

# (s,S) STOK SİSTEMLERİNDE EN UYGUN STOK POLİTİKASININ BELİRLENMESİ İÇİN SİMÜLASYON UYGULAMASI ÜZERİNE TEORİK BİR ÇALIŞMA ÖRNEĞİ

Süleyman ŞAHİN<sup>(\*)</sup>

**Özet:** Bu çalışmada, tedarik süresinin ve talebin stokastik bir durum gösterdiği, elde bulundurmama maliyetine maruz ve periyodik olarak gözden geçirilen tek ürünlü bir (s,S) stok sistemi ele alınmış ve bu sistemin simülasyon metodu ile çözümü incelenmiştir. Bu stok sistemine uygun bir örnek problemin çözümü için geliştirilen bir simülasyon modeli ve FORTRAN 77 programlama dilinde hazırlanan bir bilgisayar programı verilmiştir. Daha sonra bu program çalıştırılarak; incelenen örnek problem için en düşük ortalama aylık toplam maliyeti veren (s,S) stok politikasını belirlemeye yarayacak sonuç değerleri elde edilmiştir.

**Abstract:** In this study, a single product (s,S) inventory system which is reviewed periodically is examined. This system has a shortage cost when lead time and demand are stochastic. And this system is assessed by the simulation method. A simulation model which developed for the solution of a sample problem suited to this system, and a computer program which is prepared by FORTRAN 77 language, are presented. Furthermore, the expected values that provide to determine the (s,S) inventory policy which has the lowest of average total cost per month are obtained by running this program.

## I. Giriş

Bir üretim sisteminde mamule dolaylı veya dolaysız olarak katılan bütün fiziksel varlıkları ve mamulün kendisini içine alan stok kavramı (Kobu, 1999: 292), kullanılmayı veya satılmayı bekleyerek belli bir süre atıl durumda tutulan, ekonomik değere sahip kaynaklar (malzeme veya mallar) olarak tanımlanmaktadır (Gençyılmaz, 1988: 9).

İşletmelerin üretim politikaları ile ilgili önemli konulardan biri olan stok problemi, bir stok sisteminde en iyi işletme kararlarının alınmasını gerektirir. Genelde, işletmenin üretim hızı ile satışları arasında bir denge kurabilmek için stok yapma ihtiyacı ortaya çıkar.

Diğer taraftan bütün işletmelerin (JIT modelini kullananlar dahil);

- Üretim faaliyetlerini birbirinden bağımsız olarak sürdürmek,
- Üretim talebindeki değişimi karşılamak,
- Üretim planındaki esnekliğe müsaade etmek,
- Ham madde teslim zamanındaki değişimi (muhtemel gecikmeleri) dikkate alarak bir emniyet stok seviyesi belirlemek,
- Ekonomik satın alma/sipariş miktarının avantajından istifade etmek

<sup>(\*)</sup> Arş.Gör. İnönü Üniversitesi İİBF İşletme Bölümü

gibi amaçlarından dolayı stok bulundurdıkları (Chase, 1995: 547) ve bunun için de stokla ilgili bazı maliyetleri yüklenmek zorunda kaldıkları bilinmektedir.

Bu maliyetler de genel olarak şu üç grup içerisinde ele alınmaktadır:

- 1) Sipariş maliyetleri (satın alma maliyeti + sabit maliyet)
- 2) Elde bulundurma maliyeti
- 3) Elde bulundurmama maliyeti (Kobu, 1999: 300; Gençyılmaz, 1988: 22-30; Öztürk, 1992: 198; Taha, 2000: 434; Chase, 1995: 548).

Buna göre stok bulundurma yoluyla gerçekleşen amaçlar ve faydalar ile yüklenen maliyetler arasındaki dengenin sağlanabilmesi için uygun bir stok seviyesi ve stok politikası belirlenmesi gerekmektedir. İşte stok kontrolünün de amacı, stok bulundurma konusunda işletmenin katlanacağı maliyetler ile elde edeceği faydalar arasında işletme için en uygun denge noktasını bulan stok modelini belirlemek ve bunu sağlayacak stok sistemini kurup işletmektir denilebilir.

## II. Bir Stok Simülasyonu

Çalışmada incelenecek olan stok sistemi, stok durumu periyodik olarak gözden geçirilen bir (s,S) stok sistemidir. Bu stok sistemi Liu ve Yang (1999), Ehrhardt (1984), Shaoxiang ve Lambrecht (1996), Bookbinder ve Çakanyıldırım (1999) ve Anupindi ve Tayur (1998) gibi bir çok yazar tarafından ele alınmıştır. Bunlardan Liu ve Yang (1999)'ın ömür süreleri üstel dağılım özelliği taşıyan mallar için sürekli gözden geçirilen bir (s,S) stok modelini incelediklerini, Ehrhardt (1984)'ın ise tedarik süresi stokastik olan dinamik bir stok modeli için en uygun (s,S) politikalarını araştıran bir çalışma yaptığını görmek mümkündür. Shaoxiang ve Lambrecht (1996) çalışmalarında, üretim/sipariş maliyetinin (K) ve üretim kapasitesi/sipariş miktarının (CP) değişik durumları için (s,S) politikaları ile ilgili yapılan çalışmalar hakkında bazı bilgiler sunmuşlardır. Şöyle ki:  $K = 0$  ve  $CP = \infty$  durumu için temel stok (base-stock) politikasının uygulandığını;  $K = 0$  ve  $CP < \infty$  durumu için düzenlenmiş (modified) temel stok politikasının optimal çözüm sağladığını;  $K > 0$  ve  $CP = \infty$  durumu için ise, (s,S) politikasının optimal politika olduğunu -bu konuda yapılan çalışmalara atıfta bulunarak- ifade etmişlerdir. Diğer taraftan Şahin (1990: 1-6), bu stok sistemi ile ilgili geniş bilgiler vermiş ve periyodik gözden geçirme sistemleri için en uygun stok politikasının (s,S) olduğunu belirtmiştir. Bu görüşü destekleyenler arasından Ishigaki (1991) ve Zheng (1991) tarafından yapılan çalışmalar örnek olarak verilebilir.

Stok kontrolünde simülasyon kullanımı oldukça yaygın olarak işlenmekte ve uygulanmaktadır. Bunlardan Silver (1985: 368-324) ve Payne (1982: 106-11) tarafından yapılan çalışmalarda, en uygun stok modelinin ve stok düzenleme politikasının belirlenmesinde simülasyon kullanımı ile ilgili bazı temel bilgilerin verilmiştir. Ayrıca, Baykoç ve Erol (1998) tarafından SLAM II simülasyon dili ile yapılan JIT üretim sistemi ile ilgili çalışma ve Savsar (1997) tarafından SIMAN simülasyon dili ile yapılan JIT üretim

sistemindeki Kanban uygulamasına yönelik çalışma da örnek olarak verilebilir. Buradaki çalışmada ise, üretilen tek bir ürün için uygulanan alternatif stok düzenleme politikalarından en uygun politikanın hangisi olduğunu belirlemek amacıyla simülasyonun nasıl kullanılabileceği gösterilecektir.

