanır.

Ardışık çözümlemede, testi gerçekleştirmek amacıyla, α , β birinci ve ikinci tip hata olasılıklarının belirlenmesi gerekir.

$$\alpha = P(H_0 \text{ ret}/H_0 \text{ doğru iken})$$

$$\beta = P(H_1 \text{ ret}/H_1 \text{ doğru iken})$$

biçimindedir.

Ardışık Olasılık Oran Testi için α ve β 'ya bağlı olan A ve B ($0 < B < A < \infty$) sabit değerlerinin belirlenmesi gerekir. $A = (1 - \beta)/\alpha$ ve $B = \beta/(1 - \alpha)$ olarak alınır. Daha sonra, aşağıda verilen üç karardan birine ulaşılabilir:

1. $\lambda_m \leq B$ ise, H_0 kabul edilerek sürece son verilir.
2. $\lambda_m \geq A$ ise, H_0 ret edilerek sürece son verilir.
3. $B < \lambda_m < A$ ise, gözlemleri yetersizliğine karar verilip bir gözlem daha eklenerek süreç sürdürülür.

λ_m oranı, A ve B sabit değerlerine dayanarak yapılan yukarıdaki teste “ardışık olasılık oran testi” denir. λ_m oranı, logaritma yardımıyla daha kolay olarak bulunabileceğinden olabilirlik oranının logaritması alınarak,

$$\ln \lambda_m = \sum_{i=1}^m \ln \left[\frac{f_{\theta_1}(x_i)}{f_{\theta_0}(x_i)} \right] \quad (2.15)$$

elde edilir. Burada

$$Z_i = \ln \left[\frac{f_{\theta_1}(x_i)}{f_{\theta_0}(x_i)} \right] \quad (2.16)$$

ile gösterilirse yukarıdaki eşitlik (2.17) biçimine dönüşür.

$$\ln \lambda_m = \sum_{i=1}^m Z_i \quad (2.17)$$

Böylece m 'inci aşamada $\ln \lambda_m$ bulunarak A ve B değerlerinin e tabanındaki logaritmalarıyla karşılaştırılır.

1. $\sum_{i=1}^m Z_i \leq \ln B$ ise, H_0 kabul edilerek sürece son verilir.
2. $\sum_{i=1}^m Z_i \geq \ln A$ ise, H_0 ret edilerek sürece son verilir.
3. $\ln B < \sum_{i=1}^m Z_i < \ln A$ ise, gözlemlerin yetersizliğine karar verilerek süreç sürdürülür.

2.3.1 Poisson dağılımı için Wald tipi ardışık çözümleme

Poisson dağılımlı bir kitlede $H_0: \lambda = \lambda_0$, $H_1: \lambda = \lambda_1$, ($\lambda_0 < \lambda_1$) hipotezini ardışık olasılık oran testi ile test edelim.

Poisson dağılımı

$$f_\lambda(x) = \frac{e^{-\lambda} \lambda^x}{x!}, \quad x = 0, 1, 2, \dots$$

olduğuna göre H_0 ve H_1 hipotezi doğru olduğunda maksimum olabilirlik oranı,

$$\begin{aligned} \lambda(m) &= \prod_{i=1}^m \frac{f_{\lambda_1}(x_i)}{f_{\lambda_0}(x_i)} \\ &= \prod_{i=1}^m \frac{e^{-\lambda_1} \lambda_1^{x_i}}{e^{-\lambda_0} \lambda_0^{x_i}} \\ &= \prod_{i=1}^m e^{\lambda_0 - \lambda_1} \left(\frac{\lambda_1}{\lambda_0} \right)^{x_i} \end{aligned}$$

eşitliğinin logaritması alınarak

$$\begin{aligned} \ln \lambda(m) &= \sum_{i=1}^m \left[(\lambda_0 - \lambda_1) + x_i \ln \left(\frac{\lambda_1}{\lambda_0} \right) \right] \\ &= m(\lambda_0 - \lambda_1) + \ln \left(\frac{\lambda_1}{\lambda_0} \right) \sum_{i=1}^m x_i \end{aligned} \quad (2.18)$$

biçiminde elde edilir.

Eğer,

$$m(\lambda_0 - \lambda_1) + \ln\left(\frac{\lambda_1}{\lambda_0}\right) \sum_{i=1}^m x_i \leq \ln B \quad (2.19)$$

ise H_0 kabul edilerek sürece son verilir. Yukarıdaki eşitsizlik $\sum_{i=1}^m x_i$ 'ye göre düzenlenirse,

$$\sum_{i=1}^m x_i \leq \frac{\ln B + m(\lambda_1 - \lambda_0)}{\ln\left(\frac{\lambda_1}{\lambda_0}\right)} \quad (2.20)$$

olmaktadır. Benzer şekilde,

$$m(\lambda_0 - \lambda_1) + \ln\left(\frac{\lambda_1}{\lambda_0}\right) \sum_{i=1}^m x_i \geq \ln A$$

ise H_0 ret edilerek sürece son verilir. Yukarıdaki eşitsizlik $\sum_{i=1}^m x_i$ 'ye göre düzenlenirse,

$$\sum_{i=1}^m x_i \geq \frac{\ln A + m(\lambda_1 - \lambda_0)}{\ln\left(\frac{\lambda_1}{\lambda_0}\right)}$$

bulunur. $(m, \sum_{i=1}^m x_i)$ noktasına göre karar verilir.

2.4 Markov Zincirleri

Stokastik süreç $\{X_t, t \in T\}$ raslantı değişkeninin kümesi olarak tanımlanır. Sürecin parametresi T 'ye de parametre veya indis kümesi denir. Süreç eğer;

$$T = \{0, \mp 1, \mp 2, \dots\} \text{ ya da } T = \{0, 1, 2, \dots\}$$

gibi kesikli değerler alıyorsa sürece kesikli parametrelili süreç,

$$T = \{t: -\infty < t < +\infty\} \text{ ya da } T = \{t: t \geq 0\}$$

gibi sürekli değerler alıyorsa sürekli parametrelili süreç adı verilir.

X_t raslantı değişkeninin değer bölgesine durum uzayı denilir. Genellikle durum uzayı E harfi ile gösterilir. X_t 'nin alabileceği her değer bir "durum" dur.

Diğer bir deyişle eğer $h > 0$ için, $P(x - h < X_t < x + h)$ olasılığı pozitif olacak şekilde T 'de bir t değeri varsa, x gerçel sayısına stokastik sürecin bir olası değeri ya da durumu denir. X_t raslantı değişkeni kesikli ise, stokastik süreç kesikli durum süreci, sürekli ise sürekli durum süreci adını alır.

$\{X_t, t \geq 0\}$, durum uzayı $E = \{0, 1, \dots, N\}$ olan sürekli parametrelili bir süreç olsun. Aşağıdaki özellik geçerli olduğunda bu sürece Markov zinciri denir. Her bir $0 \leq t_1 < \dots < t_k < t$ ve $n_1, \dots, n_k \in D$ için,

$$P(X_t = j / X_{t_1} = n_1, \dots, X_{t_k} = n_k) = P(X_t = j / X_{t_k} = n_k)$$

dir. Bu tanıma göre, Markov sürecinin herhangi bir t anında durumu belli olduğunda $\{X_s, s < t\}$ ve $\{X_s, s > t\}$ birbirinden bağımsızdır. $\{X_t, t \geq 0\}$ Markov zinciri için,

$$p_{ij}(t) = P(X_t = j / X_0 = i)$$

ihtimaline i durumundan j durumuna geçiş ihtimali denir. Bunlar aşağıdaki özellikleri sağlar: Her bir $i, j \in D$ için

$$p_{ij}(t) \geq 0, \quad \sum_{j=0}^N p_{ij}(t) = 1$$

dir. Kabul edelim ki

$$p_{ij}(0) = \delta_{ij} = \begin{cases} 1, & i = j \\ 0, & i \neq j. \end{cases}$$

Varsayalım ki,

$$a_{ij} = p'_{ij}(0) = \lim_{h \rightarrow 0} \frac{p_{ij}(h) - p_{ij}(0)}{h} \quad (2.21)$$

türevleri mevcuttur. a_{ii} 'ye i . durumundan çıkış oranı, $a_{ij}(i \neq j)$ 'ye ise i . durumdan j . duruma geçiş oranı denir ve bunlar aşağıdaki özellikleri sağlarlar:

$$a_{ij} \geq 0 \ (i \neq j), \quad a_{ii} = - \sum_{j \neq i} a_{ij} \leq 0 \quad (2.22)$$

$p_{ij}(t)$ geçiş ihtimalleri için diferansiyel denklemler elde edelim. $\{X_t, t \geq 0\}$, durum uzayı $E = \{0, 1, \dots, N\}$ olan bir Markov zinciri olsun. Bunun

$$p_j(t) = P(X_t = j), \quad j \in D$$

ihtimali için (2.23)'deki diferansiyel denklemler (Kolmogorov denklemleri) doğrudur:

$$p'_j(t) = \sum_{i=0}^N p_i(t) a_{ij}, \quad (i, j \in D), \quad (2.23)$$

limiti mevcuttur ve

$$\sum_{i=0}^N p_i a_{ij} = 0, \quad \sum_{j=0}^N p_j = 1 \quad (j = 0, 1, \dots, N)$$

denklem sisteminin çözümü olarak bulunur (Gross and Harris, 1985).

τ_i , Markov zincirinin i . durumda kalma süresi olsun. Bu takdirde τ_i üstel dağılıma sahiptir, yani

$$P(\tau_i < t) = \begin{cases} 1 - e^{-\lambda_i t}, & t \geq 0 \\ 0, & \text{d.d.} \end{cases}$$

2.4.1 Gömülü Markov zinciri

$\{X(t), t \geq 0\}$ olmak üzere sürekli parametrelili Markov zincirinin herhangi bir durumunda gerçekte geçirilen zamanı görmezden gelirsek ve sadece gerçekte ortaya çıkan geçiş dizisini dikkate alırsak gömülü Markov zinciri olarak adlandırabileceğimiz yeni bir kesikli Markov zinciri tanımlayabiliriz.

Sürekli parametrelili bir Markov zinciri tarafından ziyaret edilen n . durumu Y_n ile gösterelim. O zaman $\{Y_n, n = 0, 1, 2\}$ sürekli Markov zincirinden türetilen Gömülü Markov zinciridir. Yutucu duruma sahip olmayan sürekli Markov zinciri için, gömülü Markov zincirinde bir adım geçiş olasılığı $w_{ij} = P(Y_{n+1} = j / Y_n = i)$ ile gösterilirse, $i = j$ olduğunda geçiş ihtimali sıfıra eşit ve $i \neq j$ olduğunda geçiş ihtimali

$$w_{ij} = \frac{q_{ij}}{\sum_{j \neq i} q_{ij}} \quad (2.24)$$

dir. Q matrisi sürekli Markov zincirinin geçiş oranları matrisi ve q_{ij} bu matrisin ij . elemanıdır. Yutucu durum olmadığı için paydadaki $\sum_{j \neq i} q_{ij}$ sıfır olamaz. Eğer gömülü Markov zincirinin geçiş olasılık matrisini W ile gösterirsek, o zaman

$$W = I - D_Q^{-1}Q \quad (2.25)$$

matrisine sahip oluruz. Burada $D_Q = \text{diag}(Q)$, köşegen elemanları sıfır olmayan diagonal matris Q diagonal matrisinin elemanlarına eşittir. W 'nin tüm elemanları $0 \leq w_{ij} \leq 1$ aralığındadır ve tüm i 'ler için $\sum_{j, j \neq i} w_{ij} = 1$ dir. Böylece W kesikli Markov zinciri için geçiş ihtimali matrisinin özelliklerine sahiptir.

2.5 Kuyruk Sistemleri

Kuyruk teorisi iletişim, bilgisayar dizaynı, üretim süreçleri, ulaşım ve servis sistemlerinde bir çok uygulamaya sahiptir. Günlük hayat da hizmet alabilmek için kuyrukta bekleyen insanlar ya da araçlarla karşılaşmak her zaman mümkündür. Örneğin hastanelerdeki hasta kuyruğu, otomatik para çekme makinelerinde maaş almak için bekleyenlerin oluşturduğu kuyruğun yanı sıra askeri alanlarda da kuyruk teorisinin bir çok uygulaması mevcuttur (Pala, Beloso ve Hines, 1966).

Yöneylem Araştırması ve Sistem Analizi disiplinlerinin en önemli yöntemlerinden biri olan kuyruk teorisinin ortaya çıkış tarihi her iki disiplinden çok öncelere gider. Bekleme hattı problemini inceleyen ilk çalışmalar 1909 yılında A.K. Erlang tarafından telefon şebekeleri üzerinde yapılmıştır.

Kuyruk teorisindeki temel aktörler müşteri ve hizmet verendir. Müşteriler bir kaynaktan gelir. Müşteri hizmet yerine vardığında hemen hizmet alır ya da kanal (server) meşgulse kuyrukta bekler. Bir kanal hizmetini tamamladığında sırada bekleyen müşteri varsa bu müşteriye hizmete alır. Kuyruk boş ise hizmet yeri yeni müşteri gelinceye kadar boş kalır.

Kuyruk sisteminin yapısı aşağıdaki beş kısımdan oluşur.

- 1) Müşteri gelişi
- 2) Hizmet süresi
- 3) Kanal sayısı
- 4) Sistemdeki maksimum müşteri sayısı
- 5) Hizmet disiplini

Çeşitli kuyruk sistemleri vardır. En yaygın ve basit olanı gelişler Poisson ve hizmet süresi üstel dağılmış olan tek kanallı bir kuyruk sistemi olan $M/M/1$ ve çok kanallı $M/M/c$ dir (Torun, 2002).

c kanallı bir kuyruk sisteminde müşterilerin sisteme geliş anları $t_1, t_2, \dots, t_{n-1}, t_n$ olsun. t_n , n . müşterinin sisteme geliş anı olmak üzere u_n 'ler raslantı değişkenleridir. $\{u_1, u_2, \dots, u_n\}$ dizisine müşteri akımı denir. $u_1, u_2, \dots, u_n$ 'lerin bağımsız ve aynı dağılıma sahip (i.d.d.) olduğu varsayılır. $u_i (i = 1, 2, \dots, n)$ 'ler i.d.d. sahip olsun. $u_i (i = 1, 2, \dots, n)$ 'lerin dağılım fonksiyonunu $P(u_i < t) = A(t)$ ile gösterelim. $A(t)$ keyfi yada belli bir olasılık fonksiyonu olabilir. $A(t)$ keyfi olduğunda giriş akımına recurent akım denir ve GI ile gösterilir.

En çok kullanılan giriş akımları:

$$1) \quad \text{Eğer } A(t) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

ise buna Poisson akımı denir. Bu akım M ile ifade edilir.

$$2) \quad u_i = D \text{ (sabit deterministik) Her bir } u_i \text{ sabittir.}$$

Bu tür akıma Deterministik Giriş Akımı denir ve bu akım D ile gösterilir.

$$3) \quad u_i = \eta_1 + \eta_2 + \dots + \eta_k$$

$\eta_1, \eta_2, \dots, \eta_k$ 'lar bağımsızdır ve aynı üstel dağılımına sahiptir.

$$P(\eta_k < t) = 1 - e^{-\lambda t}$$

Bu tür dağılıma Erlang akımı denir ve bu giriş akımı E_k ile gösterilir. $A(t)$ 'nin verilmesiyle giriş akımı belirlenir.

Hizmet süresi sisteme gelen her bir müşteri belli bir zaman içinde hizmet alıyorsa bu zamana hizmet süresi denir. Sistemin tam belli olabilmesi için hizmet süresinin verilmesi gerekir. Bu süre genellikle bir raslantı değişkenidir.

$\xi_i : i$ 'ci müşterinin hizmet süresini göstermek üzere, $\xi_1, \xi_2, \dots$ 'ler bağımsız raslantı değişkeni olup aynı dağılıma sahiptirler.

Hizmet süresinin dağılım fonksiyonu $P(\xi_i < t) = B(t)$ ile gösterilsin. $B(t)$ keyfi olduğunda hizmet süresi G ile gösterilir. Özel olarak $B(t)$ aşağıdaki gibi seçilebilir,

$$1) \xi_i \text{'ler üstel dağılıma uyarsa } B(t) = \begin{cases} 1 - e^{-\mu t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

$$2) \xi_i \text{'ler sabitse } \xi_i = D = a \leq b(t)$$

$$B(t) = \begin{cases} 1 & a < t \\ 0 & a \geq t \end{cases}$$

3) ξ_i 'ler Erlang dağılımına uyarsa $\xi_i = E_k$ olur.

$$B(t) = E_k(t).$$

Hizmet disiplini: Sisteme gelen müşteri belli bir kurala göre hizmet alacaktır. Bu kurala hizmet disiplini denir. Çeşitli hizmet disiplinleri kullanılmaktadır (Taha, 2000).

- a) FIFO (İlk gelen ilk hizmeti alır)
- b) LIFO (Son gelen hizmeti alır)
- c) Random (Tesadüfi olarak)
- d) Priority (Öncelikli müşteri ne zaman sisteme gelirse beklemeden hizmetini alır)

c kanaldan birkaçı boş olabilir ve müşteri bir tanesinde hizmet almaya başlar; yani kanallardan birinde yer alır. Eğer kanalların hepsi dolu ise bekleme hattı vardır. Bekleme hattı sonluda olabilir sonsuzda olabilir. Bu müşteri, bekleme hattının sonunda yer alır. FIFO'da bekleme hattının en önünde yer alan hizmeti alır, LIFO'da bekleme hattının sonunda yer alan ilk hizmeti alır, Random'da ise tesadüfi olarak hizmet alır. Bekleme hattındaki müşteri sonlu da olabilir sonsuz da olabilir. Gelen müşteri sistemde N 'den fazla müşteriyle karşılaşarsa hizmet almadan sistemi terk eder.

Servis sistemlerinin göstergeleri: Her bir servis sisteminin karakteristiklerini özetlemek için aşağıdaki notasyon kullanılır.

$A/B/c/N/d$

Burada,

A :Gelişler arası sürenin dağılım fonksiyonunu (giriş akımını) gösterir.

B : Hizmet süresinin dağılım fonksiyonunu gösterir.

c : Kanal sayısı

N : Kuyruktaki en fazla müşteri sayısı

d : Hizmet disiplini

2.5.1 M/G/1 Kuyruk modeli

Müşteri gelişleri λ parametrelili Poisson süreci ve müşteriler tek kanaldan servis almış olsun. Servis zamanlarının dağılım fonksiyonu $P(S_n \leq x) = B(x)$ olmak üzere, $\{S_n, n = 1, 2, 3, \dots\}$ bağımsız ve aynı dağılıma sahip rasgele değişkenlerden meydana gelsin. S_n , n . müşterinin servis zamanı olsun. $Q(t)$, t zamanında sistemdeki müşteri sayısı olsun ve $t_0=0$ olmak üzere $t_1, t_2, \dots$ müşterilerin ayrılış zamanları olsun. Yukarıda tanımlandığı gibi bu ayrılış noktalarında müşterilerin kalan servis zamanları sıfırdır. n . ayrılıştan hemen sonra sistemde kalan müşteri sayısı $Q_n = Q(t_n + 0)$ olarak alınır. Q_n 'nin $n = 0, 1, 2, \dots$ noktalarında bir Markov zinciri olduğunu gösterebiliriz (Bhat, 2008).

X_n 'i S_n 'inci servis anındaki müşteri sayısı olarak alalım. Geliş sürecinin Poisson olduğu varsayımıyla,

$$\frac{B(t + dt) - B(t)}{dt} = b(t) \text{ ve } P(t \leq S_n \leq t + dt) = B(t + dt) - B(t) = b(t)dt$$

olmak üzere,

$$\begin{aligned} k_j &= P(X_n = j) = \int_0^\infty P(X_n = j | S_n) P(t \leq S_n \leq t + dt) \\ &= \int_0^\infty P(X_n = j | S_n) b(t) dt \\ &= \int_0^\infty e^{-\lambda t} \frac{\lambda t^j}{j!} dB(t), \quad j = 0, 1, 2, \dots \end{aligned} \tag{2.26}$$

dir.

Q_n ve Q_{n+1} arasındaki ilişkiyi dikkate aldığımızda,

$$Q_{n+1} = \begin{cases} Q_n + X_{n+1} - 1 & \text{Eğer } Q_n > 0 \text{ ise} \\ X_{n+1} & \text{Eğer } Q_n = 0 \text{ ise} \end{cases} \tag{2.27}$$

elde ederiz.

(2.27)'den görebiliriz ki X_{n+1} rasgele değişkeni t_n 'den önceki herhangi bir olaya bağlı değildir. X_{n+1} i.i.d. olduğu için ne de Q_n 'e bağlıdır. Markov özelliğinin bir adım bağımlılığı geçerli olur ve böylece $\{Q_n, n = 0, 1, 2, \dots\}$ bir Markov

zinciridir. Onun parametre uzayı ayrılış noktalarında oluşturulur ve $E = \{0,1,2, \dots\}$ olmak üzere E sistemdeki müşterilerin sayısını gösteren durum uzayı olmak üzere, Q_n gömülü Markov zinciri olarak bilinir.

$$P_{ij}^{(n)} = P(Q_n = j | Q_0 = i), \quad i, j \in E \quad (2.28)$$

alınırsa

$$P_{ij}^{(1)} = P_{ij}$$

yazılır. (2.26) ve (2.27)'deki k_j 'nin tanımından aşağıdaki ifadeleri yazabiliriz.

$$\begin{aligned} P_{ij} &= P(Q_{n+1} = j | Q_n = i) \\ &= \begin{cases} P(i + X_{n+1} - 1 = j) & \text{Eğer } i > 0 \text{ ise} \\ P(X_{n+1} = j) & \text{Eğer } i = 0 \text{ ise} \end{cases} \\ &= \begin{cases} k_{j-i+1} & \text{Eğer } i > 0 \\ k_j & \text{Eğer } i = 0 \end{cases} \end{aligned} \quad (2.29)$$

Markov zinciri için geçiş olasılığı matrisi

$$P = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & \dots \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ . \end{matrix} & \begin{bmatrix} k_0 & k_1 & k_2 & \dots \\ k_0 & k_1 & k_2 & \dots \\ 0 & k_0 & k_1 & \dots \\ 0 & 0 & k_0 & \dots \\ 0 & 0 & 0 & \dots \end{bmatrix} \end{matrix} \quad (2.30)$$

olarak elde edilir.

2.5.2 $M/G/1/N$ Kuyruk modeli

Yukarıda $M/G/1$ kuyruk modeli tanıtıldı. Şimdi sistemdeki müşteri sayısı için kapasite N olsun. Çünkü gömülü Markov zinciri için durum uzayı ayrılıştan hemen sonra sistemdeki sayıdır ve N durum uzayına dahil değildir. Bundan dolayı durum uzayı $E = \{0,1,2,\dots,N-1\}$ dir. Böylece (2.27)'deki ifadenin karşılığı

$$Q_{n+1} = \begin{cases} \min(Q_n + X_{n+1} - 1, N - 1) & \text{Eğer } Q_n > 0 \text{ ise} \\ \min(X_{n+1}, N - 1) & \text{Eğer } Q_n = 0 \text{ ise} \end{cases} \quad (2.31)$$

olur. k_j 'nin olasılık dağılımı kullanılarak aşağıdaki geçiş matrisi elde edilir.

$$P = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & \dots & N-1 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ . \\ . \\ . \\ N-1 \end{matrix} & \begin{bmatrix} k_0 & k_1 & k_2 & \dots & 1 - \sum_0^{k-2} k_j \\ k_0 & k_1 & k_2 & \dots & 1 - \sum_0^{k-2} k_j \\ 0 & k_0 & k_1 & \dots & 1 - \sum_0^{k-3} k_j \\ 0 & 0 & k_0 & \dots & . \\ 0 & 0 & 0 & \dots & . \\ 0 & 0 & 0 & \dots & . \\ 0 & 0 & 0 & \dots & 1 - k_0 \end{bmatrix} \end{matrix} \quad (2.32)$$

$G/M/1/N$ kuyruğunda P geçiş matrisinin hesaplanmasına ile ilgili bir örnek Ek B'de verilmiştir.

2.5.3 $G/M/1$ Kuyruk modeli

$t_0 = 0, t_1, t_2, \dots$ zaman anında gelen müşteriler olsun ve tek servisten hizmet alsınlar. $n = 1, 2, 3, \dots$ değerleri için $Z_n = t_{n+1} - t_n$ farkı a ortalamalı $A(\cdot)$ dağılım fonksiyonlu bağımsız ve aynı dağılıma sahip rasgele değişkenler olarak dağılsın. Ayrıca servis zamanı dağılımını $1/\mu$ ortalamasıyla alalım. Sistem geleneksel olarak $GI/M/1$ (GI = genel bağımsız) sembolüyle gösterilir ve $G/M/1$ şekline de indirgenebilir.

Herhangi bir t zamanındaki müşteri sayısı $Q(t)$ olsun. $n = 1, 2, 3, \dots$ için $Q(t_n - 0) = Q_n$ olarak tanımlayalım. Böylece Q_n n . gelişten önceki hali hazırdaki müşteri sayısı olur. X_n 'i Z_n zaman periyodunda gelişler arasındaki servis tamamlayanların sayısı olarak tanımlayalım O zaman

$$b_j = P(X_j = j) = \int_0^\infty e^{-\lambda t} \frac{\lambda t^j}{j!} dA(t) \quad (2.33)$$

olur. Q_n ile Q_{n+1} arasındaki ilişkileri dikkate aldığımızda aşağıdaki ifadeyi yazabiliriz.

$$Q_{n+1} = \begin{cases} Q_n + 1 - X_{n+1} & \text{Eğer } Q_n + 1 - X_{n+1} > 0, \\ 0 & \text{Eğer } Q_n + 1 - X_{n+1} \leq 0. \end{cases} \quad (2.34)$$

X_{n+1} ayrılışların potansiyel sayısı olduğu için $Q_n + X_{n+1} - 1 < 0$ olmalıdır. Açıkça Q_{n+1} n 'den daha önceki index parametrelili herhangi rasgele değişkenine bağlı değildir; böylece $n = 0, 1, 2, \dots$ değerleri için Q_n kuyruk uzunluğu sürecinde gömülü bir Markov zinciridir. (2.34)'den

$$P_{ij} = b_{i-j+1}, \quad j > 0, \\ P_{i0} = \sum_{r=i+1}^{\infty} b_r \quad (2.35)$$

alındığında

$$P_{ij} = P(Q_{n+1} = j | Q_n = i) \\ P_{ij} = \begin{cases} P(X_{n+1} = i - j + 1) & \text{Eğer } j > 0 \text{ ise} \\ P(X_{n+1} \geq i + 1) & \text{Eğer } j = 0 \text{ ise} \end{cases} \quad (2.36)$$

geçiş ihtimallerini elde ederiz. Geçiş olasılığı matrisi,

$$P = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & \dots \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ \vdots \\ \vdots \end{matrix} & \begin{bmatrix} \sum_1^{\infty} b_r & b_0 & 0 & 0 & \dots \\ \sum_2^{\infty} b_r & b_1 & b_0 & 0 & \dots \\ \sum_3^{\infty} b_r & b_2 & b_1 & b_0 & \dots \\ . & . & . & . & \dots \\ . & . & . & . & \dots \\ . & . & . & . & \dots \end{bmatrix} \end{matrix} \quad (2.37)$$

olarak elde edilir.

Burada dikkatimizi müşterilerin servisten ayrılış noktalarında kalan müşteri sayısı ile ya da iki geliş arası servis alan müşteri sayıları sınırlandıracağız. Bu şekilde ayrılış noktalarında ya da iki geliş arasında sistemi gözlemlemek daha az bilgi tutulmasını gerektirdiğinden gözlemlenen değerler diğer kuyruk sistemlerini gözlemlerken tutulan bilgi miktarından daha az olmaktadır yani gelişler arası süreleri, servisler arası geçen süreleri vs. tutmaya gerek yoktur. Bu şekilde $M/G/1$, $GI/M/1$, $GI/M/c$, $M^k/G/1$ gibi kuyruk sistemleri Markov özelliğine sahip olmamasına rağmen gömülü Markov zinciri şeklinde tanımlanabilmektedirler.

3. KUYRUK SÜRECİ VE ARDIŞIK OLASILIK ORAN TESTİ

Bir çok kuyruk uygulamasının büyük önem ve ilgiye sahip performans özelliği ortalama geliş oranı λ 'nın ortalama servis oranı μ 'ye oranı olan trafik yoğunluğudur. Rao ve ark. (1984) gözlem olarak sistemdeki müşteri sayılarını kullanan $M/G/1$ ve $GI/M/c$ kuyruk sistemlerinde trafik yoğunluğundaki değişiklikleri belirlemek için Ardışık Olasılık Oran Testini kullanmışlardır. Bu testin amacı trafik yoğunluğundaki değişiklikleri mümkün olduğu kadar hızlı tespit etmek ve uygun düzeltici eylemleri almaktır. Yani trafik yoğunluğunu istenilen düzeyde (ρ_0) tutmaktır. Örneğin trafik yoğunluğu ρ , istenilen düzey ρ_0 'dan ρ_1 'e değiştiğinde ($\rho_1 > \rho_0$), uygun eylem ρ_1 'den istenilen düzey ρ_0 'a trafik yoğunluğunu geri getirmektir. Bunu sağlamak için servis oranı μ artırılabilir ya da kanal sayısı artırılabilir. Gelişleri kontrol edemediğimiz için servis oranında bir değişikliğe gitmek gerekir. Bu değişimi aşağıdaki şekilde inceleriz.

$t_0 < t_1 < t_2 < \dots < t_n$ olmak üzere sürecin durumunun değişme anlarında Markov özelliği sergileyen $Q(t)$ sürecini alalım, yani

$$\begin{aligned} P\{Q(t_{n+1}) = j \mid Q(t_n) = i, Q(t_{n-1}) = i_{n-1}, \dots, Q(t_0) = i_0\} \\ = P\{Q(t_{n+1}) = j \mid Q(t_n) = i\} \end{aligned}$$

dir.

$M/G/1$ kuyruğunda t_n n . müşterinin servis tamamlama süreci olsun ve $GI/M/c$ kuyruğunda t_n , n . müşterinin geliş aralığı olsun. $M/M/c$ kuyruğu sürekli zaman Markov özelliği olmasına rağmen kontrol için geliş noktalarında süreci incelemek kolaydır.

Kolaylık için $Q_n = Q(t_n)$ şeklinde kullanırız. $E = \{0, 1, 2, \dots, N\}$ durum uzayında tanımlı Markov zincirinin geçiş ihtimali matrisi,

$$P_{ij}(\rho) = P\{Q_{n+1} = j \mid Q_n = i; \rho\}$$

şeklindedir.

ρ parametresinin değeri hariç P olasılık matrisi varsayalım. $H_1: \rho = \rho_1$ alternatif hipotezine karşı $H_0: \rho = \rho_0$ hipotezini test edelim.

$Q_0, Q_1, Q_2, \dots, Q_n$ gözlemlerinin dizisini ele alalım. H_0 ve H_1 hipotezleri altında bu gözlem dizisinin birleşik olasılığı,

$$\Pr \{Q_0, Q_1, Q_2, \dots, Q_n; \rho_i\} = P(Q_0; \rho_i) \prod_{j=1}^n P(Q_j \setminus Q_{j-1}; \rho_i), \quad i = 0, 1$$

şeklinde verilir. O zaman olabilirlik oranı,

$$L = \frac{P(Q_0; \rho_1) \prod_{j=1}^n P(Q_j \setminus Q_{j-1}; \rho_1)}{P(Q_0; \rho_0) \prod_{j=1}^n P(Q_j \setminus Q_{j-1}; \rho_0)} \quad (3.1)$$

dır.

$$Z_0 = \ln \frac{P(Q_0; \rho_1)}{P(Q_0; \rho_0)} \quad (3.2)$$

ve

$$Z_r = \ln \frac{P(Q_r \setminus Q_{r-1}; \rho_1)}{P(Q_r \setminus Q_{r-1}; \rho_0)}, \quad (r \geq 1) \quad (3.3)$$

olarak alalım. Bu takdirde $H_1: \rho = \rho_1$ alternatif hipotezine karşı $H_0: \rho = \rho_0$ hipotezini test etmek için Wald (1947)'ın Ardışık Olasılık Oran Testini kullanırız. α ve β birinci ve ikinci tip hata olasılıkları olmak üzere $A = (1 - \beta)/\alpha$ ve $B = \beta/(1 - \alpha)$ olarak alalım. Sırasıyla $i = 0, 1, 2, \dots$ değerleri için ve $n \geq 1$ olmak şartıyla Q_i değerleri gözlemlendiğinde,

- 1) Eğer $\sum_0^n Z_r \leq \ln B$ ise H_0 kabul,
- 2) Eğer $\sum_0^n Z_r \geq \ln A$ ise H_1 kabul,
- 3) Eğer $\ln B < \sum_0^n Z_r < \ln A$ ise Q_{n+1} . gözlemi alarak devam et

kararlarından birine ulaşılır. Yöntemin detayları Ek A'da verildi.

