T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DİNAMİK PROGRAMLAMA ve İSTATİSTİKSEL BAZLI UYGULAMALAR

EYLÜL YILDIRIM

YÜKSEK LİSANS TEZİ
İSTATİSTİK ANABİLİM DALI
İSTATİSTİK PROGRAMI

DANIŞMAN
DOÇ. DR. ATIF A. EVREN

İSTANBUL, 2016

T.C.
YILDIZ TEKNİK ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

DİNAMİK PROGRAMLAMA ve İSTATİSTİKSEL BAZLI UYGULAMALAR

Eylül YILDIRIM tarafından hazırlanan tez çalışması 21.04.2016 tarihinde aşağıdaki jüri tarafından Yıldız Teknik Üniversitesi Fen Bilimleri İstatistik Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Tez Danışmanı

Doç. Dr. Atıf A. EVREN
Yıldız Teknik Üniversitesi

Jüri Üyeleri

Prof. Dr. A. Hakan BÜYÜKLÜ
Yıldız Teknik Üniversitesi _____

Doç. Dr. Atıf A. EVREN
Yıldız Teknik Üniversitesi _____

Yrd. Doç. Dr. Yavuz ILGAZ
İstanbul Üniversitesi _____

ÖNSÖZ

Yaşamın her alanında var olan karar verme süreci, neticede istenilene ulaşma ya da ulaşamamayı belirlemesi yönünden oldukça önemli bir yer teşkil etmektedir. Bu süreçte, seçeneklerin irdelenmesi yapılacak seçimde belirleyici rol oynamaktadır. Çok aşamalı bir şekilde ifadesi mümkün olan durumlarda, dinamik programlama, en iyi karara varmada kullanılan etkin bir yöntemdir. Dönüşüm denklemleri ile farklı problem türlerine çözüm sağlamaktadır. Böylece işletmelerin; bütçeleme, kargo yükleme, en kısa yol, yatırım, stok, işgücü gibi birbirinden farklı konularda sorunlarına en çok faydayı almayı sağlayacak kararı almalarına sebep olmaktadır.

Günlük hayatta da sıkça karşılaşabileceğimiz uygulama alanlarına, etkin ve kolay çözümler sunmasını dinamik programlamayı tez konusu seçmemdeki neden olarak gösterebilirim.

Tez çalışmamda değerli bilgi ve deneyimleriyle beni yönlendiren danışman hocam Sayın Doç. Dr. Atıf A.EVREN'e çalışmam süresince göstermiş olduğu sabır ve anlayış için teşekkürlerimi sunarım. Destek ve yardımlarını esirgemeyen değerli aileme ve sözlüme teşekkür ederim.

Nisan 2016,

Eylül YILDIRIM

İÇİNDEKİLER

Sayfa

SİMGE LİSTESİ vii

KISALTMA LİSTESİ vii

ŞEKİL LİSTESİ viii

ÇİZELGE LİSTESİ ix

ÖZET xi

ABSTRACT xiii

BÖLÜM 1

GİRİŞ 1

1.1 Literatür Özeti 1
1.2 Tezin Amacı 1
1.3 Hipotez 2

BÖLÜM 2

YÖNEYLEM ARAŞTIRMASININ TARİHSEL GELİŞİMİ VE TANIMI 3

2.1 Yöneylem Araştırması Tarihçesi 2
Yöneylem Araştırması Tanımı 4
2.3 Yöneylem Araştırması Kapsamında İncelenen Modeller 5
2.3.1 Doğrusal Programlama 6
2.3.2 Doğrusal Olmayan Programlama 7

BÖLÜM 3

DİNAMİK PROGRAMLAMA 8

3.1 Dinamik Programlamanın Doğuşu 8
3.2 Dinamik Programlamanın Tanımı 9
3.3 Dinamik Programlamanın Çözüm Yöntemleri 10
3.4 Dinamik Programlama Çeşitleri 12
3.5 Dinamik Programlamanın Yapısı 12

3.5.1 Dinamik Programlamanın Temel Kavramları 12
3.5.2 Dinamik Programlama Formülasyonu 13
3.6 Dinamik Programlama için Geçerlilik Özellikleri 15
3.7 Dinamik Programlamanın Avantaj ve Dezavantajları 15

BÖLÜM 4

YAYGIN DİNAMİK PROGRAMLAMA UYGULAMALARI 17

4.1 En Kısa Yol Problemi 18
4.2 Kargo Yükleme (Knapsack-Sırt Çantası) Modeli 23
4.3 Stokastik Dinamik Programlama Problemi 27
3.4.1 Şans Oyunu 27
4.4 Doğrusal Programlama Modelinin Dinamik Programlama ile Çözümü 30
4.5 Doğrusal Olmayan Modelin Dinamik Programlama ile Çözümü 34

BÖLÜM 5

DİNAMİK PROGRAMLAMA İLE BİR İŞGÜCÜ MİKTARI MODELİ UYGULAMASI 38

5.1 Problemin Çözümü 39

BÖLÜM 6

SONUÇ VE ÖNERİLER 54

KAYNAKLAR 56

EK - A

EK AYNI PROBLEMİN LINGO İLE ÇÖZÜMÜ 58

ÖZGEÇMİŞ 60

SİMGE LİSTESİ

C_1	fazla işçi bulundurma maliyetini,
C_2	yeni bir işçi alma durumundaki maliyeti
C_i	n. aşamadaki getiri değeri
$f_n(s_n,x_n)$	n. aşamadaki s_n durumu ve x_n kararı altındaki getiri,
$f_n^*(s_n)$	optimal getiri değeri,
N	aşama sayısı,
n	geçerli aşama,
p_i	geçiş olasılığı
s_n	n. aşama için geçerli durum değeri,
x_n	n. aşama için karar değişkeni,
x_n^*	optimal x_n değeri,

KISALTMA LİSTESİ

br birim

CP Kritik yol üzerindeki toplam süre rassal değişkeni

opt optimum

Pb Para birimi

PERT Program evaluation and review tecnique (Program değerlendirme ve gözden geçirme tekniği)

ŞEKİL LİSTESİ

Sayfa

Şekil 3. 1 Deteministik dinamik programlamanın basit yapısı 14
Şekil 3. 2 Stokastik (olasılıklı) dinamik programlamanın basit yapısı 15
Şekil 4. 1 En kısa yol problem ağı 18
Şekil 4. 2 LINGO maksimizasyon modeli çözüm raporu 26
Şekil 5. 1 Faaliyet ağı şeması 45
Şekil 5. 2 Erken ve geç başlama zaman gösterimi 46

ÇİZELGE LİSTESİ

Sayfa

Çizelge 3. 1 Dinamik Programlamanın tablosal gösterimi 14
Çizelge 4. 1 Dördüncü aşamada en kısa yol 20
Çizelge 4. 2 Üçüncü aşamada en kısa yol 20
Çizelge 4. 3 İkinci aşamada en kısa yol 20
Çizelge 4. 4 Birinci aşamada en kısa yol 21
Çizelge 4. 5 Normal dağılıma uygun veri setinden çekilen değerlere göre en kısa yol 21
Çizelge 4. 6 Tek-düze dağılıma uygun veri setinden çekilen değerlere göre en kısa yol 22
Çizelge 4. 7 Çanta ağırlık durumu 24
Çizelge 4. 8 Birinci aşama 25
Çizelge 4. 9 İkinci aşama 25
Çizelge 4. 10 Üçüncü aşma 25
Çizelge 4. 11 Beşinci aşama 28
Çizelge 4. 12 Dördüncü aşama 28
Çizelge 4. 13 Üçüncü aşama 29
Çizelge 4. 14 İkinci aşama 29
Çizelge 4. 15 İkinci aşama 31
Çizelge 4. 16 Simpleks yöntemi ile çözümü 32
Çizelge 4. 17 Simpleks yöntemi ile son değer 33
Çizelge 4. 18 İkinci aşama 33
Çizelge 4. 19 Simpleks yöntemi ile son değer 33
Çizelge 4. 20 İkinci aşama 34
Çizelge 5. 1 (i) ve (i-1) dönemlerine ait hesaplanacak işçi sayıları 40
Çizelge 5. 2 Sekizinci aşama 40
Çizelge 5. 3 Yedinci aşama 41
Çizelge 5. 4 Altıncı aşama 41
Çizelge 5. 5 Beşinci aşama 42
Çizelge 5. 6 Dördüncü aşama 42
Çizelge 5. 7 Üçüncü aşama 43
Çizelge 5. 8 İkinci aşama 43
Çizelge 5. 9 Birinci aşama 44
Çizelge 5. 10 Fabrika faaliyet çizelgesi 45
Çizelge 5. 11 Toplam boşluk miktarları 46
Çizelge 5. 12 Faaliyet süreleri beklenen değerleri ve varyansları 47

Çizelge 5. 13 İş makinesi bakımı yapılmama ve yapılma durumlarında p_{ij},r_{ij} değerleri 48
Çizelge 5. 14 Durumlar bazında bakım yapılmama ve yapılma faaliyetlerine göre getiri 50
Çizelge 5. 15 Üçüncü aşamada elde edilen getiri değerleri 50
Çizelge 5. 15 İkinci aşamada elde edilen getiri değerleri 50
Çizelge 5. 16 Birinci aşamada elde edilen getiri değerleri 51

ÖZET

DİNAMİK PROGRAMLAMA ve İSTATİSTİKSEL BAZLI UYGULAMALAR

Eylül YILDIRIM

İstatistik Anabilim Dalı

Yüksek Lisans Tezi

Tez Danışmanı: Doç. Dr. Atıf A. EVREN

"Kıt kaynakların etkin kullanımı" felsefesini esas alan yöneylem araştırması kapsamında incelenen modellerden biri olan dinamik programlama, oldukça geniş bir yelpazede uygulama alanı içermektedir. Problemi daha küçük alt bölümlere ayırma yolunu seçmesi ve hem deterministik hem de olasılıklı modellerde çözülebilme imkânı sağlaması dinamik programlamanın dikkat çeken özelliklerindendir.

Tez çalışmamda, yöneylem araştırması tekniği ışığı altında çözümler sunan dinamik programlamak onusunda çalışma yapılmıştır. Bu çalışma altı bölümden oluşmaktadır. Birinci bölüm, literature özeti, tezin amacı ve hipotez başlıklarından oluşmaktadr. İkinci bölümde, yöneylem araştırmasının tarihsel gelişimi ve tanımı yer almaktadır. Üçüncü bölümde, dinamik programlama konusu; yöntemin doğuşu, tanımı, çözüm yöntemleri, çeşitleri, yapısı-formülasyonu, geçerlik özellikleri, avantaj ve dezavantajları ile detaylandırılmıştır. Dördüncü bölümde, literatürde sıklıkla karşılaşılan dinamik programlama uygulamaları deterministic ve olasılıklı modellerle incelenmiştir. Beşinci bölümde ise gemi inşa faaliyetinde bulunan bir fabrikada işçi alımı stratejisi dinamik programlama yöntemi ile değerlendirilmiştir. Altıncı bölümde yapılan araştırma ve uygulamaların sonuçları ile yorum-öneriler yer almaktadır. Dinamik programlama çözüm sonuçları Excel ve LINGO programları kullanılarak desteklenmiştir.

AnahtarKelimeler: dinamik programlama, deterministik dinamik programlama, olasılıklı dinamik programlama.

ABSTRACT

DYNAMIC PROGRAMMING and STATISTICAL PRACTICES

Eylül YILDIRIM

Department of Statistics

Msc. Thesis

Thesis Adviser: Assoc. Prof. Dr. Atıf A. EVREN

Dynamic programming is a branch of operation research which bases on the methodology of 'effective usage of scarce sources'. It separates the problems into small parts. Also, the capability of solving both deterministic and stochastic models is a critical feature of dynamic programming.

In my thesis, dynamic programming that is a part of operation research methodology is investigated. The thesis has six parts. The first part includes summary of literature, the aim of thesis and the hypothesis. The historical evolution and definition of operation research is given in the second part. In the third part, I discuss dynamic programming details; birth of the method, definitions, procedures of solutions, kinds, formulations, attiributes of validity, advantages and disadvantages. In the fourth part, deterministic and stochastic models of dynamic programming that are mostly used in literature are figured out. In the fifth part, the strategy of employing workers in a factory of vessel construction is analyzed with dynamic programming. The last part includes the results of investigation with comments and suggestions. The solutions are provided with the help of Excel and LINGO programs.

KeyWords: dynamic programming, deterministic dynamic programming, stochastic dynamic programming.

BÖLÜM 1

GİRİŞ

1.1 Literatür Özeti

Stuart E.Dreyfus, dinamik programlama adını ve yöntemin ortaya çıkışını Bellman'ın bıraktığı makale-çalışmalarından ve otobiyografisinde yazılanlarla öğrendiğimizi 'Richard Bellman On TheBirth Of Dynamic Programming (Richard Bellman Dinamik Programlamanın Doğuşu)' adlı yazısında ele almıştır.1949 yazında çok aşamalı karar süreçleri ile tanışması yöntemin başlangıcına sebep olduğu da belirtilmektedir. Kuram olarak Abraham Wald'ın Sıralama Analizi Kuramı ile ilişkilendirildiği literatürde yer almaktadır.Sonraki dönemlerde ise HerbertSimon'ın belirsizlik altında ve kuadratik fonksiyonla dinamik programlama çalışması, S.Karlin ve H.N. Shapiro'nun karar süreçleri ve fonksiyonel denklemleri çalışmaları literatürde yer alan açıklayıcı ve yöntemi geliştirici çalışmalar arasındadır. Yöneylem çalışması kapsamında hem doğrusal hem de doğrusal olmayan modellerde geniş uygulama alanına sahip olduğu görülmektedir. Birçok uygulama Y.Turgay'ın dinamik programlama çalışmasında da örneklendirilmektedir. Zaman faktörünü dikkate alarak, belirsizlik altındaki süreçlere ve yönetim alanında çözümler sunan bir tekniktir.

1.2 Tezin Amacı

Dinamik programlama yöntemini deterministik ve olasılıklı modellerde inceleyerek uygulanabilirliği üzerine çalışmalar yapmak, istatistiksel yöntemleri kullanarak çözümlerin yorum gücünü arttırmak ve bir uygulama sahası olarak seçilen fabrikada işgücü istihdamı için belirlenen yönetici stratejisini değerlendirerek, maliyet azaltıcı

alternatif sunabilmek, istatistiksel analiz yöntemi kullanarak kıyaslama yapılan rakip fabrika ile aradaki farkı netleştirerek, iyileştirmeye yönelik öneriler sunabilmektir.