İncelenen örnek stok problemi, Law ve Kelton (1991: 25-30) tarafından ele alınmış ve bu problem için bir simülasyon modeli geliştirilmiştir. Bu çalışmada ise, incelenen stok problemini çözmek üzere onların hazırladığı simülasyon modelinin FORTRAN Bilgisayar Programı, bazı ilave ve değişiklikler yapılmak suretiyle FORTRAN 77 ile çalıştırılmış ve en uygun stok düzenleme politikasını belirlemeye yarayacak stok maliyet hesaplarını veren sonuç raporu elde edilmiştir.

Bilindiği gibi, simülasyon uygulamalarında kullanılan birçok simülasyon dili mevcuttur ve genel olarak;

- a) Sürekli simülasyon dilleri
- b) Kesikli simülasyon dilleri

biçiminde iki kısma ayrılmaktadırlar.

Kesikli simülasyon dilleri;

- 1) Olay sıralamalı (event scheduling)
- 2) Sürece yönelik (process orient)

biçiminde iki genel bölüme ayrılmaktadır. Bu iki bölüm arasındaki fark ise; sürece yönelik dillerde kullanıcıyı karmaşık hesaplamalar ve mantıksal hatalardan kurtarmak için *kaynak*, *kuyruk* ve *hizmet* gibi faaliyetlerin otomatik olarak yürütülmesidir. Olay sıralamalı belli başlı diller SIMSCRIPT, SLAM ve SIMAN' dir. Bu üç dil zamanla sürece yönelik hale gelerek FORTRAN ve C gibi üst seviye dillerinin modellerde kullanımına izin vermektedirler. Sürece yönelik en eski dil olarak GPSS ve yeni bir simülasyon dili olarak SIMNET II örnek olarak verilebilir (Taha, 2000: 698-699).

Diğer taraftan, günümüzde kesikli simülasyon olaylarına yönelik hazırlanmış bir çok simülasyon yazılım programı mevcuttur. Örnek olarak; Arena, AweSim!, Extend, Extend+BPR, Extend+Manufacturing, GPSS/H Professional, GPSS/H Run-time, GPSS/H Personal, GPSS/H Student, GPSS/PC, iThink, MAST, MODSIM III, SIMPROCESS, SIMSCRIPT II.5, SIMUL8, SLAM II, STELLA, Taylor II verilebilir. (The Consummate Design Center, 2002).

#### A. Problemin Belirlenmesi

Law ve Kelton (1991: 25-30) tarafından incelenen stok problemi şöyledir: Tek bir kalem/çeşit mal üreten bir şirket, gelecek  $n$  kadar ay süresince, her ay için ne miktar mal stok edeceğine karar vermek istemektedir. Herhangi iki talep arasında geçen süre, ortalaması (0.1) ay olan *bağımsız ve özdeş dağılımlı* (independent identically distributed) bir *üstel* (exponential) değişkendir. Talep miktarı ( $D$ ), *bağımsız ve özdeş dağılımlı* bir tesadüfi

değişkendir ve talebin gerçekleşmesinden bağımsızdır. Ayrıca D'nin farklı durumları ve bunların gerçekleşme ihtimalleri (P(D)) şöyledir:

D	P(D)
1	1/6
2	1/3
3	1/3
4	1/6

Stok seviyesi periyodik olarak (her ay başında) gözden geçirilmektedir. Şirket, kendisine mal temin eden (tedarikçi) firmadan ne kadar mal alacağına elindeki stok durumuna bağlı olarak karar vermektedir ve  $Z$  kadar mal için;  $K + iZ$  kadar bir maliyet yüklenmektedir. Burada  $K$  sabit sipariş verme maliyeti ve  $i$  birim satın alma maliyetidir. İncelenen örnek stok probleminde  $K = 32 \$$  ve  $i = 3 \$$  olduğu kabul edilmektedir.

Genel olarak bir siparişin verilmesi ile alınması arasında *teslim gecikmesi* veya *tedarik süresi* denilen bir süre geçer. Şirket bu süreyi, 0.5 ay ile 1 ay arasında değişen *düzgün* (uniform) dağılımlı bir tesadüfi değişken olarak kabul etmektedir.

Şirket ne kadar mal sipariş vereceğine karar vermek için  $(s, S)$  stok politikasını kullanmaktadır. Şöyle ki; periyodik olarak her ay başında gözlemlenen stok seviyesi, "yeniden sipariş verme (reorder point)" noktasından küçük ( $I < s$ ) ise,  $S - I$  kadar mal sipariş edilmektedir. Eğer ( $I \geq s$ ) ise, o zaman sipariş verilmemektedir.

$$Z = \begin{cases} S - I ; & \text{eğer } I < s \text{ ise} \\ 0 ; & \text{eğer } I \geq s \text{ ise} \end{cases} \quad (1)$$

Bir talep olduğunda, o andaki stok seviyesi talepten büyük veya talep miktarına eşitse, talep derhal karşılanmaktadır. Stok seviyesi talepten küçük ise, o zaman talebin karşılanamayan kısmı gelecekteki teslimatlardan karşılanmak üzere bekletilmektedir (Bu durumda yeni stok seviyesi negatif bir tamsayı durumuna gelir). Bir sipariş teslim alındığında, bekletilen talep miktarı bundan düşülmekte, geriye kalan kısım mevcut stok üzerine ilave edilmektedir. Bu duruma göre ele alınan stok politikasında *sipariş verme maliyetlerine* ilave olarak, *elde bulundurma* ve *elde bulundurmama maliyetleri* de söz konusudur.

Yukarıda açıklanan durumları ifade eden bazı notasyonlar ve bunların açıklamaları şöyledir:

$I(t)$  :  $t$  zamanındaki stok seviyesi olup; pozitif, sıfır veya negatif olabilir.