Eğer $Q_0 = i_0$ olarak alınırsa ve n_{ij} , $i \rightarrow j$ 'ye geçiş sayısı olarak gösterir ve n . geçişi gösterirse o zaman (3.1)'de verilen olabilirlik oranı

$$L = \prod_{i,j} P_{ij}^{n_{ij}}(\rho_1) / \prod_{i,j} P_{ij}^{n_{ij}}(\rho_0) \quad (3.4)$$

şeklinde indirgenir ve

$$\ln L = \sum_{i,j} n_{ij} \ln \left[\frac{P_{ij}(\rho_1)}{P_{ij}(\rho_0)} \right] \quad (3.5)$$

şeklinde yazılabilir. O zaman Ardışık Olasılık Oran Testi için süreklilik bölgesi

$$\ln B < \sum_{i,j} n_{ij} \ln \left[\frac{P_{ij}(\rho_1)}{P_{ij}(\rho_0)} \right] < \ln A \quad (3.6)$$

dir.

$M/E_k/1$, $E_k/M/c$ ve $M/M/c/N$ kuyrukları için olabilirlik oranının logaritmasını a ve c_{ij} , ρ_1 , ρ_0 ve P_{ij} geçiş ihtimali parametrelerine bağlı olan sabitler olmak üzere sonuç olarak

$$\ln L = an + \sum_{i,j} n_{ij} c_{ij} \quad (3.7)$$

şeklinde yazabiliriz. Böylece (3.6)

$$\ln B - an < \sum_{i,j} n_{ij} c_{ij} < \ln A - an \quad (3.8)$$

şekline indirgenir. Testin süreklilik bölgesi $\ln A - an$ ve $\ln B - an$ doğruları ile sınırlanır.

3.1 $M/G/1$ ve $GI/M/c$ Kuyrukları İçin Ardışık Olasılık Oran Testi

3.1.1 $M/G/1$ Kuyruk modeli

Birim zamanda gelişlerin λ oranında Poisson sürecine göre olduğu bir tek servis kanallı kuyruk sistemini ele alalım. Müşterilerin servis zamanları $B(x)$ dağılımı ile bağımsız ve aynı dağılımlı rasgele değişkenlerdir. Bu sistem için servis bitiş noktalarında kuyruk uzunlukları gömülü Markov zinciri formundadır. Q_n n . ayrılan

müşterinin arkasında kalan müşteri sayısı olacaktır. Kuyruk sayısının kapasitesi N ile sınırlıdır. Gömülü Markov zincirinin geçiş ihtimalleri matrisi

$$k_n = P\{\text{servis periyodu esnasında } n \text{ geliş}\}$$

$$= \int_0^\infty e^{-\lambda t} \frac{\lambda t^j}{j!} dB(t)$$

olmak üzere

$$P = \{P_{ij}(\rho)\} = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & \dots & N-1 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ . \\ \vdots \\ . \\ . \\ N-1 \end{matrix} & \left[\begin{array}{ccccc} k_0 & k_1 & k_2 & \dots & 1 - \sum_0^{k-2} k_j \\ k_0 & k_1 & k_2 & \dots & 1 - \sum_0^{k-2} k_j \\ 0 & k_0 & k_1 & \dots & 1 - \sum_0^{k-3} k_j \\ 0 & 0 & k_0 & \dots & . \\ \vdots & \vdots & \vdots & & \\ 0 & 0 & 0 & \dots & . \\ 0 & 0 & 0 & \dots & . \\ 0 & 0 & 0 & \dots & 1 - k_0 \end{array} \right] \end{matrix}$$

şeklinde verilir.

Bu form Ardışık Olasılık Oran Testi yöntemini çözmek için çok genel olduğu için servis zamanı Erlang dağılımına sahip olan kuyruk modelinde bu durumu özelleştireceğiz ve modelin Poisson - Erlang karışımı olan sürekli olasılık yoğunluk fonksiyonunu kesikli negatif binom dağılımına dönüştürülür. (Shahbazov, 2005; Karlis, 2005).

3.1.2 $M/E_k/1$ Kuyruk modeli

Teorem 3.1.2.1 Burada $k \geq 1$ ve tam sayı olmak üzere

$$\frac{dB(x)}{dx} = \frac{(k\mu x)^{k-1}}{(k-1)!} e^{-k\mu x} k\mu, \quad (x > 0)$$

dir. $k_n = P\{\text{servis periyodu esnasında } n \text{ geliş}\}$

$$= \int_0^{\infty} e^{-\lambda t} \frac{\lambda t^j}{j!} dB(t)$$

olmak üzere,

$$k_n = \binom{n+k-1}{n} \left(\frac{\rho}{\rho+k} \right)^n \left(\frac{k}{\rho+k} \right)^k \quad (3.9)$$

dir.

İspat:

$$\begin{aligned} k_n &= \int_0^{\infty} \frac{e^{-\lambda t} (\lambda t)^n}{n!} \frac{(k\mu t)^{k-1}}{(k-1)!} e^{-k\mu t} k\mu d(t) \\ &= \frac{k\mathcal{M}}{n! (k-1)!} \int_0^{\infty} e^{-t(\lambda+k\mu)} \lambda^n t^n k\mu^{(k-1)} t^{k-1} d(t) \\ &= \frac{k\mu \lambda^n k\mu^{(k-1)}}{n! (k-1)!} \int_0^{\infty} e^{-t(\lambda+k\mu)} t^{n+k-1} d(t) \rightarrow \text{Gama dağılımı } \Gamma(n+k) \\ &= \frac{(n+k-1)! k\mu \lambda^n k\mu^{(k-1)}}{n! (k-1)!} \left(\frac{1}{\lambda+k\mu} \right)^{n+k} \\ &= \binom{n+k-1}{n} \left(\frac{1}{\lambda+k\mu} \right)^n \left(\frac{1}{\lambda+k\mu} \right)^k (k\mu)^k \lambda^n \\ &= \binom{n+k-1}{n} \left(\frac{\lambda}{\lambda+k\mu} \right)^n \left(\frac{k\mu}{\lambda+k\mu} \right)^k \end{aligned}$$

Pay ve paydayı μ ile bölersek,

$$= \binom{n+k-1}{n} \left(\frac{\rho}{\rho+k\mu} \right)^n \left(\frac{k\mu}{\rho+k\mu} \right)^k$$

eşitliğini elde ederiz ve

$$\begin{aligned} P_{ij}(\rho) &= k_{j-i+1}, & i &= 1, 2, \dots, N \\ & & j &= 0, 1, \dots, N-1 \\ & & j &\geq i-1, \end{aligned}$$

$$P_{0j}(\rho) = k_j, \quad j = 0, 1, \dots, N-1,$$

$$P_{iN}(\rho) = 1 - \sum_0^{N-i} k_n, \quad i = 1, 2, \dots, N,$$

$$P_{0N}(\rho) = 1 - \sum_0^{N-1} k_n$$

dir. Sadeleştirmelerle;

$$P_{iN}(\rho) = \left(\frac{k}{\rho+k}\right)^k \left\{ \left(\frac{\rho+k}{k}\right)^k - \sum_{n=0}^{N-i} \binom{n+k-1}{n} \left(\frac{\rho}{\rho+k}\right)^n \right\}$$

eşitliğini elde ederiz.

$P_{ij}(\rho)$ 'nin bu değerleri için olabilirlik oranının logaritması,

$$\begin{aligned} \ln L &= \sum_{i,j} n_{ij} \ln \left[\frac{P_{ij}(\rho_1)}{P_{ij}(\rho_0)} \right] \\ &= an + \sum_{i,j} n_{ij} c_{ij} \end{aligned}$$

olacaktır. Burada,

$$a = k \ln \left(\frac{\rho_0 + k}{\rho_1 + k} \right),$$

$$\begin{aligned} c_{ij} &= \ln \left\{ \frac{\rho_1}{\rho_0} \cdot \frac{\rho_0 + k}{\rho_1 + k} \right\}^{j-i+1} \quad i = 1, 2, \dots, N \\ &\quad j = 0, 1, \dots, N-1 \\ &\quad j \geq i-1, \end{aligned}$$

$$c_{0j} = \ln \left\{ \frac{\rho_1}{\rho_0} \cdot \frac{\rho_0 + k}{\rho_1 + k} \right\}^j \quad j = 0, 1, \dots, N-1, \quad (3.10)$$

$$c_{iN} = \ln \left\{ \frac{\left(\frac{\rho_1 + k}{k}\right)^k - \sum_{r=0}^{N-i} \binom{r+k-1}{r} \left(\frac{\rho_1}{\rho_1 + k}\right)^r}{\left(\frac{\rho_0 + k}{k}\right)^k - \sum_{r=0}^{N-i} \binom{r+k-1}{r} \left(\frac{\rho_0}{\rho_0 + k}\right)^r} \right\} \quad i = 1, 2, \dots, N,$$

$$c_{0N} = c_{1N}$$

olarak alınır. Bu katsayıları aşağıdaki şekilde elde ederiz.

$$\begin{aligned}
\ln L &= \sum_{i,j} n_{ij} \ln \frac{P_{ij}(\rho_1)}{P_{ij}(\rho_0)} \\
&= \sum_{i,j} n_{ij} \ln \left[\frac{\binom{n+k-1}{n} \left(\frac{\rho_1}{\rho_1+k}\right)^n \left(\frac{k}{\rho_1+k}\right)^k}{\binom{n+k-1}{n} \left(\frac{\rho_0}{\rho_0+k}\right)^n \left(\frac{k}{\rho_0+k}\right)^k} \right] \\
&= \sum_{i,j} n_{ij} \ln \left[\frac{\left(\frac{\rho_1}{\rho_1+k}\right)^n \left(\frac{k}{\rho_1+k}\right)^k}{\left(\frac{\rho_0}{\rho_0+k}\right)^n \left(\frac{k}{\rho_0+k}\right)^k} \right] \\
&= \sum_{i,j} n_{ij} \ln \frac{\left(\frac{k}{\rho_1+k}\right)^k}{\left(\frac{k}{\rho_0+k}\right)^k} + \sum_{i,j} n_{ij} \ln \frac{\left(\frac{\rho_1}{\rho_1+k}\right)^n}{\left(\frac{\rho_0}{\rho_0+k}\right)^n} \\
an &= \sum_{i,j} n_{ij} \ln \frac{\left(\frac{k}{\rho_1+k}\right)^k}{\left(\frac{k}{\rho_0+k}\right)^k} = nkl \ln \left(\frac{\rho_0+k}{\rho_1+k}\right) \\
c_{ij} &= \ln \frac{\left(\frac{\rho_1}{\rho_1+k}\right)^n}{\left(\frac{\rho_0}{\rho_0+k}\right)^n} = \ln \left\{ \frac{\rho_1}{\rho_0} \cdot \frac{\rho_0+k}{\rho_1+k} \right\}^{j-i+1} \quad n = j - i + 1 \\
c_{0j} &= \ln \frac{\left(\frac{\rho_1}{\rho_1+k}\right)^{n-1}}{\left(\frac{\rho_0}{\rho_0+k}\right)^{n-1}} = \ln \left\{ \frac{\rho_1}{\rho_0} \cdot \frac{\rho_0+k}{\rho_1+k} \right\}^j \\
n-1 &= j - i + 1 - 1 = j - 0 + 1 - 1 = j
\end{aligned}$$

3.1.3 GI/M/c Kuyruk modeli

Bu bölümde gelişler arası zaman dağılımı $A(x)$ olan yinemeli yönteme göre gelişler arası görülen bir kuyruğu dikkate alacağız. Sistemde $c \geq 1$ kanal olduğunu varsayalım ve servis zamanları μ^{-1} ortalamalı sahip bağımsız ve aynı dağılımlı negatif üstel dağılıma sahip olsun. Kuyruk kapasitesi n ile sınırlanır.

Burada kuyruk uzunluğundaki başarılı gelişler Markov zinciri şeklinde gözlemlenir. Markov zincirinin geçiş ihtimalleri aşağıdaki şekilde verilir.

$$\begin{aligned}
 P_{ij} &= 0, \quad (i+1 < j) \\
 P_{ij} &= \int_0^\infty \binom{i+1}{i+1-j} (1-e^{-\mu t})^{i+1-j} e^{-j\mu t} dA(t), \quad (j \leq i+1 \leq c) \\
 P_{ij} &= \binom{c}{c-j} \frac{(c\mu)^{i+1-j}}{(i-c)!} \int_0^\infty \int_0^t \{(1-e^{-\mu(t-v)})^{c-j} \\
 &\quad e^{-\mu(t-v)j} v^{i-c} e^{-c\mu v}\} dv dA(t), \quad (i+1 > c > j) \\
 P_{ij} &= \int_0^\infty e^{-c\mu t} \frac{(c\mu t)^{i-c+1}}{(i+1-j)!} dA(t), \quad (i+1 \geq j \geq c) \quad (3.11)
 \end{aligned}$$

Gelecek bölümde geliş dağılımları Erlang olan özel bir durumu göstereceğiz.

3.1.4 E_k/M/c Kuyruk modeli

Burada geliş dağılımları $k \geq 1$ tamsayıları için

$$\frac{dA(x)}{dx} = \frac{k\lambda x^{k-1}}{(k-1)!} e^{-k\lambda x} k\lambda \quad (x > 0)$$

olarak verilir.

Sistem kapasitesini N ile sınırladığımızdan dikkat edilmelidir ki $(N-1)$ ve N . satır geçiş ihtimalleri matrisi aynı olacaktır.

$$P_{Nj} = P_{N-1,j} \quad j = 0, 1, \dots, N$$

P_{ij} için yukarıdaki ifadelerde $dA(x)$ yerine koyarak ve $\rho = \lambda/\mu$ alındığında;

$$a = k \ln \left(\frac{\rho_1}{\rho_0} \cdot \frac{\rho_0 + k}{\rho_1 + k} \right),$$

$$c_{ij} = \ln \left\{ \frac{\sum_{r=0}^{i+1-j} (-1)^r \binom{i+1-j}{r} \left(\frac{k\rho_1 + c}{k\rho_1 + r + j} \right)^k}{\sum_{r=0}^{i+1-j} (-1)^r \binom{i+1-j}{r} \left(\frac{k\rho_0 + c}{k\rho_0 + r + j} \right)^k} \right\} \quad (j \leq i+1 \leq c)$$

$$c_{ij} = \ln \frac{g_{ij}(\rho_1)}{g_{1j}(\rho_0)}, \quad (i+1 > c > j),$$

olarak ve

$$\begin{aligned} g_{ij}(\rho) = & \sum_{r=0}^{c-j-1} (-1)^r \binom{c-j}{r} \left(c - r - j \right)^{i-c+1} \left[\left(\frac{(k\rho + c)}{k\rho + r + j} \right)^k \right. \\ & - \sum_{m=0}^{i-c} \binom{k+m-1}{m} \left(\frac{c-r-j}{k\rho + c} \right)^m \Big\} \\ & + (-1)^{s-j} \binom{k+i-c}{i-c+1} \left(\frac{c}{k\rho + c} \right)^{i-c+1} \Big], \end{aligned}$$

$$c_{ij} = \ln \left\{ \frac{c + k\rho_0}{c + k\rho_1} \right\}^{i+1-j} \quad (i+1 \geq j \geq c),$$

$$c_{Nj} = c_{N-1,j}$$

alındığında;

$$\ln L = an + \sum_{i,j} n_{ij} c_{ij}$$

olarak elde edilir.

Sonlu kapasiteli $M/M/c$ kuyruğu için sonuçlar $k = 1$ alınarak $E_k/M/c$ kuyruğundan elde edilebilir.

3.2 Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayısı

Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayısı bağımsız gözlemler için Temel Tanımlar ve Wald Teoremi ile çözülür. Wald (1947)'ın temel tanımlarına benzer ve $Z_r = h(Q_{r-1}, Q_r)$ tipindeki değişkenler için bazı teoremler Phatarfod (1965) tarafından çözülmüştür ve bu sonuçlar Ardışık Olasılık Oran Testindeki Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayısını hesaplamak için kullanılmıştır.

$r \geq 1$ olmak üzere $Q_{r-1} = i$ ve $Q_r = j$ ise $\{Z_r\}$ rasgele değişkeninin dizisi $Z_r = h_{ij}$ olarak alınsın. $P(t) = [P_{ij} \exp\{th_{ij}\}]$ olarak alınsın. $P(t)$ matrisinin gizil kökleri $i = 0, 1, \dots, N$ için $\lambda_i(t)$ olsun. Perron (1907) ve Thornett (1961) pozitif matrisler teoremini kullanarak tüm gerçel t değerleri için $\lambda_0(t)$ gerçel kökünün pozitif olduğunu ve $|\lambda_i(t)|$ 'den daha büyük olduğunu göstermiştir. İlgi çekici bir özellik olarak $\lambda_0(t)$ değeri bağımsız gözlemler için elde edilen Z_r 'nin moment çıkaran fonksiyonu $M(t)$ 'nin yerine kullanılabilir. Daha fazlası $\lambda_0(t_0) = 1$ eşitliğinde $t \neq 0$ farklı olmak üzere sadece ve sadece tek gerçel değer vardır.

Phatarfod (1965) tarafından Karakteristik İşlem Eğrisinin yaklaşık formülü $t_0(\rho)$, $\lambda_0(t, \rho) = 1$ eşitliğinin sıfırdan farklı gerçel kökü olduğunda aşağıdaki şekilde verilmiştir.

$$L(\rho) \cong \frac{A^{t_0(\rho)} - 1}{A^{t_0(\rho)} - B^{t_0(\rho)}} \quad \text{Eğer } t_0(\rho) \neq 0 \text{ ise} \quad (3.12)$$

$$\cong \frac{\ln A}{\ln A - \ln B} \quad \text{Eğer } t_0(\rho) = 0 \text{ ise} \quad (3.13)$$

Ortalama Örnek Sayısının yaklaşık değeri

$$E(n; \rho) \cong \frac{L(\rho) \ln B + \{1 - L(\rho)\} \ln A}{\lambda'_0(0)} \quad \text{Eğer } \lambda'_0 \neq 0 \quad (3.14)$$

$$\cong \frac{L(\rho) \{\ln B\}^2 + \{1 - L(\rho)\} \{\ln A\}^2}{\lambda''_0} \quad \text{Eğer } \lambda'_0 = 0 \quad (3.15)$$

şeklinde verilir.

Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayılarını hesaplayabilmek için

$$P(t) = \left\{ p_{ij}(\rho) \left[\frac{p_{ij}(\rho_1)}{p_{ij}(\rho_0)} \right]^t \right\}$$

matrisinin ne büyük kökü $\lambda_0(t, \rho)$ 'ı, onun $t = 0$ 'da türevini ve $\lambda_0(t, \rho) = 1$ denklemin sıfırdan farklı kökü olan $t(\rho)$ 'u tanımlamak gerekir.

Rao ve Harishchandra $\lambda_0(t, \rho) = 1$ eşitliğinde $t(\rho)$ köklerini hesaplamak ve genel durumlarda $t = 0$ noktasında $\lambda_0(t, \rho)$ 'nın türevi için nümerik metodlar geliştirmişlerdir.

Nümerik analiz tekniklerini kökleri bulmak için kullanılmıştır. Alternatif olarak grafiksel yöntem ile $\lambda_0(t, \rho)$ 'ler t 'nin fonksiyonu olarak çizilebilir. $t = 0$ noktasında $\lambda_0(t, \rho)$ 'nın türevini değerlendirmek için beş noktalı nümerik türev bağıntısı kullanılmıştır (McCormick ve Salvadori, 1964).

Eğer biz $\lambda_0(t, \rho)$ 'nın değerlerini, $\lambda_{-2}, \lambda_{-1}, \lambda_0, \lambda_1, \lambda_2$ kökleri için verilen bir h değeri için $-2h, -h, 0, h, 2h$ olarak eksen üzerinde eşit aralıklı beş noktada gösterirsek o zaman $t = 0$ noktasında $\lambda_0(t, \rho)$ 'nın birinci ve ikinci değerleri için yaklaşık değerleri

$$\lambda'_0(t, \rho) \cong \frac{1}{12h} \{\lambda_{-2} - 8\lambda_{-1} + 8\lambda_1 - \lambda_2\} \quad (3.16)$$

ve

$$\lambda''_0(t, \rho) \cong \frac{1}{12h^2} \{-\lambda_{-2} + 16\lambda_{-1} - 30\lambda_0 + 16\lambda_1 - \lambda_2\} \quad (3.17)$$

dır.

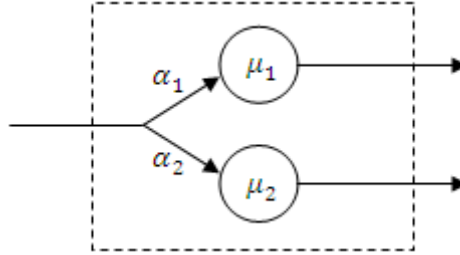
Bu yaklaşımlara karşı gelen hatalar sırasıyla $h^4 \lambda_0''(0, \rho)/30$ ve $h^4 \lambda_0'''(0, \rho)/90$ olarak verilir. Daha küçük h seçimi yaklaşık hatayı indirgeyebilir.

4. FAZ TİPLİ KUYRUK MODELLERİ İÇİN ARDIŞIK TEST

Tek kanallı seri ya da paralel fazlardan oluşan hizmet sistemlerini analiz etmek için bazı dağılımların bilinmesine ihtiyaç vardır. Bu bağlamda üstel dağılım yaygın bir biçimde kullanılır. Bunun nedeni dağılımın belleksizlik özelliğindendir. Fakat bazen hizmet yerleri birden fazla üniteden meydana gelebilir, dolayısıyla bu da bizi genel dağılımlı modeller oluşturmaya götürebilir. Bu durumda faz tipi dağılımlardan bahsetmek doğru olacaktır (Stewart, 2009).

4.1 Hyperexponential ($M/H_2/1$) Kuyruk Modeli

Gelişler poisson, servis süreleri üstel dağılıma uygun olan bu kuyruk modelinde servise giren müşteri $\alpha_1 + \alpha_2 = 1$ olmak şartıyla iki kanaldan birini α_1 , diğerini α_2 olasılığı ile seçer. Herhangi bir kanalda hizmetini aldıktan sonra sistemi terk eder. Herhangi bir zamanda serviste sadece bir müşteri olabilir. Bu kuyruk modeli aşağıdaki şekilde gösterilebilir.



Şekil 4.1: Hyperexponential ($M/H_2/1$) kuyruk modeli.

Teorem 4.1.1 Bu modelin servis zamanının olasılık yoğunluk fonksiyonu

$$\frac{dB(x)}{dx} = (\alpha_1 \mu_1 e^{-\mu_1 x} + \alpha_2 \mu_2 e^{-\mu_2 x}), \quad (x \geq 0)$$

dir.

$$k_n = P\{\text{servis periyodu esnasında } n \text{ geliş}\}$$

$$= \int_0^\infty e^{-\lambda t} \frac{\lambda t^j}{j!} dB(t)$$

olmak üzere,

$$k_n = \alpha_1 \left(\frac{\rho_1}{\rho_1 + 1} \right)^n \left(\frac{1}{\rho_1 + 1} \right) + \alpha_2 \left(\frac{\rho_2}{\rho_2 + 1} \right)^n \left(\frac{1}{\rho_2 + 1} \right) \quad (4.1)$$

dir.

İspat:

$$\begin{aligned} k_n &= \int_0^\infty \frac{e^{-\lambda t} (\lambda t)^n}{n!} (\alpha_1 \mu_1 e^{-\mu_1 t} + \alpha_2 \mu_2 e^{-\mu_2 t}) d(t) \\ &= \int_0^\infty \frac{e^{-\lambda t} (\lambda t)^n}{n!} \alpha_1 \mu_1 e^{-\mu_1 t} d(t) + \int_0^\infty \frac{e^{-\lambda t} (\lambda t)^n}{n!} \alpha_2 \mu_2 e^{-\mu_2 t} d(t) \\ &= \frac{\alpha_1 \mu_1 \lambda^n}{n!} \int_0^\infty e^{-\lambda t} t^n e^{-\mu_1 t} d(t) + \frac{\alpha_2 \mu_2 \lambda^n}{n!} \int_0^\infty e^{-\lambda t} t^n e^{-\mu_2 t} d(t) \\ &= \frac{\alpha_1 \mu_1 \lambda^n}{n!} \int_0^\infty e^{-t(\lambda + \mu_1)} t^n d(t) + \frac{\alpha_2 \mu_2 \lambda^n}{n!} \int_0^\infty e^{-t(\lambda + \mu_2)} t^n d(t) \\ &= \frac{\alpha_1 \mu_1 \lambda^n}{n!} \left(\frac{1}{\lambda + \mu_1} \right)^{n+1} \Gamma(n+1) + \frac{\alpha_2 \mu_2 \lambda^n}{n!} \left(\frac{1}{\lambda + \mu_2} \right)^{n+1} \Gamma(n+1) \\ &= \frac{\alpha_1 \mu_1 \lambda^n n!}{n!} \left(\frac{1}{\lambda + \mu_1} \right)^{n+1} + \frac{\alpha_2 \mu_2 \lambda^n n!}{n!} \left(\frac{1}{\lambda + \mu_2} \right)^{n+1} \\ &= \alpha_1 \left(\frac{\lambda}{\lambda + \mu_1} \right)^n \left(\frac{\mu_1}{\lambda + \mu_1} \right) + \alpha_2 \left(\frac{\lambda}{\lambda + \mu_2} \right)^n \left(\frac{\mu_2}{\lambda + \mu_2} \right) \\ &= \alpha_1 \left(\frac{\lambda/\mu_1}{\lambda/\mu_1 + \mu_1/\mu_1} \right)^n \left(\frac{\mu_1/\mu_1}{\lambda/\mu_1 + \mu_1/\mu_1} \right) + \alpha_2 \left(\frac{\lambda/\mu_2}{\lambda/\mu_2 + \mu_2/\mu_2} \right)^n \left(\frac{\mu_2/\mu_2}{\lambda/\mu_2 + \mu_2/\mu_2} \right) \\ &= \alpha_1 \left(\frac{\rho_1}{\rho_1 + 1} \right)^n \left(\frac{1}{\rho_1 + 1} \right) + \alpha_2 \left(\frac{\rho_2}{\rho_2 + 1} \right)^n \left(\frac{1}{\rho_2 + 1} \right) \end{aligned}$$

Burada,

$$P_{ij}(\rho) = k_{j-i+1}, \quad i = 1, 2, \dots, N$$

$$j = 0, 1, \dots, N - 1$$

$$j \geq i - 1,$$

$$P_{0j}(\rho) = k_j, \quad j = 0, 1, \dots, N - 1,$$

$$P_{iN}(\rho) = 1 - \sum_0^{N-i} k_n, \quad i = 1, 2, \dots, N,$$

$$P_{0N}(\rho) = 1 - \sum_0^{N-1} k_n$$

ve

$$P_{iN}(\rho) = 1 - \sum_0^{N-i} k_n, \quad i = 1, 2, \dots, N, \text{ olmak üzere}$$

$$P_{iN}(\rho) = 1 - \sum_{r=0}^{N-i} \alpha_1 \left(\frac{\rho_1}{\rho_1 + 1} \right)^r \left(\frac{1}{\rho_1 + 1} \right) + \alpha_2 \left(\frac{\rho_2}{\rho_2 + 1} \right)^r \left(\frac{1}{\rho_2 + 1} \right)$$

dir. $P_{ij}(\rho)$ ' nin bu değerleri için olabilirlik oranının logaritması

$$\begin{aligned} \ln L &= \sum_{i,j} n_{ij} \ln \frac{P_{ij}(\rho_1)}{P_{ij}(\rho_0)} \\ &= an + \sum_{i,j} n_{ij} c_{ij} \end{aligned}$$

$$\ln L = \sum_{i,j} n_{ij} \ln \left(\frac{\alpha_1 \left(\frac{\rho_{1(1)}}{\rho_{1(1)} + 1} \right)^n \left(\frac{1}{\rho_{1(1)} + 1} \right) + \alpha_2 \left(\frac{\rho_{2(1)}}{\rho_{2(1)} + 1} \right)^n \left(\frac{1}{\rho_{2(1)} + 1} \right)}{\alpha_1 \left(\frac{\rho_{1(0)}}{\rho_{1(0)} + 1} \right)^n \left(\frac{1}{\rho_{1(0)} + 1} \right) + \alpha_2 \left(\frac{\rho_{2(0)}}{\rho_{2(0)} + 1} \right)^n \left(\frac{1}{\rho_{2(0)} + 1} \right)} \right) \quad (4.2)$$

olacaktır. Burada,

$$i = 1, 2, \dots, N,$$

$$j = 0, 1, \dots, N - 1$$

$$j \geq i - 1 \text{ değerleri için}$$

$$c_{ij} = \ln \left(\frac{\alpha_1 \left(\frac{\rho_{1(1)}}{\rho_{1(1)} + 1} \right)^{j-i+1} \left(\frac{1}{\rho_{1(1)} + 1} \right) + \alpha_2 \left(\frac{\rho_{2(1)}}{\rho_{2(1)} + 1} \right)^{j-i+1} \left(\frac{1}{\rho_{2(1)} + 1} \right)}{\alpha_1 \left(\frac{\rho_{1(0)}}{\rho_{1(0)} + 1} \right)^{j-i+1} \left(\frac{1}{\rho_{1(0)} + 1} \right) + \alpha_2 \left(\frac{\rho_{2(0)}}{\rho_{2(0)} + 1} \right)^{j-i+1} \left(\frac{1}{\rho_{2(0)} + 1} \right)} \right), \quad (4.3)$$

$$j = 0, 1, \dots, N - 1 \text{ değerleri için}$$

$$c_{0j} = \ln \left(\frac{\alpha_1 \left(\frac{\rho_{1(1)}}{\rho_{1(1)} + 1} \right)^j \left(\frac{1}{\rho_{1(1)} + 1} \right) + \alpha_2 \left(\frac{\rho_{2(1)}}{\rho_{2(1)} + 1} \right)^j \left(\frac{1}{\rho_{2(1)} + 1} \right)}{\alpha_1 \left(\frac{\rho_{1(0)}}{\rho_{1(0)} + 1} \right)^j \left(\frac{1}{\rho_{1(0)} + 1} \right) + \alpha_2 \left(\frac{\rho_{2(0)}}{\rho_{2(0)} + 1} \right)^j \left(\frac{1}{\rho_{2(0)} + 1} \right)} \right),$$

$i = 1, 2, \dots, N$ değerleri için

$$c_{iN} = \ln \left\{ \frac{1 - \sum_{r=0}^{N-i} \left(\alpha_1 \left(\frac{\rho_{1(1)}}{\rho_{1(1)} + 1} \right)^r \left(\frac{1}{\rho_{1(1)} + 1} \right) + \alpha_2 \left(\frac{\rho_{2(1)}}{\rho_{2(1)} + 1} \right)^r \left(\frac{1}{\rho_{2(1)} + 1} \right) \right)}{1 - \sum_{r=0}^{N-i} \left(\alpha_1 \left(\frac{\rho_{0(1)}}{\rho_{0(1)} + 1} \right)^r \left(\frac{1}{\rho_{0(1)} + 1} \right) + \alpha_2 \left(\frac{\rho_{0(1)}}{\rho_{0(1)} + 1} \right)^r \left(\frac{1}{\rho_{0(1)} + 1} \right) \right)} \right\}, \quad (4.4)$$

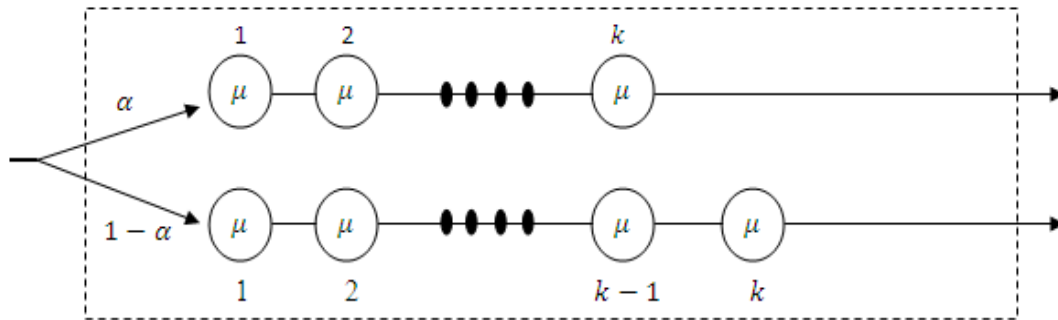
ve

$$c_{0N} = c_{1N}$$

olarak alınır.