1.3 Hipotez

Deterministik modellerde olduğu gibi olasılıklı modellerde de dinamik programlama yönteminin kolay uygulanabilirliğini, değişkenlerin bilinmesi ve olasılıklı olması durumunda istikrarlı öneriler sağlayabildiğini vurgulamaktır.

BÖLÜM 2

YÖNEYLEM ARAŞTIRMASININ TARİHSEL GELİŞİMİ VE TANIMI

2.1 Yöneylem Araştırması Tarihçesi

II. Dünya Savaşı zamanında İngiliz ordusunda Alman hava kuvvetlerinin saldırılarına karşı koymak amacıyla görevlendirilen bazı bilim adamı ve mühendislerin çalışmaları 'Yöneylem (harekât) Araştırması' olarak adlandırılmıştır. Yöneylem araştırması kapsamında, Alman hava saldırılarını önceden tespitte yetersiz kalan mevcut radar ağının yerinin yeniden belirlenmesi ve erken uyarı sisteminin getirilmesi sağlanmıştır. Yapılan istatistiki çalışmalar ile gemi kaybını en aza indirmek için optimal konvoy ve eskort büyüklüğü tespit edilmiştir. Alman denizaltılarını etkisiz hale getirebilmede sualtı bombaları için optimal patlama derinliğinin bulunması belirlenen bir matematiksel model ile sağlanmıştır [1].

Sonraki yıllarda G.B. Dantzig tarafından geliştirilen doğrusal programlama yaklaşımı ve Simpleks çözüm yöntemi A.B.D. 'nin askeri faaliyetlerinin planlamasında kullanılmıştır. Bu yaklaşım iktisadi unsurlarda Robert Dorfman tarafından uyarlanmıştır. Daha sonraki çalışmalarda iktisatçı ve matematikçiler 'Dualite Teorisini' geliştirmişlerdir. 1950'lerden sonra ise R.Bellman dinamik programlama ve H.Kuhn ile A. Tucker doğrusal olmayan programlama modellerini geliştirmişlerdir[2].

Türkiye'de ise oluşturulan ilk yöneylem araştırması birimi 1954'te İlmi İstişare Kurulu Müdürlüğü'dür. 1958'de ARGE olarak isim değişikliğine gidilen ve 1970 yılına kadar Genel Kurmay Başkanlığı'na bağlı olarak çalışmalarını sürdüren kurum, sonrasında Milli Savunma Bakanlığı'na bağlı olarak faaliyetlerine devam etmiştir. Sivil kurumlarda ise ilk olarak TÜBİTAK bünyesinde, eğitim alanında ilk uygulamalar ise İstanbul Teknik Üniversitesi ve Ortadoğu Teknik Üniversitesi'nde başlatılmış, Kara Harp Okulu müfredatında da yer alan yöneylem araştırması sonraki yıllarda lisans, yüksek lisans ve doktora düzeyinde diğer üniversitelerde de yerini almıştır [2].

Savaş dönemi başlatılan yöneylem araştırması tekniği, sonraki dönemlerde üretim, iktisat, işletme, finans alanları gibi oldukça geniş bir yelpazede kullanılmış ve geliştirilmiştir.

2.2 Yöneylem Araştırması Tanımı

Yöneylem araştırması en genel ve yaygın olarak 'kıt kaynakların etkin kullanımı' şeklinde ifade edilmektedir. Etkin kullanım kavramı optimizasyona yönlendirir. Dolayısıyla belirli bir amaç doğrultusunda eldeki kaynakları en verimli şekilde kullanarak problem çözümünü sağlamaktadır.

Yöneylem araştırması sistemlerin karşılaştıkları problemlerde, disiplinler arası bir ekiple, bilimsel metotları kullanarak ve problemin kontrol edilebilir unsurları ile ilgili alternatifleri değerlendirmek suretiyle optimal çözümü bulmayı amaçlar. Yöneylem araştırması gerçek hayattan kaynaklanan ve çoğunlukla sınırlı kaynakların paylaştırıldığı deterministik ve olasılıklı problemlerin modellenmesi ve optimal kararın verilmesi ile ilgilenir[3].

Yöneylem araştırması; rakamlara dönüştürülmüş bir sağduyu, yönetim bilimi, karar analizi, tasarım analizi, işletmeler için araştırma tekniği ve mevcut kaynaklardan en üst düzeyde yararlanarak bilimsel tekniklerin problem çözümünü gerçekleştirmede kullanılması literatürde yer alan tanımlardır.

Bu tanımlardan yola çıkarak; karar vermede, eldeki kaynakları en iyi kullanarak sınırlamaların zorluğunu en aza indiren ve bilimsel bir metot olan yöneylem araştırması tekniği kullanımı etkinlik sağladığı söylenebilir.

Yöneylem araştırması yöntemi, problemin tanımlanması, matematiksel modelin belirlenmesi, modelin çözümünün yapılması, modelin test edilmesi ve değer tahmininde kullanılması, çözümün uygulanması adımlarını izlemektedir.

Yöneylem araştırması savaş zamanı ortaya koyduğu savunma stratejileriyle başlamakla birlikte, günlük yaşamdan süreçlerde ve stratejik kararlar almada da en uygun sonuca ulaşmaya yardımcı alternatifler sunmaktadır.

2.2 Yöneylem Araştırması Kapsamında İncelenen Modeller

Deterministik ve stokastik (olasılıksal) modeller olarak ayrılır. Deterministik modeller için;

- Doğrusal Programlama,
    - Simpleks Yöntem,
    - Ulaştırma Modelleri,
    - Şebeke Modelleri,
    - Tamsayılı Programlama,
    - Düzeltilmiş Simpleks Yöntem,
    - İki-Faz Yönetimi,
    - Parametrik Programlama,
- Doğrusal Olmayan Programlama,
    - Klasik Optimizasyon Kuramı,
    - Lagrange Çarpanları Yöntemi,
    - Kuhn-Tucker Yöntemi,
    - Kuadratik Programlama,
    - Ayrılabilir Programlama,
- Oyun Kuramı,
- Envanter Modelleri,
- *Dinamik Programlama,*
- Çok Amaçlı Karar Verme,

stokastik (olasılıksal) modeller için ise;

- Belirsizlik Altında Karar Verme,
- Oyun Kuramı,
- Markov Zincirleri,
- *Dinamik Programlama,*
- Kuyruk Kuramı,
- Benzetim Modelleri,
- Tahmin Modelleri,
- Çok Amaçlı Karar Verme,

şeklinde sınıflandırılır.

2.3.1 Doğrusal Programlama

Kar maksimizasyonu ya da maliyet minimizasyonu şeklinde olan belirli bir amaca, belirli kısıtlar altında, kıt kaynakların en verimli şekilde kullanılmasını sağlayarak oluşturulan matematiksel yöntem doğrusal programlama adını alır.

$$Z = CX \tag{2.1}$$

$$AX \leq B$$

$$x \geq 0$$

$$Z = CX = [c1, c2, \dots, cn][x1, x2, \dots, xn]$$

Gerçek bir sistemi matematiksel ifade etmede bilgi kaybını en aza indirmek için bazı varsayımların sağlanması gerekmektedir. Doğrusallık varsayımı; her bir karar değişkeninin amaç fonksiyonuna katkısının, diğer karar değişkenlerinin değerlerinden farklı olarak o karar değişkenin değeri ile orantılı olmasını gösterir. Toplanabilirlik varsayımı; karar değişkenlerinin aldıkları değere göre oluşan katkıların toplanabilmesini ifade ederken doğrusallık varsayımının doğal bir sonucu olmaktadır. Bölünebilirlik varsayımı; karar değişkenlerinin sürekli değerler alabilmesidir. Belirlilik; doğrusal programlama modelinde amaç fonksiyona katkıların, kullanılan kaynak miktarlarının ve kıt kaynakların mevcut miktarlarının önceden bilinmesi yani parametrelerin bilinmesini gösterir.

2.3.2 Doğrusal Olmayan Programlama

Amaç fonksiyonu ya da kısıtları doğrusal olmadığı genel durumlar doğrusal olmayan programlama yöntemleri ile incelenir. $g_i(x) \leq b_i$ kısıtları altında $z = f(x)$ doğrusal olmayan fonksiyonu optimum yapan $x= (x_1, x_2,..,x_n)$ vektörü bulunur.Doğrusal olmayan programlamada fonksiyonun konveks ya da konkav oluşuna göre işlem yapılmaktadır. Gerçekte sistemleri doğrusal programlama modelleriyle çözmek her zaman mümkün olmamaktadır. Endüstri ve işletmelerde karşılaşılan problemlere ait koşulların veya amaç fonksiyonunun ya da her ikisinin doğrusal olmaması, durumu daha zor ve karmaşık hale getirmektedir. Doğrusal olmayan yöntemde simpleks yöntem gibi tek bir algoritma ile bütün problemlerin optimal kararına varılamaz.

BÖLÜM 3

DİNAMİK PROGRAMLAMA

3.1 Dinamik Programlamanın Doğuşu

Dinamik programlama, 1950-1960 yılları arasında geliştirilen yöneylem analizi alanındaki tekniklerden biridir. Dinamik programlama kuramı, Abraham Wald'ın Sıralama Analizi Kuramı –Statistical Sequential Analysis- ile ilişkilendirilmektedir. Richard Ernest Bellman tarafından geliştirilmiş ve isimlendirilmiştir [5].

Bellman, 1949 yazında RAND adında Santa Monica'da eğitim, sağlık, hukuk alanlarında strateji geliştiren bir araştırma-geliştirme kuruluşunda göreve başlamıştır. Burada kendisine önerilen çok aşamalı karar verme süreçleri ile ilgili çalışmaya karar verir. Dinamik programlamanın temellerini oluşturan çalışmaları neticesinde söz konusu yöneylem araştırması tekniklerinden olan çalışmanın ismini verirken planlama kelimesini birçok bakımdan olumsuz olarak değerlendirmiştir. Süreci en iyi ifade için dinamik ve programlama kelimelerinin kullanılmasını otobiyografisinde belirtmektedir [6].

Ardışık sistemlerde bir sonraki adımda en iyi kararı vermeyi sağlayan problem çözme sürecini tarif etmek için kullanılan dinamik programlama, 1953'lerde şimdiki kullanımı olan büyük kararlar içinde küçük alt kararlar alma yöntemi haliyle Bellman eşitliği şeklinde literatürde yerini almaktadır. Uygulamada ise en iyi örneklerden birini Andrew Vazsonyi vermektedir(Scientific Programming in Business andIndustry) [5].

Richard Bellman, dinamik programlama deyimini dönüşüm fonksiyonlarının kullanıldığı karar problemlerinin deterministik ve stokastik türlerini birleştirerek oluşturduğu yaklaşımı adlandırmak için kullanmıştır. Çok aşamalı karar süreçleri ile de

bağlantılı olması bu adı alma nedenlerinden biridir [6]. Harvard, UCLA, Pittsburgh ve daha birçok üniversite konu büyük ilgi görmüştür, 1958 yılında Boston'daki yöneylem araştırması yıllık toplantısının ana konusu olmuştur.

Bellman'ın optimizasyon prensibi Pontryagin'in maksimizasyon prensibi aynı problemlere farklı yaklaşım getirmiştir. Pontryagin maksimizasyon prensibinde, herhangi bir optimum için gerekli şartları sağlarken, Bellman dinamik programlama yönteminde belirli bir optimuma ulaşmada gerekli algoritmayı sağlamaktadır. Optimalite prensibinde 'Optimal bir politikanın şöyle bir özelliği vardır; ilk durum ve karar ne olursa olsun, geriye kalan kararlar, ilk kararın ortaya çıkarttığı durumu da dikkate alarak, belirli bir optimal politika kurmak zorundadır.' şeklinde ifade etmektedir. Bu kurama göre, ardı sıra gelen iki adım arasında kurulan ilişki en uygun değeri verecek biçimdedir.

R.Bellman ile başlanan dinamik programlama çalışmalarına Rutherford Aris, Stuart E. Dreyfus, Averill M. Law, Eric V. Denardo, SvenDano, Robert E. Larson, Reginald Lee, George L. Nemhauser, A. Kaufman, R. Cruon, L.G. Mitten ve D.J. White katkıda bulunan isimlerdir.

3.2 Dinamik Programlamanın Tanımı

Dinamik programlama birden çok işlemi sıralı elde eden özel bir planlama yöntemi olarak yöneylem araştırmasının optimumlaştırma yöntemleri arasında yer almaktadır.

Dinamik programlama, problemi aşamalar halinde çözer ve her bir aşamada en uygun çözümü bulmaya çalışır. Farklı aşamalardaki hesaplamalar, sonuçta optimum bir çözümün elde edildiği yinelenen hesaplamalarla ilişkilendirilerek elde edilir[7].

En iyi çözüme ulaşmada birçok matematiksel programlama yöntemlerinde aşamalara ayırma yöntemine gidilmektedir. Fakat dinamik programlama yineleme yöntemini kullanır. Sıralı ilişki durumu söz konusu olduğundan aşamalar tek başına çözüm oluşturmazken, optimum çözüme ulaşacak bilgiyi içermektedir[8].

Sıralı kararların alındığı problemlerin içinde optimal bir stratejinin bulunduğu çözüme ulaşmada dinamik programlama modeli kullanılır.

Optimizasyon modelleri felsefesi ile yapılmak istenen iş, karar değişkenlerinin belirli bir amaç fonksiyonu maksimize ya da minimize edecek değerlerin bulunmasıdır.

Dinamik bir modelde sorun genellikle bir değişim kalıbı içinde optimal politikalarının ortaya çıkarılmasını içerir.

Dinamik programlama, doğrusallık şartı aramamaktadır. Belirlenen problemi bir bütün sistem olarak kabul eder. Sistem parçaları ardışık işlem görür ve ardışık iki işlem arasında fonksiyonel bağlantı kurulur [9].

Dinamik programlama, büyük ve çok sayıda karar değişkeni olan sorunları, ardışık küçük sorunlara bölerek çözmek için geliştirilmiş bir yöntemdir. Yöntemin birçok uygulamasında zaman içinde ardışık kararlar bulunması yani modelde zaman değişkeninin açık olarak ele alınması, yönteme dinamik adının verilmesine neden olmaktadır. Dinamik programlamada her alt sorun ayrı ayrı izlenerek ele alınır ve sonuçta tüm sorun optimal olacak biçimde çözülür. Başka bir ifadeyle, dinamik programlama 'sıralı kararların yapılması zorunlu olan büyük ve karmaşık sorunlara uygulanacak kantitatif analiz tekniği' dir [10].