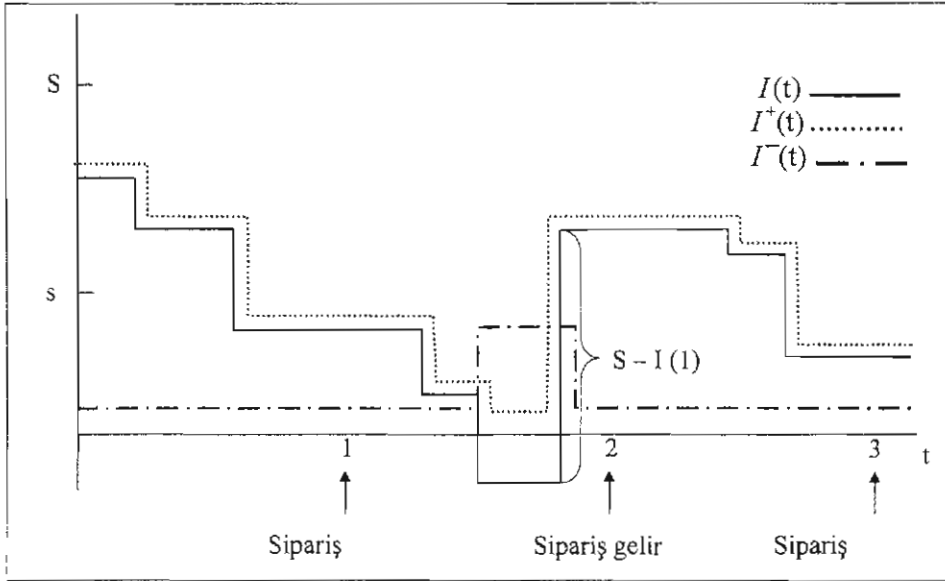
$I^+(t)$  :  $t$  zamanındaki stokta mevcut olan mal sayısıdır.

$$(I^+(t) = \max. \{ I(t), 0 \} \text{ ve } I^+(t) \geq 0) \quad (2)$$

$I^-(t)$  : t zamanındaki birikmiş karşılanamayan talep bakiyesidir.

$$(I^-(t) = \max. \{-I(t), 0\} \text{ ve } I^-(t) \geq 0) \quad (3)$$

Bu notasyonların ifade ettikleri durumlar,  $t \geq 0$  için Şekil 1'de gösterilmiştir. Bir talep olduğu anda,  $I(t)$  seviyesi talep miktarı kadar azalmaktadır.



Şekil 1: t Zamanına Göre  $I(t)$ ,  $I^+(t)$  ve  $I^-(t)$ 'nin Gerçekleşme Durumları

Bu stok modelinde şirketin maliyet hesaplamaları ile ilgili varsayımlar ise şöyledir:

$n$  aylık dönem için aylık ortalama stoktaki mal miktarı

$$\int_0^n I^+(t) dt / n \quad (4)$$

kadardır. Birim mal başına aylık elde bulundurma maliyeti  $h = 1\$$  olmak üzere;  $n$  aylık dönem için ortalama elde bulundurma maliyeti

$$\frac{\int_0^n I^+(t) dt}{n} \quad (5)$$

ifadesi ile hesaplanmaktadır.

$n$  aylık dönem içerisinde, aylık ortalama bekletilen talep miktarı

$$\int_0^n I^-(t) dt / n \quad (6)$$

kadardır. Birim mal başına aylık elde bulundurmama maliyeti  $\pi = 5\text{\$}$  olmak üzere; şirketin  $n$  aylık dönem için *ortalama elde bulundurmama maliyeti* ise aşağıdaki formülle hesaplanmaktadır:

$$\pi \frac{\int_0^n I^-(t) dt}{n} \quad (7)$$

Başlangıçtaki stok seviyesinin  $I(0) = 60$  olduğu ve verilmiş herhangi bir siparişin bulunmadığı farz edilmektedir. Ele alınan stok sisteminin  $n = 120$  ay (10 yıl) için simüle edilmesi öngörülmektedir. Aylık ortalama toplam maliyet -ki bu maliyet, aylık ortalama sipariş maliyeti ile aylık ortalama elde bulundurma ve bulundurmama maliyetlerinin toplamına eşittir- esasına dayalı olarak, Tablo 1’de verilen dokuz adet stok sipariş politikasının karşılaştırılması söz konusudur.

Tablo 1: *Değişik (s,S) Stok Politikaları*

<b>S</b>	20	20	20	20	40	40	40	60	60
<b>s</b>	40	60	80	100	60	80	100	80	100

### B. Geliştirilen FORTRAN Programı

Ele alınan (s,S) stok sistemi ile ilgili olarak Law ve Kelton (1991: 25-30) tarafından geliştirilen stok modeli, modelin FORTRAN 77 ile çalıştırılabilmesi için bazı değişiklikler ve ilâveler yapılmak suretiyle yeniden düzenlenmiştir. Düzenlenen bu stok modelinde kullanılacak olan olay tipleri ve bu olay tiplerine ait olay tanımları Tablo 2’de verilmiştir:

Tablo 2: *Olay Tipleri ve Tanımları*

<i>Olay Tipi</i>	<i>Olayın Tanımı</i>
1	Şirketin verdiği siparişin gelmesi
2	Müşteriden talep gelmesi
3	Simülasyonun $n$ ay sonunda sona ermesi
4	Stok ve sipariş durumunun ay başında değerlemesinin yapılması

Programda simülasyonu sona erdirmek için 4. olay yerine 3. olay tercih edilecektir. Çünkü, TIME = 120 ay olduğunda “simülasyonun sonu” ile “stok değerlemesi” olayları planlanacak ve bu anda ilk olarak önceki olay icra edilecektir. Çünkü simülasyon, TIME = 120 ay’dan sonraya sarkınca, sipariş verme durumu ile stok seviyesinin gözden geçirilmesinin bir anlamı yoktur.

Buna göre 4. Tip olaydan önce 3. Tip olayın yerine getirilmesi garanti edilmiştir. Çünkü zamanlama programı TIMING sayesinde, aynı bir zaman noktası için iki veya daha fazla olayın planlanmış olması durumunda - ki buna "zaman düğümü" denilir - olaylardan olay numarası daha küçük olana öncelik verilecektir. Genelde bir simülasyon modeli, zaman düğümlerinin meydana gelebilmesi söz konusu olduğunda, yapılması gereken bir sipariş ile ilgili olayları programlamak için düzenlenmelidir.

İncelenen stok modeli ile ilgili FORTRAN 77 programında kullanılan alt programlar, terimler ve notasyonlarla ilgili olarak gerekli bazı açıklamalar Tablo 3.a ve Tablo 3.b'de verilmiştir.