4.2 Mixed Erlang ($M/mixE_k/1$) Kuyruk Modeli

Gelişler poisson, servis süreleri Erlang-2 dağılıma uygun olan bu kuyruk modelinde servise giren müşteri iki kanaldan birini α_1 , diğerini $1 - \alpha_1$ olasılığı ile seçer. Herhangi bir kanalda hizmetini aldıktan sonra sistemi terk eder. Herhangi bir zamanda serviste sadece bir müşteri olabilir. Bu kuyruk modeli aşağıdaki şekilde gösterilebilir.



Şekil 4.2: Mixed Erlang ($M/mixE_k/1$) kuyruk modeli.

Teorem 4.2.1 Burada $k \geq 2$ ve tam sayı olmak üzere modelin olasılık yoğunluk fonksiyonu

$$\frac{dB(x)}{dx} = \alpha \frac{\mu(\mu x)^{k-2} e^{-\mu x}}{(k-2)!} + (1-\alpha) \frac{\mu(\mu x)^{k-1} e^{-\mu x}}{(k-1)!}, \quad (x \geq 0)$$

dir.

$$k_n = P\{\text{servis periyodu esnasında } n \text{ geliş}\}$$

$$= \int_0^\infty e^{-\lambda t} \frac{\lambda t^n}{n!} dB(t)$$

olmak üzere,

$$k_n = \left(\frac{\rho}{\rho+k}\right)^n \left(\frac{k}{\rho+k}\right)^{k-1} \frac{(n+k-2)!}{n!(k-2)!} \left(\alpha + (1-\alpha) \frac{(n+k-1)}{(k-1)} \left(\frac{k}{\rho+k}\right) \right) \quad (4.5)$$

dir.

İspat:

$$\begin{aligned} k_n &= \int_0^\infty \frac{e^{-\lambda t} (\lambda t)^n}{n!} \left(\alpha \frac{(k\mu t)^{k-2}}{(k-2)!} e^{-k\mu t} k\mu + (1-\alpha) \frac{(k\mu t)^{k-1}}{(k-1)!} e^{-k\mu t} k\mu \right) dt \\ &= \int_0^\infty \frac{e^{-\lambda t} (\lambda t)^n}{n!} \alpha \frac{(k\mu t)^{k-2}}{(k-2)!} e^{-k\mu t} k\mu dt + \int_0^\infty \frac{e^{-\lambda t} (\lambda t)^n}{n!} (1-\alpha) \frac{(k\mu t)^{k-1}}{(k-1)!} e^{-k\mu t} k\mu dt \\ &= \frac{\alpha k\mu \lambda^n (k\mu)^{k-2}}{n!(k-2)!} \int_0^\infty e^{-\lambda t} t^n e^{-k\mu t} t^{k-2} dt \\ &\quad + \frac{(1-\alpha) k\mu \lambda^n (k\mu)^{k-1}}{n!(k-1)!} \int_0^\infty e^{-\lambda t} t^n e^{-k\mu t} t^{k-1} dt \\ &= \frac{\alpha k\mu \lambda^n k\mu^{(k-2)}}{n!(k-2)!} \int_0^\infty e^{-t(\lambda+k\mu)} t^{n+k-2} dt + \frac{\alpha k\mu \lambda^n k\mu^{(k-1)}}{n!(k-1)!} \int_0^\infty e^{-t(\lambda+k\mu)} t^{n+k-1} dt \\ &= \frac{\alpha \lambda^n (k\mu)^{k-1}}{n!(k-2)!} \int_0^\infty e^{-t(\lambda+k\mu)} t^{n+k-2} dt + \frac{\alpha \lambda^n (k\mu)^k}{n!(k-1)!} \int_0^\infty e^{-t(\lambda+k\mu)} t^{n+k-1} dt \\ &= \frac{k\mu \lambda^n k\mu^{(k-1)}}{n!(k-1)!} \int_0^\infty e^{-t(\lambda+k\mu)} t^{n+k-1} dt \rightarrow \text{Gama dağılımı } \Gamma(n+k) \\ &= \frac{\alpha \lambda^n (k\mu)^{k-1}}{n!(k-2)!} \left(\frac{1}{\lambda+k\mu}\right)^{n+k-1} \Gamma(n+k-1) + \frac{(1-\alpha) \lambda^n (k\mu)^k}{n!(k-1)!} \left(\frac{1}{\lambda+k\mu}\right)^{n+k} \Gamma(n+k) \end{aligned}$$

$$\begin{aligned}
&= \frac{(n+k-2)! \propto \lambda^n (k\mu)^{k-1}}{n! (k-2)!} \left(\frac{1}{\lambda + k\mu} \right)^{n+k-1} + \frac{(n+k-1)! (1-\alpha) \lambda^n (k\mu)^k}{n! (k-1)!} \left(\frac{1}{\lambda + k\mu} \right)^{n+k} \\
&= \binom{n+k-2}{n} \left(\frac{1}{\lambda + k\mu} \right)^n \left(\frac{1}{\lambda + k\mu} \right)^{k-1} \propto (k\mu)^{k-1} \lambda^n \\
&\quad + \binom{n+k-1}{n} \left(\frac{1}{\lambda + k\mu} \right)^n \left(\frac{1}{\lambda + k\mu} \right)^k (1-\alpha) (k\mu)^k \lambda^n \\
&= \alpha \binom{n+k-2}{n} \left(\frac{\lambda}{\lambda + k\mu} \right)^n \left(\frac{k\mu}{\lambda + k\mu} \right)^{k-1} + (1-\alpha) \binom{n+k-1}{n} \left(\frac{\lambda}{\lambda + k\mu} \right)^n \left(\frac{k\mu}{\lambda + k\mu} \right)^k
\end{aligned}$$

Pay ve paydayı μ ile bölünürse,

$$\begin{aligned}
&= \alpha \binom{n+k-2}{n} \left(\frac{\lambda/\mu}{(\lambda + k\mu)/\mu} \right)^n \left(\frac{(k\mu)/\mu}{(\lambda + k\mu)/\mu} \right)^{k-1} + (1-\alpha) \binom{n+k-1}{n} \left(\frac{\lambda/\mu}{(\lambda + k\mu)/\mu} \right)^n \left(\frac{(k\mu)/\mu}{(\lambda + k\mu)/\mu} \right)^k \\
&= \alpha \binom{n+k-2}{n} \left(\frac{\rho}{\rho + k} \right)^n \left(\frac{k}{\rho + k} \right)^{k-1} + (1-\alpha) \binom{n+k-1}{n} \left(\frac{\rho}{\rho + k} \right)^n \left(\frac{k}{\rho + k} \right)^k
\end{aligned}$$

ve ortak parantezlere alınırsa,

$$\begin{aligned}
&= \left(\frac{\rho}{\rho + k} \right)^n \left(\frac{k}{\rho + k} \right)^{k-1} \left(\alpha \frac{(n+k-2)!}{n! (k-2)!} + (1-\alpha) \frac{(n+k-1)(n+k-2)!}{n! (k-1)(k-2)!} \left(\frac{k}{\rho + k} \right) \right) \\
&= \left(\frac{\rho}{\rho + k} \right)^n \left(\frac{k}{\rho + k} \right)^{k-1} \frac{(n+k-2)!}{n! (k-2)!} \left(\alpha + (1-\alpha) \frac{(n+k-1)}{(k-1)} \left(\frac{k}{\rho + k} \right) \right)
\end{aligned}$$

olur. Burada,

$$\begin{aligned}
P_{ij}(\rho) &= k_{j-i+1}, & i &= 1, 2, \dots, N \\
& & j &= 0, 1, \dots, N-1 \\
& & j &\geq i-1,
\end{aligned}$$

$$P_{0j}(\rho) = k_j, \quad j = 0, 1, \dots, N-1,$$

$$P_{iN}(\rho) = 1 - \sum_0^{N-i} k_n, \quad i = 1, 2, \dots, N,$$

$$P_{0N}(\rho) = 1 - \sum_0^{N-1} k_n$$

ve

$$P_{iN}(\rho) = 1 - \sum_0^{N-i} k_n, \quad i = 1, 2, \dots, N, \text{ olmak üzere}$$

$$P_{iN}(\rho) = 1 - \sum_{r=0}^{N-i} \left(\frac{\rho}{\rho+k} \right)^r \left(\frac{k}{\rho+k} \right)^{k-1} \binom{n+k-2}{n} \left(\alpha + (1 - \alpha) \frac{(r+k-1)}{(k-1)} \left(\frac{k}{\rho+k} \right) \right)$$

dir.

$P_{ij}(\rho)$ 'nin bu değerleri için olabilirlik oranının logaritması,

$$\begin{aligned} \ln L &= \sum_{i,j} n_{ij} \ln \frac{P_{ij}(\rho_1)}{P_{ij}(\rho_0)} \\ &= an + \sum_{i,j} n_{ij} c_{ij} \end{aligned}$$

$\ln L$

$$= \sum_{i,j} n_{ij} \ln \left(\frac{\left(\frac{\rho_1}{\rho_1+k} \right)^n \left(\frac{k}{\rho_1+k} \right)^{k-1} \frac{(n+k-2)!}{n!(k-2)!} \left(\alpha + (1-\alpha) \frac{(n+k-1)}{(k-1)} \left(\frac{k}{\rho_1+k} \right) \right)}{\left(\frac{\rho_0}{\rho_0+k} \right)^n \left(\frac{k}{\rho_0+k} \right)^{k-1} \frac{(n+k-2)!}{n!(k-2)!} \left(\alpha + (1-\alpha) \frac{(n+k-1)}{(k-1)} \left(\frac{k}{\rho_0+k} \right) \right)} \right)$$

$\boxed{\ln L}$

$$= \sum_{i,j} n_{ij} \ln \left(\frac{\left(\frac{\rho_1}{\rho_1+k} \right)^n \left(\frac{k}{\rho_1+k} \right)^{k-1} \left(\alpha + (1-\alpha) \frac{(n+k-1)}{(k-1)} \left(\frac{k}{\rho_1+k} \right) \right)}{\left(\frac{\rho_0}{\rho_0+k} \right)^n \left(\frac{k}{\rho_0+k} \right)^{k-1} \left(\alpha + (1-\alpha) \frac{(n+k-1)}{(k-1)} \left(\frac{k}{\rho_0+k} \right) \right)} \right) \quad (4.6)$$

olacaktır. Burada

$$\begin{aligned} &= n(k-1) \ln \left(\frac{\rho_0+k}{\rho_1+k} \right) \\ &+ \sum_{i,j} n_{ij} \ln \left(\frac{\rho_1}{\rho_0} \cdot \frac{\rho_0+k}{\rho_1+k} \right)^n \end{aligned}$$

$$+ \sum_{i,j} n_{ij} \ln \left(\frac{\alpha + (1 - \alpha) \frac{(n+k-1)}{(k-1)} \left(\frac{k}{\rho_1 + k} \right)}{\alpha + (1 - \alpha) \frac{(n+k-1)}{(k-1)} \left(\frac{k}{\rho_0 + k} \right)} \right) \quad (4.7)$$

eşitliğini elde ederiz.

Burada,

$$a = (k-1) \ln \left(\frac{\rho_0 + k}{\rho_1 + k} \right), \quad (4.8)$$

$i = 1, 2, \dots, N$,

$j = 0, 1, \dots, N-1$

$j \geq i-1$ değerleri için

$$c_{ij} + b_{ij} = \ln \left(\frac{\rho_1}{\rho_0} \cdot \frac{\rho_0 + k}{\rho_1 + k} \right)^{j-i+1} + \ln \left(\frac{\alpha + (1 - \alpha) \frac{(j-i+k)}{(k-1)} \left(\frac{k}{\rho_1 + k} \right)}{\alpha + (1 - \alpha) \frac{(j-i+k)}{(k-1)} \left(\frac{k}{\rho_0 + k} \right)} \right), \quad (4.9)$$

$j = 0, 1, \dots, N-1$ değerleri için

$$c_{0j} + b_{0j} = \ln \left(\frac{\rho_1}{\rho_0} \cdot \frac{\rho_0 + k}{\rho_1 + k} \right)^j + \ln \left(\frac{\alpha + (1 - \alpha) \frac{(j+k-1)}{(k-1)} \left(\frac{k}{\rho_1 + k} \right)}{\alpha + (1 - \alpha) \frac{(j+k-1)}{(k-1)} \left(\frac{k}{\rho_0 + k} \right)} \right),$$

$i = 1, 2, \dots, N$ değerleri için

c_{iN}

$$= \ln \left\{ \frac{1 - \sum_{r=0}^{N-i} \left(\frac{\rho_1}{\rho_1 + k} \right)^r \left(\frac{k}{\rho_1 + k} \right)^{k-1} \binom{r+k-2}{r} \left(\alpha + (1 - \alpha) \frac{(r+k-1)}{(k-1)} \left(\frac{k}{\rho_1 + k} \right) \right)}{1 - \sum_{r=0}^{N-i} \left(\frac{\rho_0}{\rho_0 + k} \right)^r \left(\frac{k}{\rho_0 + k} \right)^{k-1} \binom{r+k-2}{r} \left(\alpha + (1 - \alpha) \frac{(r+k-1)}{(k-1)} \left(\frac{k}{\rho_0 + k} \right) \right)} \right\}$$

ve

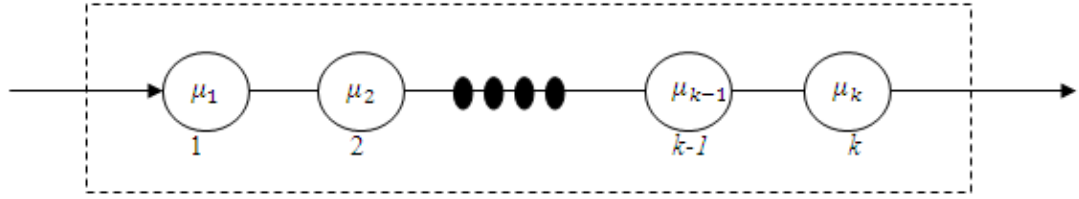
$$c_{0N} = c_{1N} \quad (4.10)$$

olarak alınır.

4.3 Hypoexponential ($M/HYPO_k/1$) Kuyruk Modeli

Gelişler poisson, servis süreleri Erlang- k dağılıma uygun olan bu kuyruk modelinde servise giren müşteri sırasıyla her bir kanalda hizmetini aldıktan sonra sistemi terk

eder. Herhangi bir zamanda serviste sadece bir müşteri olabilir. Bu kuyruk modeli aşağıdaki şekilde gösterilebilir.



Şekil 4.3: Hypoexponential ($M/HYPO_k/1$) kuyruk modeli.

Teorem 4.3.1 $k \geq 2$ ve tam sayı olmak üzere bu modelin iki kanal için olasılık yoğunluk fonksiyonu

$$\frac{dB(x)}{dx} = \frac{\mu_1 \mu_2}{\mu_1 - \mu_2} (e^{-\mu_2 x} - e^{-\mu_1 x}), \quad (x > 0)$$

dir.

$k_n = P\{\text{servis periyodu esnasında } n \text{ geliş}\}$

$$= \int_0^\infty e^{-\lambda t} \frac{\lambda t^n}{n!} dB(t)$$

olmak üzere,

$$k_n = \frac{1}{\mu_1 - \mu_2} \left[\mu_1 \left(\frac{\rho_2}{\rho_2 + 1} \right)^n \left(\frac{1}{\rho_2 + 1} \right) - \mu_2 \left(\frac{\rho_1}{\rho_1 + 1} \right)^n \left(\frac{1}{\rho_1 + 1} \right) \right] \quad (4.11)$$

dir.

İspat:

$$\begin{aligned} k_n &= \int_0^\infty \frac{e^{-\lambda t} (\lambda t)^n}{n!} \frac{\mu_1 \mu_2}{\mu_1 - \mu_2} (e^{-\mu_2 t} - e^{-\mu_1 t}) dt \\ &= \frac{\mu_1 \mu_2}{\mu_1 - \mu_2} \int_0^\infty \frac{e^{-\lambda t} (\lambda t)^n}{n!} (e^{-\mu_2 t} - e^{-\mu_1 t}) dt \\ &= \frac{\mu_1 \mu_2}{\mu_1 - \mu_2} \frac{\lambda^n}{n!} \int_0^\infty e^{-\lambda t} t^n (e^{-\mu_2 t} - e^{-\mu_1 t}) dt \end{aligned}$$

$$\begin{aligned}
&= \frac{\mu_1 \mu_2}{\mu_1 - \mu_2} \frac{\lambda^n}{n!} \left[\int_0^\infty e^{-t(\lambda+\mu_2)} t^n dt - e^{-t(\lambda+\mu_1)} t^n dt \right] \\
&= \frac{\mu_1 \mu_2}{\mu_1 - \mu_2} \frac{\lambda^n n!}{n!} \left[\left(\frac{1}{\lambda + \mu_2} \right)^{n+1} - \left(\frac{1}{\lambda + \mu_1} \right)^{n+1} \right] \\
&= \frac{\mu_1 \mu_2}{\mu_1 - \mu_2} \left[\left(\frac{\lambda}{\lambda + \mu_2} \right)^n \left(\frac{1}{\lambda + \mu_2} \right) - \left(\frac{\lambda}{\lambda + \mu_1} \right)^n \left(\frac{1}{\lambda + \mu_1} \right) \right] \\
&= \frac{\mu_1}{\mu_1 - \mu_2} \left(\frac{\rho_2}{\rho_2 + 1} \right)^n \left(\frac{\mu_2}{\lambda + \mu_2} \right) - \frac{\mu_2}{\mu_1 - \mu_2} \left(\frac{\rho_1}{\rho_1 + 1} \right)^n \left(\frac{\mu_1}{\lambda + \mu_2} \right) \\
&= \frac{1}{\mu_1 - \mu_2} \left[\mu_1 \left(\frac{\rho_2}{\rho_2 + 1} \right)^n \left(\frac{1}{\rho_2 + 1} \right) - \mu_2 \left(\frac{\rho_1}{\rho_1 + 1} \right)^n \left(\frac{1}{\rho_1 + 1} \right) \right]
\end{aligned}$$

olur. Burada,

$$\begin{aligned}
P_{ij}(\rho) &= k_{j-i+1}, & i &= 1, 2, \dots, N \\
& & j &= 0, 1, \dots, N-1 \\
& & j &\geq i-1,
\end{aligned}$$

$$P_{0j}(\rho) = k_j, \quad j = 0, 1, \dots, N-1,$$

$$P_{iN}(\rho) = 1 - \sum_0^{N-i} k_n, \quad i = 1, 2, \dots, N,$$

$$P_{0N}(\rho) = 1 - \sum_0^{N-1} k_n$$

ve

$$P_{iN}(\rho) = 1 - \sum_0^{N-i} k_n, \quad i = 1, 2, \dots, N \text{ olmak üzere}$$

$$P_{iN}(\rho) = 1 - \left(\sum_{r=0}^{N-i} \frac{1}{\mu_1 - \mu_2} \left[\mu_1 \left(\frac{\rho_2}{\rho_2 + 1} \right)^r \left(\frac{1}{\rho_2 + 1} \right) + \mu_2 \left(\frac{\rho_1}{\rho_1 + 1} \right)^r \left(\frac{1}{\rho_1 + 1} \right) \right] \right)$$

dir. $P_{ij}(\rho)$ 'nin bu değerleri için olabilirlik oranının logaritması,

$$\ln L = \sum_{i,j} n_{ij} \ln \frac{P_{ij}(\rho_1)}{P_{ij}(\rho_0)}$$

$$= an + \sum_{i,j} n_{ij} c_{ij}$$

olacaktır. Burada,

$\ln L$

$$= \sum_{i,j} n_{ij} \ln \left[\frac{\frac{1}{\mu_{1(1)} - \mu_{2(1)}} \left(\mu_{1(1)} \left(\frac{\rho_{2(1)}}{\rho_{2(1)} + 1} \right)^n \left(\frac{1}{\rho_{2(1)} + 1} \right) - \mu_{2(1)} \left(\frac{\rho_{1(1)}}{\rho_{1(1)} + 1} \right)^n \left(\frac{1}{\rho_{1(1)} + 1} \right) \right)}{\frac{1}{\mu_{1(0)} - \mu_{2(0)}} \left(\mu_{1(0)} \left(\frac{\rho_{2(0)}}{\rho_{2(0)} + 1} \right)^n \left(\frac{1}{\rho_{2(0)} + 1} \right) - \mu_{2(0)} \left(\frac{\rho_{1(0)}}{\rho_{1(0)} + 1} \right)^n \left(\frac{1}{\rho_{1(0)} + 1} \right) \right)} \right]$$

$\ln L$

$$= n \ln \left[\frac{\mu_{1(0)} - \mu_{2(0)}}{\mu_{1(1)} - \mu_{2(1)}} \right] + \sum_{i,j} n_{ij} \ln \left[\frac{\left(\mu_{1(1)} \left(\frac{\rho_{2(1)}}{\rho_{2(1)} + 1} \right)^n \left(\frac{1}{\rho_{2(1)} + 1} \right) - \mu_{2(1)} \left(\frac{\rho_{1(1)}}{\rho_{1(1)} + 1} \right)^n \left(\frac{1}{\rho_{1(1)} + 1} \right) \right)}{\left(\mu_{1(0)} \left(\frac{\rho_{2(0)}}{\rho_{2(0)} + 1} \right)^n \left(\frac{1}{\rho_{2(0)} + 1} \right) - \mu_{2(0)} \left(\frac{\rho_{1(0)}}{\rho_{1(0)} + 1} \right)^n \left(\frac{1}{\rho_{1(0)} + 1} \right) \right)} \right] \quad (4.12)$$

$i = 1, 2, \dots, N$

$j = 0, 1, \dots, N - 1$

$j \geq i - 1$ değerleri için,

$$c_{ij} = \ln \left[\frac{\left(\mu_{1(1)} \left(\frac{\rho_{2(1)}}{\rho_{2(1)} + 1} \right)^{j-i+1} \left(\frac{1}{\rho_{2(1)} + 1} \right) - \mu_{2(1)} \left(\frac{\rho_{1(1)}}{\rho_{1(1)} + 1} \right)^{j-i+1} \left(\frac{1}{\rho_{1(1)} + 1} \right) \right)}{\left(\mu_{1(0)} \left(\frac{\rho_{2(0)}}{\rho_{2(0)} + 1} \right)^{j-i+1} \left(\frac{1}{\rho_{2(0)} + 1} \right) - \mu_{2(0)} \left(\frac{\rho_{1(0)}}{\rho_{1(0)} + 1} \right)^{j-i+1} \left(\frac{1}{\rho_{1(0)} + 1} \right) \right)} \right] \quad (4.13)$$

$j = 0, 1, \dots, N - 1$ değerleri için,

c_{0j}

$$= \ln \left[\frac{\left(\mu_{1(1)} \left(\frac{\rho_{2(1)}}{\rho_{2(1)} + 1} \right)^j \left(\frac{1}{\rho_{2(1)} + 1} \right) - \mu_{2(1)} \left(\frac{\rho_{1(1)}}{\rho_{1(1)} + 1} \right)^j \left(\frac{1}{\rho_{1(1)} + 1} \right) \right)}{\left(\mu_{1(0)} \left(\frac{\rho_{2(0)}}{\rho_{2(0)} + 1} \right)^j \left(\frac{1}{\rho_{2(0)} + 1} \right) - \mu_{2(0)} \left(\frac{\rho_{1(0)}}{\rho_{1(0)} + 1} \right)^j \left(\frac{1}{\rho_{1(0)} + 1} \right) \right)} \right] \quad (4.14)$$

$i = 1, 2, \dots, N$ için değerleri,

c_{0j}

$$= \ln \left[\frac{1 - \left(\sum_{r=0}^{N-i} \mu_{1(1)} \left(\frac{\rho_{2(1)}}{\rho_{2(1)} + 1} \right)^r \left(\frac{1}{\rho_{2(1)} + 1} \right) - \mu_{2(1)} \left(\frac{\rho_{1(1)}}{\rho_{1(1)} + 1} \right)^r \left(\frac{1}{\rho_{1(1)} + 1} \right) \right)}{1 - \left(\sum_{r=0}^{N-i} \mu_{1(0)} \left(\frac{\rho_{2(0)}}{\rho_{2(0)} + 1} \right)^r \left(\frac{1}{\rho_{2(0)} + 1} \right) - \mu_{2(0)} \left(\frac{\rho_{1(0)}}{\rho_{1(0)} + 1} \right)^r \left(\frac{1}{\rho_{1(0)} + 1} \right) \right)} \right]$$

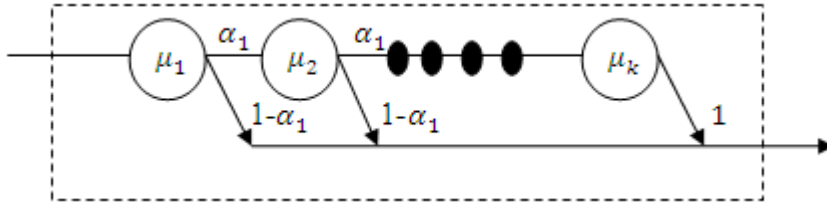
ve

$$c_{0N} = c_{1N} \quad (4.15)$$

olarak alınır.

4.4 Coxian ($M/C_k/1$) Kuyruk Modeli

Gelişler poisson, servis süreleri üstel dağılıma uygun olan bu kuyruk modelinde servise giren müşteri ilk kanalda hizmetini μ_1 servis oranıyla aldıktan sonra α_1 olasılığı ile diğer kanala geçer ya da $1 - \alpha_1$ olasılığı sistemi terk eder ve bu işlem diğer kanallarda da böyle devam eder. Herhangi bir zamanda serviste sadece bir müşteri olabilir. Bu kuyruk modeli aşağıdaki şekilde gösterilebilir.



Şekil 4.4: Coxian ($M/C_k/1$) kuyruk modeli.

Teorem 4.4.1 $k \geq 2$ ve tam sayı olmak üzere bu kuyruk modelinin olasılık yoğunluk fonksiyonu

σ_{nx1} satır vektörü,

$S_{n \times n}$ kare matris,

$S_{1 \times n}^0$ sütun vektörü olmak üzere

$k_n = P\{\text{servis periyodu esnasında } n \text{ geliş}\}$

$$\frac{dB(x)}{dx} = \sigma e^{Sx} S^0, \quad (x > 0)$$

olmak üzere,

$$k_n = \sigma \left(\frac{\rho}{\rho I - D} \right)^n \left(\frac{1}{\lambda I - S} \right) S^0 \quad (4.16)$$

dir.

İspat :

$$\begin{aligned}
k_n &= \int_0^\infty \frac{e^{-\lambda t} (\lambda t)^n}{n!} \sigma e^{Sx} S^0 d(t) \\
&= \int_0^\infty \frac{\lambda^n t^n}{n!} \sigma e^{-t(\lambda I - S)} S^0 d(t) \\
&= \frac{\lambda^n}{n!} \sigma \left\{ \int_0^\infty e^{-t(\lambda I - S)} t^n d(t) \right\} S^0 \\
&= \frac{\lambda^n}{n!} \sigma \frac{1}{(\lambda I - S)^{n+1}} n! S^0 \\
&= \lambda^n \sigma \frac{1}{(\lambda I - S)^{n+1}} S^0 \\
&= \sigma \left(\frac{\lambda}{\lambda I - S} \right)^n \left(\frac{1}{\lambda I - S} \right) S^0
\end{aligned}$$

$D = S/M$ olarak alırsak ve ilk kesirde pay ve paydayı μ ile bölünürse üsteki eşitliği

$$= \sigma \left(\frac{\rho}{\rho I - D} \right)^n \left(\frac{1}{\lambda I - S} \right) S^0$$

olarak yazabiliriz. Burada,

$$\begin{aligned}
P_{ij}(\rho) &= k_{j-i+1}, & i &= 1, 2, \dots, N \\
& & j &= 0, 1, \dots, N-1 \\
& & j &\geq i-1,
\end{aligned}$$

$$P_{0j}(\rho) = k_j, \quad j = 0, 1, \dots, N-1,$$

$$P_{iN}(\rho) = 1 - \sum_0^{N-i} k_n, \quad i = 1, 2, \dots, N,$$

$$P_{0N}(\rho) = 1 - \sum_0^{N-1} k_n$$

ve

$$P_{iN}(\rho) = 1 - \sum_0^{N-i} k_n, \quad i = 1, 2, \dots, N, \text{ olmak üzere}$$

$$P_{iN}(\rho) = 1 - \sum_{r=0}^{N-i} \sigma(\rho(\rho I - D)^{-1})^r (\lambda I - S)^{-1} S^0$$

dir. $P_{ij}(\rho)$ 'nin bu deęerleri iin olabilirlik oranının logaritması

$$\begin{aligned} \ln L &= \sum_{i,j} n_{ij} \ln \frac{P_{ij}(\rho_1)}{P_{ij}(\rho_0)} \\ &= an + \sum_{i,j} n_{ij} c_{ij} \end{aligned}$$

olacaktır. Burada,

$$\begin{aligned} \ln L &= \sum n_{ij} \ln \left(\frac{\sigma_1 \left(\frac{\rho_1}{\rho_1 I - D} \right)^n \left(\frac{1}{\lambda_1 I - S} \right) S_1^0}{\sigma_0 \left(\frac{\rho_0}{\rho_0 I - D} \right)^n \left(\frac{1}{\lambda_0 I - S} \right) S_0^0} \right) \\ \ln L &= \sum n_{ij} \ln \left(\frac{\sigma_1 (\rho_1 (\rho_1 I - D_1)^{-1})^n (\lambda_1 I - S_1)^{-1} S_1^0}{\sigma_0 (\rho_0 (\rho_0 I - D_0)^{-1})^n (\lambda_0 I - S_0)^{-1} S_0^0} \right) \end{aligned} \quad (4.17)$$

$$i = 1, 2, \dots, N$$

$$j = 0, 1, \dots, N - 1$$

$$j \geq i - 1,$$

$$c_{ij} = \ln \left(\frac{\sigma_1 (\rho_1 (\rho_1 I - D_1)^{-1})^{j-i+1} (\lambda_1 I - S_1)^{-1} S_1^0}{\sigma_0 (\rho_0 (\rho_0 I - D_0)^{-1})^{j-i+1} (\lambda_0 I - S_0)^{-1} S_0^0} \right) \quad (4.18)$$

$$j = 0, 1, \dots, N - 1,$$

$$c_{0j} = \ln \left(\frac{\sigma_1 (\rho_1 (\rho_1 I - D_1)^{-1})^j (\lambda_1 I - S_1)^{-1} S_1^0}{\sigma_0 (\rho_0 (\rho_0 I - D_0)^{-1})^j (\lambda_0 I - S_0)^{-1} S_0^0} \right) \quad (4.19)$$

$$i = 1, 2, \dots, N \text{ iin}$$

$$c_{iN} = \ln \left(\frac{1 - \sum_{r=0}^{N-i} \sigma_1 (\rho_1 (\rho_1 I - D_1)^{-1})^r (\lambda_1 I - S_1)^{-1} S_1^0}{1 - \sum_{r=0}^{N-i} \sigma_0 (\rho_0 (\rho_0 I - D_0)^{-1})^r (\lambda_0 I - S_0)^{-1} S_0^0} \right) \quad (4.20)$$

$$c_{0N} = c_{1N} \text{ olarak alınır.}$$

5. SAYISAL ÖRNEK

5.1 Erlang ($M/E_k/1$) Kuyruk Modeli İçin Sayısal Örnek

Örnek 5.1.1

$$\rho_0 = 0.1, \rho_1 = 0.9$$

$\alpha = 0.05, \beta = 0.10, A = (1 - \beta)/\alpha = 18, B = \beta/(1 - \alpha) = 0.1053$ değerleri için

$$H_0: \rho = 0.1$$

$$H_1: \rho = 0.9$$

hipotezini test edelim.