Dinamik programlama, bazı sorunların optimal sonucunu bulmada kullanılan güçlü, esnek bir yöntemdir. Dinamik programlamanın en çok uygulandığı işletme konuları; işgücü miktarı problemi, sermaye bütçeleme problemi, satış problemi, en kısa yol problemi, güvenilirlik problemi, fiyat stratejisi belirleme problemi, yatırım problemi, üretim planlama problemi, kargo yükleme problemi, stok kontrol problemi, dağıtım problemi, optimal parçalama problemi olarak sıralanabilir.

Dinamik programlama böl ve yönet yöntemine benzemekle birlikte böl-yönet metodundaki alt problemlerin bağımsız olma şartını aramamaktadır. Her alt problemi bir kez çözer ve çözümleri bir tabloda saklar, aynı alt problemin ortaya çıkması durumunda tablodaki değeri kullanır.

Dinamik programlama uygulanabilmesi için özyinelemeli çalışan bir yöntem olduğundan problem uzayının küçük olması gerekir. Toplam alt problem sayısının giriş verisinin boyutu türünden polinomsal olması gerekir.

3.3 Dinamik Programlamanın Çözüm Yöntemleri

Dinamik programlama probleminin çözümüne uygun bir modelin kurulması ile başlanır. Dinamik programlama ile ilgili problemler konuları ve biçimleri açısından

birbirlerinden çok farklı olabildiklerinden tamamına uygun çözüm içeren tek bir yöntem yoktur.

Tablosal yöntemde, tüm aşamalar için bütün durumlar göz önünde bulundurularak tüm seçenekler belirlenir. Seçeneklerden en iyileri tabloya yerleştirilir. Uygun olan seçenekler dikkate alınarak çözüm yapılması sağlanır.

Analitik yöntemde ise her bir aşamada tek bir değişkene bağlı olarak en iyileme yapılan dönüşüm denklemleri kullanılır. Dönüşüm denklemlerinin her bir aşamada türevleri alınarak aşamalar için optimal değer bulunur. Tablosal yöntemde kesikli, analitik yöntemde sürekli veriler kullanılır.

Dinamik programlama problemleri çözümü problemin yapısı ve araştırmacının yaklaşımına bağlı olarak ileriye doğru ya da geriye doğru yineleme yöntemini kullanır. İleriye doğru hesaplamada ilk aşamadan başlanıp n. aşamaya doğru hesaplanarak en iyi değer elde edilirken, geriye doğru hesaplamada n. aşamadan başlanarak ilk aşamaya doğru ilerlenerek bulunan yöntemdir. Geriye doğru yinelemeyi en iyi örnekleyen durum;

İki arkadaş bir masada otururken akıllarına bir oyun gelir. Bu oyunun kurallarına göre sırası gelen oyuncu masadan ancak 1,2 veya 3 kibrit çöpü alabilir.Masada toplam otuz kibrit çöpü olduğuna ve en son kibrit çöpünü alan oyuncu yenileceğine göre,ilk oyuncu kazanması için nasıl bir strateji izlemesi gerekir?

Geriye doğru çözüm yöntemine göre;kazanabilmek için en sonunda sıra karşıdaki oyuncuda iken bir kibrit çöpü kalması gerekir, dolayısıyla karşıdaki oyuncu 1,2 veya 3 kibrit çöpü de alsa sıra bize geldiğinde alacağımız kibrit sayısını ayarlayıp ona bir kibrit bırakırız.Kibrit sayısı dörtlü modlarla ilerlediğinde oyun bizim kontrolümüzde sona erer.Bu durumda kibrit sayıları,sıra karşıdaki oyuncuda iken 1,5,9,13,17,21,25,29 olmalıdır ve eğer ilk biz bir kibrit alırsak 29 ile karşıdaki oyuncu başlar ve oyunu kazanmış oluruz[11].

İleriye doğru çözüm yöntemi, Bhavnoni ve Chen tarafından 1962 yılında Bellman'ın geliştirdiği optimalite kuramına eşdeğer olarak ortaya çıkarılmıştır. Bhavnoni ve Chen'e göre, önce gelen kararlar son kararı izleyen duruma göre optimal bir politika oluşturmalıdır.

3.4 Dinamik Programlama Çeşitleri

Deterministik ve stokastik olmak üzere iki çeşit dinamik programlama problemi modeli bulunmaktadır. Çok aşamalı bir karar verme sürecinde, karara etki eden tüm faktörler tam olarak bilinirse süreç deterministiktir. Süreçle ilgili geçiş fonksiyonunda bir rassallık var ise stokastik yapılı dinamik programlamadan bahsedilir. Stokastik süreçlerin dinamik programlamada analitik ve tablosal yöntemlerle çözülebilmesi için durumların beklenen değerlerinin hesaplanma işleminin yapılması gerekir.

Deterministik ve stokastik dinamik programlama problemleri sonlu ve sonsuz durumlara göre dört şekilde sınıflandırılır. Her iki tarafı kapalı durum; sol tarafı açık, sağ tarafı kapalı durum; sağ tarafı açık,sol tarafı kapalı durum; her iki tarafı açık durum olarak çözüm yapılabilinir.

3.5 Dinamik Programlama Yapısı

3.5.1. Dinamik Programlamanın Temel Kavramları

Aşama; küçük alt problemlere ayrıştırılan büyük problemlerde her alt problem çözümlenmesi gereken bir karar problemi olarak ele alınır. Karar verilen her nokta da aşama olarak adlandırılır. Bu aşamalar, birbirleriyle bağlantılı bir süreç olduğundan herhangi bir alt grupta verilen yanlış karar, sonucu etkiler. Aşama sayısı sürecin uzunluğuna bağlıdır ve sürecin kesikli ya da sürekli olması aşama yapısını belirler.

Durum değeri, sistemin mevcut durumundan başlayarak özel bir yol izlendiğinde ortaya çıkan getirilerin toplam bir fonksiyonudur[12]. Bir aşamadaki durum, daha önceki aşamada verilen kararın bir sonucudur [13].

Dönüşüm fonksiyonları, aşamalar arasında verilen karara göre bağlantı sağlayan matematiksel ifadeler olmakla birlikte ileriye ya da geriye doğru en iyileme olmasına göre farklılık gösterirler. Bir aşamanın girdisi önceki aşamanın uygun değer çözümü olduğundan bazı ortak kısıtlarla alt problemler birbirleriyle ilişkilendirilirler.

Ardışık en iyileme, dinamik programlama yaklaşımı için gerekli olan, her aşama bir karar probleminin çözümüyle biri diğerini izleyen kararlarda uygulanmakta olup, tüm aşamalarla birlikte, başlangıç karar probleminin en uygun çözümünü veren fonksiyonel bağlantılar ile bir en iyileme tekniği bulmaktadır [14].

Karar, problemdeki her bir aşamayı tamamlamak için alternatifler arasından yapılan seçimdir.

Optimal politika, sürecin her bir aşaması için verilen kararların bir sırasıdır.

3.5.2 Dinamik Programlama Formülasyonu

Dinamik programlama, birbiriyle ilişkili sıralı kararları vermede en iyileme prosedürü içeren matematiksel bir model kullanır. Ancak, doğrusal programlamanın tersine standart bir matematiksel formül içermez. Farklı problem tipleri için geliştirilebilen özel denklemleri kullanır [15].

Dinamik programlama literatürü değişmez bir şekilde geriye doğru yineleme yöntemini kullanmaktadır. Geriye doğru yinelemenin hesaplama bakımından daha etkili olması tercih sebebi olmaktadır [4].

Formülasyon ;

N : aşama sayısı,

n : geçerli aşama,

s_n : n. aşama için geçerli durum değeri,

x_n : n. aşama için karar değişkeni,

x_n^* : optimal x_n değeri,

$f_n(s_n,x_n)$: n. aşamadaki s_n durumu ve x_n kararı altındaki getiri,

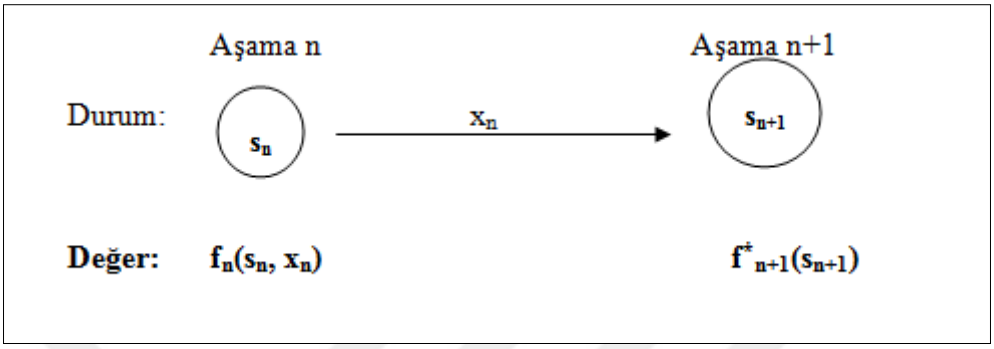
$f_n^*(s_n)$: optimal getiri değeri,

$f_n^*(s_n) = \max \{ f_n(s_n,x_n) \}$ veya $f_n^*(s_n) = \min \{ f_n(s_n,x_n) \}$

Çizelge 3.1Dinamik programlamanın çözümünün tablosal gösterimi

$s_n \setminus x_n$	$f_n(s_n,x_n)$			$f_n^*(s_n)$	x_n^*
	x_{n1}	x_{n2}…..	x_{nj}		

n. aşamadan başlanan çözüm değerleri Tablo 3.1'de gösterildiği gibi yerleştirilerek ilk aşamaya kadar devam ettirilir. İlk aşamadaki tablo değerleri problem için optimum çözümü vermektedir.

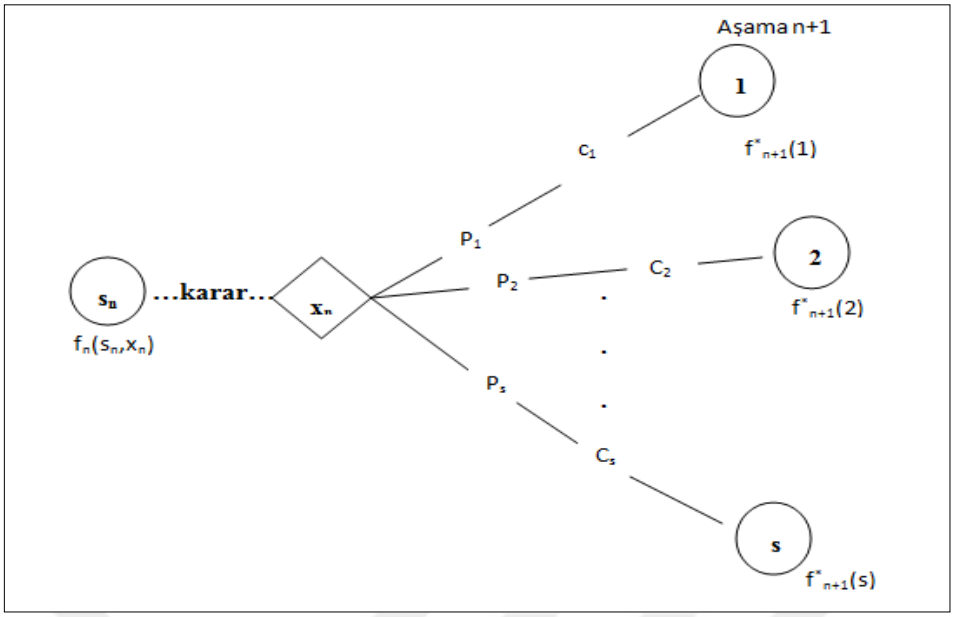


Şekil 3.1 Deterministik dinamik programlamanın basit yapısı

Şekil 3.1'de gösterilen deterministik dinamik programlama şemasını incelediğimizde, bir aşamadan başlanarak sonraki aşamalara geçildiği ve bu aşamaların birbirlerinden bağımsız birer optimizasyon problemi modelleri oldukları söylenebilir.

Bir sonraki adımda durum değerinin belirli olmadığı, olasılıklı dağılım içermesi durumunda (Şekil 3.2) ise C_i, n. aşamadaki getiri değerini, i durumuna geçiş olasılığını p_i, durum değişkenini s_n ve n. aşamadaki karar değişkenini x_n gösterirken, olasılıklı dinamik programlama modeli formülü;

$f_n(s_n, x_n) = \sum p_i[C_i + f^*_{n+1}(i)]$şeklinde ifade edilir. (3.1)



Şekil 3.2 Stokastik (olasılıklı) dinamik programlamanın basit yapısı

3.6 Dinamik Programlama İçin Geçerlilik Özellikleri

1.Ayrılabilme Özelliği:

Her plan için her durum değeri o anki getiri ve bir önceki aşamadaki durum değerlerinin bir fonksiyonu olarak ifade edilebilmesidir.

2. Optimallik Özelliği:

Optimum planın, birbirini izleyen durum ve hareketleri içine alan adımlardan oluşmasıdır.

3.7 Dinamik Programlamanın Avantaj ve Dezavantajları

Dinamik programlama yöntemi ile çözüm önerisi getirilebilen problem türleri oldukça geniş çerçevededir. Doğrusal ve doğrusal olmayan yapıdaki problemler ile kesikli ya da sürekli zamanlarda tanımlanan problemleri de çözebilme imkânı tanımaktadır. Ayrıca, veriler olasılıklı ya da olasılıklı olmayan özellikler taşıması durumunda da dinamik programlama yöntemi uygulanabilmektedir.

Yöntem optimalite felsefesini içinde barındırdığı için gereksiz seçenekleri çözüm aşamasında hesaba dâhil etmeyerek kolaylık sağlamaktadır.

Karar verme sürecini problemi alt problemler halinde ayrıştırdığı için hesaplama kolaylığını beraberinde getirmektedir. Değişkenlerle her aşamada karşılaşılsa da sayıları artmamaktadır.

Sistem parametrelerindeki değişikliklere karşın esnek yapıda olması nedeniyle uygulanacak modelin duyarlılık analizi yapılabilmesine de olanak sağlamaktadır.

Dinamik programlama yönteminin dezavantajlarına yönelik ise karşılaşılan bütün problemler için geçerli olan ortak bir matematiksel model yapısı içermeme durumu söylenebilir. Farklı her problem için o probleme özgü matematiksel yapının baştan oluşturulmasını gerektirir.

Ayrıca alt problemlere ayrıştırılarak azaltılan boyut sorununun, çözümü oluşturacak seçeneklerin durum değişkeni ile üssel olarak artmasından dolayı tamamen kaldırılamamış olması da dezavantajlı olmasına neden olmaktadır.