Tablo 3.a: FORTRAN 77 Değişkenleri ve Alt programlar

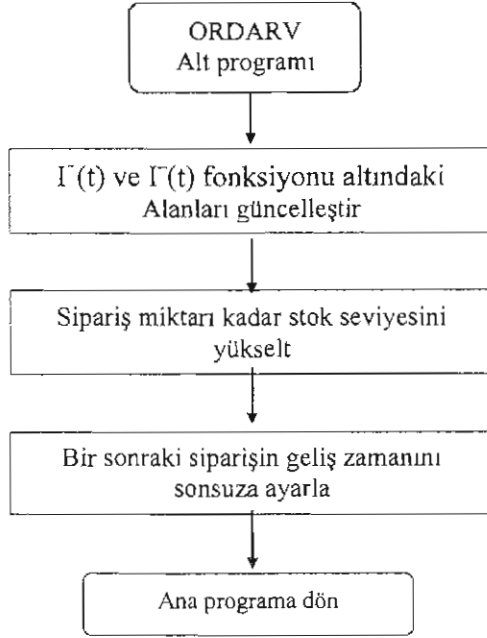
Alt programlar	Gördüğü Fonksiyon
INIT	Başlangıç programı.
TIMING	Zamanlama programı.
ORDARV	1. Tip olayları işleten olay denetleme programı.
DEMAND	2. Tip olayları işleten olay denetleme programı.
REPORT	3. Tip olayları işleten olay denetleme programı (rapor üretici).
EVALU8	4. Tip olayları işleten olay denetleme programı.
UPDATE	Stok seviyesi değiştiğinde veya simülasyon sona erdiğinde $I(t)$ ve $I'(t)$ fonksiyonlarının altındaki alanları güncelleştiren Alt program.
RANDI(Z)	Fonksiyon PROBD(I) ; (I=1,2,...,NVALUE) ile ilişkili olarak 1 ile NVALUE (pozitif tamsayı) aralığında tesadüfi bir tamsayı üreten fonksiyon. Burada Z sahte deęişkendir. [ X bir tesadüfi tamsayı ise, X' in I' ya eşit veya ondan daha az bir deęer almasının hesabı PROBD(I) ile yapılır.] PROBD(I) ; (I=1,2,...,NVALUE) ve NVALUE deęerleri Ana programa yerleştirilir ve buradaki COMMON/RANDOM/NVALUE,PROBD(25) satırı ile de RANDI programına gönderme yapılır. NVALUE en fazla 25 deęerini alabildiğine dikkat edilmeli.
RANUN(Z)	Tesadüfi sayı üretici ;U(0,1) aralığındaki U sayısını üretir.
EXPON(RMEAN)	REMEAN ortalama ile üstel dağılımlı tesadüfi deęişken üretici.
UNIFRM(A;B)	(A;B) aralığında düzgün dağılımlı sürekli tesadüfi deęişken üreten fonksiyon. Burada A ve B gerçel (real) deęer olarak alınmalıdır. Ayrıca A deęeri B' den küçük olmalıdır (A<B).

Tablo 3.b: FORTRAN 77 Değişkenleri ve Alt programlar (Devamı)

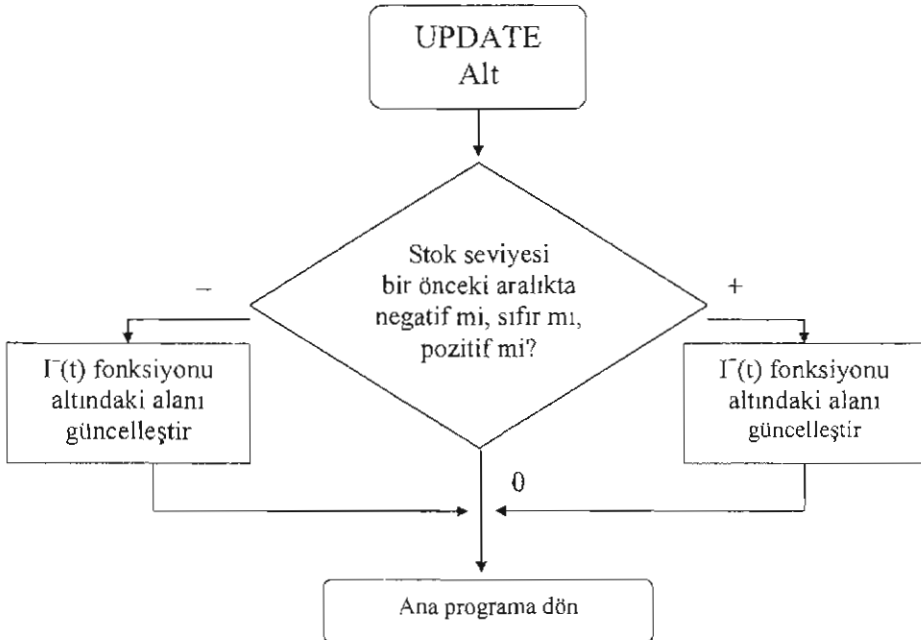
Girdi Parametreleri	Açıklaması
BIGS	(s,S) stok politikasındaki S sayısı (ikinci terim).
H	Ortalama elde bulundurma maliyeti ile ilgili h çarpanı.
INCRMC	Sipariş edilen her kalem mal için ödenen birim i maliyeti.
INITIL	Başlangıçtaki stok seviyesi.
MDEMDT	Her iki talep arasında geçen ortalama süre.
Model Değişkenleri	Açıklaması
NMNTHS	Aylar ile ifade edilen simülasyon uzunluğu.
NPOLCY	İncelenen stok politikası sayısı.
NVALUE	Meydana gelebilecek talep durumu sayısı.
PI	Ortalama elde bulundurmama maliyeti ile ilgili $\pi$ çarpanı.
PROBD(I)	Talebin I ' dan küçük veya eşit olması ihtimali.
SETUPC	Sipariş verme (sabit K ) maliyeti.
SMALLS	(s,S) stok politikasındaki s sayısı (birinci terim).
AMINUS	$I(t)$ fonksiyonu altındaki alan.
AMOUNT	Sipariş edilen mal miktarı: Z
APLUS	$I^+(t)$ fonksiyonu altındaki alan.
DSIZE	Talep miktarı.
INVLEV	(t) anındaki stok seviyesi: l(t)
NEVNTS	Stok modeli için olay tipi sayısı.
NEXT	Meydana gelecek bir sonraki olayın tipi (1,2,3 veya 4).
TIME	Simülasyon saati.
TLEVNT	Stok seviyesini değiştiren en son olayın zamanı.
TNE(I)	I tipi olayın meydana gelme zamanı (I=1,2,3,4).
TORDC	Toplam sipariş maliyeti.
TSLE	En son olaydan itibaren geçen süre.
Çıktı Değişkenleri:	Açıklaması
ACOST	Aylık ortalama toplam maliyet.
AHLDC	Aylık ortalama elde bulundurma maliyeti.
AORDC	Aylık ortalama sipariş maliyeti.
ASHRC	Aylık ortalama elde bulundurmama maliyeti.