$k = 2$ olsun

$t_0 = 0, t_1, t_2, t_3, \dots$ $M/E_k/1$ kuyruğunda müşterilerin ayrılış noktaları olsun.

1,0,2,3,2,1,1,0,1,1 müşterilerin ayrılış noktalarında kalan müşteri sayıları olsun. E durum uzayı $\{0,1,2,3\}$ olsun. i satır değeri j sütun değeri olmak üzere n_{ij} , $i \rightarrow j$ 'ye geçişlerin sayısını göstermek üzere,

Çizelge 5.1: $N = 4$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.

	0	1	2	3	
0	0	1	1	0	$n_0 = 2$
1	2	2	0	0	$n_1 = 4$
2	0	1	0	1	$n_2 = 2$
3	0	0	1	0	$n_3 = 1$
	Toplam				9

$$n_{00} = 0, n_{01} = 1, n_{02} = 1, n_{03} = 0$$

$$n_{10} = 2, n_{11} = 2, n_{12} = 1, n_{13} = 0$$

$$n_{20} = 0, n_{21} = 1, n_{22} = 0, n_{23} = 1$$

$$n_{30} = 0, n_{31} = 0, n_{32} = 1, n_{33} = 0$$

dir.

$$i = 1, 2, \dots, N,$$

$$j = 0, 1, \dots, N - 1,$$

$$j \geq i - 1$$

değerleri için,

$$c_{10} = \ln \left\{ \frac{\rho_1}{\rho_0} \cdot \frac{\rho_0 + k}{\rho_1 + k} \right\}^{j-i+1} = \ln \left\{ \frac{0.9}{0.1} \cdot \frac{0.1 + 2}{0.9 + 2} \right\}^{0-1+1} = \ln \left\{ \frac{0.9}{0.1} \cdot \frac{2.1}{2.9} \right\}^0 = 0$$

$$c_{11} = \ln \left\{ \frac{0.9}{0.1} \cdot \frac{0.1 + 2}{0.9 + 2} \right\}^{1-1+1} = \ln \left\{ \frac{0.9}{0.1} \cdot \frac{0.1 + 2}{0.9 + 2} \right\}^1 = \ln \left\{ \frac{1.89}{0.29} \right\} = \ln\{6.52\} = 1.875$$

$$c_{12} = \ln\{6.52\}^2 = 3.75$$

$$c_{20} = \ln\{6.52\}^{-1} = 0 \quad j \geq i - 1 \text{ özelliğinden}$$

$$c_{21} = \ln\{6.52\}^0 = 0$$

$$c_{22} = \ln\{6.52\}^1 = 1.875$$

$$c_{30} = \ln\{6.52\}^{-2} = 0 \quad j \geq i - 1 \text{ özelliğinden}$$

$$c_{31} = \ln\{6.52\}^{-1} = 0 \quad j \geq i - 1 \text{ özelliğinden}$$

$$c_{32} = \ln\{6.52\}^0 = 0$$

bulunur.

$$j = 0, 1, \dots, N - 1 \text{ değerleri için,}$$

$$c_{00} = \ln \left\{ \frac{\rho_1}{\rho_0} \cdot \frac{\rho_0 + k}{\rho_1 + k} \right\}^j = \ln \left\{ \frac{0.9}{0.1} \cdot \frac{0.1 + 2}{0.9 + 2} \right\}^0 = \ln\{6.52\}^0 = 0$$

$$c_{01} = \ln \left\{ \frac{\rho_1}{\rho_0} \cdot \frac{\rho_0 + k}{\rho_1 + k} \right\}^j = \ln \left\{ \frac{0.9}{0.1} \cdot \frac{0.1 + 2}{0.9 + 2} \right\}^1 = \ln\{6.52\} = 1.875$$

$$c_{02} = \ln\{6.52\}^2 = 3.75$$

bulunur.

$$i = 1, 2, \dots, N \text{ değerleri için,}$$

$$c_{13} = \ln \left\{ \frac{\left(\frac{\rho_1 + k}{k}\right)^k - \sum_{r=0}^{N-i} \binom{r+k-1}{r} \left(\frac{\rho_1}{\rho_1 + k}\right)^r}{\left(\frac{\rho_0 + k}{k}\right)^k - \sum_{r=0}^{N-i} \binom{r+k-1}{r} \left(\frac{\rho_0}{\rho_0 + k}\right)^r} \right\} = 6.04$$

$$c_{23} = 4.19$$

$$c_{33} = 2.37$$

bulunur.

$$c_{03} = c_{13} = 6.04 \text{ dir.}$$

Bu sonuçları ve n adet geçiş değerini aldığımızda,

$c_{00} =$	0	$n_{11} =$	0	$\Rightarrow$	0
$c_{01} =$	1.875	$n_{01} =$	1	$\Rightarrow$	1.875
$c_{02} =$	3.75	$n_{02} =$	1	$\Rightarrow$	3.75
$c_{03} =$	6.04	$n_{03} =$	0	$\Rightarrow$	0
$c_{10} =$	0	$n_{11} =$	2	$\Rightarrow$	0
$c_{11} =$	1.875	$n_{11} =$	2	$\Rightarrow$	3.75
$c_{12} =$	3.75	$n_{12} =$	0	$\Rightarrow$	0
$c_{13} =$	6.04	$n_{13} =$	0	$\Rightarrow$	0
$c_{20} =$	0	$n_{20} =$	0	$\Rightarrow$	0
$c_{21} =$	0	$n_{21} =$	1	$\Rightarrow$	0
$c_{22} =$	1.875	$n_{22} =$	0	$\Rightarrow$	0
$c_{23} =$	4.19	$n_{23} =$	1	$\Rightarrow$	4.19
$c_{30} =$	0	$n_{30} =$	0	$\Rightarrow$	0
$c_{31} =$	0	$n_{31} =$	0	$\Rightarrow$	0
$c_{32} =$	0	$n_{32} =$	1	$\Rightarrow$	0
$c_{33} =$	2.38	$n_{33} =$	0	$\Rightarrow$	0
Toplam:					13.57

$$a = k \ln \left(\frac{\rho_0 + k}{\rho_1 + k} \right)$$

$$= 2 \ln \left(\frac{2.1}{2.9} \right) = -0.65$$

$$an = 9 \cdot -0.65 = -5.81$$

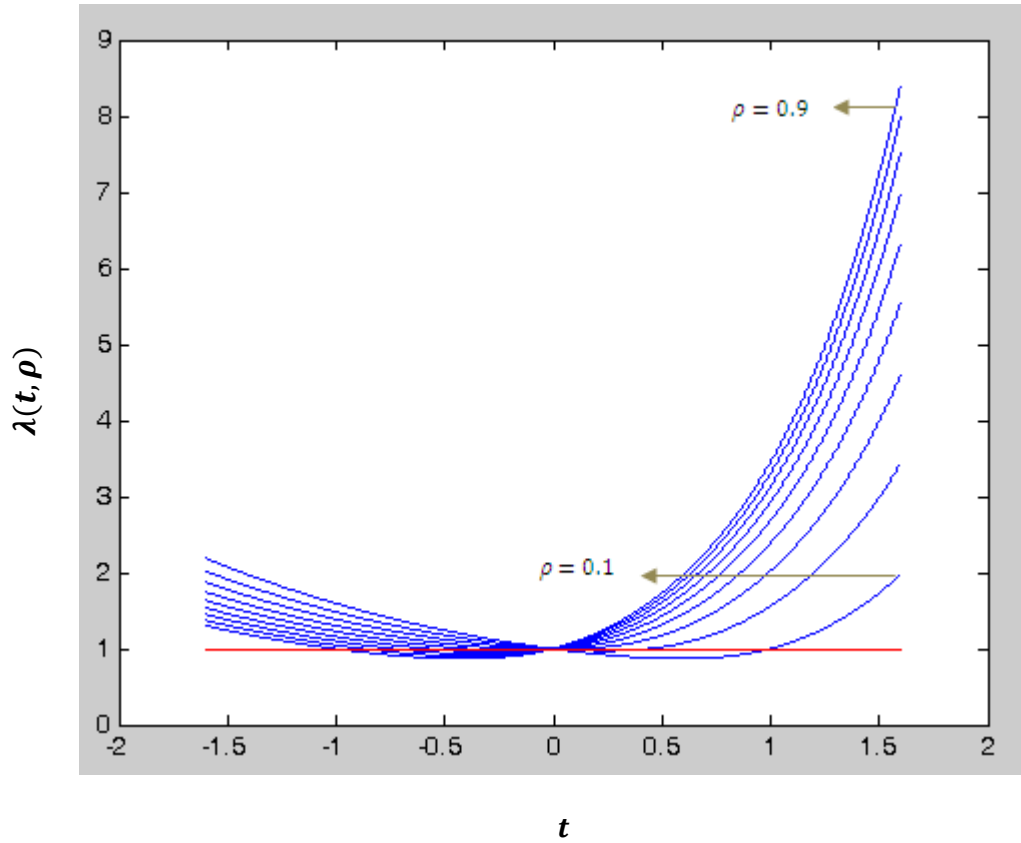
$$\ln L = an + \sum_{i,j} n_{ij} c_{ij}$$

$$\ln B - an < \sum_{i,j} n_{ij} c_{ij} < \ln A - an \quad \text{karar bölgesinden,}$$

13.57 > 8.7

olduğundan H_0 hipotezi ret edilir.

Karakteristik İşlem Eğrisini Çizmek ve Ortalama Örneklem Sayılarını bulmak için $P(t)$ matrisinin en büyük kökü $\lambda_0(t, \rho)$ sabit t ve ρ değerleri için hesaplanır. $\lambda_0(t, \rho)$ 'nın grafiği ρ 'nun farklı değerleri için t değerlerine göre Şekil 5.1'deki gibi çizilir.



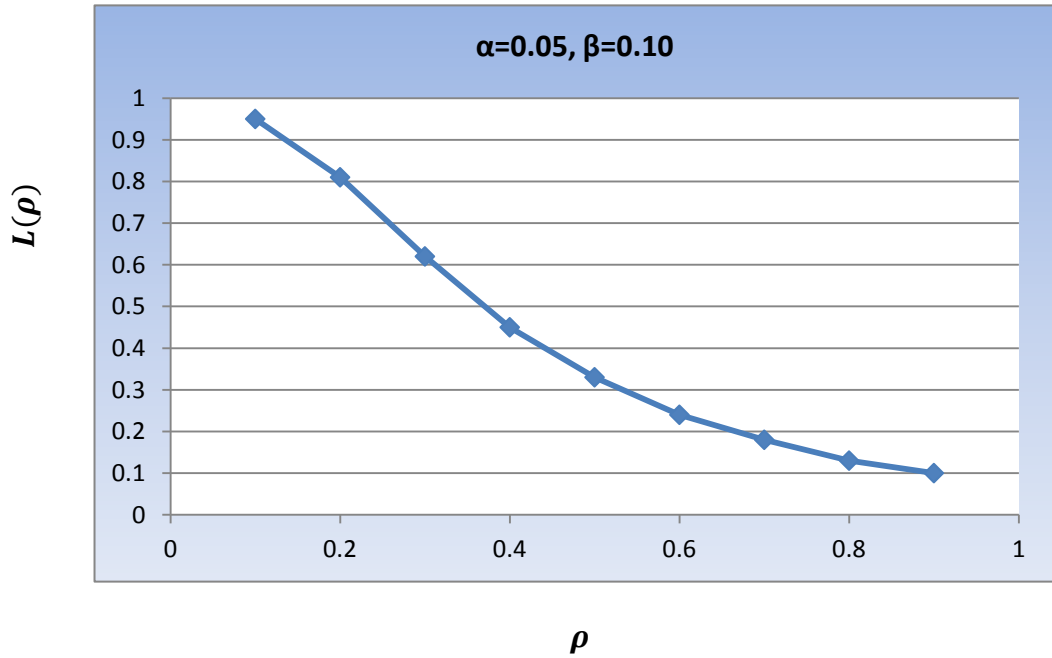
Şekil 5.1: $H_0: \rho = 0.1$, $H_1: \rho = 0.9$ testi için t 'ye karşı gelen $\lambda(t, \rho)$ 'nın grafiği.

Şekil 5.1'den anlaşılacağı gibi her bir ρ 'ya karşı gelen eğri $\lambda(t, \rho) = 1$ noktasında t 'yi sıfırdan hariç tek bir noktada kesmektedir. $t = 0$ noktasında $\lambda(t, \rho)$ 'nın türevleri $h=0.01$ alınarak (3.16) ve (3.17)'deki formüller vasıtasıyla hesaplanır. Karakteristik İşlem Eğrisi (3.12) ve (3.13)'deki, Ortalama Örneklem Sayısı ise (3.14) ve (3.15)'deki formüllerle hesaplanır. $H_0: \rho = 0.1$, $H_1: \rho = 0.9$ testi için Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları Çizelge 5.2'de verilmiştir.

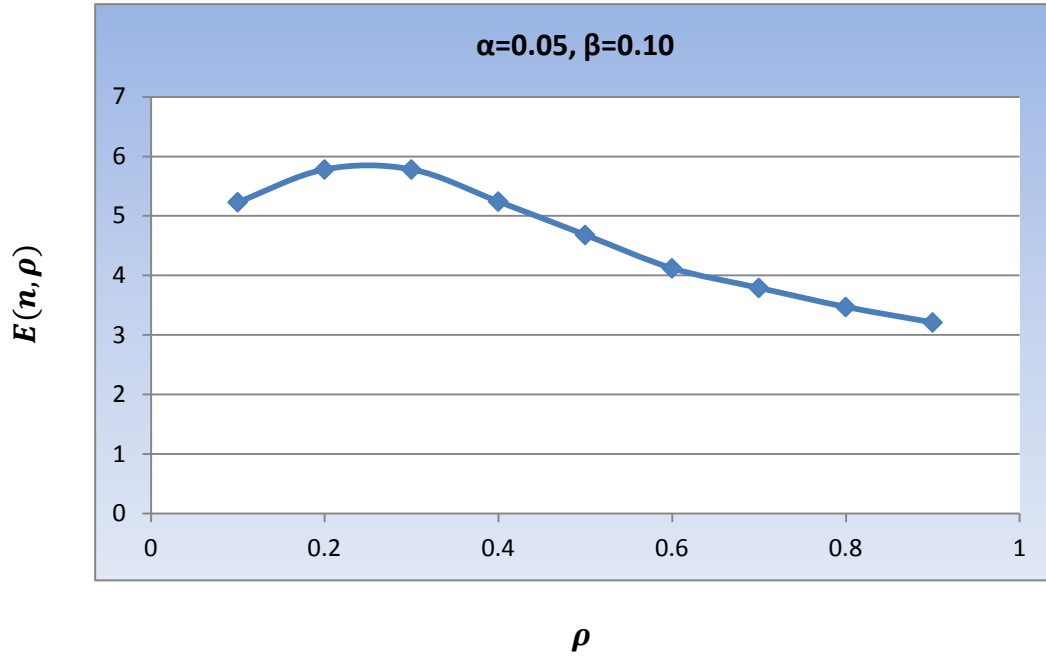
Çizelge 5.2: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.

ρ	$t(\rho)$	$\lambda'_0(0, \rho)$	$L(\rho)$	$E(n, \rho)$
0.1	1	-0.3811	0.95	5.23
0.2	0.4410	-0.2150	0.81	5.78
0.3	0.0900	-0.0504	0.62	5.78
0.4	-0.1750	0.1092	0.45	5.24
0.5	-0.3890	0.2602	0.33	4.68
0.6	-0.5710	0.3998	0.24	4.12
0.7	-0.7300	0.5265	0.18	3.79
0.8	-0.8718	0.6398	0.13	3.47
0.9	-1	0.7400	0.10	3.21

Aşağıdaki şekillerde Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları şematik olarak gösterilmiştir.



Şekil 5.2: $H_0: \rho = 0.1, H_1: \rho = 0.9$ testi için Karakteristik İşlem Eğrisi.



Şekil 5.3: $H_0: \rho = 0.1$, $H_1: \rho = 0.9$ testi için Ortalama Örneklem Sayısı.

5.2 Hyperexponential ($M/H_2/1$) Kuyruk Modeli İçin Sayısal Örnek

Örnek 5.2.1

$$\alpha_1 = 0.3 \quad \alpha_2 = 0.7$$

$$\mu_1 = 0.4 \quad \mu_2 = 0.3$$

$$\lambda_1 = 0.9 \quad \lambda_2 = 0.5$$

$$N = 7$$

değerleri alınarak ortalama ρ değeri aşağıdaki formülle hesaplanır.

$$\rho = \frac{\lambda}{\left(\frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2}\right)}$$

$$\rho_0 = 1.3, \rho_1 = 0.7$$

$\alpha = 0.05$, $\beta = 0.10$, $A = (1 - \beta)/\alpha = 18$, $B = \beta/(1 - \alpha) = 0.1053$ değerleri için

$$H_0: \rho = 1.3$$

$$H_1: \rho = 0.7$$

hipotezini test edelim.

$t_0 = 0, t_1, t_2, t_3, \dots M/H_2/1$ kuyruğunda müşterilerin ayrılış noktaları olsun.

0 0 0 3 2 1 0 1 0 6 6 5 6 5 5 4 4 3 3 2 1 2 2 1 0 0 4 3 2 1 3 4 3 2
2 1 4 3 2 3 2 2 1 0 1 0 1 0 0

gözlem değerleri müşterilerin 49 ayrılış noktasında kalan müşteri sayıları olsun.

E durum uzayı $\{0,1,2,3,4,5,6\}$ olur.

Çizelge 5.3’de i satır değeri j sütun değeri olmak üzere, n_{ij} $i \rightarrow j$ ’ye geçişlerin sayısını

Çizelge 5.3: $N = 7$ olmak üzere $i \rightarrow j$ ’ye geçişlerin sayısı.

	0	1	2	3	4	5	6	
0	4	3	0	1	1	0	1	$n_0 = 10$
1	6	0	1	1	1	0	0	$n_1 = 9$
2	0	6	3	1	0	0	0	$n_2 = 10$
3	0	0	6	1	1	0	0	$n_3 = 8$
4	0	0	0	4	1	0	0	$n_4 = 5$
5	0	0	0	0	1	1	1	$n_5 = 3$
6	0	1	0	0	0	2	1	$n_6 = 3$
	Toplam=							48

ve Çizelge 5.4 c_{ij} katsayılarını göstermek üzere,

Çizelge 5.4: Test için hesaplanan c_{ij} katsayıları.

	0	1	2	3	4	5	6
0	0.180121	-0.21514	-0.59559	-0.9606	-1.31182	-1.65269	-2.19273
1	0.180121	-0.21514	-0.59559	-0.9606	-1.31182	-1.65269	-2.19273
2	0	0.18012	-0.21514	-0.59559	-0.9606	-1.31182	-1.85218
3	0	0	0.18012	-0.21514	-0.59559	-0.9606	-1.50591
4	0	0	0	0.18012	-0.21514	-0.59559	-1.15013
5	0	0	0	0	0.18012	-0.21514	-0.78144
6	0	0	0	0	0	0.18012	-0.39798

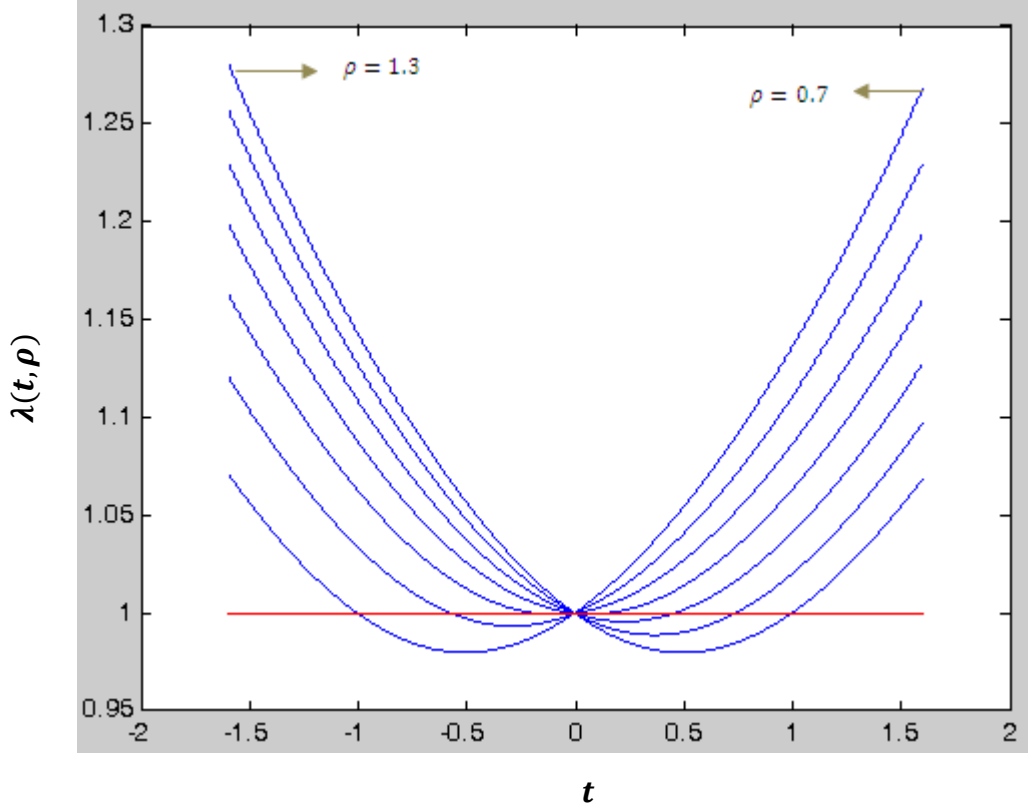
$$\ln L = \sum_{i,j} n_{ij} c_{ij} = -6.42$$

bulunur.

$\ln B < \sum_{i,j} n_{ij} c_{ij} < \ln A$ karar bölgesinden, $-2.2513 < \sum_{i,j} n_{ij} c_{ij} < 2.89$ olup

$-6.42 < -2.25$ olduğundan H_0 hipotezi kabul edilir.

Karakteristik İşlem Eğrisini Çizmek ve Ortalama Örneklem Sayılarını bulmak için $P(t)$ matrisinin en büyük kökü $\lambda_0(t, \rho)$ sabit t ve ρ değerleri için hesaplanır. $\lambda_0(t, \rho)$ 'nın grafiği ρ 'nun farklı değerleri için t değerlerine göre Şekil 5.4'deki gibi çizilir.



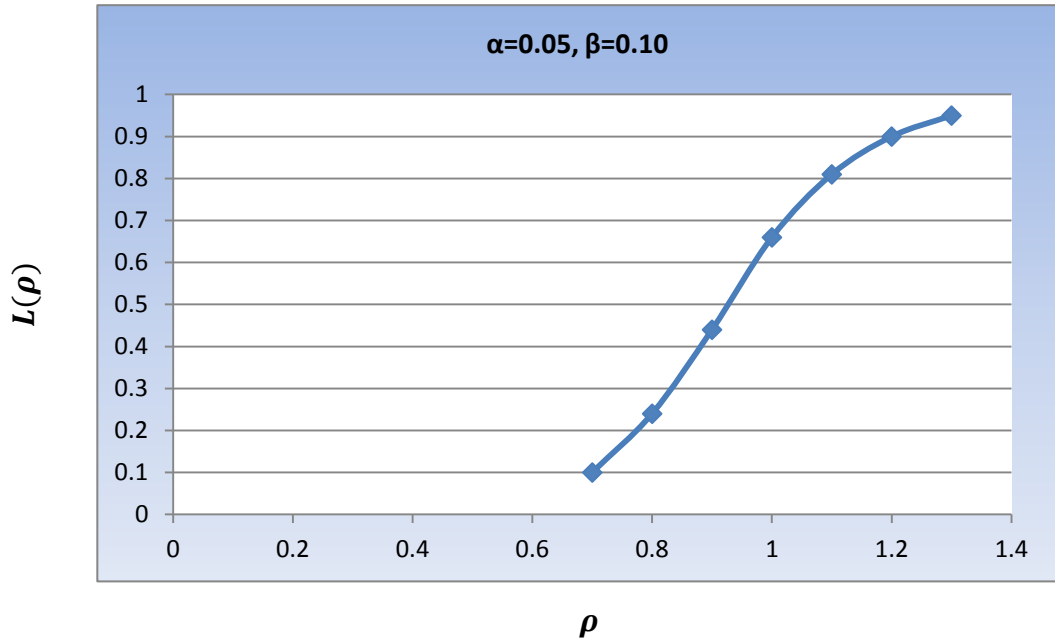
Şekil 5.4: $H_0: \rho = 1.3, H_1: \rho = 0.7$ testi için t 'ye karşı gelen $\lambda(t, \rho)$ 'nın grafiği.

Şekil 5.4'den anlaşılacağı gibi her bir ρ 'ya karşı gelen eğri $\lambda(t, \rho) = 1$ noktasında t 'yi sıfırdan hariç tek bir noktada kesmektedir. $t = 0$ noktasında $\lambda(t, \rho)$ 'nın türevleri $h=0.01$ alınarak (3.16) ve (3.17)'deki formüller vasıtasıyla hesaplanır. Karakteristik İşlem Eğrisi (3.12) ve (3.13)'deki, Ortalama Örneklem Sayısı ise (3.14) ve (3.15)'deki formüllerle hesaplanır. $H_0: \rho = 1.3, H_1: \rho = 0.7$ testi için Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları Çizelge 5.5'de verilmiştir.

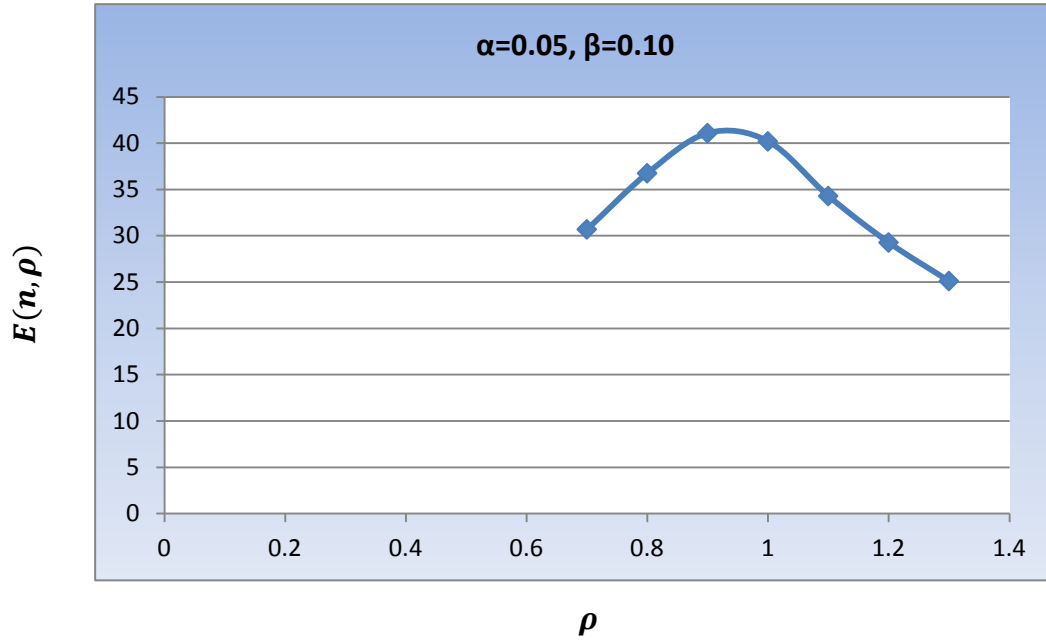
Çizelge 5.5: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.

ρ	$t(\rho)$	$\lambda'_0(0, \rho)$	$L(\rho)$	$E(n, \rho)$
1.3	1	-0.0794	0.95	25.11
1.2	0.74	-0.0598	0.90	29.27
1.1	0.46	-0.0375	0.81	34.29
1.0	0.16	-0.0124	0.66	40.19
0.9	-0.20	0.0154	0.44	41.08
0.8	-0.57	0.0456	0.24	36.75
0.7	-1	0.0775	0.10	30.69

Aşağıdaki şekillerde Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları şematik olarak gösterilmiştir.



Şekil 5.5: $H_0: \rho = 1.3$, $H_1: \rho = 0.7$ testi için Karakteristik İşlem Eğrisi.



Şekil 5.6: $H_0: \rho = 1.3$, $H_1: \rho = 0.7$ testi için Ortalama Örneklem Sayısı.

Örnek 5.2.2

$\alpha_1 = 0.3$ $\alpha_2 = 0.7$

$\lambda_1 = 0.3$ $\lambda_2 = 1.6$ $\mu_1 = 0.9$ $\mu_2 = 0.8$

$N = 5$

değerleri alınarak ortalama ρ değeri aşağıdaki formülle hesaplanır.

$$\rho = \frac{\lambda}{\left(\frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2}\right)}$$

$\rho_0 = 0.3$, $\rho_1 = 1.5$

$\alpha = 0.1$, $\beta = 0.01$, $A = (1 - \beta)/\alpha = 9.9$, $B = \beta/(1 - \alpha) = 0.01$ değerleri için

$H_0: \rho = 0.3$

$H_1: \rho = 1.5$

hipotezini test edelim.

$t_0 = 0, t_1, t_2, t_3, \dots M/H_2/1$ kuyruğunda müşterilerin ayrılış noktaları olsun.

0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 1 0 0 0 0 1 1 0 0 0 0 0 0 0

gözlem değerleri müşterilerin 50 ayrılış noktasında kalan müşteri sayıları olsun.

E durum uzayı $\{0,1,2,3,4\}$ olur.

Çizelge 5.6'da i satır değeri j sütun değeri olmak üzere, n_{ij} $i \rightarrow j$ 'ye geçişlerin sayısını

Çizelge 5.6: $N = 5$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.

	0	1	2	3	4	
0	39	3	1	0	0	$n_0 = 43$
1	4	1	0	0	0	$n_1 = 5$
2	0	1	0	0	0	$n_2 = 1$
3	0	0	0	0	0	$n_3 = 0$
4	0	0	0	0	0	$n_4 = 0$
	Toplam					49

ve Çizelge 5.7 c_{ij} katsayılarını göstermek üzere,

Çizelge 5.7: Test için hesaplanan c_{ij} katsayıları.

	0	1	2	3	4
0	-0.3118	0.8590	2.0045	3.1243	4.5661
1	-0.3118	0.8590	2.0045	3.1243	4.5661
2	0	-0.3118	0.8590	2.0045	3.4595
3	0	0	-0.3118	0.8590	2.3311
4	0	0	0	-0.3118	1.1778

$$\ln L = \sum_{i,j} n_{ij} c_{ij} = -8.28$$

bulunur.