Analitik çözüm yönteminin kullanıldığı bazı durumlarda uygun çözüm bulunamadığı gibi bazı durumlarda çözüm için diğer metotlar daha kısa sürede cevap verebilmektedir. Kısıtlayıcı sayısı artması durumunda matematiksel hesaplamaları yapmak güçleşir.

BÖLÜM 4

YAYGIN DİNAMİK PROGRAMLAMA UYGULAMALARI

Dinamik programlama modellerinde; problemin aşamalara ayrılması, her bir aşama için alternatif ve durumların tanımlanması işlemi yapılması ortak özellik olarak gösterilebilir.

İki yer arasındaki en kısa mesafenin izlenebilmesi amacıyla geçiş yolları arasında bir alternatif sunan en kısa yol problemleri, dinamik programlama yönteminin kolaylıkla uygulandığı alandır. Geçiş yolları aşamalara ayrılarak her bir aşamada kendi içinde optimum sonuç aranır ve yol rotası çıkarılır. Yöntem ile çok karmaşık görünen ağlarda bile çözümü parça parça görmeyi sağlar.[16]

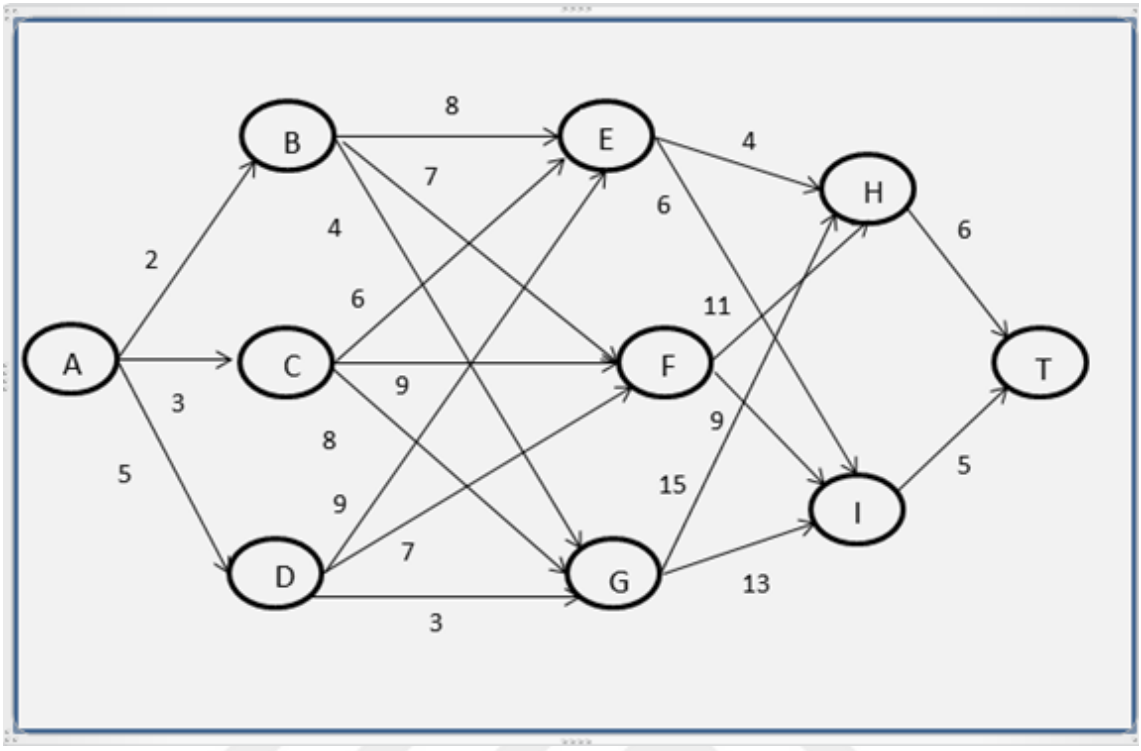
Çok aşamalı bir karar verme süreci olarak incelenebilen yükleme problemleri (sırt çantası, knapsack) , sınırlı hacme sahip taşıtlara kargoların en çok geliri (en çok fayda) sağlayacak şekilde doldurulması ile ilgilenir. Model tamsayılı doğrusal programlama modeli şeklinde ve dinamik programlama yöntemi ile çözülebilmektedir.

Aşamalardaki durum ve getiri fonksiyonlarının olasılıklı olduğu şans oyunları, stok modelleri ve markov süreçleri gibi problemlerde dinamik programlama yöntemi kullanılan diğer problemlerdir.

Maksimizasyon ve minimizasyon problemleri doğrusal ya da doğrusal olmayan modelde ifade edildiğinde de yine değişkenler birer aşama, kısıtlar durum değişkeni şeklinde fonksiyona dahil edildiğinde optimum sonuç alınabilmektedir.

İşgücü miktarı modellerinde, maliyet parametreleri göz önünde bulundurularak gerekli işgücü miktarını (işe alım ya da işten çıkarma durumu) hesaplamada dinamik programlama modeli kullanılabilmektedir [17]

4.1En Kısa Yol Problemi



Şekil 4.1 En kısa yol problem ağı

Şekil 4.1'de verilen örnekte, A noktasından T noktasına varış için amaç,en kısa yolu kullanmaktır.[1] Düzenek, 10 düğüm 20 alternatif yolda dinamik programlama yöntemi kullanılarak 4 aşamada çözüm yapılır. Her aşamada optimum sonuca ulaşarak bir sonraki aşamaya geçilir. Bu durumda geçiş noktalarını belirlemede geriye doğru yaklaşımla her bir aşamadaki adımlar belirlenir. Karar verilecek aşamalar;

[1]Şebeke ağıProf.Dr.H.Kemal SEZEN Yöneylem Araştırması kitabından alınmıştır.

[2]LINGO çözümü ektedir.

4.Aşama

H → T (6 br)

I → T (5 br)

3.Aşama

E → H (4 br)

E → I (6 br)

F → H (11 br)

F → I (9 br)

G → H (15 br)

G → I (13 br)

2.Aşama

B → E (8 br)

B → F (7 br)

B → G (4 br)

1.Aşama

A → B (2 br)

A → C (3 br)

A → D (5 br)

C → E (6 br)

C → F (9 br)

C → G (8 br)

D → E (9 br)

D → F(7 br)

D → G(3 br)

şeklinde gösterilebilir. Dört aşamaya ayrılan sistemdeki durum değerlerini karar tablolarını kullanarak hesaplama kolaylığı sağlanır. Karar tablolarında aşama sayısı (n), karar değişkenleri (x_n), durum değeri (s_n) ile gösterilir. s_n durumunda en iyi karara karşılık gelen değeri $f_n^*(s_n,x_n)$ ile $f_n(s_n,x_n)$ değeri en iyileyen x_n değerini x_n^* ve $f_n^*(s_n)$ ile gösterilir.

Problemin çözümüne 4.aşamadan başlandığında,

Çizelge 4.1 Dördüncü aşamada en kısa yol

s / x_4	T	$f^*_4(s)$	x_4
H	6	6	T
I	5	5	T

Dördüncü aşamada T varış noktasına $f^*_4(s) = f^*_4(s,T) = c_{s,j}$ formülü ile H noktasından 6 birim, I noktasından 5 birim mesafeleri başlangıç tablosuna yerleştirilir. H, I noktalarına ise E, F, G noktalarından geçiş olduğu verilmektedir. Hangi yoldan gidileceğine karar vermek için üçüncü aşamada $f_n(s,x_n) = c_{s,xn} + f^*_{n+1}(x_n)$ formülü kullanılarak hesaplanan değerler tabloya yerleştirilir.

Çizelge 4.2 Üçüncü aşamada en kısa yol

	$\mathbf{f_3(s,x_3) = c_{sx3} + f^*_4(x_3)}$			
$\mathbf{s \setminus x_3}$	**H**	**I**	$\mathbf{f^*_3(s)}$	$\mathbf{x^*_3}$
E	4+6=10	6+5=11	10	H
F	11+6=17	9+5=14	14	I
G	15+6=21	13+5=18	18	I

Hesaplanan Çizelge 4.2'de elde edilen değerlere göre E'den en kısa yol 10 birim ile H noktasına, F'den en kısa yol 14 birim ile I noktasına, G'den en kısa yol 18 birim ile I noktasına olduğu söylenebilir.

Çizelge 4.3 İkinci aşamada en kısa yol

	$\mathbf{f_2(s,x_2) = c_{sx2} + f^*_3(x_2)}$				
$\mathbf{s \setminus x_2}$	**E**	**F**	**G**	$\mathbf{f^*_2(s)}$	$\mathbf{x^*_2}$
B	8+10=18	7+14=21	4+18=22	18	E
C	6+10=16	9+14=23	8+18=26	16	E
D	9+10=19	14+7=21	3+18=21	19	E

Çizelge 4.3'te elde edilen değerlere göre ikinci aşamada en kısa yol; B'den 18 birim, C'den 16 birim ve D'den 19 birim ile E noktasınadır.

Çizelge 4.4 Birinci aşamada en kısa yol

s / x_1	$f_1(s,x_1) = c_{sx1} + f^*_2(x_1)$				
	B	**C**	**D**	$f^*_1(s)$	x^*_1
A	2+18 = 20	3+16 = 19	5+19 = 24	19	C

Başlangıçta A noktasından, B noktasından geçiş yapılarak optimum yol izlendiğinde toplam yol 20 birim, C noktası tercih edildiğinde 19 birim, D noktası tercih edildiğinde ise 24 birim uzunluk ölçülmüştür. Bu durumda en kısa yol rotası A-C-E-H-T ile mesafe 19 birimdir.

Düğümler arasındaki mesafelerin sabit olmaması, herhangi bir sebep ile uzaklıkların değişkenlik göstermesi durumunda sonuçlar incelenmiştir. Değişkenlik olması varsayımında normal dağılım ve tek-düze dağılıma uygun simülasyon yapılmıştır. İzlenecek yolun karşılaştırması Tablo 4.5 ve Tablo 4.6'da özet şeklinde verilen değerlere göre yapılabilir.

Çizelge 4.5 Normal dağılıma uygun veri setinden çekilen değerlere göre en kısa yol

GEÇİŞ NOKTALARI	Normal Dağılım				
	Standart Sapma = 2	Standart Sapma = 5	Standart Sapma = 10	Standart Sapma = 15	Standart Sapma = 20
	Ortalama Mesafe (br)	Ortalama Mesafe (br)	Ortalama Mesafe (br)	Ortalama Mesafe (br)	Ortalama Mesafe (br)
H-T	6.14	5.84	4.26	7.46	7.06
I-T	4.9	4.56	5.03	2.09	6.21
E-H	3	4.28	4.14	2.26	1.74
E-I	5.83	7.64	7.03	7.48	8.79
F-H	11.07	11.49	9.57	10.74	17.46
F-I	8.54	9.85	8.86	10.93	9.72
G-H	15.12	16.26	15.67	12.47	13.55
G-I	12.71	13.38	15.63	11.13	14.3
B-E	7.37	7.95	9.72	9.41	10.13
B-F	6.83	7.76	7.39	6.15	7.39
B-G	4.22	4.36	2.76	3.19	0.34
C-E	5.57	7.06	9.35	7.87	6.11
C-F	9.53	8.59	10.11	5.08	9.19
C-G	8.64	8.28	7.02	6.09	8.04
D-E	9.15	8.98	8.59	9.07	2.79
D-F	7.52	7.2	8.24	7.71	4.21
D-G	2.82	2.6	2.13	1.84	2.47
A-B	2.14	1.58	2.62	0.76	3.01
A-C	2.9	3.18	1.56	4.65	1.67
A-D	4	3.86	1.85	1.79	1.84
EN KISA YOL	A-C-E-H-T	A-B-E-H-T	A-D-E-H-T	A-D-G-I-T	A-D-E-H-T
TOPLAM MESAFE	17.61 br	18.37 br	18.84 br	16.85 br	13.43 br

Çizelge 4.5'te verilen durum özetine göre normal dağılıma uygun düzenekte standart sapmanın 2 br olması halinde en kısa yol toplamda 17.61 br ile A-C-E-H-T geçiş yollarından oluşmaktadır. Standart sapmanın 5 br olması durumunda 18.37 br ile A-B-E-H-T, 10 br olması halinde 18.84 br ile A-D-E-H-T, 15 br iken A-D-G-I-T, 20 br'e çıkardığımızda ise 13.43 br toplam mesafe ile A-D-E-H-T tercih edilir.

Çizelge 4.6 Tek-düze dağılıma uygun veri setinden çekilen değerlere göre en kısa yol

GEÇİŞ NOKTALARI	Tek Düze Dağılım				
	Standart Sapma = 2	Standart Sapma = 5	Standart Sapma = 10	Standart Sapma = 15	Standart Sapma = 20
	Ortalama Mesafe (br)	Ortalama Mesafe (br)	Ortalama Mesafe (br)	Ortalama Mesafe (br)	Ortalama Mesafe (br)
H-T	5.8	5.95	7.04	6.7	7.09
I-T	5.23	5.11	5.02	4.08	5.05
E-H	3.65	4.24	4.1	4.41	4.4
E-I	6.45	5.65	5.83	3.62	7.71
F-H	11.09	11.55	11.01	11.89	12.77
F-I	9.16	8.85	10.52	8.66	9.81
G-H	14.87	14.62	15.06	17.45	16.17
G-I	13.1	13.75	13.05	12.43	16.47
B-E	8.12	7.92	8.11	8.58	7.75
B-F	6.86	6.65	8.53	7.2	6.3
B-G	4.05	4.18	2.7	1.67	6.15
C-E	5.95	5.29	5.31	6.66	6.32
C-F	8.97	9.52	9.11	9.75	10.23
C-G	8.12	8.07	7.32	6.93	7.35
D-E	8.96	9.13	8.72	5.68	7.92
D-F	7.02	6.91	6.62	6.56	8.56
D-G	3.17	2.93	2.96	2.95	4
A-B	2.33	2.77	1.71	4.3	4.41
A-C	3.43	2.96	3.6	3.99	2.09
A-D	5.08	4.83	6.56	6.47	5.93
EN KISA YOL	A-C-E-H-T	A-C-E-H-T	A-C-E-I-T	A-C-E-I-T	A-C-E-H-T
TOPLAM MESAFE	18.83 br	18.44 br	19.76 br	18.35 br	19.9 br

Çizelge 4.6'da tek-düze dağılıma uygun düzenekte en kısa yol için standart sapma 2 ve 5 br iken toplam mesafeler 18.83 ve 18.44 br ile A-C-E-H-T geçiş yollarına karar verilir. Standart sapma 10 ve 15 birime çıkarıldığında 19.76 ve 18.35 birim ile A-C-E-I-T rotası, 20 br olduğu durumda ise 19.9br ile A-C-E-H-T yolunu en kısa yol olarak söylemek mümkündür.