Aşağıda bazı alt programların mantıksal akış şemaları verilmiştir:

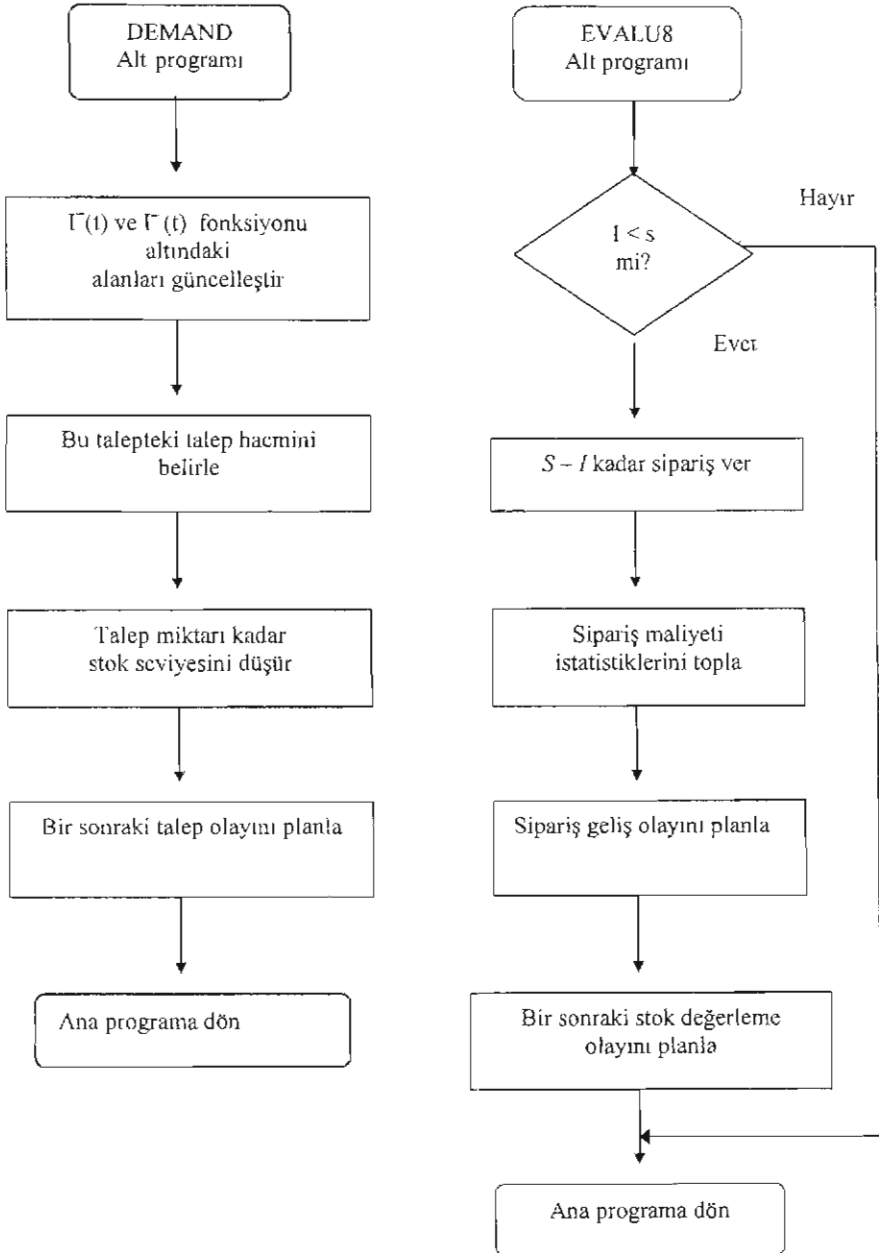




Şekil 2: ORDARV Alt programı Mantıksal Akış Şeması



Şekil 3: UPDATE Alt programı Mantıksal Akış Şeması



Şekil 4: DEMAND ve EVALU8 Alt programlarının Mantıksal Akış Şemaları

Yukarıda verilen mantıksal akış şemalarına göre düzenlenmiş olan simülasyon modelinin FORTRAN 77 programlama dili ile yazılmış kodları ise şu şekildedir:

```

C
C *** MAIN PROGRAM
C
      INTEGER AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,
      *SMALLS
      INTEGER I, NPOLICY
      REAL AMINUS, APLUS, H, INCRMC, MDEMDT, PI, SETUPC,
      *TIME,TLEVNT, TNE(4), TORDC
      COMMON /MODEL/AMINUS, AMOUNT, APLUS, BIGS, H, INCRMC,
      *INITIL,INVLEV, MDEMDT, NEVNTS, NEXT, NMNTHS, PI,
      *SETUPC, SMALLS, TIME,TLEVNT, TNE, TORDC
      REAL Q1,X1
      COMMON /RANDOM/ NVALUE, PROBD(25)
      COMMON /RANDEG/ Q1,X1,K1
      OPEN(5,FILE='GIRDI.DAT',STATUS='OLD')
      OPEN(6,FILE='SONUC.TXT',STATUS='NEW')
C
C *** INITIALIZE RANDOM FUNCTION VARIABLES
C
      Q1=37.0
      K1=6
      X1=1.0
C
C *** SPECIFY THE NUMBER OF EVENT TYPES FOR THE TIMING
C *** ROUTINE.
C
      NEVNTS=4
C
C *** READ INPUT PARAMETERS
C
      READ(5,10) INITIL,NMNTHS,NPOLCY,NVALUE
      10 FORMAT(4I10)
      READ(5,20) MDEMDT,SETUPC,INCRMC,H,PI
      20 FORMAT(5F10.0)
      READ(5,30) (PROBD(I), I=1,NVALUE)
      30 FORMAT(8F10.0)
C
C *****
C
C *** PRINT REPORT HEADING
C
      WRITE(6,40)
      40 FORMAT(' ',5X,'SINGLE-PRODUCT INVENTORY SYSTEM')
      WRITE(6,50) INITIL
      50 FORMAT(' ',5X,'INITIAL INVENTORY LEVEL', 21X,I3,'
      *ITEMS')
      WRITE(6,60) NVALUE
      60 FORMAT(' ',5X,'NUMBER OF DEMAND SIZES', 22X,I3)
      WRITE(6,70) (PROBD(I), I=1,NVALUE)