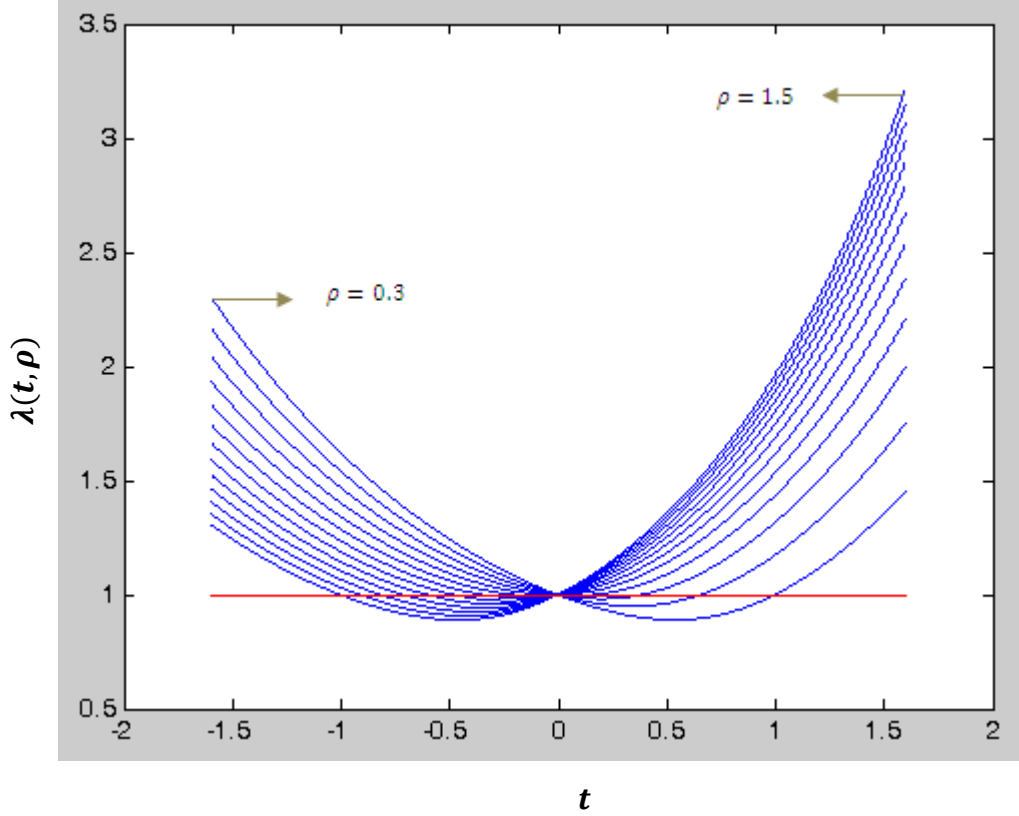
$$\ln B < \sum_{i,j} n_{ij} c_{ij} < \ln A \quad \text{karar bölgesinden,}$$

$$-4.5 < \sum_{i,j} n_{ij} c_{ij} < 2.29 \quad \text{olup}$$

$$-8.28 < -4.5 \quad \text{olduğundan } H_0 \text{ hipotezi kabul edilir.}$$

Karakteristik İşlem Eğrisini Çizmek ve Ortalama Örneklem Sayılarını bulmak için $P(t)$ matrisinin en büyük kökü $\lambda_0(t, \rho)$ sabit t ve ρ değerleri için hesaplanır.

$\lambda_0(t, \rho)$ 'nın grafiği ρ 'nun farklı değerleri için t değerlerine göre Şekil 5.7'deki gibi çizilir.



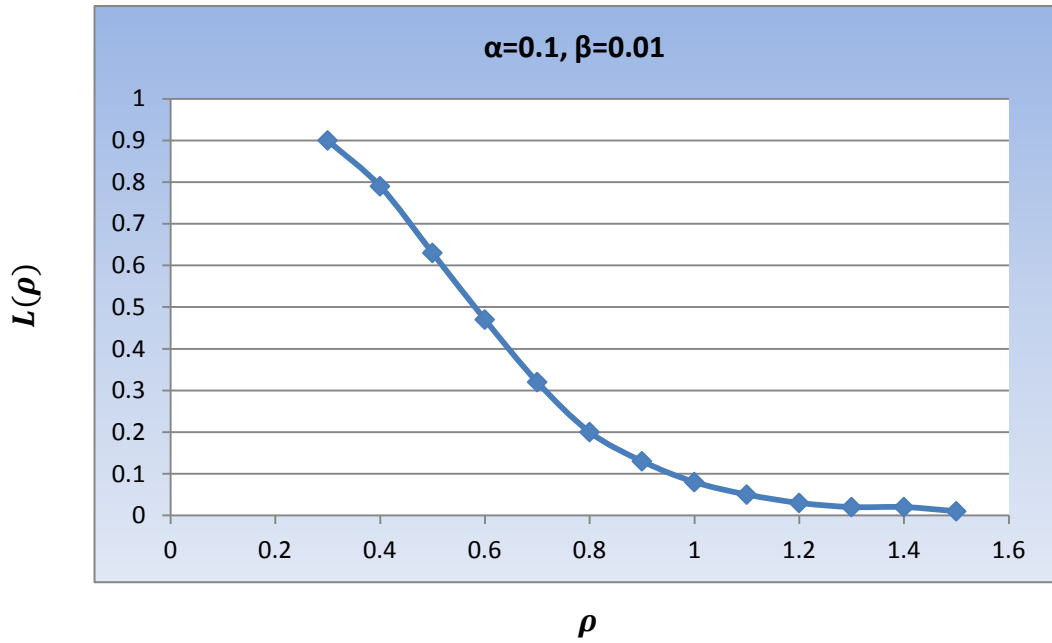
Şekil 5.7: $H_0: \rho = 0.3$, $H_1: \rho = 1.5$ testi için t ' ye karşı gelen $\lambda(t, \rho)$ ' nın grafiği.

Şekil 5.7'den anlaşılacağı gibi her bir ρ 'ya karşı gelen eğri $\lambda(t, \rho) = 1$ noktasında t 'yi sıfırdan hariç tek bir noktada kesmektedir. $t = 0$ noktasında $\lambda(t, \rho)$ 'nın türevleri $h=0.01$ alınarak (3.16) ve (3.17)'deki formüller vasıtasıyla hesaplanır. Karakteristik İşlem Eğrisi (3.12) ve (3.13)'deki, Ortalama Örneklem Sayısı ise (3.14) ve (3.15)'deki formüllerle hesaplanır. $H_0: \rho = 0.3$, $H_1: \rho = 1.5$ testi için Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları Çizelge 5.8'de verilmiştir.

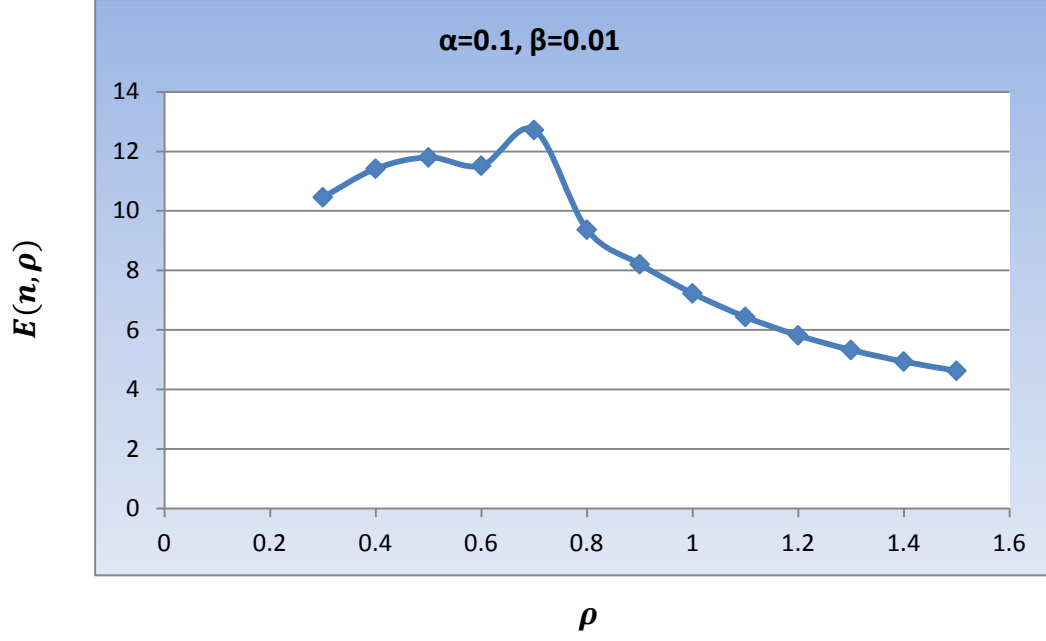
Çizelge 5.8: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.

ρ	$t(\rho)$	$\lambda'_0(0, \rho)$	$L(\rho)$	$E(n, \rho)$
0.3	1	-0.3652	0.90	10.46
0.4	0.66	-0.2675	0.79	11.42
0.5	0.39	-0.1705	0.63	11.80
0.6	0.16	-0.0761	0.47	11.52
0.7	0.03	0.0136	0.32	12.72
0.8	-0.20	0.0970	0.20	9.37
0.9	-0.34	0.1730	0.13	8.21
1.0	-0.48	0.2413	0.08	7.23
1.1	-0.60	0.3019	0.05	6.44
1.2	-0.71	0.3554	0.03	5.82
1.3	-0.81	0.4025	0.02	5.33
1.4	-0.91	0.4438	0.02	4.94
1.5	-1	0.4802	0.01	4.63

Aşağıdaki şekillerde Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları şematik olarak gösterilmiştir.



Şekil 5.8: $H_0: \rho = 0.3$, $H_1: \rho = 1.5$ testi için Karakteristik İşlem Eğrisi.



Şekil 5.9: $H_0: \rho = 0.3$, $H_1: \rho = 1.5$ testi için Ortalama Örneklem Sayısı.

Örnek 5.2.3

$$\alpha_1 = 0.4 \quad \alpha_2 = 0.6$$

$$\mu_1 = 0.6 \quad \mu_2 = 1.8$$

$$\lambda_1 = 0.5 \quad \lambda_2 = 0.8$$

$$N = 4$$

değerleri alınarak ortalama ρ değeri aşağıdaki formülle hesaplanır.

$$\rho = \frac{\lambda}{\left(\frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2}\right)}$$

$$\rho_0 = 0.5, \rho_1 = 0.8$$

$\alpha = 0.005$, $\beta = 0.05$, $A = (1 - \beta)/\alpha = 19.9$, $B = \beta/(1 - \alpha) = 0.05$ değerleri için

$$H_0: \rho = 0.5$$

$$H_1: \rho = 0.8$$

hipotezini test edelim.

$t_0 = 0, t_1, t_2, t_3, \dots M/H_2/1$ kuyruğunda müşterilerin ayrılış noktaları olsun.

0 0 0 3 2 1 0 1 0 3 3 2 2 1 0 0 1 3 2 1 1 0 1 0 0 0 0 0 0 0 0 1 0
0 0 0 0 1 2 1 1 0 0 2 1 0 0 gözlem değerleri müşterilerin 48 ayrılış
noktasında kalan müşteri sayıları olsun.

E durum uzayı $\{0,1,2,3\}$ olur.

Çizelge 5.9'da i satır değeri j sütun değeri olmak üzere, n_{ij} $i \rightarrow j$ 'ye geçişlerin sayısını

Çizelge 5.9: $N = 4$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.

	0	1	2	3	
0	17	5	1	2	$n_0 = 25$
1	8	2	1	1	$n_1 = 12$
2	0	5	1	0	$n_2 = 6$
3	0	0	3	1	$n_3 = 4$
	Toplam				47

ve Çizelge 5.10 c_{ij} katsayılarını göstermek üzere,

Çizelge 5.10: Test için hesaplanan c_{ij} katsayıları.

	0	1	2	3
0	-0.1587	0.11495	0.34835	0.74499
1	-0.1587	0.11495	0.34835	0.74499
2	0	-0.1587	0.11495	0.52381
3	0	0	-0.1587	0.28001

$\ln L = \sum_{i,j} n_{ij} c_{ij} = -1.1062$ bulunur. $\ln B < \sum_{i,j} n_{ij} c_{ij} < \ln A$ karar bölgesinden,

$-2.9907 < -1.1062 < 5.2470$ olduğundan sistem tekrar gözlenir ve 49 ayrılış noktasında aşağıdaki değerler bulunur.

0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 0 1 0 1 1 0 1 0 2 1 0 0 0 0 0 1
0 0 0 0 0 0 0 0 0 2 1 0 0 2 2

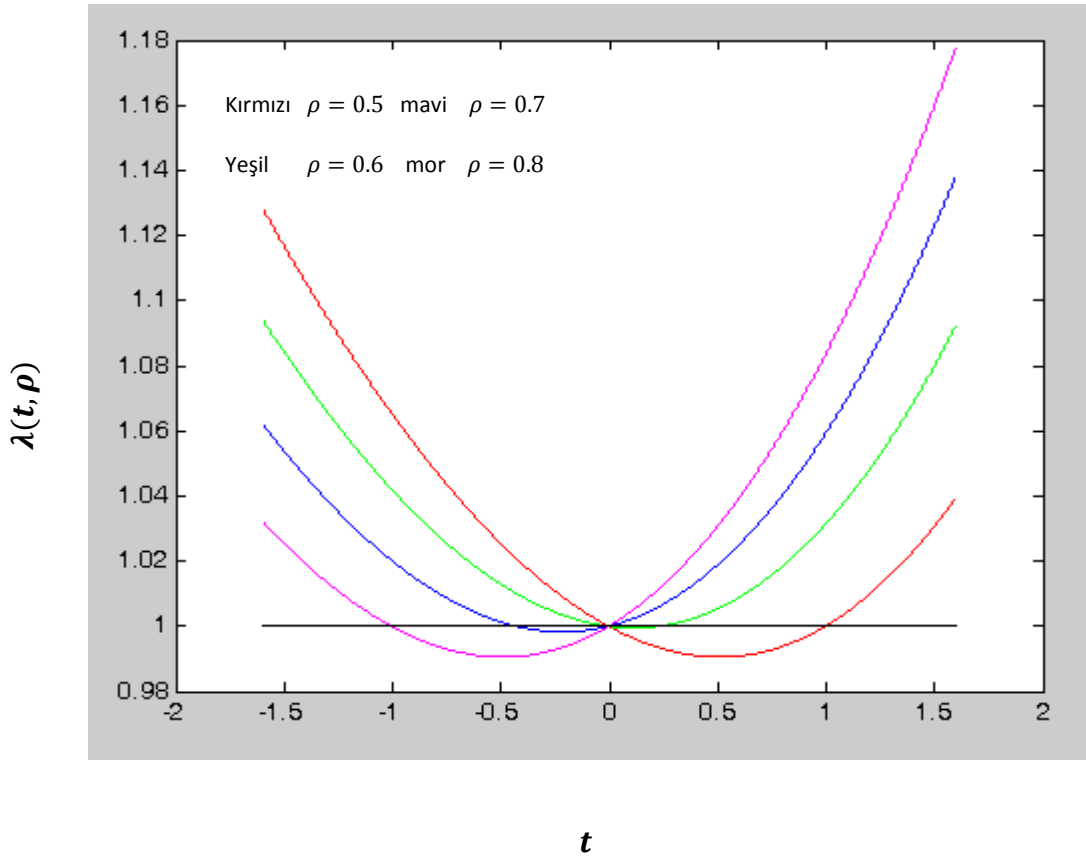
Çizelge 5.11'de i satır değeri j sütun değeri olmak üzere, n_{ij} $i \rightarrow j$ 'ye geçişlerin sayısını gösterir.

Çizelge 5.11: $N = 4$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.

	0	1	2	3	
0	42	11	4	2	$n_0 = 59$
1	16	6	1	1	$n_1 = 24$
2	0	7	2	0	$n_2 = 9$
3	0	0	3	1	$n_3 = 4$
	Toplam				96

$\ln L = \sum_{i,j} n_{ij} c_{ij} = -4.3517$ bulunur. $\ln B < \sum_{i,j} n_{ij} c_{ij} < \ln A$ karar bölgesinden, $-4.3517 < -2.9907$ olduğundan H_0 hipotezi kabul edilir.

Karakteristik İşlem Eğrisini Çizmek ve Ortalama Örneklem Sayılarını bulmak için $P(t)$ matrisinin en büyük kökü $\lambda_0(t, \rho)$ sabit t ve ρ değerleri için hesaplanır. $\lambda_0(t, \rho)$ 'nın grafiği ρ 'nun farklı değerleri için t değerlerine göre Şekil 5.10'daki gibi çizilir.



Şekil 5.10: $H_0: \rho = 0.5$, $H_1: \rho = 0.8$ testi için t 'ye karşı gelen $\lambda(t, \rho)$ 'nın grafiği.

Şekil 5.10'dan anlaşılaçağı gibi her bir ρ 'ya karşı gelen eğri $\lambda(t, \rho) = 1$ noktasında t 'yi sıfırdan hariç tek bir noktada kesmektedir. $t = 0$ noktasında $\lambda(t, \rho)$ 'nın türevleri $h=0.01$ alınarak (3.16) ve (3.17)'deki formüller vasıtasıyla hesaplanır. Karakteristik İşlem Eğrisi (3.12) ve (3.13)'deki, Ortalama Örneklem Sayısı ise (3.14) ve (3.15)'deki formüllerle hesaplanır. $H_0: \rho = 0.5$, $H_1: \rho = 0.8$ testi için Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları $\alpha = 0.005$, $\beta = 0.05$ için Çizelge 5.12'de verilmiştir ve $\alpha = 0.05$, $\beta = 0.1$ için Çizelge 5.13'de verilmiştir.

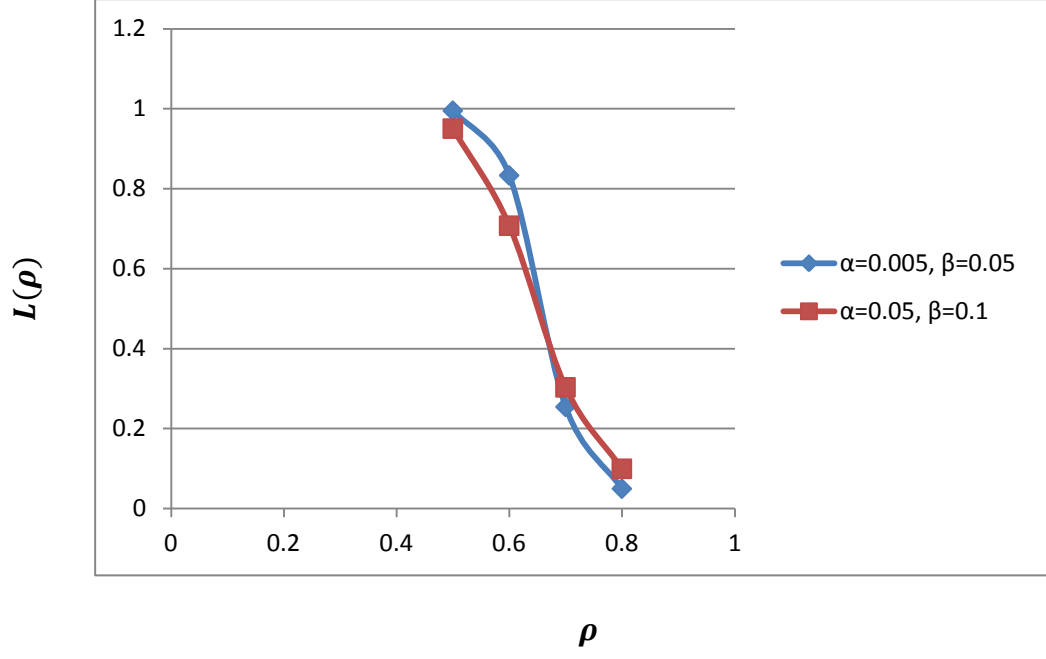
Çizelge 5.12: $\alpha = 0.005$, $\beta = 0.05$ için Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.

ρ	$t(\rho)$	$\lambda'_0(0, \rho)$	$L(\rho)$	$E(n, \rho)$
0.5	1	-0.0359	0.9951	82.14
0.6	0.24	-0.0087	0.8332	184.79
0.7	-0.43	0.0165	0.2543	191.34
0.8	-1	0.0395	0.0496	122.40

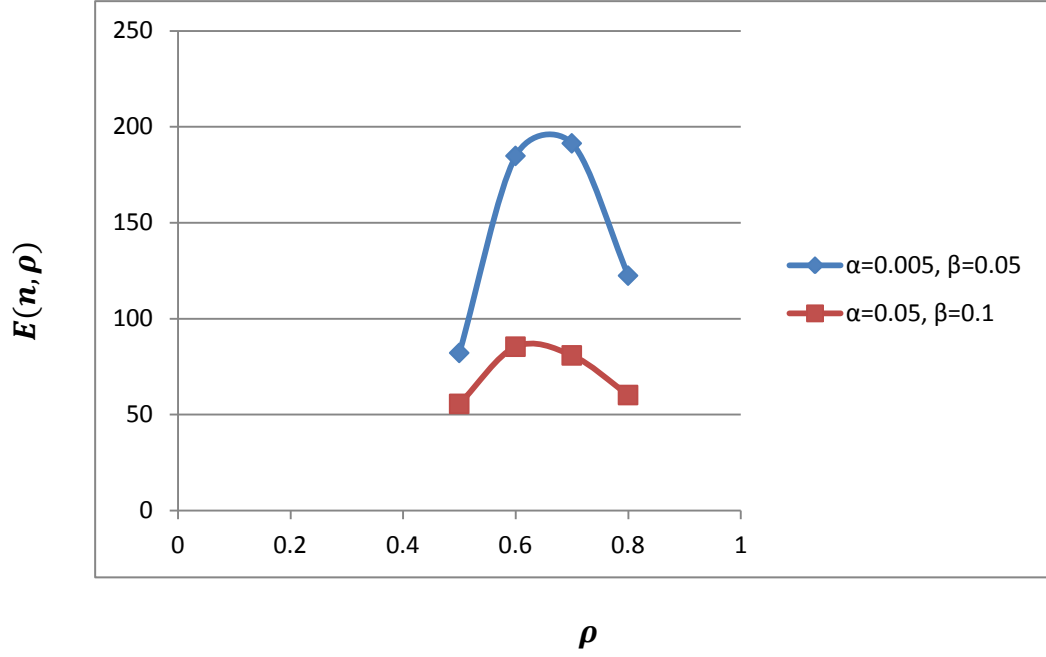
Çizelge 5.13: $\alpha = 0.05$, $\beta = 0.1$ Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.

ρ	$t(\rho)$	$\lambda'_0(0, \rho)$	$L(\rho)$	$E(n, \rho)$
0.5	1	-0.0359	0.9503	55.57
0.6	0.24	-0.0087	0.7074	85.38
0.7	-0.43	0.0165	0.3031	80.85
0.8	-1	0.0395	0.0994	60.19

Aşağıdaki şekillerde Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları iki ayrı hata olasılıkları için aynı şemada verilmiştir.



Şekil 5.11: $H_0: \rho = 0.5$, $H_1: \rho = 0.8$ testi için Karakteristik İşlem Eğrisi.



Şekil 5.12: $H_0: \rho = 0.5$, $H_1: \rho = 0.8$ testi için Ortalama Örneklem Sayısı.

Çizelgelerden ve Şekil 5.11 ve 5.12'den anlaşılacağı gibi hata olasılıkları küçüldükçe daha fazla örneklem sayısına ihtiyaç duyulmakta ve testin gücü artmaktadır.

5.3 Mixed Erlang ($M/mixE_k/1$) Kuyruk Modeli İçin Sayısal Örnek

Örnek 5.3.1

$\alpha_1 = 0.5$, $\alpha_2 = 0.5$, $\lambda = 0.1$, $\mu = 1.5$, $N = 10$

değerleri alınarak ortalama ρ değeri aşağıdaki formülle hesaplanır.

$$\rho = \frac{\lambda}{\left(\frac{\alpha}{\mu} + \frac{1-\alpha}{\mu}\right)} = \frac{\lambda}{\mu}$$

$\alpha = 0.1$, $\beta = 0.01$, $A = (1 - \beta)/\alpha = 9.9$, $B = \beta/(1 - \alpha) = 0.01$ değerleri için

$H_0: \rho_0 = 0.1$

$H_1: \rho_1 = 0.9$

hipotezini test edelim.

$t_0 = 0, t_1, t_2, t_3, \dots M/mixE_k/1$ kuyruğunda müşterilerin ayrılış noktaları olsun.

0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0

gözlem değerleri müşterilerin 25 ayrılış noktasında kalan müşteri sayıları olsun.

E durum uzayı $\{0,1,2,3,4,5,6,7,8,9\}$ olur. Çizelge 5.14'de i satır değeri j sütun değeri olmak üzere, n_{ij} $i \rightarrow j$ 'ye geçişlerin sayısını

Çizelge 5.14: $N = 10$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı.

	0	1	2	3	4	5	6	7	8	9	
0	20	2	0	0	0	0	0	0	0	0	22
1	2	0	0	0	0	0	0	0	0	0	2
2	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0
	Toplam										24

ve Çizelge 5.15 c_{ij} katsayılarını göstermek üzere,

Çizelge 5.15: Test için hesaplanan c_{ij} katsayıları.

	0	1	2	3	4	5	6	7	8	9
0	-0,15	2.05	4.29	6.53	8.78	11.04	13.29	15.55	17.81	20.11
1	-0,15	2.05	4.29	6.53	8.78	11.04	13.29	15.55	17.81	20.11
2	0	-0,15	2.05	4.29	6.53	8.78	11.04	13.29	15.55	17.85
3	0	0	-0,15	2.05	4.29	6.53	8.78	11.04	13.29	15.60
4	0	0	0	-0,15	2.05	4.29	6.53	8.78	11.04	13.34
5	0	0	0	0	-0,15	2.05	4.29	6.53	8.78	11.29
6	0	0	0	0	0	-0,15	2.05	4.29	6.53	8.85
7	0	0	0	0	0	0	-0,15	2.05	4.29	6.61
8	0	0	0	0	0	0	0	-0,15	2.05	4.37
9	0	0	0	0	0	0	0	0	-0,15	2.17

$\ln L = \sum_{i,j} n_{ij} c_{ij} = 0.76$ bulunur.

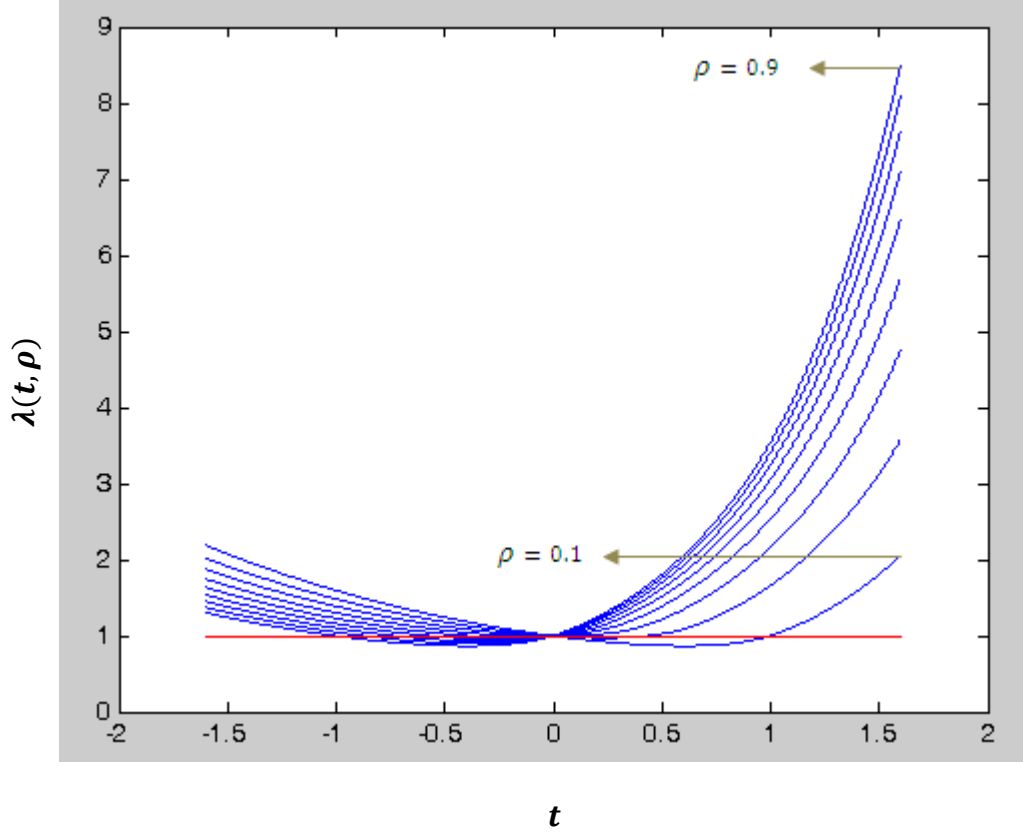
$\ln B < \sum_{i,j} n_{ij} c_{ij} < \ln A$ karar bölgesinden,

$$3.63 < \sum_{i,j} n_{ij} c_{ij} < 10.42$$

$$0.76 < 3.60$$

olduğundan H_0 hipotezi kabul edilir.

Karakteristik İşlem Eğrisini Çizmek ve Ortalama Örneklem Sayılarını bulmak için $P(t)$ matrisinin en büyük kökü $\lambda_0(t, \rho)$ sabit t ve ρ değerleri için hesaplanır. $\lambda_0(t, \rho)$ 'nın grafiği ρ 'nun farklı değerleri için t değerlerine göre Şekil 5.13'deki gibi çizilir.

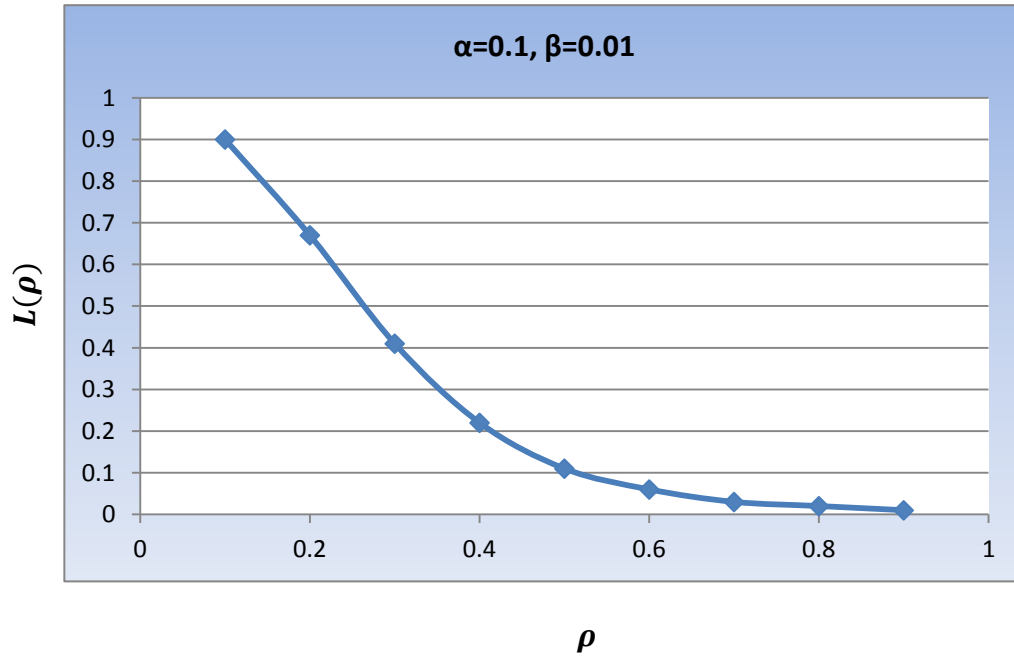


Şekil 5.13: $H_0: \rho = 0.1$, $H_1: \rho = 0.9$ testi için t' ye karşı gelen $\lambda(t, \rho)$ ' nın grafiği

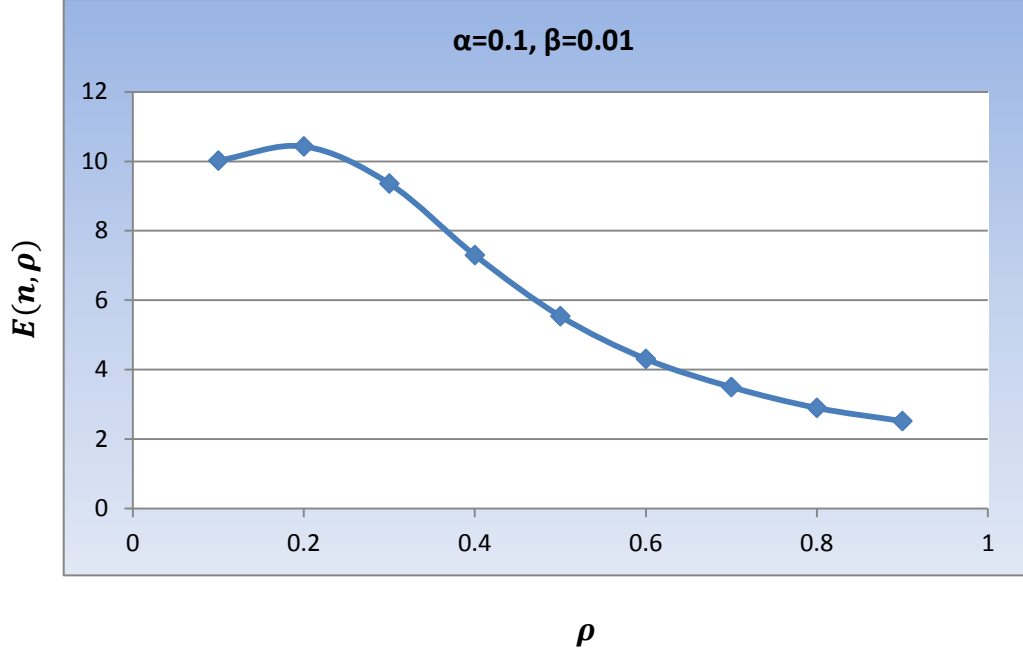
Şekil 5.13'den anlaşılabacağı gibi her bir ρ 'ya karşı gelen eğri $\lambda(t, \rho) = 1$ noktasında t 'yi sıfırdan hariç tek bir noktada kesmektedir. $t = 0$ noktasında $\lambda(t, \rho)$ ' nın türevleri $h=0.01$ alınarak (3.16) ve (3.17)'deki formüller vasıtasıyla hesaplanır. Karakteristik İşlem Eğrisi (3.12) ve (3.13)'deki, Ortalama Örneklem Sayısı ise (3.14) ve (3.15)'deki formüllerle hesaplanır. $H_0: \rho = 0.1$, $H_1: \rho = 0.9$ testi için Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları $\alpha = 0.1$, $\beta = 0.01$ için Çizelge 5.16'da verilmiştir.