İnceleme neticesinde değişkenlik ölçüsü 2 birim iken her iki dağılımda simülasyon öncesindeki sonuçla farklılık göstermemektedir. Sapma miktarı arttıkça normal dağılıma uygun düzenekte farklılık göstermekle birlikte tek-düze dağılıma uygun düzenekte tercih edilen yol daha istikrarlı olduğu söylenebilir.

4.2 Kargo Yükleme Modeli (Knapsack-sırt çantası problemi)

Kargo yükleme modeli, belirli bir kapasitedeki alana en çok faydayı sağlayacak şekilde yükleme yapılması durumuyla ilgilidir. Bir asker ya da yürüyüşçünün sırt çantasına alması gereken en önemli eşyalarını belirlemesi şeklinde düşünüldüğünde literatürde sırt çantası modeli olarak da yer almaktadır.

Geriye doğru yineleme denklemi;

Amaç fonksiyonu:

$$\text{maks.}\, z = r_1 m_1 + r_2 m_2 + \cdots + r_n m_n \tag{4.1}$$

Kısıtlar:

$$w_1 m_1 + w_2 m_2 + \cdots + w_n m_n \leq W$$

$$m_1, m_2, \ldots, m_n \geq 0 \text{ ve tamsayı}$$

N farklı yük, W kapasite,m_i yüklemede i.yükten kaç birim olacağını gösterir.

Dinamik programlama formülasyonu içinde

1) i.aşama, i yükü ile ifade edilir; i=1,2,…,n
2) i yükün sayısı m_i ile gösterilirken W/w_i'den küçük ya da ona eşit en büyük tamsayı olarak değerler alır, gelir ise $r_i m_i$ şeklinde ifade edilir,
3) i. aşamadaki durum ise i,i+1,..,n. aşamalarda alacak ağırlık x_i ile gösterilir.

Maksimum geliri, faydayı sağlayacak yol iki adımda incelenir.

$$\underline{\text{1.adım:}} f_i(x_i) = \text{maks}\{r_i m_i + f_{i+1}(x_{i+1})\}, \quad i = 1,2,\ldots,n \tag{4.2}$$

$$m_i = 0,1,\ldots W/w_i$$

$$x_i = 0,1,\ldots,W$$

ve $f_{n+1}(x_{n+1}) \equiv 0'$dır.

$$\underline{\text{2.adım:}} f_i(x_i) = \text{maks}\{r_i m_i + f_{i+1}(x_i - w_i m_i)\}, \quad i = 1,2,\ldots,n \tag{4.3}$$

$$m_i = 0,1,\ldots W/w_i$$

$$x_i = 0,1,\ldots,W$$

Yürüyüşe çıkan bir kişi yiyecek, ilk yardım çantası ve giyecek gibi üç şeyi yanında götürmek istemektedir. Yanına aldığı sırt çantasının kapasitesi 3 kg, her bir yiyecek 1 kg, ilk yardım çantası ¼ kg, her bir parça giysi ½ kg'dır. Yürüyüşçü yiyecek, ilk yardım çantası ve giyeceklerin önceliklerini 3, 4 ve 5 olarak belirlemiştir. Yürüyüşçü her bir parçadan en az bir adet ve ilk yardım çantasından ikiden fazla almaması gerektiğini önceki yürüyüşlerden bilmektedir.[1] Bu durumda her bir gerekli malzemeden kaçar adet alması gerektiğinin hesabını sırt çantası (knapsack) modeli ile çözersek;

w_i;ağırlıkları,

r_i; öncelik puanlarını,

m_i; malzeme sayısını,

x_i; çantanın alabileceği ağırlık miktarını, gösterdiği durumda aşağıdaki tablo ile özetleyebiliriz.

Çizelge 4.7 Çanta ağırlık durumu

		w_i	r_i
1	Yiyecek	1 kg	3 puan
2	İlk Yardım	¼ kg	4 puan
3	Giyecek	½ kg	5 puan

Kısıtlar; $m_1, m_2, m_3 \geq 1$ (4.4)

$m_2 \leq 2$

1. aşama

Birinci aşamada giyecek için tablo yapıldığında malzeme sayısı(m) diğer malzemelerden de alması gerektiğinden en fazla 3 olabilmektedir.

Getiri ise $f_3(x_3)=\text{maks}\ \{5m_3\}$ (4.5)

[1]Soru metni HamdyA.Taha Yöneylem Araştırması kitabından alınmıştır.

Çizelge 4.8 Birinci aşama

x_3	m_3=1	m_3=2	m_3=3	Optimal Çözüm	
				$f_3(x_3)$	m_3^*
1	5	10	-	10	2
2	5	10	15	15	3
3	5	10	15	15	3

2.aşama

İkinci aşamada ilk yardım malzemesi için 2'den fazla olmaması kısıtı altında;

$$f_2(x_2) = \text{maks}\{4m_2 + f_3(x_2 - 1/4\, m_2\} \quad (4.6)$$

Çizelge 4.9 İkinci aşama

x_2	m_2=1	m_2=2	Optimal Çözüm	
			$f_2(x_2)$	m_2^*
1	4+5=9	8+5=13	13	2
2	4+15=19	8+15=23	23	2
3	4+15=19	8+15=23	23	2

3.aşama

Üçüncü aşamada yiyecek malzemesi çanta kapasitesi ve diğer malzemelerin de alınması gerektiğinden en fazla 2 olmakta,

$$f_1(x_1) = \text{maks}\{3m_1 + f_2(x_1 - 1\, m_1\} \quad (4.7)$$

Çizelge 4.10 Üçüncü aşama

x_1	m_1=1	m_1=2	Optimal Çözüm	
			$f_1(x_1)$	m_1^*
1	-	-	-	-
2	3+13=16	-	16	1
3	3+23=26	6+13=19	26	1

Sonuç olarak; Yiyecek = 1 birim

İlk Yardım = 2 birim

Giyecek = 3 birim, optimal puan 26 birim olmaktadır.

Soru, doğrusal programlamada maksimizasyon problemi olarak modellenirse;

Değişken x_i:çantaya alınacak malzeme miktarı (adet) (i=yiyecek,ilkyardım,giyecek);

Amaç fonksiyonu

$$\text{maksz} = 3.\text{xyiyecek} + 4.\text{xilkyardım} + 5.\text{xgiyecek} \qquad (4.8)$$

Kısıtlar;

$\text{xyiyecek} + 0{,}25.\text{xilkyardım} + 0{,}50.\text{xgiyecek} \leq 3$,(çantanın kapasitesinin 3kg olması)

$\text{xilkyardım} \leq 2$, (önceki yürüyüşten bilinen miktar)

$\text{xyiyecek}, \text{xilkyardım}, \text{xgiyecek} \geq 1$, (her birinden alınması koşulu)

Kurulan model LINGO programı ile çözülmesi durumunda model denklemi ve çözüm raporu şekil 4.2'de verilmektedir.

```
max=3*xyiyecek+4*xilkyardim+5*xgiyecek;
1*xyiyecek+0.25*xilkyardim+0.5*xgiyecek<=3;
xilkyardim>=1;
xgiyecek>=1;
xyiyecek>=1;
xilkyardim<=2;

end
```

```
Global optimal solution found.
Objective value:                              26.00000
Infeasibilities:                              0.000000
Total solver iterations:                             0
Elapsed runtime seconds:                          0.08

Model Class:                                        LP

Total variables:                     3
Nonlinear variables:                 0
Integer variables:                   0

Total constraints:                   6
Nonlinear constraints:               0

Total nonzeros:                     10
Nonlinear nonzeros:                  0


                        Variable           Value        Reduced Cost
                          XYIYECEK        1.000000            0.000000
                       XILKYARDIM        2.000000            0.000000
                          XGIYECEK        3.000000            0.000000

                             Row    Slack or Surplus      Dual Price
                               1        26.00000            1.000000
                               2        0.000000            10.00000
                               3        1.000000            0.000000
                               4        2.000000            0.000000
                               5        0.000000           -7.000000
                               6        0.000000            1.500000
```

Şekil 4.2 LINGO maksimizasyon modeli ve çözüm raporu

Çözüm raporuna göre amaç fonksiyonunun 26 puan, yiyecek miktarının 1 adet, ilk yardım miktarının 2 adet ve giyecek miktarının 3 adet olması durumunda en iyi çözüme ulaşıldığı görülmektedir. Modelde aylak ve artık değişken değerleri (slack or surplus) sütununda, ilk satırda amaç fonksiyonu z değerini, diğer satırlarda kısıtlara ilişkin S_1=0, S_2=1, S_3=2, S_4=0,S_5=0 değerlerini vermektedir. İkil değişkenlerin (dual price) modelde aldıkları değerlere göre de sağ taraf sabitlerindeki bir birimlik artış ya da azalışın amaç fonksiyonuna(maksimum getiri) ne kadar etki edeceğini göstermektedir. Burada ikinci satır çanta kapasitesini içeren ilk kısıta denk geldiğinden kapasitedeki bir birimlik artış elde edilecek faydayı 10 birim arttırmakta olduğu söylenebilir.

Dinamik programlama ve simpleks yöntemlerinin sırt çantası modeline verdiği çözümler karşılaştırıldığında, uygulamada (herhangi bir program kullanılmaksızın çözüldüğünde) işlemsel açıdan dinamik programlamanın daha kolaylık sağladığı görülmekle birlikte elde edilen bilgi ve süreç ile ilgili yorumda simpleks yönteminin daha etkin olduğu anlaşılmaktadır.

4.3 Stokastik Dinamik Programlama Modeli Problemi

4.3.1 Şans Oyunu

Çevresi 1'den n'ye kadar numaraların olduğu bir çarkla oynanan Rus ruleti oyununun bir çeşididir. Oyunda çarkın bir kez çevrildiğinde i numarasında durma olasılığı p_i'dir. Oyuncu m kez çevirme hakkına sahip olabilmek için x para birimi ödemektedir. Oyuncunun getirisi son çevirmede gelen sayının iki katı olduğu durumunda en iyi getiriyi elde etmek için strateji belirlenir.

Dinamik programlama modeli;

i aşamasındaki sistemin j durumu ile ifade edildiğinde,

$f_i(j)$; son çevirmenin sonucu j iken ve oyun j aşamasındayken beklenen getiriyi gösterir.

Beklenen maksimum getiri ise oyun biterse 2j, oyun devam ederse $\sum p_k f_{i+1}$'dir. İlk çevirmede oyun yeni başlamış olduğundan j=0'dır ve ilk aşamada

$f_1(0)=p_1f_2(1)+p_2f_2(2)+\ldots+p_nf_2(n)$ olur. (4.9)

Yinelenen hesaplamalar f_{m+1} ile başlar ($f_{m+1}=2j$) ve $f_1(0)$ ile sona erer. $f_1(0)$ m çevirmenin beklenen getirisi olduğundan ve oyunun maliyeti x para birimi olarak verildiğinden beklenen getiri $f_1(0) - x$'dir.

Çevresi 1'den 5'e kadar numaralandırılmış bir rulet tekerleği için tekerleğin i sayısında durma olasılığı p_1=0.35, p_2=0.30, p_3=0.20, p_4=0.10, p_5=0.05 iken oyuncuların maksimum 4 çevirme için 3pb ödemekte ise optimum strateji ve getirisini hesaplayalım;

Çizelge 4.11 Beşinci aşama

4.Çevirmenin Sonucu	**Optimum Çözüm**	
J	**$f_5(j)$**	**Karar**
1	2	Bitir
2	4	Bitir
3	6	Bitir
4	8	Bitir
5	10	Bitir

4.aşama

$$f_4(j) = \text{maks}\{2j, p_1 f_5(1) + p_2 f_5(2) + p_3 f_5(3) + p_4 f_5(4) + p_5 f_5(5)\} \quad (4.10)$$

$$= \text{maks}\{2j, 0.35 * 2 + 0.30 * 4 + 0.20 * 6 + 0.10 * 8 + 0.05 * 10\}$$

$$= \text{maks}\{2j, 4.4\}$$

Çizelge 4.12 Dördüncü aşama

3.Çevirmenin Sonucu	**Beklenen Getiri**		**Optimum Çözüm**	
J	**Bitir**	**Çevir**	**$f_4(j)$**	**Karar**
1	2	4.4	4.4	Çevir
2	4	4.4	4.4	Çevir
3	6	4.4	6	Bitir
4	8	4.4	8	Bitir
5	10	4.4	10	Bitir

3.Aşama

$$f_3(j) = \text{maks}\{2j, p_1 f_4(1) + p_2 f_4(2) + p_3 f_4(3) + p_4 f_4(4) + p_5 f_4(5)\} \quad (4.11)$$

$$= \text{maks}\{2j, 0.35 * 4.4 + 0.30 * 4.4 + 0.20 * 6 + 0.10 * 8 + 0.05 * 10\}$$

$$= \text{maks}\{2j, 5.36\}$$

Çizelge 4.13 Üçüncü aşama

2.Çevirmenin Sonucu	**Beklenen Getiri**		**Optimum Çözüm**	
J	**Bitir**	**Çevir**	**$f_3(j)$**	**Karar**
1	2	5.36	5.36	Çevir
2	4	5.36	5.36	Çevir
3	6	5.36	6	Bitir
4	8	5.36	8	Bitir
5	10	5.36	10	Bitir

2.Aşama

$$f_2(j) = \text{maks}\{2j, p_1 f_3(1) + p_2 f_3(2) + p_3 f_3(3) + p_4 f_3(4) + p_5 f_3(5)\} \quad (4.12)$$

$$= \text{maks}\{2j, 0.35 * 5.36 + 0.30 * 5.36 + 0.20 * 6 + 0.10 * 8 + 0.05 * 10\}$$

$$= \text{maks}\{2j, 5.984\}$$

Çizelge 4.14 İkinci aşama

1.Çevirmenin Sonucu	**Beklenen Getiri**		**Optimum Çözüm**	
J	**Bitir**	**Çevir**	**$f_5(j)$**	**Karar**
1	2	5.984	5.984	Çevir
2	4	5.984	5.984	Çevir
3	6	5.984	6	Bitir
4	8	5.984	8	Bitir
5	10	5.984	10	Bitir

1.Aşama

$$f_1(j) = \text{maks}\{2j, p_1 f_2(1) + p_2 f_2(2) + p_3 f_2(3) + p_4 f_2(4) + p_5 f_2(5)\} \quad (4.13)$$

$$= \text{maks}\{2j, 0.35 * 5.984 + 0.30 * 5.984 + 0.20 * 6 + 0.10 * 8 + 0.05 * 10\}$$

$$= \text{maks}\{2j, 6.3896\}$$

Beklenen kazanç: 6.3896 – 3 =3.3896 pb.