```

```

70 FORMAT(' ',5X,'DISTRIBUTION FUNCTION OF DEMAND
  *SIZES', 5X,8(F5.3,3X))
  WRITE(6,80) MDEMDT
80 FORMAT(' ',5X,'MEAN INTERDEMAND TIME',21X,F5.2,'
  *MONTHS')
  WRITE(6,90) NMNTHS
90 FORMAT(' ',5X,'LENGHT OF THE SIMULATION',20X,I3,'
  *MONTHS')
  WRITE(6,100) SETUPC, INCRMC, H, PI
100 FORMAT(' ',5X,'K = ',F5.1,3X,'I = ',F5.1,3X,'H =
  ',F5.1,3X,'PI = ',F5.1)
  WRITE(6,110)
110 FORMAT(//)
  WRITE(6,120)
120 FORMAT(' ',7X,'POLICY',6X,'AVERAGE COST',5X,'AVERAGE'
  *'ORDERING COST',5X,'AVERAGE HOLDING COST',5X,'AVERAGE
  *', 'SHORTAGE COST')
C
C *****
C
C *** RUN THE SIMULATION VARYING THE INVENTORY POLICY.
C
  DO 190 I=1, NPOLCY
C
C *** READ THE INVENTORY POLICY.
C
  READ(5,130) SMALLS, BIGS
130 FORMAT(2I10)
C
C *** INITIALIZE THE SIMULATION.
C
  CALL INIT
C
C *** DETERMINE THE NEXT EVENT.
C
140 CALL TIMING
C
C *** CALL THE APPROPRIATE EVENT ROUTINE
C
  GO TO (150, 160, 180, 170), NEXT
150 CALL ORDARV
  GO TO 140
160 CALL DEMAND
  GO TO 140
170 CALL EVALUS
  GO TO 140
180 CALL REPORT
190 CONTINUE
  WRITE(6,200)
200 FORMAT(' ')
  STOP
  END

```

```

SUBROUTINE INIT
  INTEGER AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,
  *SMALLS REAL AMINUS, APLUS, H, INCRMC, MDEMDT, PI,
  *SETUPC, TIME,TLEVNT, TNE(4), TORDC
  COMMON /MODEL/AMINUS, AMOUNT, APLUS, BIGS, H, INCRMC,
  *INITIL,INVLEV, MDEMDT, NEVNTS, NEXT, NMNTHS, PI,
  *SETUPC, SMALLS, TIME,TLEVNT, TNE, TORDC
C
C *** INITIALIZE THE SIMULATION CLOCK
C
  TIME = 0.
C
C *** INITIALIZE THE STATE VARIABLES.
C
  INVLEV=INITIL
  TLEVNT=0.
C
C *** INITIALIZE THE STATISTICAL COUNTERS.
C
  TORDC=0.
  APLUS=0.
  AMINUS=0.
C
C *** INITIALIZE THE EVENT LIST. SINCE NO ORDER IS
C *** OUTSTANDING, THE TIME OFF THE NEXT ORDER ARRIVAL
C *** OFF THE NEXT ORDER ARRIVAL IS SET TO 'INFINITY'.
C
  TNE(1)=1.E-30
  TNE(2)=EXPON(MDEMDT)
  TNE(3)=NMNTHS
  TNE(4)=0.
  RETURN
  END

SUBROUTINE TIMING
  INTEGER AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,
  *SMALLS REAL AMINUS, APLUS, H, INCRMC, MDEMDT, PI,
  *SETUPC, TIME,TLEVNT, TNE(4), TORDC
  REAL RMIN
  COMMON /MODEL/AMINUS, AMOUNT, APLUS, BIGS, H, INCRMC,
  *INITIL, INVLEV, MDEMDT, NEVNTS, NEXT, NMNTHS, PI,
  *SETUPC, SMALLS, TIME, TLEVNT, TNE, TORDC
  RMIN=1.E+29
  NEXT=0
C
C *** DETERMINE THE EVENT TYPE OF THE NEXT EVENT TO OCCUR.
C
  DO 10 I=1,NEVNTS
  IF (TNE(I).GE.RMIN) GO TO 10
  RMIN=TNE(I)
  NEXT=I
  10 CONTINUE
C

```

```

C *** IF THE EVENT LIST IS EMPTY (I.E. NEXT=0), STOP THE
C *** SIMULATION. OTHERWISE, ADVANCE THE SIMULATION CLCCK.
C
      IF (NEXT.GT.0) GO TO 30
      WRITE(6,20)
20  FORMAT(' ',5X,'EVENT LIST EMPTY')
      STOP
30  TIME=TNE(NEXT)
      RETURN
      END

      SUBROUTINE ORDARV
      INTEGER AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,
      *SMALLS REAL AMINUS, APLUS, H, INCRMC, MDEMDT, PI,
      *SETUPC, TIME,TLEVNT, TNE(4), TORDC
      COMMON /MODEL/AMINUS, AMOUNT, APLUS, BIGS, H, INCRMC,
      *INITIL,INVLEV, MDEMDT, NEVNTS, NEXT, NMNTHS, PI,
      *SETUPC, SMALLS, TIME,TLEVNT, TNE, TORDC
C
C *** UPDATE 'APLUS' AND 'AMINUS'
C
      CALL UPDATE
C
C *** INCREMENT THE INVENTORY LEVEL BY THE AMOUNT ORDERED.
C
      INVLEV=INVLEV+AMOUNT
C
C *** SINCE NO ORDER IS NOW OUTSTANDING, SET THE TIME OF
C *** THE NEXT ORDER ARRIVAL TO 'INFINITY'.
C
      TNE(1)=1.E-30
      RETURN
      END

      SUBROUTINE DEMAND
      INTEGER AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,
      *SMALLS
      INTEGER DSIZE
      REAL AMINUS, APLUS, H, INCRMC, MDEMDT, PI, SETUPC,
      *TIME,TLEVNT, TNE(4), TORDC
      COMMON /MODEL/AMINUS, AMOUNT, APLUS, BIGS, H, INCRMC,
      *INITIL,INVLEV, MDEMDT, NEVNTS, NEXT, NMNTHS, PI,
      *SETUPC, SMALLS, TIME,TLEVNT, TNE, TORDC
C
C *** UPDATE 'APLUS' AND 'AMINUS'
C
      CALL UPDATE
C
C *** GENERATE THE DEMAND SIZE
C
      DSIZE=RANDI(2)
C
C *** DECREMENT THE INVENTORY LEVEL BY THE DEMAND SIZE.