Çizelge 5.16: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.

ρ	$t(\rho)$	$\lambda'_0(0, \rho)$	$L(\rho)$	$E(n, \rho)$
0.1	1	-0.3815	0.90	10.02
0.2	0.44	-0.2164	0.67	10.43
0.3	0.09	-0.0513	0.41	9.36
0.4	-0.18	0.1137	0.22	7.30
0.5	-0.39	0.2785	0.11	5.54
0.6	-0.57	0.4421	0.06	4.31
0.7	-0.73	0.6013	0.03	3.50
0.8	-0.87	0.7507	0.02	2.90
0.9	-1	0.8832	0.01	2.52



Şekil 5.14: $H_0: \rho = 0.1, H_1: \rho = 0.9$ testi için Karakteristik İşlem Eğrisi.



Şekil 5.15: $H_0: \rho = 0.1, H_1: \rho = 0.9$ testi için Ortalama Örneklem Sayısı.

5.4 Hypoexponential ($M/HYPO_k/1$) Kuyruk Modeli İçin Sayısal Örnek

Örnek 5.4.1

$$\lambda_1 = 0.5, \lambda_2 = 0.2, \mu_1 = 0.7, \mu_2 = 0.3, N = 7$$

değerleri alınarak ortalama ρ değeri aşağıdaki formülle hesaplanır.

$$\rho = \frac{\lambda}{\left(\frac{\alpha}{\mu} + \frac{1-\alpha}{\mu}\right)} = \frac{\lambda}{\mu}$$

$\alpha = 0.05, \beta = 0.10, A = (1 - \beta)/\alpha = 18, B = \beta/(1 - \alpha) = 0.1053$ değerleri için

$$H_0: \rho_0 = 0.1$$

$$H_1: \rho_1 = 0.04$$

hipotezini test edelim.

$t_0 = 0, t_1, t_2, t_3, \dots$ $M/HYPO_k/1$ kuyruğunda müşterilerin ayrılış noktaları olsun.

3 2 1 0 4 4 4 3 3 2 1 0 0 0 0 0 0 0 0 0 1 1 1 2 2 1 0 0 0 0 0 1 0
0 0 0 0 0 0 1 0 0 1 1 0 0 0 2 1 0 1 1 1 1 0 4 3 2 1 4 3 2 1 2 1 1 0
0 0 1 0 0 0 0 0 1 2 2 2 1 1 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0

gözlem değerleri müşterilerin 100 ayrılış noktasında kalan müşteri sayıları olsun.

E durum uzayı $\{0,1,2,3,4,5,6,7\}$ olur.

Çizelge 5.17’de i satır değeri j sütun değeri olmak üzere, n_{ij} , $i \rightarrow j$ ’ye geçişlerin sayısını

Çizelge 5.17: $N = 7$ olmak üzere $i \rightarrow j$ ’ye geçişlerin sayısı.

	0	1	2	3	4	5	6	
0	41	9	1	0	2	0	0	53
1	13	8	3	0	1	0	0	25
2	0	8	3	0	0	0	0	11
3	0	0	4	1	0	0	0	5
4	0	0	0	3	2	0	0	5
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
	toplam							99

ve Çizelge 5.18 c_{ij} katsayılarını göstermek üzere

Çizelge 5.18. Test için hesaplanan c_{ij} katsayıları.

	0	1	2	3	4	5	6
0	0.76	0.24	-0.26	-0.75	-1.23	-1.70	-0.06
1	0.76	0.24	-0.26	-0.75	-1.23	-1.70	-0.06
2	0	0.76	0.24	-0.26	-0.75	-1.23	-0.09
3	0	0	0.76	0.24	-0.26	-0.75	-0.12
4	0	0	0	0.76	0.24	-0.26	-0.15
5	0	0	0	0	0.76	0.24	-0.16
6	0	0	0	0	0	0.76	-0.12

$\ln L = \sum_{i,j} n_{ij} c_{ij} = 53.13$ bulunur.

$\ln B < \sum_{i,j} n_{ij} c_{ij} < \ln A$ karar bölgesinden,

$$-2.25 < \sum_{i,j} n_{ij} c_{ij} < 2.89$$

$$53.13 > 2.89$$

olduğundan H_0 hipotezi ret edilir.

Karakteristik İşlem Eğrisini Çizmek ve Ortalama Örneklem Sayılarını bulmak için $P(t)$ matrisinin en büyük kökü $\lambda_0(t, \rho)$ sabit t ve ρ değerleri için hesaplanır. $t = 0$ noktasında $\lambda(t, \rho)$ 'nın türevleri $h=0.01$ alınarak (3.16) ve (3.17)'deki formüller vasıtasıyla hesaplanır. Karakteristik İşlem Eğrisi (3.12) ve (3.13)'deki, Ortalama Örneklem Sayısı ise (3.14) ve (3.15)'deki formüllerle hesaplanarak Çizelge 5.19'da verilmiştir.

Çizelge 5.19: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.

ρ	$t(\rho)$	$\lambda'_0(0, \rho)$	$L(\rho)$	$E(n, \rho)$
0.1	1	-0.0299	0.95	66.73
0.09	0.75	-0.0213	0.90	82.34
0.08	0.45	-0.0127	0.82	102.37
0.07	0.18	0.0041	0.67	136.10
0.06	0.20	0.0045	0.44	144.10
0.05	0.57	0.0131	0.24	127.18
0.04	-1	0.0217	0.10	109.76

5.5 Coxian ($M/C_k/1$) Kuyruk Modeli İçin Sayısal Örnek

Örnek 5.5.1

alfa1 = 0.3 , alfa2 = 0.7, $\lambda_1 = 0.1$, $\mu_{01} = 1$, $\mu_{02} = 2$, $\mu_{11} = 0.5$, $\mu_{12} = 0.5$

$N = 3$, $k = 2$

değerleri alınarak ortalama ρ değeri aşağıdaki formülle hesaplanır.

$$\rho = \frac{\lambda}{\left(\frac{1}{\mu_1} + \frac{\alpha_1}{\mu_2}\right)}$$

$\rho_0 = 0.04, \rho_1 = 0.09$ olarak bulunur.

$\rho_0 = 0.04$ için;

$\sigma_{2 \times 1} = [1 \ 0]$ satır vektörü

$S_{2 \times 2} = \begin{bmatrix} 0.1 & 0.2 \\ 0.5 & 0.6 \end{bmatrix}$ kare matris

$S_{1 \times 2}^0 = [0.3 \ 0.6]$ sütun vektörü

$I_{2 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ birim matris

$\rho_1 = 0.09$ için;

$\sigma_{2 \times 1} = [1 \ 0]$ satır vektörü

$S_{2 \times 2} = \begin{bmatrix} -1 & 0.3 \\ 0 & -2 \end{bmatrix}$ kare matris

$S_{1 \times 2}^0 = [0.7 \ 0.6]$ sütun vektörü

$I_{2 \times 2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ matris

değerleri alınarak,

$\alpha = 0.1, \beta = 0.05, A = (1 - \beta)/\alpha = 9.5, B = \beta/(1 - \alpha) = 0.05$ değerleri için

$H_0: \rho_0 = 0.04$

$H_1: \rho_1 = 0.09$

hipotezini test edelim.

$t_0 = 0, t_1, t_2, t_3, \dots M/C_k/1$ kuyruğunda müşterilerin ayrılış noktaları olsun.

1 0 1 0 0 0 0 2 2 1 0 2 1 2 1 0 2 1 1 0

gözlem değerleri müşterilerin 20 ayrılış noktasında kalan müşteri sayıları olsun.

E durum uzayı $\{0,1,2\}$ olur.

Çizelge 5.20'de i satır değeri j sütun değeri olmak üzere, n_{ij} $i \rightarrow j$ 'ye geçişlerin sayısını gösterir.

Çizelge 5.20: $N = 3$ olmak üzere $i \rightarrow j$ 'ye geçişlerin sayısı

	0	1	2	
0	3	1	3	7
1	5	1	1	7
2	0	4	1	5
	Toplam			19

ve Çizelge 5.21 c_{ij} katsayılarını göstermek üzere

Çizelge 5.21: Test için hesaplanan c_{ij} katsayıları.

	0	1	2
0	0.87	-1.88	-0.14
1	0.87	-1.88	-0.14
2	0	0.87	-0.90

$\ln L = \sum_{i,j} n_{ij} c_{ij} = 5.19$ bulunur.

$\ln B < \sum_{i,j} n_{ij} c_{ij} < \ln A$ karar bölgesinden,

$$-2.89 < \sum_{i,j} n_{ij} c_{ij} < 2.25$$

$$5.19 > 2.89$$

olduğundan H_0 hipotezi ret edilir.

Karakteristik İşlem Eğrisini Çizmek ve Ortalama Örneklem Sayılarını bulmak için $P(t)$ matrisinin en büyük kökü $\lambda_0(t, \rho)$ sabit t ve ρ değerleri için hesaplanır. $t = 0$ noktasında $\lambda(t, \rho)$ 'nın türevleri $h=0.01$ alınarak (3.16) ve (3.17)'deki formüller vasıtasıyla hesaplanır. Karakteristik İşlem Eğrisi (3.12) ve (3.13)'deki, Ortalama Örneklem Sayısı ise (3.14) ve (3.15)'deki formüllerle hesaplanarak Çizelge 5.22'de verilmiştir.

Çizelge 5.22: Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayıları.

ρ	$t(\rho)$	$\lambda'_0(0, \rho)$	$L(\rho)$	$E(n, \rho)$
0.04	1	-0.0164	0.90	145.43
0.05	0.50	-0.0087	0.73	173.99
0.06	0.11	-0.0011	0.52	314.76
0.07	-0.34	0.0065	0.40	156.05
0.08	-0.67	0.0141	0.11	119.16
0.09	-1	0.0216	0.05	92.34

Bu bölümdeki sayısal örneklerin program kodları Ek D - Ek I bölümleri arasında verilmiştir.

6. SONUÇ VE ÖNERİLER

Ardışık Olasılık Oran Testi gömülü Markov zincirinin kullanılabildiği herhangi bir stokastik sürece uygulanabilir. Rao ve ark. (1984) gözlem olarak sistemdeki müşteri sayısını kullanan kuyruk modellerinde Ardışık Olasılık Oran Testini kullanmışlardır. Bu konuda yapılmış çalışmalar oldukça sınırlıdır.

Ardışık Olasılık Oran Testinin burada kullanılmasının amacı daha kısa zamanda ve daha az örneklemle trafik yoğunluğundaki değişimleri tespit etmek ve uygun düzeltici eylemleri almaktır. Yani trafik yoğunluğunu istenilen düzeyde (ρ_0) tutmaktır. Örneğin trafik yoğunluğu ρ , istenilen düzey ρ_0 'dan ρ_1 'e değiştiğinde ($\rho_1 > \rho_0$), uygun eylem ρ_1 'den istenilen düzey ρ_0 'a trafik yoğunluğunu geri getirmektir. Bunu sağlamak için servis oranı μ artırılabilir ya da kanal sayısı artırılabilir. Benzer olarak trafik yoğunluğu istenilen düzey ρ_0 'dan ρ_1 'e değiştiğinde ($\rho_1 < \rho_0$) servis oranı μ azaltılabilir ya da kanal sayısı azaltılabilir. Bu test dikkati çekecek kadar az örneklem kullanımıyla sonuç elde etme avantajına sahiptir (Bhat,1987).

Ardışık Olasılık Oran Testinin sınırlarından birisi sonlu gömülü Markov zincirleriyle kısıtlı olmasıdır. Rao ve Bhat (1980) Ardışık Olasılık Oran Testinin sonsuz Markov zincirlerine de uygulanabileceğini göstermişlerdir. Fakat bunun yalnız $M/G/1$ ve $GI/M/1$ tipindeki tek kanallı sistemlere uygulanabileceğini, Karakteristik İşlem Eğrisi ve Ortalama Örnek Sayısının hesaplanması oldukça zor olduğu için çok kanallı ve yığın kuyruklarda uygulanamayacağını söylemişlerdir.

Bu çalışmada trafik yoğunluğunun kontrolü faz dağılımlı Hyperexponential ($M/H_2/1$), Mixed Erlang ($M/mixE_k/1$), Hypoexponential ($M/HYPO_k/1$) ve Coxian ($M/C_k/1$) kuyruk modelleri için önce teorik olarak çalışıldı ve Ardışık Olasılık Oran Testi için gereken katsayılar bulundu. Daha sonra her bir model için Matlab programıyla farklı geliş oranları (λ), servis oranları (μ) ve verilen sistem kapasitesi (N) için sistemde kalan müşteri sayıları rasgele üretildi. Trafik yoğunlukları hesaplandı. Burada elde edilen verilere ve daha önceden belirlenen α (1. tip hata) ve

β 'ya (2. tip hata) bağılı olarak basit hipotezler kurularak Ardışık Olasılık Oran Testi ile hipotezlerin kabul ya da ret edilip edilmediklerine bakıldı. Herhangi bir karara ulaşılmadıysa program ile tekrar veri üretilerek aynı yöntemlerle bir karara ulaşılmaya çalışıldı. Daha sonra program ile hipotez aralığındaki çeşitli ρ ve t değerleri için $P(t)$ matrisinin en büyük kökleri hesaplanarak grafikleri çizildi ve bu grafiklerin $\lambda_0(t, \rho) = 1$ noktasında $t \neq 0$ için t eksenini hangi noktalarda kestikleri bulundu. Elde edilen bu değerlerle Karakteristik İşlem Eğrileri ve Ortalama Örneklem Sayıları hesaplandı ve Excel programıyla grafikleri çizildi.

Sonuç olarak $\rho > 1$ ($\lambda > \mu$) olduğunda ayrışım noktalarında gözlenen kuyruk uzunluğunun maksimum (N) olabildiği, $\rho < 1$ ($\lambda < \mu$) olduğunda ise ayrışım noktalarında gözlenen kuyruk uzunluğunun minimum (N) civarlarında olduğu gözlemlendi. Örneklerde sistem kapasitesi (N) genelde küçük seçildi fakat daha büyük değerlerde seçilebilir.

Yine verilen α ve β 'ya göre örneklem büyüklüğünde farklılar gözlemlendi. Hata olasılıkları küçüldükçe daha fazla örneklem sayısına ihtiyaç duyulduğu ve testin gücü ($1 - \beta$)'nin arttığı görüldü. Daha sonra β 'yı sabit tutup ve α 'nın azaltırken örneklem sayısında arttığı görüldü. Ayrıca α 'yı sabit tutup β artarken örneklem sayısının azaldığı gözlemlendi.

Ayrıca trafik yoğunluğunu hesaplamak için gereken servis oranı (μ) faz dağılımlı kuyruk sistemleri için Ek C'de verilen formüllerle ortalama μ değeri olarak hesaplandı. Yani trafik yoğunluğunu istenilen düzeyde tutabilmek için ortalama μ değerinde artırmaya ya da azaltmaya gidilebileceği görüldü.

Bunlara ilave olarak göz önüne alınan kuyruk sistemlerini ifade eden diferansiyel denklemlerin süreçteki parametrelere göre kararlılık özellikleri incelenebilir. Ayrıca diferansiyel denklemler lineer olduğundan kuyruk sistemi modelleri lineer durum için sonuçlar vermektedir. O halde sistemi nonlineer edecek terimler ilave edilerek daha zor ve karmaşık modeller üzerine sonuçlar alınabilir. Bu durum ise bu alanda yeni çalışma alanları açabilir.

KAYNAKLAR

- Bener, A., *Matematiksel İstatistik*, Yıldız Teknik Üniversitesi Vakfı Yayın No: YTÜVAK.FE.DK-2002.002,Cilt II, İstanbul, 2002.
- Benes, V.E., A sufficient set of statistics for a simple telephone exchange model, *Bell Syst. Tech. J.*, **36**, 1957,939.
- Basava, I.V., Prabhu, N.U., Estimation in single server queues, *Nav.Res. Log. Quart.*, **28(3)**, 1981,475-487.
- Basava, I.V., Bhat, U.N., Lund, R., Maximum likelihood estimation for single server queues from waiting time, *Queueing System*, **24**, 1996,155-167.
- Bhat, U.N., A Sequential Technique For The Control of Traffic Intensity in Markovian Queues, *Annals of Operations Research*, **8**, 1987,151-164.
- Bhat, U.N., An Introduction to Queueing Theory Modelling and Analysis in Applications, Birkhäuser, Boston, 2008.
- Bhat, U.N., *Elements of Applied Stochastic Processes*, A Wiley-Interscience Publication, New York, 1972.
- Clarke, A.B., Maximum likelihood estimates in a simple queue, *Annals of Mathematical Statisticians*, **28**, 1957,1036-1040.
- Choudhury, A., Borthakur, A.C., Bayesian inference and prediction in the single server Markovian queue, *Metrika*, **67**, 2008, 371-383.
- Çıngı, H., *Ardışık Çözümleme*, Hacettepe Üniversitesi, Fen Fakültesi, İstatistik Bölümü, Ankara, 2005.
- Dave, U., Shah, Y.K., Maximum likelihood estimates in the $M/M/2$ queue with heterogeneous servers, *Journal of Operational Research*, **31**, 1980, 423-426.
- Gross, D., Harris, C.M., *Fundamentals of Queueing Theory*, 2nd ed., John Wiley & Sons, New York, 1985.
- Harishchandra, K., Rao, S.S., Statistical inference about the traffic intensity parameter of $M/E_k/1$ and $E_k/M/1$ queues, Tech. Report, *Indian Institute of Management*, Bangalore, India, 1984.
- Harishchandra, K., Rao, S.S., A note on statistical inference about the traffic intensity parameter of $M/E_k/1$ queue, *Sankhya*, **50**, 1988,144-148.
- Karlis, D., Em Algorithm for Mixed Poisson and Other Discrete Distributions, *Astin Bulletin*, Vol. **35**, 2005, No.1, 3-24.
- McCormick, M.J., Salvadori, M.G., *Numerical Methods in Fortran*, Prentice-Hall, Englewood Cliffs, NJ, 1964.
- Mood, A., *Introduction to the Theory of Statistics*, Mc-Graw Hill, New York,1974.

- Muddapur, M.V., Bayesian estimates of parameters in some queueing models, *Ann. Inst. Math.*, 1972, **24**, 327-331.
- Pala, R.E., Beloso, A.C., Hines W.W., *Waiting Line Models An Introduction to Their Theory and Application*, Reinhold Publishing Corporation, New York, 1966.
- Perron, O., Zur Theorie der Matrices, *Math. Annalen*, 1907, **64**, 248-263.
- Phatarfod, R.M., Sequential analysis of dependent observations-I, *Biometrika*, 1965, **52**, 157-165.
- Rao, S.S., Bhat, U.N., A Sequential Test for Denumerable Markov chain and application to Queues, *Tech. Report. School of Business, Virginia Commonwealth University*, Richmond, VA, 1980.
- Rao, S.S., Bhat, U.N., Harishchanra, K., Control of traffic intensity in a queue – A method based on SPRT, *Opsearch*, 1984, **21 (2)**, 63-80.
- Rao, S.S., Bhat, U.N., Statistical Analysis Of Queueing Systems, *Queueing Systems*, 1987, **1**, 217- 247.
- Sağlam, V., İki Aşamalı Tandem Sistemlerinin Analizi ve Optimizasyonu. Doktora Tezi, Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü ,Samsun, 1997.
- Shahbazov, A., *Olasılık Teorisine Giriş*, Birsen Yayınevi Ltd. Şti., İstanbul, 2005.
- Stewart, W.J., *Probability, Markov Chains, Queues, and Simulation*, Princeton University Press, United Kingdom, 2009.
- Wald, A., *Sequential Analysis*, Dover Publications, New-York, Second Edition, 1947.
- Taha, H., *Yöneylem Araştırması*, Literatür Yayıncılık, İstanbul, 2000.
- Thiagarajan, T.R., Harris, C.M., Statistical tests for exponential service from $M/G/1$ waiting time data, *Nav. Res. Logist. Quart.*, 1979, **26**, 511-520.
- Thornett, M.L., Some Results on the First Latent Root of a Transition Probability Matrix, Mathematics Research Report, University of Western Australia, 1961.
- Torun, H., İki Heterojen Kanallı Kuyruk Sistemi ve $M/M/c$ 'nin Uygulaması. Yüksek Lisans Tezi, Ondokuz Mayıs Üniversitesi Fen Bilimleri Enstitüsü, Samsun, 2002.
- Uzunoğlu, M., Kızıl, A., Onar, Ö.Ç., *Her Yönü ile Matlab*, Genişletilmiş 2. Baskı, Türkmen Kitabevi, Yayın No:219, İstanbul, 2003.
- Yüksel, İ., *Matlab ile Mühendislik Sistemlerinin Analizi ve Çözümü*, Genişletilmiş 3. Baskı, Nobel Yayın Dağıtım, İstanbul, 2004.

EKLER

Ek A: Kuyruklarda trafik yoğunluğunun kontrolü için yapılan Ardışık Olasılık Oran Testinin adımları aşağıdaki gibidir.

Adım 1: İlk Adım

- Sistem için ideal ve ya arzu edilen $\rho = \rho_0$ düzeyini tanımla,
- Kuyruk sisteminin burada çalışılan kuyruk sistemlerinden birisi olduğuna emin ol,
- Durum uzayının boyutunu N olarak belirle,
- ρ 'yu sürekli izlemek için aşağıdaki gibi iki hipotez kur :

- $H_0^{(1)}: \rho = \rho_0$
 $H_1^{(1)}: \rho = \rho_1 (\rho_1 < \rho_0)$
- $H_0^{(2)}: \rho = \rho_0$
 $H_1^{(2)}: \rho = \rho_2 (\rho_2 > \rho_0)$

i 'deki hipotez ρ 'daki aşağıya doğru değişimleri tespit eder, ii'deki hipotez ρ 'daki yukarıya doğru değişimleri tespit eder. ρ_1 ve ρ_2 'nin seçiminde bazı ön bilgiler gereklidir. $|\rho_1 - \rho_0|$ ya da $|\rho_2 - \rho_0|$ farkı büyükse ρ 'daki değişim daha hızlıdır.

- α ve β için uygun değerleri seç ve

$$A = \frac{1 - \beta}{\alpha} \text{ ve } B = \frac{\beta}{1 - \alpha} \text{ değerlerini hesapla.}$$

Adım 2: Keşif Adımı

- $Q_0 = i_0$ ilk kuyruk uzunluğu ile başla ve n . yenilenme noktası t_n 'de $Q_n = (t_n)$ olmak üzere $n=1,2,\dots$ noktalarında Q_n . kuyruk uzunluğunu gözlemle,
- n . aşamada
 $\ln L = an + \sum_{i,j} c_{ij} n_{ij}$ 'yi hesapla,
 a ve c_{ij} katsayıları (ρ_0, ρ_1) ve (ρ_0, ρ_2) için ayrı ayrı hesaplanır. Sırayla L_1 ve L_2 ile $\ln L$ 'nin bu değerleri gösterilsin.
- Eğer $L_1 \geq \ln A$ ise $H_1^{(1)}$ hipotezini kabul et, Kontrol Adımı 3(a)'ya git, Eğer $L_2 \geq \ln A$ ise $H_1^{(2)}$ hipotezini kabul et, Kontrol Adımı 3(b)'ye git, Eğer $L_1 \leq \ln B$ ve $L_2 \leq \ln B$ ise $H_0^{(1)} = H_0^{(2)}: \rho = \rho_0$ hipotezlerini kabul et, Kontrol Adımı 3(b)'ye git.
Aksi takdirde Q_{n+1} . kuyruk uzunluğunu gözlemle ve Adım 1(b) ve Adım 1(c)' yi tekrar et.

Adım 3: Kontrol Adımı

- a) Eğer Adım 2 (c)'deki $H_1^{(1)}$ kabul edilirse, bu ρ 'nun dizayn düzeyi ρ_0 'dan ρ_1 'e ($\rho_1 < \rho_0$) değiştiği anlamına gelir. Kontrol eylemi servis süresi μ 'yi azaltmak ya da kanal sayısını indirmektedir.
- b) Eğer Adım 2 (c)'deki $H_1^{(2)}$ kabul edilirse, bu ρ 'nun dizayn düzeyi ρ_0 'dan ρ_2 'e ($\rho_2 > \rho_0$) değiştiği anlamına gelir. Kontrol eylemi servis süresi μ 'yi artırmak, ek kanal eklemek ya da geliş oranı için mümkün kontrolleri yapmaktır.
- c) Eğer Adım2 (c)'deki $H_0^{(1)}$ (eşdeğeri $H_0^{(2)}$) kabul edilirse, bu ρ 'da meydana gelen hiçbir değişikliğin olmadığı anlamına gelir ve hiçbir kontrol eylemi gerektirmez. Bu durumda Adım 2 ve Adım 3 yeni bir gözlem dizisi ile tekrarlanır.

Ek B: Gömülü Markov zinciri için bir örnek:

Bir berber dükkanında bekleyen müşteriler için dört sandalye vardır ve bir berber hizmet vermektedir. Gelen müşteriler tüm sandalyeler dolu ise saçlarını kestirmeden ayrılmaktadırlar. Berberin saç kesimi tam olarak 15 dakika sürmektedir ve bekleyen müşteri oldukça herhangi boş zamanı olmamaktadır. Minimum 15 dakikalık zaman aralığında gelen müşterileri sayısı aşağıdaki dağılıma sahiptir (Bhat,1972).

Gelenlerin Sayısı	0	1	2	3	4	5 ve üstü
Olasılığı	0.20	0.70	0.07	0.02	0.01	0.00

n . saç kesiminden hemen sonra dükkanda kalan müşterilerin sayısını Q_n olarak alalım. Q_n 'in mümkün değerleri $\{0,1,2,3,4\}$ dir. (Kolaylık açısından aynı anda geliş ve ayrılış olmadığı varsayılır. Böylece berber dükkanında saç kesilirken 5 müşteri olsa bile saç kesiminden hemen sonra sadece dört müşteri kalmaktadır.)

n . saç kesilişi esnasında gelen müşterilerin sayısını X_n olarak alalım. X_n 'in dağılımı yukarıda verildiği gibidir. $Q_n = 0$ olsun. Berber bir gelişten sonra işine başlar ve X_{n+1} müşteri gelirken saç keser. Saç kesiminin sonunda kalan müşteri sayısı $\min(4, X_{n+1})$ olacaktır. (Bu noktada ayrılan müşteri X_{n+1} içinde bulunmaz.)

Sonuç olarak,

$$Q_{n+1} = \begin{cases} \min(4, Q_n + X_{n+1} - 1) & \text{eğer } Q_n > 0 \text{ ise} \\ \min(4, X_{n+1}) & \text{eğer } Q_n = 0 \text{ ise} \end{cases}$$

Q_{n+1} sadece Q_n ve X_n 'e (onun dağılımı bilinir ve sistem haricidir) bağlıdır ve ($r < n$) olmak üzere herhangi bir Q_r 'ye bağlı değildir. Böylece $\{Q_n, n = 0,1,2, \dots\}$ olmak üzere onun geçiş ihtimali matrisi aşağıdaki gibi elde edilir.

Eğer $Q_n = 0$ ise o zaman $j = 0,1,2, \dots$ için $P(Q_{n+1} = j) = P(X_{n+1} = j)$ olasılığı

$$P_{00} = 0.20, P_{01} = 0.70, P_{02} = 0.07, P_{03} = 0.02, P_{04} = 0.01$$

değerleri elde edilir.

Eğer $Q_n > 0$ ise

$i = 1$ olduğunda $P(Q_{n+1} = j) = P[\min(4, X_{n+1}) = j] = P(X_{n+1} = j)$ olasılığı

$$P_{10} = 0.20, P_{11} = 0.70, P_{12} = 0.07, P_{13} = 0.02, P_{14} = 0.01$$

değerlerini verir,

$i = 2$ olduğunda $P(Q_{n+1} = j) = P[\min(4, X_{n+1} + 1) = j]$ olasılığı

$$P_{20} = 0, P_{21} = 0.20, P_{22} = 0.70, P_{23} = 0.07, P_{24} = 0.02 + 0.01$$

değerlerini verir,

$i = 3$ olduğunda $P(Q_{n+1} = j) = P[\min(4, X_{n+1} + 2) = j]$ olasılığı

$$P_{30} = 0, P_{31} = 0, P_{32} = 0.20, P_{33} = 0.70, P_{34} = 0.07 + 0.02 + 0.01$$

değerlerini verir,

$i = 4$ olduğunda $P(Q_{n+1} = j) = P[\min(4, X_{n+1} + 3) = j]$ olasılığı

$$P_{40} = 0, P_{41} = 0, P_{42} = 0, P_{43} = 0.20, P_{44} = 0.70 + 0.07 + 0.02 + 0.01$$

değerlerini verir.

Böylece geçiş ihtimali matrisi

$$P = \begin{matrix} & \begin{matrix} 0 & 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} 0.2 & 0.7 & 0.07 & 0.02 & 0.01 \\ 0.2 & 0.7 & 0.07 & 0.02 & 0.01 \\ 0 & 0.2 & 0.70 & 0.07 & 0.03 \\ 0 & 0 & 0.20 & 0.70 & 0.10 \\ 0 & 0 & 0 & 0.20 & 0.80 \end{bmatrix} \end{matrix}$$

olarak verilir.

Ek C: Faz dağılımlı kuyruk modellerinde trafik yoğunluğu aşağıdaki şekilde hesaplanır.

Hyperexponential ($M/H_2/1$) Kuyruk Modeli

$$\rho = \frac{\lambda}{\left(\frac{\alpha_1}{\mu_1} + \frac{\alpha_2}{\mu_2}\right)}$$

Mixed Erlang ($M/mixE_k/1$) Kuyruk Modeli

$$\rho = \frac{\lambda}{\left(\frac{\alpha}{\mu} + \frac{1-\alpha}{\mu}\right)} = \frac{\lambda}{\mu}$$

Hypoexponential ($M/HYPO_k/1$) Kuyruk Modeli

$$\rho = \frac{\lambda}{\left(\frac{1}{\mu_1} + \frac{1}{\mu_2}\right)}$$

Coxian ($M/C_k/1$) Kuyruk Modeli

$$\rho = \frac{\lambda}{\left(\frac{1}{\mu_1} + \frac{\alpha_1}{\mu_2} + \frac{\alpha_1\alpha_2}{\mu_3} + \dots + \frac{\alpha_1\alpha_2\dots\alpha_{k-1}}{\mu_k}\right)}$$

Ek D: $M/E_k/1$ kuyruk modelinin hipotez için karar veren simülasyon programı Matlab programında yapılmıştır. Dört program dosyasından oluşmakta ve function şekline aşağıdaki ilk programdan çağrılmaktadır (Uzunoğlu ve Onar, 2003, Yüksel, 2004).