Sonuç olarak izlenecek strateji; oyun başında tek seçenek çevirmektir. İlk çevirmenin sonucunda 3-4-5 gelirse bitirilmeli, 1-2 gelirse çevirmeye devam edilmelidir. İkinci ve

üçüncü çevirmenin sonunda 3-4-5 gelirse bitirilmeli, 1-2 gelirse devam edilmelidir. Beklenen getiri ise 3.3896pb. olmaktadır.

4.4 Doğrusal Programlama Modelinin Dinamik Programlama Çözümü

Maks $z = 15x_1+10x_2$ (4.14)

$x_1 + 2x_2 \leq 6$

$3x_1+x_2 \leq 8$

$x_1 \geq 0,\ x_2 \geq 0$

Dinamik programlama formülasyonuna göre x_1,x_2 katsayıları dikkate alındığında aşama sayısı 2 ve x_n n.aşamadaki karar değişkenidir. Durum değişkeni s_n ise kalan faaliyetlere tahsisi için kullanılabilir kaynak miktarı olarak tanımlanabilir, kısıtlarda sağ taraf sabitleri R değişkeni ile gösterildiğinde;

$s_n=(R_1,R_2)$ (4.15)

$s_1=(6,8)$

$s_2=(6-x_1 , 8-3x_1)$ (4.16)

olur.

Dinamik programlama ile çözümü ise;

$f_2(R_1,R_2,x_2)$, 2.aşamada (R_1,R_2) durumunda karar değişkeni x_2 için 2.aktivitenin z'ye katkısı$=10x_2$,

$f_1(6,8,x_1)$, karar değişkeni x_1 ve sonrasında optimum karar değişkeni 2.aşamada alınacak iken 1.aşamada ve (6,8) durumunda 1 ve 2.aktivitenin z' ye katkısı

$=15x_1+\max\{10x_2\}_{x2<=12,2x2<=18-3x1,x2>=0}$

$f_n^*(R_1,R_2)=\text{maks}.f_n(R_1,R_2,x_n)$ (4.17)

(1) $f_2^*(R_1,R_2)=\text{maks}.\{10x_2\}$ (4.18)

$2x_2 \leq R_1$

$x_2 \leq R_2$

$x_2 \geq 0$

(2) $f_1(6,8,x_1)=15x_1+f_2^*(6-x_1 , 8-3x_1)$, (4.19)

(3) $f_1^*(6,8)=$ maks.$\{15x_1+f_2^*(6-x_1, 8-3x_1)\}$ (4.20)

$x_1 \leq 6$

$3x_1 \leq 8$, $x_1 \geq 0$

Aşama 2

$2x_2 <= R_1$, $x_2 <= R_2$, $x_2 >= 0$ şartlarını sağlayan en büyük x_2 değeri için $R_1/2$ ve R_2 değerlerinin minimumu araştırılır;

Çizelge 4.15 İkinci aşama

R_1,R_2	$f_2^*(R_1,R_2)$	x_2^*
$R_1 \geq 0$, $R_2 \geq 0$	10 min.$\{ R_1/2, R_2\}$	min.$\{ R_1/2, R_2\}$

$(R_1,R_2)= (6-x_1, 8-3x_1)$, (4.21)

$f_2^*(6-x_1, 8-3x_1)=10\min.\{ R_1/2, R_2\}=10\min.\{ (6-x_1)/2, 8-3x_1\}$ (4.22)

$f_1^*(6,8)=$ maks.$\{15x_1+10\min.\{ (6-x_1)/2, 8-3x_1\}\}_{0<=x1<=8/3}$ (4.23)

$0 \leq x_1 \leq 2$ ----------$(6-x_1)/2$ ---------- $15x_1 +10(6-x_1)/2$ (4.24)

$2 \leq x_1 \leq 8/3$----------$(8-3x_1)$ ---------- $15x_1 +10(8-3x_1)$ (4.25)

Aralıklar iki kısım halinde incelenir ve denklemler çözümlendiğinde değişkenlerin optimum değerleri $x_1=2$, $x_2=2$ bulunarak maksimum z değeri 50 olmaktadır.

Aynı soruyu simpleks yöntemi ile çözdüğümüzde elde edilen sonuçlar ise tabloda gösterildiği gibidir.

Çizelge 4.16 Simpleks yöntemi ile çözümü

Hedef Hücre (En Büyük)			
Hücre	**Ad**	**İlk Değer**	**Son Değer**
F4		50	50

Hücre	**Ad**	**Son Değer**	**Azaltılmış Maliyet**	**Amaç Katsayı**	**İzin Verilen Artış**	**İzin Verilen Azalış**
B3	Z	2	0	15	15	10
C3	Z	2	0	10	20	5

Sınırlamalar

Hücre	**Ad**	**Son Değer:**	**Gölge Ücret**	**Sınırlama Sağ Taraf**	**İzin Verilen Artış**	**İzin Verilen Azalış**
F7	kısıtlar	6	3	6	10	3,33
F8		8	4	8	10	5

Simpleks metodunda da değişkenler 2 değerini aldığında optimum sonuç değeri 50 sonucunu alır.

Doğrusal programlama modeli sorusunda kısıt katsayılarının standart sapması simülasyon ile normal dağılıma ait örneklemden çekilen veri setinden alındığı durumda simpleks yöntemi ve dinamik programlama yöntemi ile çözüldüğü durumda karşılaştırma yapılırsa;

$$\text{Maks } z = 14{,}76x_1 + 10{,}09x_2 \quad (4.26)$$

$$0{,}87x_1 + 1{,}99x_2 \leq 6$$

$$2{,}95x_1 + 0{,}75x_2 \leq 8$$

$$x_1 \geq 0,\ x_2 \geq 0$$

Çizelge 4.17 Simpleks yöntemi ile son değer

Hedef Hücre (En Büyük)			
Hücre	**Ad**	**İlk Değer**	**Son Değer**
F4		53,0713	53,0713

Ayarlanabilir Hücreler

Hücre	**Ad**	**İlk Değer**	**Son Değer**
B3	Z	2,1886	2,1886
C3	Z	2,0583	2,0583

Standart sapması 1 olan veri setinin ortalaması alındığında elde edilen değerlere uygulanan simpleks yöntemi çözümü sonucu 2,18 ve 2,05 ile optimum sonuca ulaşıldığında maksimum sonucu 53,07 olmaktadır.

Çizelge 4.18 İkinci aşama

R_1,R_2	$f_2^*(R_1,R_2)$	x_2^*
$R_1\geq 0$, $R_2\geq 0$	10,09 min.{ $R_1/1.99$, $R_2/0.75$}	min.{ $R_1/1.99$, $R_2/0.75$}

$$(R_1,R_2)= (6-0.87x_1, 8-2.95x_1), \quad (4.27)$$

$$f_2^*(6-0.87x_1, 8-2.95x_1)= 10.09 \text{ min.}\{ R_1/1.99, R_2/0.75\} = 10,09 \text{ min.}\{ (6-0.87x_1)/1.99, 8-2.95x_1/0.75\} \quad (4.28)$$

$$f_1^*(6,8)= \text{maks.}\{14.76x_1+10.09\text{min.}\{ 6-0.87x_1)/1.99, 8-2.95x_1/0.75\}\}_{0<=x1<=2,18} \quad (4.29)$$

$$0\leq x_1\leq 2.18 \text{ ----------}(6-0.87x_1)/1.99 \text{ ----------} 14.76x_1 +10.09(6-0.87x_1)/1.99 \quad (4.30)$$

değişkenlerin optimum değerleri $x_1=2.18$, $x_2=2.06$ bulunarak maksimum z değeri 52.962 olmaktadır.

$$\text{Maks } z =15.26x_1 + 9.28x_2 \quad (4.31)$$

$$1.94x_1 +2.93x_2 \leq 6$$

$$2.07x_1 + 1.65x_2 \leq 8$$

$$x_1\geq 0, x_2\geq 0$$

Çizelge 4.19 Simpleks yöntemi ile son değer

Hedef Hücre (En Büyük)			
Hücre	**Ad**	**İlk Değer**	**Son Değer**
F4		49,08	47,196
Ayarlanabilir Hücreler			
Hücre	**Ad**	**İlk Değer**	**Son Değer**
B3	Z	2	3,093
C3	Z	2	0

Standart sapması 2 olan veri setinin ortalaması alındığında elde edilen değerlere uygulanan simpleks yöntemi çözümü sonucu 3.09 ve 0 ile optimum sonuca ulaşıldığında maksimum sonucu 47.19 olmaktadır.

Çizelge 4.20 İkinci aşama

R_1,R_2	$f_2^*(R_1,R_2)$	x_2^*
$R_1\geq 0, R_2\geq 0$	9.28 min.{ $R_1/1.94, R_2/2.07$}	min.{ $R_1/1.94, R_2/2.07$}

$(R_1,R_2)= (6-1.94x_1 , 8-2.07x_1)$, (4.32)

$f_2^*(6-1.94x_1, 8-2.07x_1) = 9.28$ min.{ $R_1/1.94, R_2/2.07$}

$= 9.28$ min.{ $(6-1.94x_1)/2.93, 8-2.07x_1/1.65$} (4.33)

$f_1^*(6,8)=$ maks.$\{15.26x_1+9.28\min\{(6-1.94x_1)/2.93, 8-2.07x_1/1.65\}\}_{0<=x1<=3}$ (4.34)

$0\leq x_1\leq 3$ ----------$(6-1.94x_1)/2.93$---------- $15.26x_1 + 9.28(6-1.94x_1)/2.93$ (4.35)

değişkenlerin optimum değerleri $x_1=3$, $x_2=0.06$ bulunarak maksimum z değeri 46.34 olmaktadır.

Dinamik programlama ve simpleks algoritması yöntemi ile doğrusal programlama modelleri çözüldüğünde oldukça yaklaşık değerler vererek sonuçlanmaktadır.

4.5 Doğrusal Olmayan Model Çözümü

Doğrusal olmayan bir modeli dinamik programlama yöntemi ile çözerken diğer birçok problem çözümünde olduğu gibi ilk adımda problemin tanımlanması yapılır. Ardından problem birkaç aşamaya ayrıştırılır ve her bir aşamada karar verilecek olan durum değişkenleri tanımlanır. Sonraki adımda, geçiş (dönüşüm) fonksiyonları belirlenir. İleriye ya da geriye dönük yineleme yöntemlerinden birine karar verilir. Bütün aşamalarda alınan kararlar ile optimal sonuca ulaşılır.

$$\max z = f_1(x_1) + f_2(x_2) + .. + f_n(x_n) \quad (4.36)$$

$$a_1x_1 + a_2x_2 + .. + a_nx_n = b$$

$$x_i \geq 0 i = 1,2,..,n$$

Yukarıdaki gibi bir model Lagrange ya da ayrık programlama yöntemleri ile çözülebilmesi yanında dinamik programlama yöntemi ile de çözülebilir. Amaç maksimizasyon olan bu modelde n aşama, s_n durum değişkenleri olarak gösterildiğinde, dönüşüm fonksiyonları;

$$s_n = a_1x_1 + .. + a_nx_n \quad (4.37)$$

$$s_{n-1} = a_1x_1 + .. + a_{n-1}x_{n-1} = s_n - a_nx_n \quad (4.38)$$

.

.

$$s_2 = a_1x_1 + a_2x_2 = s_3 - a_3x_3 \quad (4.39)$$

$$s_1 = a_1x_1 = s_2 - a_2x_2 \quad (4.40)$$

şeklinde yazılır.

Yinelemeli formülü ise;

$$F_n(x_n) = \max\{f_n(x_n) + F_{n-1}(s_{n-1})\} \quad (4.41)$$

$$F_1(S_1) = \max\{f_1(x_1)\} \quad (4.42)$$

$$F_2(S_2) = \max\{f_2(x_2) + f_1(s_1)\} \quad (4.43)$$

..

$$F_{n-1}(s_{n-1}) = \max\{f_{n-1}(x_{n-1}) + F_{n-2}(s_{n-2})\} \quad (4.44)$$

olarak ifade edilir.

Aşağıdaki gibi doğrusal olmayan kısıtlı model denkleminin

$$\text{maks. } z = 3x^2 - x^3 + 5y^2 - y^3 \tag{4.45}$$

$$x + 2y \leq 4$$

$$x, y \geq 0$$

dinamik programlama ile çözümünde iki değişkenli bir model olduğundan aşama sayısı iki olarak belirlenir ve durum değişkenleri kısıtlara göre dönüşüm denklemleri ile ifade edildiğinde, yineleme fonksiyonları hesaplanır;

$$s_1 = x \tag{4.46}$$

$$s_2 = x + 2y \leq 4 \tag{4.47}$$

$$s_1 = x = s_2 - 2y \tag{4.48}$$

$$\text{maksz} = y^2(5 - y) + x^2(3 - x) \text{olarak denklemi düzenlediğimizde,} \tag{4.49}$$

$$F_1(s_1) = \text{maks.}\{x^2(3 - x)\} = \text{maks.}\{(s_2 - 2y)^2(3 - s_2 + 2y)\} \tag{4.50}$$

denklemi açıp 2.dereceden türevini aldığımızda;

$$\left((s_2 - 2y)^2(3 - s_2 + 2y)\right)'' = 0 \tag{4.51}$$

$$\left((s_2^2 - 4s_2y + 4y^2)(3 - s_2 + 2y)\right) =$$

$$\left(3s_2^2 - s_2^3 + 2ys_2^2 - 12ys_2 + 4ys_2^2 - 8y^2s_2 + 12y^2 - 4y^2s_2\right)$$

$$\left(3s_2^2 - s_2^3 + 2ys_2^2 - 12ys_2 + 4ys_2^2 - 8y^2s_2 + 12y^2 - 4y^2s_2\right)'' = 24 - 24s_2 + 48y \tag{4.52}$$

$$24 - 24s_2 + 48y = 0 \quad y = (s_2 - 1)/2 \tag{4.53}$$

$$F_1(s_1) = (s_2 - 2y)^2(3 - s_2 + 2y) = \left(s_2 - \frac{2(s_2-1)}{2}\right)^2 \left(3 - s_2 + \frac{2(s_2-1)}{2}\right) = 2 \tag{4.54}$$

$$F_2(s_2) = \text{maks}\{y^2(5 - y) + F_1(s_1)\} \tag{4.55}$$

$$(y^2(5 - y))'' = 10 - 6y \tag{4.56}$$

$$10 - 6y + F_1(s_1) = 10 - 6y + 2 = 0 \tag{4.57}$$

$$= 10 - \frac{6(s_2 - 1)}{2} + 2 = 0 \qquad s_2 = 5 \rightarrow y = 2$$

$$y = 2,\ x + 2y \leq 4 \rightarrow x = 0 \tag{4.58}$$

$x = 0\,, y = 2$için amaç fonksiyonu z en büyük değeri 12 olmaktadır.