```

```

C
      INVLEV=INVLEV-DSIZE
C
C *** SCHEDULE THE NEXT DEMAND.
C
      TNE(2)=TIME+EXPON(MDEMDT)
      RETURN
      END

      SUBROUTINE EVALU8
      INTEGER AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,
      *SMALLS REAL AMINUS, APLUS, H, INCRMC, MDEMDT, PI,
      *SETUPC, TIME,TLEVNT, TNE(4), TORDC
      COMMON /MODEL/AMINUS, AMOUNT, APLUS, BIGS, H, INCRMC,
      *INITIL,INVLEV, MDEMDT, NEVNTS, NEXT, NMNTHS, PI,
      *SETUPC, SMALLS, TIME,TLEVNT, TNE, TORDC
C
C *** IF THE INVENTORY LEVEL IS LESS THAN 'SMALLS' PLACE AN
C *** ORDER FOR 'BIGS'-'INVLEV' ITEMS.
C
      IF(INVLEV.GE.SMALLS) GO TO 10
      AMOUNT=BIGS-INVLEV
      TORDC=TORDC+SETUPC+(INCRMC*AMOUNT)
C
C *** SCHEDULE THE ARRIVAL OF THE ORDER.
C
      TNE(1)=TIME+UNIFRM(0.5,1.0)
C
C *** SCHEDULE THE NEXT INVENTORY EVALUATION.
C
      10 TNE(4)=TIME+1.
      RETURN
      END

      SUBROUTINE REPORT
      INTEGER AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,
      *SMALLS REAL AMINUS, APLUS, H, INCRMC, MDEMDT, PI,
      *SETUPC, TIME,TLEVNT, TNE(4), TORDC
      REAL ACOST, AHLDC, AORDC, ASHRC
      COMMON /MODEL/AMINUS, AMOUNT, APLUS, BIGS, H, INCRMC,
      *INITIL,INVLEV, MDEMDT, NEVNTS, NEXT, NMNTHS, PI,
      *SETUPC, SMALLS, TIME,TLEVNT, TNE, TORDC
C
C *** UPDATE 'APLUS' AND 'AMINUS'
C
      CALL UPDATE
C
C *** COMPUTE ESTIMATES OF THE DESIRED MEASURES OF
C *** PERFORMANCE.
C
      AORDC=TORDC/NMNTHS
      AHLDC=H*(APLUS/NMNTHS)
      ASHRC=PI*(AMINUS/NMNTHS)

```

```

ACOST=AORDC-AHLDC+ASHRC
WRITE(6,10) SMALLS, BIGS, ACOST, AORDC, AHLDC, ASHRC
10 FORMAT(' ',5X,'(',I3,',',I3,')',8X,F6.2,16X,F6.2,19X,
*F6.2,19X,F6.2)
RETURN
END

SUBROUTINE UPDATE
INTEGER AMOUNT,BIGS,INITIL,INVLEV,NEVNTS,NEXT,NMNTHS,
*SMALLS REAL AMINUS, APLUS, H, INCRMC, MDEMDI, PI,
*SETUPC, TIME,TLEVNT, TNE(4), TORDC
REAL TSLE
COMMON /MODEL/AMINUS, AMOUNT, APLUS, BIGS, H, INCRMC,
*INITIL,INVLEV, MDEMDT, NEVNTS, NEXT, NMNTHS, PI,
*SETUPC, SMALLS, TIME,TLEVNT, TNE, TORDC
C
C *** COMPUTE THE TIME SINCE THE LAST EVENT WHICH CHANGED
C *** THE INVENTORY LEVEL.
C
TSLE=TIME-TLEVNT
TLEVNT=TIME
C
C *** DETERMINE WHETHER THE INVENTORY LEVEL DURING THE
C *** PREVIOUS INTERVAL WAS NEGATIVE, ZERO OR POSITIVE
C
IF(INVLEV) 10, 20, 30
C
C *** SINCE THE INVENTORY LEVEL DURING THE PREVIOUS
C *** INTERVAL WAS NEGATIVE, UPDATE 'AMINUS'.
C
10 AMINUS=AMINUS+(-INVLEV*TSLE)
C
C *** THE INVENTORY LEVEL DURING THE PREVIOUS INTERVAL WAS
C *** ZERO.
C
20 RETURN
C
C *** SINCE THE INVENTORY LEVEL DURING THE PREVIOUS
C *** INTERVAL WAS POSITIVE, UPDATE 'APLUS'.
C
30 APLUS=APLUS+(INVLEV*TSLE)
RETURN
END
FUNCTION RANUN(Z)
INTEGER A, P, Z, B15, B16, XHI, XALO, LEFTLO, FHI, K
DATA A/16807/, B15/32768/, B16/65536/, P/2147483647/
XHI =Z / B16
XALO = ( Z - XHI * B16 ) * A
LEFTLO = XALO / B16
FHI = XHI * A + LEFTLO
K = FHI / B15
Z = ((( XALO - LEFTLO*B16) - P ) + (FHI - K * B15 ) *
*B16 ) + K

```



```

IF ( Z.LT. 0) Z = Z+ P
RANUN = FLOAT ( Z) * 4.656612875E-10
RETURN
END
FUNCTION EXPON(RMEAN)
REAL RMEAN, U
C
C *** GENERATE A U(0,1) RANDOM VARIABLE. THE FORM OF THIS
C *** STATEMENT DEPENDS ON THE COMPUTER USED.
C
U=RANUN(60000)
C
C *** GENERATE AN EXPONENTIAL RANDOM VARIABLE WITH MEAN
C *** RMEAN.
C
EXPON=-RMEAN * ALOG(U)
RETURN
END
FUNCTION RANDI(Z)
INTEGER I, N1
REAL U
COMMON /RANDOM/ NVALUE, PROBD(25)
C
C *** GENERATE A U(0,1) RANDOM VARIABLE. THE FORM OF THIS
C *** STATEMENT DEPENDS ON THE COMPUTER USED.
C
U=RANUN(60000)
C
C *** GENERATE A RANDOM INTEGER BETWEEN 1 AND NVALUE IN
C *** ACCORDANCE WITH DISTRIBUTION FUNCTION 'PROBD'.
C
N1=NVALUE-1
DO 10 I=1, N1
IF(U.GE.PROBD(I)) GO TO 10
RANDI=I
RETURN
10 CONTINUE
RANDI=NVALUE
RETURN
END
FUNCTION UNIFORM(A, B)
REAL A,B,U
C
C *** GENERATE A U(0,1) RANDOM VARIABLE. THE FORM OF THIS
C *** STATEMENT DEPENDS ON THE COMPUTER USED.
C
U=RANUN(60000)
C
C *** GENERATE A U(A, B) RANDOM VARIABLE.
C
UNIFORM=A+(U*(B-A))
RETURN
END