1. Aşağıdaki kodlar bilgisayar ortamında dizi.m olarak kaydedilmiştir.

```
% ayrılış anlarında sistemdeki kalan müşteri sayısı
[x,lambda,mu]=erlang(9,0.8,2);
%  $n_{ij}$  geçiş sayısı
boyut=9;
uz=length(x);
v=num2str(x);
    maxx=max(x)
    minn=min(x)
    for k=1:boyut
        for l=1:boyut
            P(k,l)=0
        end
    end
    for k=1:boyut
        for l=1:boyut
            W(k,l)=0
        end
    end
m=1
tk='';
for h=1:uz-1
    for i=h:h
        tk='';
        for j=i:i+1
            tk=strcat(tk,num2str(x(j)))
            a=findstr(v,tk)
            b=length(a)
            end
        [y]=tk
        w=str2num(y(1))+1
        c=str2num(y(2))+1
        P(w,c)=b
        j=0;
    end
    say(h)=str2num(tk)
```

```

end
for h=1:uz-1
    for i=h:h
        tk='';
        b=0
        n=1
        for j=i:i+1
            if n==1
                x1=x(j)
            else
                y1=x(j)
            end
            tk=strcat(tk,num2str(x(j)))
            n=n+1
        end
        for sy=1:uz-1
            if say(sy)==str2num(tk)
                b=b+1
            end
            [y]=tk
            w=x1+1
            c=y1+1
            P(w,c)=b
            j=0;
        end
    end
end
tb=0
for i=1:boyut
    for j=1:boyut
        tb=tb+P(i,j)
    end
end
% hipotez testi
ro1=0.9;
ro0=lambda;
k=1;
for i=2:boyut
    for j=1:boyut-1
        zz=((j-1)-(i-1)+1)
    end
end

```

```

C=((ro1*(ro0+k))/(ro0*(ro1+k)))^zz
if j>=i-1
W(i,j)=log(C)
else
W(i,j)=0
end
end
end
for i=1:1
for j=1:boyut-1
C=((ro1*(ro0+k))/(ro0*(ro1+k)))^(j-1)
W(i,j)=log(C)
end
end
eks=3
for i=2:eks+1
eks1=eks-(i-1)
z1=((ro1+k)/k)^k
z0=((ro0+k)/k)^k
for j=eks+1:eks+1
top1=0;
top0=0;
for ic=0:eks1
carpus=1;
carpalt1=1;
carpalt2=1;
r=ic+k-1
if r==0
carpus=1
else
if k==1
carpus=1
else
for t=1:r
carpus=carpus*t
end
end
end
r=ic
if r==0

```

```

        carpalt1=1
    else
        if k==1
            carpus=1
        else
            for t=1:r
                carpalt1=carpalt1*t
            end
        end
    end
    r=k-1
    if r==0
        carpalt2=1
    else
        if k==1
            carpalt2=1
        else
            for t=1:r
                carpalt2=carpalt2*t
            end
        end
    end
    end
    zus1=(carpus/(carpalt1*carpalt2))
    zro1=zus1*((ro1/(ro1+k))^ic)
    zro0=zus1*((ro0/(ro0+k))^ic)
    top1=top1+zro1
    top0=top0+zro0
    end
    bir=z1-top1
    iki=z0-top0
    C=bir/iki
    W(i,j)=log(C)
end
end
W(1,boyut)=W(2,boyut);
Toplam=0
for i=1:boyut
    for j=1:boyut
        Toplam=Toplam+ W(i,j)*P(i,j)
    end
end

```

```

end
% karar
alfa=0.05;
beta=0.10;
A=(1-beta)/alfa;
B=beta/(1-alfa);
lg=log((ro0+k)/(ro1+k));
kg=2*lg;
an=tb*k*lg;
alt=log(B)-an;
ust=log(A)-an;
if Toplam<alt
    Karar='kabul'
end
if Toplam>ust
    Karar='ret'
end
if Toplam>=alt && Toplam<=ust
    Karar='devam'
    %y=input('Gözlem Değeri Giriniz:');
    yeni_erlang(x,lambda,mu);
end

```

2. erlang.m fonksiyonu

```

function [X,lambda,mu] =erlang(n,lam,mu1)
global nextarrival nextdeparture current say_ayrilis X
lambda=lam;
mu=mu1;
nextarrival= (-1/lambda)*log(rand);
now= 0;
nextdeparture= inf;
current= 0;
say_ayrilis=0;
for i=1:20
if nextarrival < nextdeparture,
    now= nextarrival;
    if current<n
        current= current + 1;
        disp(sprintf('Arrival at : %f (current length %d)', now,
current));
    end
end
end

```

```

end
nextarrival= now + (-1/lambda)*log(rand);
if current == 1,
nextdeparture= now + (-1/mu)*log(rand);
end
else
now= nextdeparture;
current= current - 1;
say_ayrilis=say_ayrilis+1;
% ayrılış anlarında sistemdeki kalan müşteri sayısı
X(say_ayrilis)=current;
disp(sprintf('Departure at : %f (current length %d)', now,
current));
if current > 0,
nextdeparture= now + (-1/mu)*log(rand);
else
nextdeparture= inf;
end
end
end
y=input('Gözlem Değeri Giriniz:');
end

```

3. dizi_devam.m fonksiyonu

```

function [P] = dizi_devam(x,lambda,mu)
% hipotez sonucu devam ise ayrılış anlarında sistemdeki kalan
müşteri sayısı
boyut=9;
uz=length(x);
v=num2str(x);
maxx=max(x)
minn=min(x)
for k=1:boyut
for l=1:boyut
P(k,l)=0
end
end
for k=1:boyut
for l=1:boyut
W(k,l)=0
end
end

```

```

end
m=1
tk='';
for h=1:uz-1
    for i=h:h
        tk='';
        for j=i:i+1
            tk=strcat(tk,num2str(x(j)))
            a=findstr(v,tk)
            b=length(a)
        end
        [y]=tk
        w=str2num(y(1))+1
        c=str2num(y(2))+1
        P(w,c)=b
        j=0;
    end
    say(h)=str2num(tk)
end
for h=1:uz-1
    for i=h:h
        tk='';
        b=0
        n=1
        for j=i:i+1
            if n==1
                x1=x(j)
            else
                y1=x(j)
            end
            tk=strcat(tk,num2str(x(j)))
            n=n+1
        end
        for sy=1:uz-1
            if say(sy)==str2num(tk)
                b=b+1
            end
        end
        [y]=tk
        w=x1+1
        c=y1+1
    end
end

```

```

        P(w,c)=b
    j=0;
    end
end
end
tb=0
for i=1:boyut
    for j=1:boyut
        tb=tb+P(i,j)
    end
end
ro1=0.9;
ro0=lambda;
k=1;
for i=2:boyut
    for j=1:boyut-1
        zz=((j-1)-(i-1)+1)
        C=((ro1*(ro0+k))/(ro0*(ro1+k)))^zz
        if j>=i-1
            W(i,j)=log(C)
        else
            W(i,j)=0
        end
    end
end
end
for i=1:1
    for j=1:boyut-1
        C=((ro1*(ro0+k))/(ro0*(ro1+k)))^(j-1)
        W(i,j)=log(C)
    end
end
eks=boyut-1
for i=2:eks+1
    eks1=eks-(i-1)
        z1=((ro1+k)/k)^k
        z0=((ro0+k)/k)^k
    for j=boyut:boyut
        top1=0;
        top0=0;
        for ic=0:eks1

```

```

carpus=1;
carpalt1=1;
carpalt2=1;
    r=ic+k-1
    if r==0
        carpus=1
    else
        if k==1
            carpus=1
        else
            for t=1:r
                carpus=carpus*t
            end
        end
    end
end
r=ic
    if r==0
        carpalt1=1
    else
        if k==1
            carpus=1
        else
            for t=1:r
                carpalt1=carpalt1*t
            end
        end
    end
end
r=k-1
    if r==0
        carpalt2=1
    else
        if k==1
            carpalt2=1
        else
            for t=1:r
                carpalt2=carpalt2*t
            end
        end
    end
end
bir=z1-top1

```

```

        iki=z0-top0
        C=bir/iki
        if i==2
            W(i,j)=log(C)
        end
        W(i,j)=log(C)
    end
end
W(1,boyut)=W(2,boyut);
Toplam=0
for i=1:boyut
    for j=1:boyut
        Toplam=Toplam+ W(i,j)*P(i,j)
    end
end
disp(sprintf('Departure at : %f (current length %d)', Toplam));
alfa=0.05
beta=0.10
A=(1-beta)/alfa
B=beta/(1-alfa)
lg=log((ro0+k)/(ro1+k))
kg=2*lg
an=tb*k*lg
alt=log(B)-an
ust=log(A)-an
if Toplam<alt
    Karar='kabul'
end
if Toplam>ust
    Karar='ret'
end
if Toplam>alt && Toplam<ust
    Karar='devam'
    %y=input('Gözlem Değeri Giriniz:');
    yeni_erlang(n,lambda,mu);
end
end

```

4. yeni_erlang.m fonksiyon dosyası

```
function [X,lambda,mu] = yeni_erlang(n,lambda,mu)
global nextarrival nextdeparture current say_ayrilis X
for i=1:20
if nextarrival < nextdeparture,
    now= nextarrival;
    if current<n
        current= current + 1;
        disp(sprintf('Arrival at : %f (current length %d)', now,
current));
    end
    nextarrival= now + (-1/lambda)*log(rand);
    if current == 1,
        nextdeparture= now + (-1/mu)*log(rand);
    end
else
    now= nextdeparture;
    current= current - 1;
    say_ayrilis=say_ayrilis+1;
    % hipotez sonucu devam ise ayrılış anlarında sistemdeki kalan
    müşteri sayısı
    X(say_ayrilis)=current;
    disp(sprintf('Departure at : %f (current length %d)', now,
current));
    if current > 0,
        nextdeparture= now + (-1/mu)*log(rand);
    else
        nextdeparture= inf;
    end
end
end
end
disp('ikinci kısım');
y=input('Gözlem Değeri Giriniz:');
dizi_devam(X,lambda,mu);
end
```

Ek E: Hyperexponential ($M/H_2/1$) kuyruk modelinin hipotez için karar veren similasyon programı Matlab programında yapılmıştır. Dört program dosyasından oluşmakta ve function şekline aşağıdaki ilk programdan çağrılmaktadır.

1. Aşağıdaki kodlar bilgisayar ortamında dizi.m olarak kaydedilmiştir.

```
[x,lambda1,lambda2,mu1,mu2]=erlang(4,0.5,0.8,0.6,1.8);
boyut=4;
uz=length(x);
v=num2str(x);
    maxx=max(x)
    minn=min(x)
    for k=1:boyut
        for l=1:boyut
            P(k,l)=0;
        end
    end
    for k=1:boyut
        for l=1:boyut
            W(k,l)=0;
        end
    end
m=1;
tk='';
for h=1:uz-1
    for i=h:h
        tk='';
            for j=i:i+1
                tk=strcat(tk,num2str(x(j)));
                a=findstr(v,tk);
                b=length(a);
            end
            [y]=tk;
                w=str2num(y(1))+1;
                c=str2num(y(2))+1;
                P(w,c)=b
            j=0;
        end
    say(h)=str2num(tk);
    end
    for h=1:uz-1
        for i=h:h
```

```

tk='';
b=0;
n=1;
    for j=i:i+1
        if n==1
            x1=x(j)
        else
            y1=x(j)
        end
        tk=strcat(tk,num2str(x(j)));
        n=n+1;
    end
    for sy=1:uz-1
        if say(sy)==str2num(tk)
            b=b+1;
        end
        [y]=tk;
        w=x1+1;
        c=y1+1;
        P(w,c)=b
        j=0;
    end
end
end
tb=0
for i=1:boyut
    for j=1:boyut
        tb=tb+P(i,j);
    end
end
ro01=lambda1/mu1;
ro02=lambda1/mu2;
ro11=lambda2/mu1;
ro12=lambda2/mu2;
alfa1=0.4;
alfa2=0.6;
ro=lambda1*((alfa1/mu1)+(alfa2/mu2))
r1=lambda2*((alfa1/mu1)+(alfa2/mu2))
k=2;
    for i=2:boyut

```

```

    for j=1:boyut-1
        zz=((j-1)-(i-1)+1)

        pay1=(alfa1*((ro11/(ro11+1))^zz)*(1/(ro11+1)))+(alfa2*((ro12/(ro12+1))^zz)*(1/(ro12+1)))

        pay2=(alfa1*((ro01/(ro01+1))^zz)*(1/(ro01+1)))+(alfa2*((ro02/(ro02+1))^zz)*(1/(ro02+1)))

        C=pay1/pay2
        if j>=i-1
            W(i,j)=log(C)
        else
            W(i,j)=0
        end
    end
end
for i=1:1
    for j=1:boyut-1
        pay1=(alfa1*((ro11/(ro11+1))^(j-1))*(1/(ro11+1)))+(alfa2*((ro12/(ro12+1))^(j-1))*(1/(ro12+1)))
        pay2=(alfa1*((ro01/(ro01+1))^(j-1))*(1/(ro01+1)))+(alfa2*((ro02/(ro02+1))^(j-1))*(1/(ro02+1)))
        C=pay1/pay2
        W(i,j)=log(C)
    end
end
eks=boyut-1
for i=2:eks+1
    eks1=eks-(i-1)
    for j=boyut:boyut
        top1=0;
        top0=0;
        for ic=0:eks1

            zus1=(alfa1*((ro11/(ro11+1))^ic)*(1/(ro11+1)))+(alfa2*((ro12/(ro12+1))^ic)*(1/(ro12+1)))

            zalt1=(alfa1*((ro01/(ro01+1))^ic)*(1/(ro01+1)))+(alfa2*((ro02/(ro02+1))^ic)*(1/(ro02+1)))

            top1=top1+zus1
            top0=top0+zalt1
        end
        bir=1-top1
        iki=1-top0
    end
end

```

```

        C=bir/iki
        %l=i-1
        W(i,j)=log(C)
    end
end
W(1,boyut)=W(2,boyut);
Toplam=0
for i=1:boyut
    for j=1:boyut
        Toplam=Toplam+ W(i,j)*P(i,j)
    end
end

alfa=0.05
beta=0.1
A=(1-beta)/alfa
B=beta/(1-alfa)
alt=log(B)
ust=log(A)
if Toplam<alt
    Karar='kabul'
end
if Toplam>ust
    Karar='ret'
end
if Toplam>alt && Toplam<ust
    Karar='devam'
    'y=input('Gözlem Değeri Giriniz:');
    yeni_erlang(x,lambda1,lambda2,mu1,mu2);
end

```

2. erlang.m fonksiyon kodları

```

function [X,lambda1,lambda2,mu1,mu2] =Erlang(n,lam1,lam2,m1,m2)
global nextarrival nextdeparture current say_ayrilis X
lambda1=lam1;
lambda2=lam2;
mu1=m1;
mu2=m2;
nextarrival= (-1/lambda1)*log(rand);
now= 0;

```

```

nextdeparture= inf;           % infinity
current= 0;
alfa1=0.40
alfa2=1-alfa1
alfa1_yuzde=100*alfa1
alfa2_yuzde=100*alfa2
say_ayrilis=0;
for i=1:100
    y=ceil(100*rand);
    if y <= alfa1_yuzde
        sec=1;
    else
        sec=2;
    end
    if sec==1,
    if nextarrival < nextdeparture,
        now= nextarrival;
        if current<4
            current= current + 1;
            disp(sprintf('Arrival at : %f (current length %d)', now,
current));
            end
            nextarrival= now + (-1/lambda1)*log(rand);
            if current == 1,
                nextdeparture= now + (-1/mu1)*log(rand);
            end
        else
            now= nextdeparture;
            current= current - 1;
            say_ayrilis=say_ayrilis+1;
            X(say_ayrilis)=current;
            disp(sprintf('Departure at : %f (current length %d)', now,
current));
            if current > 0,
                nextdeparture= now + (-1/mu1)*log(rand);
            else
                nextdeparture= inf;
            end
        end
    end
    if sec==2,

```

```

if nextarrival < nextdeparture,
    now= nextarrival;
    if current<4
        current= current + 1;
        disp(sprintf('Arrival at : %f (current length %d)', now,
current));
    end
    nextarrival= now + (-1/lambda1)*log(rand);
    if current == 1,
        nextdeparture= now + (-1/mu2)*log(rand);
    end
else
    now= nextdeparture;
    current= current - 1;
    say_ayrilis=say_ayrilis+1;
    X(say_ayrilis)=current;
    disp(sprintf('Departure at : %f (current length %d)', now,
current));
    if current > 0,
        nextdeparture= now + (-1/mu2)*log(rand);
    else
        nextdeparture= inf;
    end
end
end
end
end
end
end

```

3. dizi_devam.m fonksiyon kodları

```

function [P] = dizi_devam(x,lambda1,lambda2,mu1,mu2)
boyut=4;
uz=length(x);
v=num2str(x);
    maxx=max(x)
    minn=min(x)
    for k=1:boyut
        for l=1:boyut
            P(k,l)=0
        end
    end
end
for k=1:boyut

```

```

        for l=1:boyut
            W(k,l)=0
        end
    end
end
m=1
tk='';
for h=1:uz-1
    for i=h:h
        tk='';
        for j=i:i+1
            tk=strcat(tk,num2str(x(j)))
            a=findstr(v,tk)
            b=length(a)
        end
        [y]=tk
        w=str2num(y(1))+1
        c=str2num(y(2))+1
        P(w,c)=b
        j=0;
    end
    say(h)=str2num(tk)
end
for h=1:uz-1
    for i=h:h
        tk='';
        b=0
        n=1
        for j=i:i+1
            if n==1
                x1=x(j)
            else
                y1=x(j)
            end
            tk=strcat(tk,num2str(x(j)))
            n=n+1
        end
        for sy=1:uz-1
            if say(sy)==str2num(tk)
                b=b+1
            end
        end
    end
end

```

```

        [y]=tk
        w=x1+1
        c=y1+1
        P(w,c)=b
        j=0;
        end
    end
end
tb=0
for i=1:boyut
    for j=1:boyut
        tb=tb+P(i,j)
    end
end
ro01=lambda1/mu1;
ro02=lambda1/mu2;
ro11=lambda2/mu1;
ro12=lambda2/mu2;
alfa1=0.4;
alfa2=0.6;
k=2;
for i=2:boyut
    for j=1:boyut-1
        zz=((j-1)-(i-1)+1)

pay1=(alfa1*((ro11/(ro11+1))^zz)*(1/(ro11+1)))+(alfa2*((ro12/(ro12+1))^zz)*(1/(ro12+1)))

pay2=(alfa1*((ro01/(ro01+1))^zz)*(1/(ro01+1)))+(alfa2*((ro02/(ro02+1))^zz)*(1/(ro02+1)))

        C=pay1/pay2
        W(i,j)=log(C)
    end
end
for i=1:1
    for j=1:boyut-1
        pay1=(alfa1*((ro11/(ro11+1))^(j-1))*(1/(ro11+1)))+(alfa2*((ro12/(ro12+1))^(j-1))*(1/(ro12+1)))
        pay2=(alfa1*((ro01/(ro01+1))^(j-1))*(1/(ro01+1)))+(alfa2*((ro02/(ro02+1))^(j-1))*(1/(ro02+1)))
        C=pay1/pay2
        W(i,j)=log(C)
    end
end

```

```

        end
    end
    eks=boyut-1
    for i=2:eks+1
        eks1=eks-(i-1)
        for j=boyut:boyut
            top1=0;
            top0=0;
            for ic=0:eks1

                zus1=(alfa1*((ro11/(ro11+1))^ic)*(1/(ro11+1)))+(alfa2*((ro12/(ro12+1))^ic)*(1/(ro12+1)))

                zalt1=(alfa1*((ro01/(ro01+1))^ic)*(1/(ro01+1)))+(alfa2*((ro02/(ro02+1))^ic)*(1/(ro02+1)))

                top1=top1+zus1
                top0=top0+zalt1
            end
            bir=1-top1
            iki=1-top0
            C=bir/iki
            W(i,j)=log(C)
        end
    end
    W(1,boyut)=W(2,boyut);
    Toplam=0
    for i=1:boyut
        for j=1:boyut
            Toplam=Toplam+ W(i,j)*P(i,j)
        end
    end
    alfa=0.05
    beta=0.1
    A=(1-beta)/alfa
    B=beta/(1-alfa)
    alt=log(B)
    ust=log(A)
    if Toplam<alt
        Karar='kabul'
    end
    if Toplam>ust
        Karar='ret'
    end
end

```

```

end
if Toplam>alt && Toplam<ust
    Karar='devam'
    y=input('Gözlem Değeri Giriniz:');
    yeni_erlang(n,lambda1,lambda2,mu1,mu2);
end
end

```

4. yeni_erlang.m fonksiyonu kodlar

```

function [X,lambda1,lambda2,mu1,mu2] =
yeni_erlang(n,lambda1,lambda2,mu1,mu2)
global nextarrival nextdeparture current say_ayrilis X
alfa1=0.4
alfa2=1-alfa1
alfa1_yuzde=100*alfa1
alfa2_yuzde=100*alfa2
for i=1:100
    y=ceil(100*rand);
    if y <= alfa1_yuzde
        sec=1;
    else
        sec=2;
    end
    if sec==1,
    if nextarrival < nextdeparture,
        now= nextarrival;
        if current<4
            current= current + 1;
            disp(sprintf('Arrival at : %f (current length %d)', now,
current));
        end
        nextarrival= now + (-1/lambda1)*log(rand);
        if current == 1,
            nextdeparture= now + (-1/mu1)*log(rand);
        end
    else
        now= nextdeparture;
        current= current - 1;
        say_ayrilis=say_ayrilis+1;
        X(say_ayrilis)=current;
    end
end

```

```

    disp(sprintf('Departure at : %f (current length %d)', now,
current));
    if current > 0,
        nextdeparture= now + (-1/mu1)*log(rand);
    else
        nextdeparture= inf;
    end
end
end
if sec==2,
if nextarrival < nextdeparture,
    now= nextarrival;
    if current<4
        current= current + 1;
    end
    disp(sprintf('Arrival at : %f (current length %d)', now,
current));
    nextarrival= now + (-1/lambda1)*log(rand);
    if current == 1,
        nextdeparture= now + (-1/mu2)*log(rand);
    end
else
    now= nextdeparture;
    current= current - 1;
    say_ayrilis=say_ayrilis+1;
    X(say_ayrilis)=current;
    disp(sprintf('Departure at : %f (current length %d)', now,
current));
    if current > 0,
        nextdeparture= now + (-1/mu2)*log(rand);
    else
        nextdeparture= inf;
    end
end
end
end
disp('ikinci kısım');
dizi_devam(X,lambda1,lambda2,mu1,mu2);
end

```

Ek F: Mixed Erlang ($M/mixE_k/1$) kuyruk modelinin hipotez için karar veren simülasyon programı Matlab programında yapılmıştır. Dört program dosyasından oluşmakta ve function şekline aşağıdaki ilk programdan çağrılmaktadır.

1. Aşağıdaki kodlar bilgisayar ortamında dizi.m olarak kaydedilmiştir.

```
[x,lambda,mu]=erlang(10,1,0.5);
boyut=10;
uz=length(x);
v=num2str(x);
Toplam=0;
    maxx=max(x)
    minn=min(x)
    for k=1:boyut
        for l=1:boyut
            P(k,l)=0
        end
    end
    for k=1:boyut
        for l=1:boyut
            W(k,l)=0
        end
    end
m=1
tk='';
for h=1:uz-1
    for i=h:h
        tk='';
        for j=i:i+1
            tk=strcat(tk,num2str(x(j)))
            a=findstr(v,tk)
            b=length(a)
        end
        [y]=tk
        w=str2num(y(1))+1
        c=str2num(y(2))+1
        P(w,c)=b
        j=0;
    end
    say(h)=str2num(tk)
end
for h=1:uz-1
```

```

for i=h:h
    tk='';
    b=0
    n=1
    for j=i:i+1
        if n==1
            x1=x(j)
        else
            y1=x(j)
        end
        tk=strcat(tk,num2str(x(j)))
        n=n+1
    end
    for sy=1:uz-1
        if say(sy)==str2num(tk)
            b=b+1
        end
        [y]=tk
        w=x1+1
        c=y1+1
        P(w,c)=b
    end
    j=0;
end
end
tb=0
for i=1:boyut
    for j=1:boyut
        tb=tb+P(i,j)
    end
end
ro1=0.9;
ro0=lambda/mu;
alfa1=0.5;
alfa2=0.5;
k=2;
for i=2:boyut
    for j=1:boyut-1
        zz=((j-1)-(i-1)+1)
        C=((ro1*(ro0+k))/(ro0*(ro1+k)))^zz
    end
end

```

```

        B=(alfa1+(alfa2*((j-1)-(i-1)+k)/(k-
1))*(k/(ro1+k)))/(alfa1+(alfa2*((j-1)-(i-1)+k)/(k-
1))*(k/(ro0+k)))
        if j>=i-1
            W(i,j)=log(C)+log(B)
        else
            W(i,j)=0
        end
    end
end
for i=1:1
    for j=1:boyut-1
        C=((ro1*(ro0+k))/(ro0*(ro1+k))^(j-1)
        B=(alfa1+(alfa2*((j-1)+k-1)/(k-
1))*(k/(ro1+k)))/(alfa1+(alfa2*((j-1)+k-1)/(k-1))*(k/(ro0+k)))
        W(i,j)=log(C)+log(B)
    end
end
eks=boyut-1
for i=2:eks+1
    eks1=eks-(i-1)
    for j=boyut:boyut
        top1=0;
        top0=0;
        for ic=0:eks1
            carpus=1;
            carpalt1=1;
            carpalt2=1;
            r=ic+k-2
            if r==0
                carpus=1
            else
                if k==2
                    carpus=1;
                else
                    for t=1:r
                        carpus=carpus*t
                    end
                end
            end
            r=ic;

```

```

        if r==0
            carpalt1=1
        else
            if k==2
                carpalt1=1;
            else
                for t=1:r
                    carpalt1=carpalt1*t
                end
            end
        end
    end

    r=k-2
    if r==0
        carpalt2=1
    else
        if k==2
            carpalt2=1;
        else
            for t=1:r
                carpalt2=carpalt2*t
            end
        end
    end

    zus1=(carpus/(carpalt1*carpalt2));
    zro1=((ro1/(ro1+k))^ic)*(k/(ro1+k))^(k-1)*zus1*(alfa1+alfa2*(ic+k-1)/(k-1)*(k/(ro1+k)))
    zro0=((ro0/(ro0+k))^ic)*(k/(ro0+k))^(k-1)*zus1*(alfa1+alfa2*(ic+k-1)/(k-1)*(k/(ro0+k)))
    top1=top1+zro1
    top0=top0+zro0
end
bir=1-top1
iki=1-top0
C=bir/iki
W(i,j)=log(C)
end

end

W(1,boyut)=W(2,boyut);
for i=1:boyut
    for j=1:boyut
        Toplam=Toplam+ W(i,j)*P(i,j)
    end
end

```

```

        end
    end
    alfa=0.1
    beta=0.01
    A=(1-beta)/alfa
    B=beta/(1-alfa)
    lg=log((ro0+k)/(ro1+k))
    kg=2*lg
    an=tb*(k-1)*lg
    alt=log(B)-an
    ust=log(A)-an
    if Toplam<alt
        Karar='kabul'
    end
    if Toplam>ust
        Karar='ret'
    end
    if Toplam>alt && Toplam<ust
        Karar='devam'
        %y=input('Gözlem Değeri Giriniz:');
        yeni_erlang(x,lambda,mu);
    end
end

```

2. erlang.m fonksiyon kodları

```

function [X,lambda,mu] =erlang(n,lam,mu1)
global nextarrival nextdeparture current say_ayrilis X
lambda=lam;
mu=mu1;
nextarrival= (-1/lambda)*log(rand);
now= 0;
nextdeparture= inf;           % infinity
current= 0;
say_ayrilis=0;
alfa1=0.50
alfa2=1-alfa1
alfa1_yuzde=100*alfa1
alfa2_yuzde=100*alfa2
for i=1:50
    y=ceil(100*rand);
    if y <= alfa1_yuzde

```

```

        sec=1;
else
    sec=2;
end
if sec==1,
if nextarrival < nextdeparture,
    now= nextarrival;
    if current<n
        current= current + 1;
        disp(sprintf('Arrival at : %f (current length %d)', now,
current));
    end
    nextarrival= now + (-1/lambda)*log(rand);
    if current == 1,
        nextdeparture= now + (-1/mu)*log(rand);
    end
else
    now= nextdeparture;
    current= current - 1;
    say_ayrilis=say_ayrilis+1;
    X(say_ayrilis)=current;
    disp(sprintf('Departure at : %f (current length %d)', now,
current));
    if current > 0,
        nextdeparture= now + (-1/mu)*log(rand);
    else
        nextdeparture= inf;
    end
end
end
if sec==2,
if nextarrival < nextdeparture,
    now= nextarrival;
    if current<n
        current= current + 1;
        disp(sprintf('Arrival at : %f (current length %d)', now,
current));
    end
    nextarrival= now + (-1/lambda)*log(rand);
    if current == 1,
        nextdeparture= now + (-1/mu)*log(rand);

```

```

    end
else
    now= nextdeparture;
    current= current - 1;
    say_ayrilis=say_ayrilis+1;
    X(say_ayrilis)=current;
    disp(sprintf('Departure at : %f (current length %d)', now,
current));
    if current > 0,
        nextdeparture= now + (-1/mu)*log(rand);
    else
        nextdeparture= inf;
    end
end
end
end
end
%X=yeni_erlang(n,lambda,mu)
end

```

3. dizi_devam.m fonksiyon kodları

```

function [P] = dizi_devam(x,lambda,mu)
boyut=4;
uz=length(x);
v=num2str(x);
Toplam=0;
    maxx=max(x)
    minn=min(x)
    for k=1:boyut
        for l=1:boyut
            P(k,l)=0
        end
    end
end
    for k=1:boyut
        for l=1:boyut
            W(k,l)=0
        end
    end
end
m=1
tk='';
for h=1:uz-1

```

```

for i=h:h
    tk='';
        for j=i:i+1
            tk=strcat(tk,num2str(x(j)))
            a=findstr(v,tk)
            b=length(a)
        end
    [y]=tk
        w=str2num(y(1))+1
        c=str2num(y(2))+1
        P(w,c)=b

    j=0;
end
say(h)=str2num(tk)
end
    for h=1:uz-1
for i=h:h
    tk='';
    b=0
    n=1
        for j=i:i+1
            if n==1
                x1=x(j)
            else
                y1=x(j)
            end
            tk=strcat(tk,num2str(x(j)))
            n=n+1
        end
        for sy=1:uz-1
            if say(sy)==str2num(tk)
                b=b+1
            end
        [y]=tk
        w=x1+1
        c=y1+1
        P(w,c)=b
    j=0;
end

```

```

end
end
    tb=0
    for i=1:boyut
        for j=1:boyut
            tb=tb+P(i,j)
        end
    end
end
ro1=0.9;
ro0=lambda/mu;
alfa1=0.5;
alfa2=0.5;
k=2;
for i=2:boyut
    for j=1:boyut-1
        zz=((j-1)-(i-1)+1)
        C=((ro1*(ro0+k))/(ro0*(ro1+k)))^zz
        B=(alfa1+(alfa2*((j-1)-(i-1)+k)/(k-1))*(k/(ro1+k)))/(alfa1+(alfa2*((j-1)-(i-1)+k)/(k-1))*(k/(ro0+k)))
        if j>=i-1
            W(i,j)=log(C)+log(B)
        else
            W(i,j)=0
        end
    end
end
end
for i=1:1
    for j=1:boyut-1
        C=((ro1*(ro0+k))/(ro0*(ro1+k)))^(j-1)
        B=(alfa1+(alfa2*((j-1)+k-1)/(k-1))*(k/(ro1+k)))/(alfa1+(alfa2*((j-1)+k-1)/(k-1))*(k/(ro0+k)))
        W(i,j)=log(C)+log(B)
    end
end
eks=boyut-1
for i=2:eks+1
    eks1=eks-(i-1)
    for j=boyut:boyut
        top1=0;
        top0=0;

```

```

for ic=0:eks1
    carpus=1;
carpalt1=1;
carpalt2=1;
    r=ic+k-2
    if r==0
        carpus=1
    else
        if k==2
            carpus=1;
        else
            for t=1:r
                carpus=carpus*t
            end
        end
    end
    r=ic;
    if r==0
        carpalt1=1
    else
        if k==2
            carpalt1=1;
        else
            for t=1:r
                carpalt1=carpalt1*t
            end
        end
    end
    r=k-2
    if r==0
        carpalt2=1
    else
        if k==2
            carpalt2=1;
        else
            for t=1:r
                carpalt2=carpalt2*t
            end
        end
    end
end

```

```

        zus1=(carpus/(carpalt1*carpalt2));
        zro1=((ro1/(ro1+k))^ic)*(k/(ro1+k))^(k-
1)*zus1*(alfa1+alfa2*(ic+k-1)/(k-1)*(k/(ro1+k)))
        zro0=((ro0/(ro0+k))^ic)*(k/(ro0+k))^(k-
1)*zus1*(alfa1+alfa2*(ic+k-1)/(k-1)*(k/(ro0+k)))
        top1=top1+zro1
        top0=top0+zro0
    end
    bir=1-top1
    iki=1-top0
    C=bir/iki
    W(i,j)=log(C)
end
end
W(1,boyut)=W(2,boyut);
for i=1:boyut
    for j=1:boyut
        Toplam=Toplam+ W(i,j)*P(i,j)
    end
end
alfa=0.1
beta=0.01
A=(1-beta)/alfa
B=beta/(1-alfa)
lg=log((ro0+k)/(ro1+k))
kg=2*lg
an=tb*(k-1)*lg
alt=log(B)-an
ust=log(A)-an
if Toplam<alt
    Karar='kabul'
end
if Toplam>ust
    Karar='ret'
end
if Toplam>alt && Toplam<ust
    Karar='devam'
    %y=input('Gözlem Değeri Giriniz:');
    yeni_erlang(x,lambda,mu);
end
end

```

4. yeni_erlang.m fonksiyonu kodlar

```
function [X,lambda,mu] = yeni_erlang(n,lambda,mu)
global nextarrival nextdeparture current say_ayrilis X
alfa1=0.30
alfa2=1-alfa1
alfa1_yuzde=100*alfa1
alfa2_yuzde=100*alfa2
for i=1:10
    y=ceil(100*rand);
    if y <= alfa1_yuzde
        sec=1;
    else
        sec=2;
    end
    if sec==1,
    if nextarrival < nextdeparture,
        now= nextarrival;
        if current<n
            current= current + 1;
            disp(sprintf('Arrival at : %f (current length %d)', now,
current));
        end
        nextarrival= now + (-1/lambda)*log(rand);
        if current == 1,
            nextdeparture= now + (-1/mu)*log(rand);
        end
    else
        now= nextdeparture;
        current= current - 1;
        say_ayrilis=say_ayrilis+1;
        X(say_ayrilis)=current;
        disp(sprintf('Departure at : %f (current length %d)', now,
current));
        if current > 0,
            nextdeparture= now + (-1/mu)*log(rand);
        else
            nextdeparture= inf;
        end
    end
end
end
if sec==2,
```

```

if nextarrival < nextdeparture,
    now= nextarrival;
    if current<n
        current= current + 1;
        disp(sprintf('Arrival at : %f (current length %d)', now,
current));
    end
    nextarrival= now + (-1/lambda)*log(rand);
    if current == 1,
        nextdeparture= now + (-1/mu)*log(rand);
    end
else
    now= nextdeparture;
    current= current - 1;
    say_ayrilis=say_ayrilis+1;
    X(say_ayrilis)=current;
    disp(sprintf('Departure at : %f (current length %d)', now,
current));
    if current > 0,
        nextdeparture= now + (-1/mu)*log(rand);
    else
        nextdeparture= inf;
    end
end
end
end
end
dizi_devam(X,lambda,mu);
end

```

Ek G: Hypoexponential ($M/HYPO_k/1$) kuyruk modelinin hipotez için karar veren simülasyon programı Matlab programında yapılmıştır. Dört program dosyasından oluşmakta ve function şekline aşağıdaki ilk programdan çağrılmaktadır.