*LINGO ile çözümü ekte verilmiştir.

BÖLÜM 5

DİNAMİK PROGRAMLAMA İLE BİR İŞ GÜCÜ MİKTARI MODELİ UYGULAMASI

Gemi inşa çalışmaları yapılan fabrikalardan oluşan Marmara Bölgesindeki bir tersanede uygulama yapılmıştır. Bu fabrikalar metal işleri ağırlıklı olmak üzere elektrik, elektronik, sac yapımı, ahşap işleri, gemi makinesi yapımı, pervane işlemleri, yükleme/indirme işleri ile sorumlu olarak işlev görmektedir. Fabrikalardaki işçi sayıları; iş yoğunluğu, iş bitirilme süresi, maliyet analizleri de göz önünde bulundurularak farklılık göstermektedir. Önceki çalışmalardan referans alınarak gemi inşa süreci 60 ayda tamamlandığı bilinmektedir. Son gemi yapımı projesi için her fabrika bilinen süre dâhilindeki işçi sayılarını iş yoğunluğuna göre ayrılan dönemler için belirlemiştir.

Uygulama için seçilen bölüm, yükleme / indirme işlemleri fabrikasıdır. Bu fabrika, bir malzemenin ya da bir makinenin iş makinesi ve teknik destekle taşınması işleminin yapıldığı bölümdür. Bulunulan fabrikada işlev gösterildiği gibi ihtiyaç halinde diğer fabrikalara da personel sevk edilir. Alınan proje sürecinde iş yoğunluğu zamanla farklılık göstermektedir.

Fabrikada fazla işçi bulundurma maliyeti ortalama 3500 TL, yeni bir işçi alma maliyeti ise ortalama 4700 TL olarak hesaplanmıştır. Daha önceki gemi inşa tecrübelerinden yola çıkılarak projenin tamamlanma süresi 60 ay olarak tespit edilmiştir. İş yoğunluğu ise ilk 18 ay için diğer aylara göre daha az olmakla birlikte altı aylık süreçlerde değişkenlik gösterdiği bilinmektedir.

Araştırma konusu olan işçi sayılarına yönelik 60 aylık süreçteki fabrika stratejisi; ilk on sekiz ay 5 işçi, sonraki ilk altı aylık dönemde 10 işçi, ikinci altı aylık dönemde 13 işçi,

üçüncü altı aylık dönemde 9 işçi, dördüncü altı aylık dönemde 11 işçi, beşinci altı aylık dönemde 16 işçi, altıncı altı aylık dönemde 12 işçi ve son altı aylık dönemde ise 14 işçi şeklindedir.

5.1 Problem Çözümü

İşçi alımı ve fazla işçi bulundurmanın maliyete neden olduğu bu durumda öngörülen işçi sayılarını dinamik programlama yöntemi ile değerlendirmek için kurulan denklem ve değişken tanımlamaları aşağıdaki gibidir.

Geriye doğru yineleme yöntemi kullanılan modelde;

C_1; fazla işçi bulundurma maliyetini,

C_2; yeni bir işçi alma durumundaki maliyeti,

i ; dönem sayısını,

b_i; en az işçi sayısını,

x_i; i.dönemdeki işçi sayısını,

gösterdiğinde optimizasyon için gerekli denklem;

$C_1(x_i-b_i)+C_2(x_i-x_{i-1})+f_i(x_{i-1})$ olarak ifade edilir.

Sınırlı sürede gerçekleşen bu faaliyette, dönem sayısı ilk on sekiz aylık bölüm birinci aşama olmak üzere, diğer altı aylık dönemler yedi aşamada inceleneceğinden toplamda sekiz aşamada problem çözümü tamamlanacaktır. Geriye doğru yineleme yöntemi nedeniyle son altı aylık 8.aşama ile başlanacaktır.

Alternatif işçi sayısı sunmak için tablo yönteminde hesaplanacak b_i, x_i ve x_{i-1} değerleri Çizelge 5.1'de gösterilmektedir.

Çizelge 5.1 (i) ve (i-1) dönemlerine ait hesaplanacak işçi sayıları

i (aşama)	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
Dönem	55.-60.Ay	49.-54.ay	43.-48.ay	37.-42.ay	31.-36.ay	25.-30. ay	19.-24. ay	1.-18. ay
b_i	14	12	16	11	9	13	10	5
x_i	14	12	16	11	9	13	10	5
x_{i-1}	12	16	11	9	13	10	5	0
			12					
	13		13	10		11		
			14			12	6	
	14		15	11				
			16			13		

$$b_1 = 5b_2 = 10b_3 = 12b_4 = 15b_5 = 11b_6 = 16\, b_7 = 12\, b_8 = 14 \quad (5.1)$$

C_1 = 3500 TL, C_2 = 4700 TL (5.2)

8.aşama

Geriye doğru yineleme yöntemi ile sekizinci aşama için öngörülen işçi sayısı b_8 =14 durumunda başlangıç çözümünde $x_8=b_8$ ve $f_8(x_7) = C_1(x_8 - b_8) + C_2(x_8 - x_7)$ denklemi ile optimizasyon aşamalarına başlanır.

$$b_8 = 14 \;\; x_8 = 14 f_8(x_7) = C_1(x_8 - b_8) + C_2(x_8 - x_7) \quad (5.3)$$

Çizelge 5.2 Sekizinci aşama(x_8 optimizasyonu)

x_7	x_8=14	f_8 (x_7)	x_8
12	3500*(0)+4700*2=9400	9400	14
13	3500*(0)+4700*(1) = 4700	4700	14
14	3500*(0)+4700*(0)=0	0	14

7.aşama

Önceki aşamada hesaplanan değerler doğrultusunda yedinci aşama için öngörülen işçi sayısı b_7= 12'deoptimum 7000 TL ile x_7=14 olduğu görülmektedir.

$$b_7 = 12 \;\; f_7(x_6) = C_1(x_7 - b_7) + C_2(x_7 - x_6) + \; f_8(x_7) \qquad (5.4)$$

Çizelge 5.3 Yedinci aşama (x_7 optimizasyonu)

x_6	x_7= 12	x_7=13	x_7=14	$f_7(x_6)$	x_7
16	3500*(0)+4700*(0)+ 9400 =9400	3500*(1)+4700*(0)+4700= 8200	3500*(2)+4700*(0)+0= 7000	7000	14

6.aşama

Altıncı aşamada öngörülen işçi sayısı b_6=16 için 7000TL değeri minimum maliyet tespit edilmiştir.

$$b_6 = 16 f_6(x_5) = C_1(x_6 - b_6) + C_2(x_6 - x_5) + \; f_7(x_6) \qquad (5.5)$$

Çizelge 5.4 Altıncı aşama (x_6 optimizasyonu)

x_5	x_6=16	$f_6(x_5)$	x_6
11	3500*(0)+ 4700*(5)+ 7000=30500	30500	16
12	3500*(0)+4700*(4)+7000=25800	25800	16
13	3500*(0)+4700*(3)+7000=21100	21100	16
14	3500*(0)+4700*(2)+7000=16400	16400	16
15	3500*(0)+4700*(1)+7000=11700	11700	16
16	3500*(0)+4700*(0)+7000=7000	7000	16

5.aşama

Beşinci aşamada öngörülen işçi sayısıb_5=11 için tablo değerlerine göre x_5=11'de minimum maliyeti 30500 TL olduğu söylenebilir.

$$b_5 = 11 f_5(x_4) = C_1(x_5 - b_5) + C_2(x_5 - x_4) + f_6(x_5) \quad (5.6)$$

Çizelge 5.5 Beşinci aşama (x_5 optimizasyonu)

x_4	x_5=11	x_5=12	x_5=13	x_5=14	x_5=15	x_5=16	$f_5(x_4)$	x_5
9	3500*(0)+ 4700*(2)+ 30500 = 39900	3500*(1)+ 4700*(3) + 25800 = 43400	3500*(2)+ 4700*(4)+ 21100 = 46900	3500*(3) + 4700*(5)+ 16400 = 50400	3500*(4) + 4700*(6)+ 11700 = 53900	3500*(5) + 4700*(7) + 7000 = 57400	39900	11
10	3500*(0)+ 4700*(1)+ 30500 = 35200	3500*(1) + 4700*(2) + 25800= 38700	3500*(2) + 4700*(3) + 21100 = 42200	3500*(3)+ 4700*(4)+ 16400=45700	3500*(4)+ 4700*(5) + 11700=49200	3500*(5) + 4700*(6) +7000=52700	35200	11
11	3500*(0)+ 4700*(0)+ 30500 = 30500	3500*(1) + 4700*(1) + 25800 = 34000	3500*(2) + 4700*(2) + 21100 = 37500	3500*(3) + 4700*(3) + 16400 = 41000	3500*(4) + 4700*(4) + 11700 = 44500	3500*(5) + 4700*(5) + 7000 =48000	30500	11

4.aşama

Dördüncü aşamada öngörülen işçi sayısı b_4=9 için x_4=11'de minimum maliyet değeri görülmektedir.

$$b_4 = 9 f_4(x_3) = C_1(x_4 - b_4) + C_2(x_4 - x_3) + f_5(x_4) \quad (5.7)$$

Çizelge 5.6 Dördüncü aşama (x_4 optimizasyonu)

x_3	x_4=9	x_4=10	x_4=11	$f_4(x_3)$	x_4
13	3500*(0)+ 4700*(0)+ 39900 =39900	3500*(1)+ 4700*(0)+ 35200 = 38700	3500*(2)+4700*(0)+ 30500= 37500	37500	11

3.aşama

Üçüncü aşamada öngörülen işçi sayısı b_3=13 içinx_2= 13 ile Çizelge değeri 37500 TL hesaplanmıştır.

$$b_3 = 13 f_3(x_2) = C_1(x_3 - b_3) + C_2(x_3 - x_2) + f_4(x_3) \quad (5.8)$$

Çizelge 5.7 Üçüncü aşama (x_3 optimizasyonu)

x_2	x_3=13	$f_3(x_2)$	x_3
10	3500*(0)+4200*(3)+ 37500 = 51600	51600	13
11	3500*(0)+4200*(2)+ 37500 = 46900	46900	13
12	3500*(0)+4200*(1)+ 37500 = 42200	42200	13
13	3500*(0)+4200*(0)+ 37500 =37500	37500	13

2.aşama

İkinci aşamada öngörülen işçi sayısıb_2=10 için x_2=10'da minimum maliyeti göstermektedir.

$$b_2 = 10 f_2(x_1) = C_1(x_2 - b_2) + C_2(x_2 - x_1) + f_3(x_2) \quad (5.9)$$

Çizelge 5.8 İkinci aşama (x_2 optimizasyonu)

x_1	x_2=10	x_2=11	x_2=12	x_2=13	$f_2(x_1)$	x_2
5	3500*(0)+ 4700*(5)+ 51600 = 75100	3500*(1)+ 4700*(6)+ 46900 = 78600	3500*(2)+ 4700*(7)+ 42200 = 82100	3500*(3)+ 4700*(8)+ 37500 = 85600	75100	10
6	3500*(0)+ 4700*(4)+ 51600 =70400	3500*(1)+ 4700*(5)+ 46900 =73900	3500*(2)+ 4700*(6)+ 42200 =77400	3500*(3)+ 4700*(7)+ 37500 = 80900	70400	10

1.aşama

İlk 18 aylık dönemi içeren birinci aşamada öngörülen işçi sayısı b_1=5 için Çizelge değerleri verilmiştir.

$$b_1 = 5 f_1(x_0) = C_1(x_1 - b_1) + C_2(x_1 - x_0) + f_2(x_1) \quad (5.10)$$

Çizelge 5.9 Birinci aşama (x_1 optimizasyonu)

x_0	x_1=5	x_1=6	$f_2(x_1)$	x_1
0	3500*(0)+4700*(5)+ 75100=98600	3500*(1)+4200*(6)+ 70400=102100	98600	5

Bu durumda ilk on sekiz ay için şirket politikası neticesinde öngörülen 5 işçi alınması, sonraki altı aylık dönem için 5 işçi daha alınması durumunda projenin 2 yıllık kısmı tamamlanır. 30'uncu ay için 3 işçi daha alınır. 36'ncı ay için şirkete 4 işçi çıkarılması yerine 2 işçi çıkarılması ve 42'nci ay için herhangi bir değişikliğe gidilmemesi önerilir.48'inci ay için öngörülen 5 işçinin alınmasını ve yine 54'üncü ay için 2 işçi çıkarılarak son dönemde değişiklik yapılmaması değerlendirilir. Başlangıçta öngörülen şirket stratejisi gereğince 60 aylık süreç için 22 işçi alım faaliyeti gerçekleştirilecek iken, 18 işçi alımı ile18.800TL maliyet kaybının önüne geçilebileceği görülmektedir.

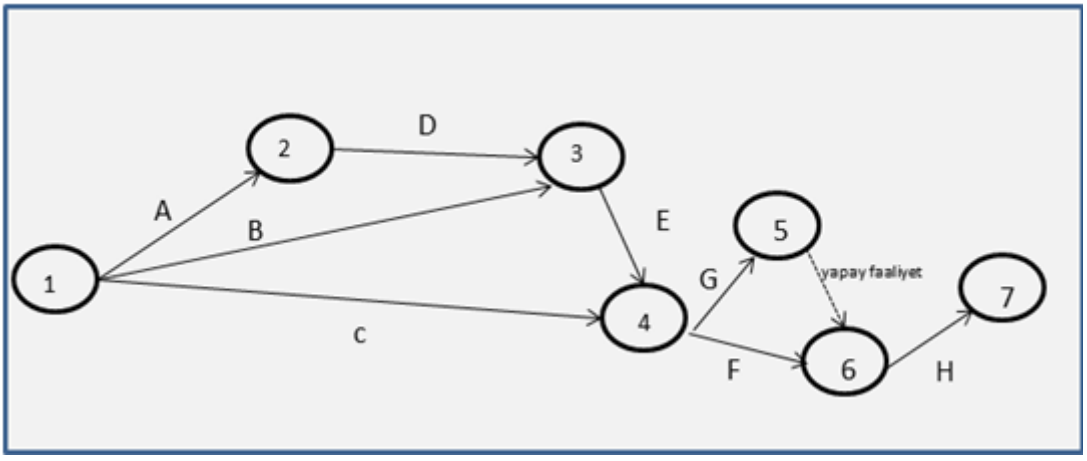
Rakip tersane niteliğinde gelişmiş bir ülkedeki gemi inşa çalışması ile kıyaslandığında işin tamamlanma süresinin 48 ay olduğu üstelik %33 daha az işçi kapasitesi ile gerçekleştirildiği tespit edilmiştir. Bu durumda projenin, uygulama alanını oluşturan yükleme/indirme fabrikasında daha kısa sürede tamamlanma olasılığı araştırılmıştır.