```

İncelenen (s,S) stok sistemi örneğinin simülasyonu için hazırlanan bu FORTRAN 77 programı, PC ortamında çalıştırıldıktan birkaç saniye sonra Tablo 4'te verilen sonuçlar elde edilmiştir. Sonuç değerleri incelendiğinde, karşılaştırılan alternatif stok politikaları arasında aylık ortalama toplam maliyeti en düşük olan stok politikasının,  $s = 20$  ve  $S = 80$  olarak belirlenmiş (20,80) stok politikası olduğu görülmektedir. Bununla birlikte, simülasyon uzunluğunun firmanın planladığı 10 yıllık sabit bir süre için olması bağlamında; gerçekten istenen şeyin, her bir stok politikası için sadece ilk 120 aylık dönemde beklenen aylık ortalama toplam maliyetini tahmin etmek olduğu dikkate alınmalıdır. Tablo 4'teki sayılar bu beklenen değerlerin tahminleridir ve her bir tahmin sadece bir tek simülasyon çalışmasına göredir. Eğer  $U(0, 1)$  aralığında farklı tesadüfi değişkenler kullanarak bu simülasyon çalışması tekrarlanırsa, elde edilecek tahminler Tablo 4'tekilerden oldukça farklı olabilir.

Tablo 4: Tek Ürünlü (s,S) Stok Sistemi için Hazırlanan FORTRAN 77 Programının Bir Defa Çalıştırılması Sonucunda Elde Edilen Sonuç Raporu

SINGLE-PRODUCT INVENTORY SYSTEM				
INITIAL INVENTORY LEVEL	60 ITEMS			
NUMBER OF DEMAND SIZES	4			
DISTRIBUTION FUNCTION OF DEMAND SIZES	.167 .500 .833 1.00			
MEAN INTERDEMAND TIME	.10 MONTHS			
LENGHT OF THE SIMULATION	120 MONTHS			
K = 32.0    I = 3.0    H = 1.0    PI = 5.0				
POLICY	AVERAGE COST	AVERAGE ORDERING COST	AVERAGE HOLDING COST	AVERAGE SHORTAGE COST
(20, 40)	126.52	97.58	9.53	19.41
(20, 60)	126.44	89.48	18.16	18.80
(20, 80)	122.83	83.11	26.84	12.88
(20,100)	123.84	79.60	35.42	8.92
(40, 60)	125.32	97.03	25.74	2.56
(40, 80)	127.44	91.09	34.44	1.91
(40,100)	132.40	86.13	45.13	1.13
(60, 80)	147.63	104.10	43.32	.20
(60,100)	141.83	88.78	53.01	.03

### III. Sonuç

Yukarıda yapılan çözüm değerlendirmesi, ele alınan stok problemi ile ilgili sadece bir tek simülasyon çalışmasına göredir. Doğal olarak, karşılaştırılan alternatif stok politikalarından hangisinin en uygun olacağını tahmin edebilmek için daha fazla simülasyon çalışması gereklidir. Bununla ilgili olarak Law ve Kelton (1990: 280-308) tarafından, iyi bir tahmin için ne kadar simülasyon çalışması yapılması gerektiği hususunda, oldukça geniş açıklayıcı bilgiler verilmiştir. Ayrıca, simülasyonla ilgili diğer literatürde de bu konu ile ilgili oldukça geniş bilgiler bulmak mümkündür.

Bu ve benzeri simülasyon örnekleri FORTRAN dilinden başka daha önce bahsedilen çeşitli simülasyon dilleri ile de yazılabilir ve daha önce verilen simülasyon yazılım programlarından istifade ederek de çözüm tablosu elde edilebilir.

Yapılan çalışma, bir (s,S) stok probleminin simülasyon yardımı ile bilgisayar ortamında nasıl çözülebileceğini göstermek amacını taşımakta olup, incelenen örnek problem sayesinde konu hakkında okuyucuya genel bir yaklaşım sunmak hedeflenmiştir. Daha detaylı bilgi almak isteyenler için de, çalışmada konu ile ilgili örnek bazı kaynaklara göndermeler yapılmıştır.

### Kaynaklar

- Anupindi, R., Tayur, S.(1998), "Managing Stochastic Multiproduct Systems: Model, Measures, and Analysis", *Operation Research*, 46(3), ss. S98-S111.
- Baykoç, Ö. F., Erol, S. (1998), "Simulation Modeling and Analysis of JIT Production System", *International Journal of Production Economics*, 55, ss.203-212
- Bookbinder, J.H., Çakanyıldırım, M.(1999), "Random Lead Times and Expedited Orders in (Q,r) Inventory Systems", *European Journal of Operational Research*, 115, ss. 300-313.
- Chase, Richard B., Aquilano, N.J. (1995), *Production and Operation Management Manufacturing and Service*, Irwin Inc., New York.
- Ehrhardt, R.(1984), "(s,S) Policies for Dynamic Inventory Model with Stochastic Lead Times", *Operation Research*, 32(1), ss. 121-131.
- Gençyılmaz, Güneş (1988), *Stok Sistemlerinin Yönetimi-1*, İ.Ü. İşletme Fak. Yayını, No:204, İstanbul.
- Ishigaki, T., Sawaki, K. (1991), "On The (s,S) Policy with Fixed Inventory Holding and Shortage Costs", *Journal of the Operation Research Society of Japan*, 34(1), ss. 47-57
- Kobu, B. (1999), *Üretim Yönetimi*, İ.Ü. İşletme Fak. Yayını, No:04, İstanbul.
- Law, A. M., Kelton, W. D. (1990), *Simulation Modeling and Analysis*, McGraw-Hill, New York.

- Liu, L., Yang, T.(1999), "An (s,S) Random Lifetime Inventory Model with a Positive Lead Time", *European Journal of Operational Research*, 113, ss. 52-63.
- Payne, J. A. (1982), *Introduction to Simulation: Programming Techniques and Method of Analysis*, McGraw-Hill, New York.
- Savsar, M. (1997), "Simulation Analysis of a Pull-push System for an Elektronik Assembly Line", *International Journal of Production Economics*, 51, ss.205-214
- Shaoxiang, C., Lambrecht, M.(1996), "X-Y Band and Modified (s,S) Policy", *Operation Research*, 44(6), ss. 1013-1019.
- Silver, E. A. (1985), *Decision Systems for Inventory Management end Production Planning*, J. Wiley, New York.
- Taha, H. A. (2000), *Yöneylem Araştırması*, 6. Basımdan Çeviri, Çev: Ş. Alp Baray ve Şakir Esnaf, Literatür Yayınları, İstanbul..
- Zheng, Y., Federgruen, A. (1991), "Finding Optimal (s,S) Policies is About as Evaluating a Single Policy", *Operation Research*, 39(4), ss. 654-665.
- The Consumate Design Center (2002), <http://www.tcdc.com./dsofts/software/simlat.htm>, date: 11.11.2002