1. Aşağıdaki kodlar bilgisayar ortamında dizi.m olarak kaydedilmiştir.

```
[x,lambda1,lambda2,mu1,mu2]=erlang(7,0.5,0.2,0.7,0.3);
boyut=7;
uz=length(x);
v=num2str(x);
    maxx=max(x)
    minn=min(x)
    for k=1:boyut
        for l=1:boyut
            P(k,l)=0
        end
    end
    for k=1:boyut
        for l=1:boyut
            W(k,l)=0
        end
    end
m=1
tk='';
for h=1:uz-1
    for i=h:h
        tk='';
        for j=i:i+1
            tk=strcat(tk,num2str(x(j)))
            a=findstr(v,tk)
            b=length(a)
        end
        [y]=tk
        w=str2num(y(1))+1
        c=str2num(y(2))+1
        P(w,c)=b

        j=0;
    end
    say(h)=str2num(tk)
end
for h=1:uz-1
```

```

for i=h:h
    tk='';
    b=0
    n=1
    for j=i:i+1
        if n==1
            x1=x(j)
        else
            y1=x(j)
        end
        tk=strcat(tk,num2str(x(j)))
        n=n+1
    end
    for sy=1:uz-1
        if say(sy)==str2num(tk)
            b=b+1
        end
    end
    [y]=tk

    w=x1+1
    c=y1+1
    P(w,c)=b

    j=0;
end
end
end
tb=0
for i=1:boyut
    for j=1:boyut
        tb=tb+P(i,j)
    end
end
M11=mu1;
M21=mu2;
M10=mu1;
M20=mu2;
ro10=lambda1/M11;
ro20=lambda1/M21;
ro11=lambda2/M10;
ro21=lambda2/M20;
ro=lambda1/((1/M10)+(1/M20))

```

```

r1=lambda2/((1/M11)+(1/M20))
for i=2:boyut
    for j=1:boyut-1
        zz=((j-1)-(i-1)+1)
        C=(M11*((ro21/(ro21+1))^zz)*(1/(ro21+1)))-(
M21*((ro11/(ro11+1))^zz)*(1/(ro11+1)))
        B=(M10*((ro20/(ro20+1))^zz)*(1/(ro20+1)))-(
M20*((ro10/(ro10+1))^zz)*(1/(ro10+1)))
        if j>=i-1
            W(i,j)=log(C/B)
        else
            W(i,j)=0
        end
    end
end
for i=1:1
    for j=1:boyut-1
        C=(M11*((ro21/(ro21+1))^(j-1))*(1/(ro21+1)))-(
M21*((ro11/(ro11+1))^(j-1))*(1/(ro11+1)))
        B=(M10*((ro20/(ro20+1))^(j-1))*(1/(ro20+1)))-(
M20*((ro10/(ro10+1))^(j-1))*(1/(ro10+1)))
        W(i,j)=log(C/B)
    end
end
eks=boyut-1
for i=2:eks+1
    eks1=eks-(i-1)
    for j=boyut:boyut
        top1=0;
        top0=0;
        for ic=0:eks1
            zro1=(M11*((ro21/(ro21+1))^ic)*(1/(ro21+1)))-(
M21*((ro11/(ro11+1))^ic)*(1/(ro11+1)))
            zro0=(M10*((ro20/(ro20+1))^ic)*(1/(ro20+1)))-(
M20*((ro10/(ro10+1))^ic)*(1/(ro10+1)))
            top1=top1+zro1
            top0=top0+zro0
        end
        bir=1-top1
        iki=1-top0
        C=bir/iki
        W(i,j)=log(C)
    end
end

```

```

        end
    end
W(1,boyut)=W(2,boyut);
Toplam=0
for i=1:boyut
    for j=1:boyut
        Toplam=Toplam+ W(i,j)*P(i,j)
    end
end
alfa=0.05
beta=0.10
A=(1-beta)/alfa
B=beta/(1-alfa)
lg=log((M10-M20)/(M11-M21))
an=tb*lg
alt=log(B)
ust=log(A)
if Toplam<alt
    Karar='kabul'
end
if Toplam>ust
    Karar='ret'
end
if Toplam>alt && Toplam<ust
    Karar='devam'
    %y=input('Gözlem Değeri Giriniz:');
    yeni_erlang(x,lambda1,lambda2,mu1,mu2);
end

```

2. erlang. m fonksiyon kodları

```

function [X,lambda1,lambda2,mu1,mu2] =Erlang(n,lam1,lam2,m1,m2)
global nextarrival nextdeparture current say_ayrilis X
%[lambda_hesap1,sermu1]=tahmin1(lam1,mu1);
%[lambda_hesap2,sermu2]=tahmin2(lam2,mu2);
lambda1=lam1;
lambda2=lam2;
mu1=m1;
mu2=m2;
%lambda=lam;

```

```

%mu=mu1;
mu_ort=mu1+mu2
nextarrival= (-1/lambda1)*log(rand);
now= 0;
nextdeparture= inf;           % infinity
current= 0;
say_ayrilis=0;
for i=1:200
if nextarrival < nextdeparture,
    now= nextarrival;
    if current<n
        current= current + 1;
        disp(sprintf('Arrival at : %f (current length %d)', now,
current));
    end
    nextarrival= now + (-1/lambda1)*log(rand);
    if current == 1,
        nextdeparture= now + (-1/mu_ort)*log(rand);
    end
else
    now= nextdeparture;
    current= current - 1;
    say_ayrilis=say_ayrilis+1;
    X(say_ayrilis)=current;
    disp(sprintf('Departure at : %f (current length %d)', now,
current));
    if current > 0,
        nextdeparture= now + (-1/mu_ort)*log(rand);
    else
        nextdeparture= inf;
    end
end
end
end

```

3. dizi_devam.m fonksiyon kodları

```

function [P] = dizi_devam(x,lambda1,lambda2,mu1,mu2)
boyut=7;
uz=length(x);
v=num2str(x);
    maxx=max(x)
    minn=min(x)

```

```

for k=1:boyut
    for l=1:boyut
        P(k,l)=0
    end
end
end
for k=1:boyut
    for l=1:boyut
        W(k,l)=0
    end
end
end
m=1
tk='';
for h=1:uz-1
    for i=h:h
        tk='';
        for j=i:i+1
            tk=strcat(tk,num2str(x(j)))
            a=findstr(v,tk)
            b=length(a)
        end
        [y]=tk
        w=str2num(y(1))+1
        c=str2num(y(2))+1
        P(w,c)=b

        j=0;
    end
    say(h)=str2num(tk)
end
for h=1:uz-1
    for i=h:h
        tk='';
        b=0
        n=1

        for j=i:i+1
            if n==1
                x1=x(j)
            else
                y1=x(j)
            end
        end
    end
end

```

```

        end
        tk=strcat(tk,num2str(x(j)))
        n=n+1
    end
    for sy=1:uz-1
        if say(sy)==str2num(tk)
            b=b+1
        end
        [y]=tk
        w=x1+1
        c=y1+1
        P(w,c)=b
    end
    j=0;
end
end
end
    tb=0
    for i=1:boyut
        for j=1:boyut
            tb=tb+P(i,j)
        end
    end
    end
    M11=mu1;
    M21=mu2;
    M10=mu1;
    M20=mu2;
    ro10=lambda1/M11;
    ro20=lambda1/M21;
    ro11=lambda2/M10;
    ro21=lambda2/M20;
    ro=lambda1*((1/M10)+(1/M20))
    r1=lambda2*((1/M11)+(1/M20))
    for i=2:boyut
        for j=1:boyut-1
            zz=((j-1)-(i-1)+1)
            C=(M11*((ro21/(ro21+1))^zz)*(1/(ro21+1)))-
            (M21*((ro11/(ro11+1))^zz)*(1/(ro11+1)))
            B=(M10*((ro20/(ro20+1))^zz)*(1/(ro20+1)))-
            (M20*((ro10/(ro10+1))^zz)*(1/(ro10+1)))
            if j>=i-1
                W(i,j)=log(C/B)
            end
        end
    end
end

```

```

        else
            W(i,j)=0
        end
    end
end
end
for i=1:1
    for j=1:boyut-1
        C=(M11*((ro21/(ro21+1))^(j-1))*(1/(ro21+1)))-(
M21*((ro11/(ro11+1))^(j-1))*(1/(ro11+1)))
        B=(M10*((ro20/(ro20+1))^(j-1))*(1/(ro20+1)))-(
M20*((ro10/(ro10+1))^(j-1))*(1/(ro10+1)))
        W(i,j)=log(C/B)
    end
end
eks=boyut-1
for i=2:eks+1
    eks1=eks-(i-1)
    for j=boyut:boyut
        top1=0;
        top0=0;
        for ic=0:eks1
            zro1=(M11*((ro21/(ro21+1))^ic)*(1/(ro21+1)))-(
M21*((ro11/(ro11+1))^ic)*(1/(ro11+1)))
            zro0=(M10*((ro20/(ro20+1))^ic)*(1/(ro20+1)))-(
M20*((ro10/(ro10+1))^ic)*(1/(ro10+1)))
            top1=top1+zro1
            top0=top0+zro0
        end
        bir=1-top1
        iki=1-top0
        C=bir/iki
        W(i,j)=log(C)
    end
end
W(1,boyut)=W(2,boyut);
Toplam=0
for i=1:boyut
    for j=1:boyut
        Toplam=Toplam+ W(i,j)*P(i,j)
    end
end
alfa=0.05

```

```

beta=0.10
A=(1-beta)/alfa
B=beta/(1-alfa)
lg=log((M10-M20)/(M11-M21))
an=tb*lg
alt=log(B)
ust=log(A)
if Toplam<alt
    Karar='kabul'
end
if Toplam>ust
    Karar='ret'
end
if Toplam>alt && Toplam<ust
    Karar='devam'
    %y=input('Gözlem Değeri Giriniz:');
    yeni_erlang(n,lambda1,lambda2,mu1,mu2);
end
end

```

4. yeni_erlang.m fonksiyonu kodlar

```

function [X,lambda1,lambda2,mu1,mu2] =
yeni_erlang(n,lambda1,lambda2,mu1,mu2)
    global nextarrival nextdeparture current say_ayrilis X
    mu_ort=mu1+mu2;
    for i=1:50
        if nextarrival < nextdeparture,
            now= nextarrival;
            if current<n
                current= current + 1;
                disp(sprintf('Arrival at : %f (current length %d)', now,
current));
            end
            nextarrival= now + (-1/lambda1)*log(rand);
            if current == 1,
                nextdeparture= now + (-1/mu_ort)*log(rand);
            end
        else
            now= nextdeparture;
            current= current - 1;
            say_ayrilis=say_ayrilis+1;
        end
    end
end

```

```
X(say_ayrilis)=current;
disp(sprintf('Departure at : %f (current length %d)', now,
current));
if current > 0,
    nextdeparture= now + (-1/mu_ort)*log(rand);
else
    nextdeparture= inf;
end
end
```

Ek H: Coxian ($M/C_k/1$) kuyruk modelinin hipotez için karar veren simülasyon programı Matlab programında yapılmıştır. Dört program dosyasından oluşmakta ve function şekline aşağıdaki ilk programdan çağrılmaktadır.

1. Aşağıdaki kodlar bilgisayar ortamında dizi.m olarak kaydedilmiştir.

```
[x,lambda,mu1,mu2]=erlang(3,0.2,1,2);
boyut=3;
uz=length(x);
v=num2str(x);
    maxx=max(x)
    minn=min(x)
    for k=1:boyut
        for l=1:boyut
            P(k,l)=0;
        end
    end
    for k=1:boyut
        for l=1:boyut
            W(k,l)=0;
        end
    end
m=1;
tk='';
for h=1:uz-1
    for i=h:h
        tk='';
            for j=i:i+1
                tk=strcat(tk,num2str(x(j)));
                a=findstr(v,tk);
                b=length(a);
            end
            [y]=tk;
            w=str2num(y(1))+1;
            c=str2num(y(2))+1;
            P(w,c)=b
            j=0;
    end
    say(h)=str2num(tk);
    end
    for h=1:uz-1
        for i=h:h
```

```

tk='';
b=0;
n=1;
    for j=i:i+1
        if n==1
            x1=x(j)
        else
            y1=x(j)
        end
        tk=strcat(tk,num2str(x(j)));
        n=n+1;
    end
    for sy=1:uz-1
        if say(sy)==str2num(tk)
            b=b+1;
        end
        [y]=tk;
        w=x1+1;
        c=y1+1;
        P(w,c)=b
        j=0;
    end
end
end
tb=0
for i=1:boyut
    for j=1:boyut
        tb=tb+P(i,j);
    end
end
alfa1=0.3;
alfa2=0.7;
R1=[1 0]
R2=[1 0]
S1=[-mu1 mu1*alfa1;0 -mu2]
S0=[0.1 0.2;0.5 0.6]
I=[1 0;0 1]
S10=[mu1*(1-alfa1);mu2*(1-alfa2)]
S00=[mu1*(1-alfa2);mu2*(1-alfa2)]
ro01=lambda/mu1;

```

```

ro02=lambda/mu2;
ro11=0.5;
ro12=0.5
ro=lambda / ((1/mu1)+(alfa1/mu2))
r1=lambda / ((1/0.5)+(alfa1/0.5))
alfa1=0.3;
alfa2=0.7;
k=2;
    for i=2:boyut
        for j=1:boyut-1
            zz=((j-1)-(i-1)+1)
            pay1=(R1*(lambda*(lambda*I-S1)^-1)^zz)*(lambda*I-S1)^-1*S10
            pay2=(R2*(lambda*(lambda*I-S0)^-1)^zz)*(lambda*I-S0)^-1*S00
            C=pay1/pay2
            if j>=i-1
                W(i,j)=log(C)
            else
                W(i,j)=0
            end
        end
    end
end
for i=1:1
    for j=1:boyut-1
        pay1=(R1*(lambda*(lambda*I-S1)^-1)^(j-1))*(lambda*I-S1)^-1*S10
        pay2=(R2*(lambda*(lambda*I-S0)^-1)^(j-1))*(lambda*I-S0)^-1*S00
        C=pay1/pay2
        if j>=i-1
            W(i,j)=log(C)
        else
            W(i,j)=0
        end
    end
end
eks=boyut-1
    for i=2:eks+1
        eks1=eks-(i-1)
            for j=boyut:boyut
                top1=0;
                top0=0;

```

```

        for ic=0:eks1
            zus1=(R1*(lambda*(lambda*I-S1)^-1)^ic)*(lambda*I-S1)^-1*S10
            zalt1=(R2*(lambda*(lambda*I-S0)^-1)^ic)*(lambda*I-S0)^-1*S00
            top1=top1+zus1
            top0=top0+zalt1
        end
        bir=1-top1
        iki=1-top0
        C=bir/iki
        W(i,j)=log(C)
    end
end
W(1,boyut)=W(2,boyut);
Toplam=0
for i=1:boyut
    for j=1:boyut
        Toplam=Toplam+ W(i,j)*P(i,j)
    end
end
alfa=0.1
beta=0.05
A=(1-beta)/alfa
B=beta/(1-alfa)
alt=log(B)
ust=log(A)
if Toplam<alt
    Karar='kabul'
    Toplam
end
if Toplam>ust
    Karar='ret'
    Toplam
end
if Toplam>alt && Toplam<ust
    Karar='devam'
    %y=input('Gözlem Değeri Giriniz:');
    yeni_erlang(x,lambda,mu1,mu2);
end

```

2. erlang.m fonksiyon kodları

```
function [X,lambda,mu1,mu2] =Erlang(n,lam,m1,m2)
global nextarrival nextdeparture nextdeparture1 nextdeparture2
current say_ayrilis X
lambda=lam;
mu1=m1;
mu2=m2;
nextarrival= (-1/lambda)*log(rand);
now= 0;
nextdeparture= inf;
nextdeparture1= inf;
nextdeparture2= inf;
current= 0;
alfa1=0.30
alfa2=1-alfa1
alfa1_yuzde=100*alfa1
alfa2_yuzde=100*alfa2
top=0
say_ayrilis=0;
for i=1:20
y=ceil(100*rand);
if y <= alfa1_yuzde
sec=1;
else
sec=2;
end
if nextarrival < nextdeparture,
now= nextarrival;
if current<n
current= current + 1;
disp(sprintf('Arrival at : %f (current length %d)', now,
current));
end
nextarrival= now + (-1/lambda)*log(rand);
if current == 1,
nextdeparture1= now + (-1/mu1)*log(rand);
if sec==1
nextdeparture2= now + (-1/mu2)*log(rand);
nextdeparture=nextdeparture1+nextdeparture2;
else
nextdeparture=nextdeparture1;
```

```

        end
    end
else
    now= nextdeparture;
    current= current - 1;
    say_ayrilis=say_ayrilis+1;
    X(say_ayrilis)=current;
    disp(sprintf('Departure at : %f (current length %d)', now,
current));
    if current > 0,
        nextdeparture1= now + (-1/mu1)*log(rand);
        if sec==1
            nextdeparture2= now + (-1/mu2)*log(rand);
            nextdeparture=nextdeparture1+nextdeparture2;
        else
            nextdeparture=nextdeparture1;
        end
    else
        nextdeparture= inf;
    end
end
end
end
end

```

3. dizi_devam.m fonksiyon kodları

```

function [P] = dizi_devam(x,lambd,mu1,mu2)
boyut=3;
uz=length(x);
v=num2str(x);
    maxx=max(x)
    minn=min(x)
    for k=1:boyut
        for l=1:boyut
            P(k,l)=0;
        end
    end
end
    for k=1:boyut
        for l=1:boyut
            W(k,l)=0;
        end
    end

```

```

end
m=1;
tk='';
for h=1:uz-1
    for i=h:h
        tk='';
        for j=i:i+1
            tk=strcat(tk,num2str(x(j)));
            a=findstr(v,tk);
            b=length(a);

            end
        [y]=tk;

        w=str2num(y(1))+1;
        c=str2num(y(2))+1;
        P(w,c)=b

        j=0;
    end
    say(h)=str2num(tk);
    end
    for h=1:uz-1
        for i=h:h
            tk='';
            b=0;
            n=1;
            for j=i:i+1
                if n==1
                    x1=x(j)
                else
                    y1=x(j);
                end
                tk=strcat(tk,num2str(x(j)));
                n=n+1;
            end
            for sy=1:uz-1
                if say(sy)==str2num(tk)
                    b=b+1;
                end
            [y]=tk;

            w=x1+1;

```

```

                c=y1+1;
                P(w,c)=b
            j=0;
        end
    end
end
end
    tb=0
    for i=1:boyut
        for j=1:boyut
            tb=tb+P(i,j);
        end
    end
    alfa1=0.3;
    alfa2=0.7;
    R1=[1 0]
    R2=[1 0]
    S1=[-mu1 mu1*alfa1;0 -mu2]
    S0=[0.1 0.2;0.5 0.6]
    I=[1 0;0 1]
    S10=[mu1*(1-alfa1);mu2*(1-alfa2)]
    S00=[mu1*(1-alfa2);mu2*(1-alfa2)]
    ro01=lambda/mu1;
    ro02=lambda/mu2;
    ro11=0.5;
    ro12=0.5
    alfa1=0.3;
    alfa2=0.7;
    k=2;
    for i=2:boyut
        for j=1:boyut-1
            zz=((j-1)-(i-1)+1)
            pay1=(R1*(lambda*(lambda*I-S1)^-1)^zz)*(lambda*I-S1)^-1*S10
            pay2=(R2*(lambda*(lambda*I-S0)^-1)^zz)*(lambda*I-S0)^-1*S00
            C=pay1/pay2
            if j>=i-1
                W(i,j)=log(C)
            else
                W(i,j)=0
            end
        end
    end
end

```

```

end
for i=1:1
    for j=1:boyut-1
        pay1=(R1*(lambda*(lambda*I-S1)^-1)^(j-1))*(lambda*I-S1)^-1*S10
        pay2=(R2*(lambda*(lambda*I-S0)^-1)^(j-1))*(lambda*I-S0)^-1*S00
        C=pay1/pay2
        W(i,j)=log(C)
    end
end
eks=boyut-1
for i=2:eks+1
    eks1=eks-(i-1)
    for j=boyut:boyut
        top1=0;
        top0=0;
        for ic=0:eks1
            zus1=(R1*(lambda*(lambda*I-S1)^-1)^ic)*(lambda*I-S1)^-1*S10
            zalt1=(R2*(lambda*(lambda*I-S0)^-1)^ic)*(lambda*I-S0)^-1*S00
            top1=top1+zus1
            top0=top0+zalt1
        end
        bir=1-top1
        iki=1-top0
        C=bir/iki
        W(i,j)=log(C)
    end
end
W(1,boyut)=W(2,boyut);
Toplam=0
for i=1:boyut
    for j=1:boyut
        Toplam=Toplam+ W(i,j)*P(i,j)
    end
end
alfa=0.1
beta=0.05
A=(1-beta)/alfa
B=beta/(1-alfa)
alt=log(B)
ust=log(A)

```

```

if Toplam<alt
    Karar='kabul'
    Toplam
end
if Toplam>ust
    Karar='ret'
    Toplam
end
if Toplam>alt && Toplam<ust
    Karar='devam'
    %y=input('Gözlem Değeri Giriniz:');
    yeni_erlang(x,lambda,mu1,mu2);
end
end

```

4. yeni_erlang.m fonksiyonu kodlar

```

function [X,lambda,mu1,mu2] = yeni_erlang(n,lambda,mu1,mu2)
global nextarrival nextdeparture nextdeparture1 nextdeparture2
current say_ayrilis X
for i=1:20
    alfa1=0.30
    alfa2=1-alfa1
    alfa1_yuzde=100*alfa1
    alfa2_yuzde=100*alfa2
    top=0
    say_ayrilis=0;
    for i=1:20
        y=ceil(100*rand);
        if y <= alfa1_yuzde
            sec=1;
        else
            sec=2;
        end
        if nextarrival < nextdeparture,
            now= nextarrival;
            if current<n
                current= current + 1;
                disp(sprintf('Arrival at : %f (current length %d)', now,
                    current));
            end
            nextarrival= now + (-1/lambda)*log(rand);

```

```

    if current == 1,
        nextdeparture1= now + (-1/mu1)*log(rand);
        if sec==1
            nextdeparture2= now + (-1/mu2)*log(rand);
            nextdeparture=nextdeparture1+nextdeparture2;
        else
            nextdeparture=nextdeparture1;
        end
    end
else
    now= nextdeparture;
    current= current - 1;
    say_ayrilis=say_ayrilis+1;
    X(say_ayrilis)=current;
    disp(sprintf('Departure at : %f (current length %d)', now,
current));
    if current > 0,
        nextdeparture1= now + (-1/mu1)*log(rand);
        if sec==1
            nextdeparture2= now + (-1/mu2)*log(rand);
            nextdeparture=nextdeparture1+nextdeparture2;
        else
            nextdeparture=nextdeparture1;
        end
    else
        nextdeparture= inf;
    end
end
end
end
dizi_devam(X,lambda,mu1,mu2);
end

```

Ek I: Aşağıdaki program kodları Karakteristik İşlem Eğrisi ve Ortalama Örneklem Sayısını hesaplamaktadır. Beş program dosyasından oluşmakta ve function şeklinde aşağıdaki ilk programdan çağrılmaktadır.

1. Aşağıdaki kodlar bilgisayar ortamında trobul.m olarak kaydedilmiştir.

```
ro=2
c=0;
alfa=0.1;
beta=0.01;
A=(1-beta)/alfa;
B=beta/(1-alfa);
% farklı trafik yoğunluğu için özdeğerlerin bulunup grafik çizimi
for b=ro:-0.1:0.9
x=(-1.6:0.0001:1.6);
n=length(x);
k=0;
c=c+1;
lmd(c)=ozdeger(b);
lmd1(c)=ozdeger2(b);
for i=1:n
    P=gecmat2(x(i),b);
    y=eig(P);
    lmd2(i)=max(y);
    if abs(lmd2(i)-1)<10^-4
        k=k+1;
        tro(k)=x(i);
    end
end
mlamda=min(lmd2);
mx=find(lmd2==mlamda);
mx=mx(1);
if x(mx)<0
    tros(c)=min(tro);
else
    tros(c)=max(tro);
end
plot(x,lmd2)
hold on;
plot(x,1,'-r')
hold on;
% karakteristik işlem sayısı
```

```

if tros(c)~=0
    L(c)=(A^tros(c)-1)/(A^tros(c)-B^tros(c))
else
    L(c)=log(A)/(log(A)-log(B))
End
% ortalama örneklem sayısı
if lmd(c)~=0
    E(c)=(L(c)*log(B)+(1-L(c))*log(A))/lmd(c)
else
    E(c)=(L(c)*log(B)+(1-L(c))*log(A))/lmd1(c)
end
end

```

2. gecmat2.m fonksiyonu kodları

```

function [P]=gecmat2(t,ro)
P1=gmam(0.9,10);
P0=gmam(2,10);
% n*n lik matris
for i=1:10
    for j=1:10
        if P0(i,j)==0
            P3(i,j)=0;
        else
            P3(i,j)=(P1(i,j)/P0(i,j))^t;
        end
    end
end
P5=gmam(ro,10);
P=P5.*P3;
End

```

3. gmat.m fonksiyonu kodları

```

function [P]=gmat(ro,n)
% n*n lik matris
for i=1:n-1
    ke(i)=(1/(1+ro)) * ((ro/(ro+1))^(i-1));
end
for i=1:2
    for j=1:(n-1)
        P(i,j)=ke(j);
    end
end

```

```

        end
    end
    for i=3:10
        dd=0;
        for j=1:(n-1)
            if j<=(i-2)
                for i1=j:(i-2)
                    P(i,j)=0;
                end
            else
                dd=dd+1;
                P(i,j)=ke(dd);
            end
        end
    end
    P1=1-sum(P')';
    P=[P,P1];
End

```

4. ozdeger.m fonksiyon kodları

```

function lmd=ozdeger(ro)
x=[-0.02,-0.01,0.01,0.02];
for i=1:4
    P=gecmat2(x(i),ro);
    y=eig(P);
    lamda(i)=max(y);
end
lmd=(1/(12*0.01))*(lamda(1)-8*lamda(2)+8*lamda(3)-lamda(4));

```

5. ozdeger2.m fonksiyon kodları

```

function lmd1 = ozdeger2(ro)
x=[-0.02,-0.01,0,0.01,0.02];
for i=1:5
    P=gecmat2(x(i),ro);
    y=eig(P);
    lamda(i)=max(y);
end
lmd1=(1/(12*(0.01^2)))*(-lamda(1)+16*lamda(2)-
30*lamda(3)+16*lamda(4)-lamda(5));
end

```

ÖZGEÇMİŞ



Ad Soyad: Müjgan ZOBu

Doğum Yeri ve Tarihi: 05/08/1969

Adres: Amasya Üniversitesi Bilgi İşlem Daire Başkanlığı

E-Posta: mujganzobu@hotmail.com

Lisans: Ondokuz Mayıs Üniversitesi Fen Edebiyat Fakültesi İstatistik Bölümü, 1986-1990

Yüksek Lisans: Gazi Üniversitesi Fen Edebiyat Fakültesi İstatistik Bölümü, 1996-2000

Mesleki Deneyim ve Ödüller:

Ondokuz Mayıs Üniversitesi Amasya Eğitim Fakültesi, 1994-2006

Amasya Üniversitesi Eğitim Fakültesi, 2006-2008

Amasya Üniversitesi Bilgi İşlem Daire Başkanlığı, 2008-....

Yayın ve Patent Listesi:

- Bayrak H., **Zobu M.**, 2001: Grafıksel Modeller ile bir Yaklaşım. *Eğitimde ve Bilimde Sağlık*, 31(40,45).

TEZDEN TÜRETİLEN SUNUMLAR

- **Zobu M.**, Sağlam V., 2012 : Kuyruklarda Trafik Yoğunluğunun Kontrolü. 13. Uluslararası Ekonometri, Yöneylem Araştırması ve İstatistik Konferansı- DAÜ, Mayıs 24-26, Gazimağusa, KKTC.
- Sağlam V., **Zobu M.**, 2012 : Sequential Probability Ratio Test in Queueing Systems of the M/G/1 and GI/M/s. 8. Uluslararası İstatistik Günleri Sempozyumu -Anadolu Üniversitesi, Ekim 11-13 ,2012 Eskişehir, Türkiye.