Yükleme fabrikasındaki faaliyetlerin listesi ve bu faaliyetlerin sıralandırılmasını sağlayacak olan öncül faaliyet olma durumları Çizelge 5.10'da verilmektedir.

Çizelge 5.10 Fabrika faaliyet çizelgesi

Faaliyet Kodu	Faaliyet	Öncül Faaliyet Kodu
A	İşçi alımı gerçekleştirme	-
B	Makine-teçhizat hazırlık ve kontrollerini yapma	-
C	Dışarıdan hazır işçi alımı	-
D	Yeni alınan işçi eğitim-stajı gerçekleştirme	A
E	Yükleme/indirme fabrikasında başlangıç çalışmalarını yapma	B-D
F	Metal fabrikasında çalışmaları yapma	E
G	Kaynak fabrikasında çalışmaları yapma	E
H	Yükleme/indirme fabrikasında proje tamamlanma çalışmalarını yapma	F-G

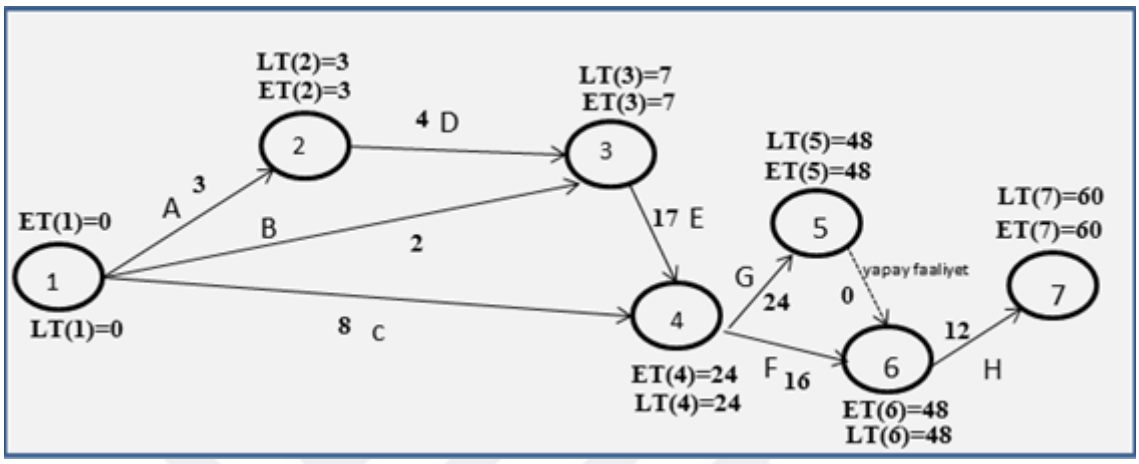
Çizelge 5.10'da verilen faaliyetlerin şema gösterimi Şekil 5.1'de yansıtılmaktadır.



Şekil 5.1 Faaliyet ağı şeması

Erken başlama zamanı ilk düğüm 0 olarak alınır. Kendinden önceki doğrudan bağlanan düğümlerde en büyük olanı ile arasındaki uzaklık toplamı ET(i,j) değerini verir. En geç başlama zamanı ise son düğümde erken başlama zamanı o düğümdeki en geç başlama zamanına eşittir. (LT(n)=ET(n)) Son düğümden başlanarak kendinden sonraki düğümlerde en küçük LT(i,j) değerinden aradaki uzaklık çıkarılır. Projenin tamamlanma

süresinin değiştirilmeksizin bir faaliyetin ne kadar attırılabileceği toplam boşluk miktarı ile anlaşılır. Toplam boşluk miktarı sıfır olan faaliyetler kritik faaliyetleri dolayısıyla kritik yolu oluştururlar. Erken başlama, geç başlama zamanlarıŞekil5.2'de gösterilmektedir.



Şekil 5.2 Erken ve geç başlama zamanları gösterimi

Çizelge 5.11 incelendiğinde toplam boşluk miktarı 0 olmayan yollardaki faaliyetler B, C ve F kritik faaliyet olmadığı bu nedenle kritik yol tercihinin A-D-E-G-H yönünde olması gerekliliği anlaşılmaktadır.

Çizelge 5.11 Toplam boşluk miktarları

Faaliyet	Geçiş Yolu	Toplam Boşluk
		ET(i)-LT(j)-tij
A	1_2	3-0-3=0
B	1_3	7-0-2=5
C	1_4	24-0-8=16
D	2_3	7-3-4=0
E	3_4	24-7-17=0
F	4_6	48-24-16=8
G	4_5	48-24-24=0
H	6_7	60-48-12=0

Elde edilen kritik yol neticesinde bu faaliyetleri içeren sürelerin, fabrika yöneticileri tarafından proje başlangıcında yapılan tahminleri PERT analizi ile değerlendirmek

adına Çizelge 5.12'de verilmektedir. Genelde faaliyet süreleri işgücü ve teknik donanıma bağlı olması nedeniyle kesin olarak bilinmemekte, geçmiş proje deneyimlerine göre tahminleme yapılmaktadır. Sürelerin beta dağılımına uyduğu varsayımı altında kritik yol faaliyetlerinin beklenen süreleri ve varyansları;

$$E(T) = \frac{(a+4m+b)}{6} \quad (5.11)$$

$$V(T) = \frac{(a-b)^2}{36} \qquad a = \text{iyimser süre}, b = \text{kötümser süre}, m = \text{olası süre} \quad (5.12)$$

Çizelge 5.12 Faaliyet sürelerinin beklenen değerleri ve varyansları

Faaliyet	Geçiş Yolu	İyimser Süre	Kötümser Süre	Olası Süre	E(T)	Var(T)
A	1_2	1 ay	5 ay	3 ay	3	0.44
D	2_3	2 ay	6 ay	4 ay	4	0.44
E	3_4	10 ay	24 ay	17 ay	17	5.44
G	4_5	16 ay	32 ay	24 ay	24	7.11
H	6_7	8 ay	16 ay	12 ay	12	1.78

Kritik yol üzerinde bulunan faaliyetlerin beklenen sürelerinin toplam süresi CP değişkenini, bitirilme zamanlarının varyansları toplamı faaliyetlerin varyansını vermektedir. CP değişkeninin normal dağılıma uyduğu varsayımı altında z tablosu yardımıyla projenin 54 ayda tamamlanma olasılığını bulunur.

$$E(CP) = 3 + 4 + 17 + 24 + 12 = 60$$

$$Var(CP) = 0.44 + 0.44 + 5.44 + 7.11 + 1.78 = 15.21$$

$$P(CP \leq 54) = P\left(\frac{CP - 60}{\sqrt{15.21}} \leq \frac{54 - 60}{\sqrt{15.21}}\right) =$$

$$P(z \leq -1.52) = 0.0643$$

54 ayda bitirilme olasılığı % 6'dır.

Bu sonuç kıyaslama yapılan rakip tersane ile arada çok büyük farklılıklar olduğunu da göz önüne sermektedir. Rakip tersanede daha az işçi miktarı ve gün bazında aynı sürede çalışılmalarına rağmen tamamlanma süresindeki 12 ay erken bitirilmesine sebep olarak işgücü verimliği etkeni düşünülmektedir. Sürecin en başına dönülüp işçi alımı

faaliyetlerinde, personel seçim aşamaları gözden geçirilmelidir. Kıyaslama neticesinde işçi eğitimi ve nitelikli oluş durumu süreci yönlendiren önemli etken olduğu, işçi sayısının fazla olmasının proje tamamlanma süresini düşürmediği söylenebilir.

İş gücünün yanı sıra teçhizatın bakım-onarım sürecinin birim başına düşen işin tamamlanma süresine etkisi merak edilmektedir. Bu kapsamda, yükleme iş makinesinin bakımı yapılması ve yapılmaması durumları markov süreci kapsamında dinamik programlama ile model denklemi;

$$f_n(i) = \text{maks}\{\sum p_{ij}^k \left[r_{ij}^k + f_{n+1}(j)\right]\} \quad (5.13)$$

Mevcut yılda birimdeki yükleme işinin tamamlanma süresi ile sonraki yılda tamamlanma süresi arasındaki geçiş olasılıkları markov süreci ile incelenmektedir. Model denklemi değişkenleri;

$f_n(i)$; aşamadaki optimum değeri,

p_{ij} ; durumlar arası geçiş olasılıklarını,

r_{ij} ; durumlar arası geçiş durumunda kazanç ve kayıp değerleri neticesinde elde edilen getiri,

k ; test edilen faaliyetler, (k_1=bakım yapılmaması, k_2=bakım yapılması)

şeklinde tanımlanır.

Geçişler üç durumda, işin optimum süreden kısa sürede tamamlanması, optimum sürede tamamlanması ve optimum süreden daha uzun sürede tamamlanması olarak alınmaktadır. Çizelge 5.13'te üç durum arasında geçiş olasılıkları ve getiri değerleri verilmektedir.

Çizelge 5.13 İş makinesi bakımı yapılmama ve yapılma durumlarında p_{ij}, r_{ij} değerleri

	DURUM	GEÇİŞ OLASILIKLARI			GETİRİ DEĞERLERİ		
BAKIM YAPILMAZSA	t < opt(t)	0.1	0.7	0.2	5400	5000	2900
	t = opt(t)	0	0.4	0.6	0	4700	1200
	t > opt(t)	0	0.2	0.8	0	2500	900
BAKIM YAPILIRSA	t < opt(t)	0.25	0.65	0.1	4400	4000	900
	t = opt(t)	0.2	0.5	0.3	5200	4200	700
	t > opt(t)	0	0.4	0.6	0	3800	400

Çizelge 5.13 değerlerini matris gösterimi ile ifade edilirse;

$P^1 = \|p_{ij}^1\| = \begin{bmatrix} 0.1 & 0.7 & 0.2 \\ 0 & 0.4 & 0.6 \\ 0 & 0.2 & 0.8 \end{bmatrix}$ bakım yapılmaması durumunda mevcut yıl ile sonraki yıl arasında geçiş olasılıkları matrisi,

$P^2 = \|p_{ij}^2\| = \begin{bmatrix} 0.25 & 0.65 & 0.1 \\ 0.2 & 0.5 & 0.3 \\ 0 & 0.4 & 0.6 \end{bmatrix}$ bakım yapılması durumunda mevcut yıl ile sonraki yıl arasında geçiş olasılıkları matrisi,

$R^1 = \|r_{ij}^1\| = \begin{bmatrix} 5400 & 5000 & 2900 \\ 0 & 4700 & 1200 \\ 0 & 2500 & 900 \end{bmatrix}$ bakım yapılmaması durumunda mevcut yıl ile sonraki yıl arasında getiri değerleri,

$R^2 = \|r_{ij}^2\| = \begin{bmatrix} 4400 & 4000 & 900 \\ 5200 & 4200 & 700 \\ 0 & 3800 & 400 \end{bmatrix}$ bakım yapılması dolayısıyla ek maliyete neden olması durumunda elde edilen getiri matrisi şeklinde gösterilir.

Problemin çözümü için;

$$v_i^k = \sum p_{ij}^k r_{ij}^k \tag{5.14}$$

$$f_N(i) = \text{maks}\{v_i^k\}$$

$$f_n(i) = \text{maks}\{v_i^k + \sum p_{ij}^k f_{n+1}(j)\} \tag{5.15}$$

$$f_n(i) = \text{maks}\{\sum p_{ij}^k [r_{ij}^k + f_{n+1}(j)]\} \tag{5.16}$$

formülasyonuna göre elde edilen değerler dinamik programlama çözüm yöntemine göre değerlendirilir. 3 yıllık süreç için geriye doğru yineleme yöntemi ile tablosal gösterimde 3 aşamada optimum sonuca ulaşılır.

$$v_1^1 = (0.1 \times 5400) + (0.7 \times 5000) + (0.2 \times 2900) = 4620$$

$$v_2^1 = (0.4 \times 4700) + (0.6 \times 1200) = 2600$$

$$v_3^1 = (0.2 \times 2500) + (0.8 \times 900) = 1220$$

Yükleme iş makinesi bakımı yapılmaması durumu; geçiş yılı için birim işin optimum süreden kısa sürede tamamlanması halinde 4620 TL, optimum sürede tamamlanması halinde 2600 TL, optimum süreden uzun sürede tamamlanması halinde ise 1220 TL getiri sağlamaktadır.

$$v_1^2 = (0.25 \times 4400) + (0.65 \times 4000) + (0.1 \times 900) = 3790$$

$$v_2^2 = (0.2 \times 5200) + (0.5 \times 4200) + (0.3 \times 700) = 3350$$

$$v_3^2 = (0.4 \times 3800) + (0.6 \times 400) = 1760$$

İş makinesi bakımı yapılması durumunda geçiş yılı için birim işin optimum süreden kısa sürede tamamlanması halinde 3790 TL, optimum sürede tamamlanması halinde 3350 TL, optimum süreden uzun sürede tamamlanmasında ise 1760 TL getiri sağlamaktadır.

Getiri hesabı Tablo 5.14'te özetlenmektedir.

Çizelge 5.14 Durumlar bazında bakım yapılmama ve yapılma faaliyetlerine göre getiri

i	v_i^1	v_i^2
t< opt (t)	4620	3790
t = opt (t)	2600	3350
t > opt (t)	1220	1760

Üçüncü aşamada hesaplanan v_{ij}^k değerleri ile alternatifler bazında maksimum getiri $f_3(i)$ sütununda yer almaktadır.

Çizelge 5.15 Üçüncü aşamada elde edilen getiri değerleri

Aşama 3				
i	k=1	k=2	$f_3(i)$	k^*
t < opt (t)	4620	3790	4620	1
t = opt (t)	2600	3350	3350	2
t > opt (t)	1220	1760	1760	2

$v_i^k + p_{i1}^k f_3(1) + p_{i2}^k f_3(2) + p_{i3}^k f_3(3)$ denklemi kullanılarak elde edilen değerler 2'nci aşama çizelgeye yerleştirilir.

t < opt (t), k=1 → $4620 + (0.1 \times 4620) + (0.7 \times 3350) + (0.2 \times 1760) = 7779$

t < opt (t), k=2 → $3790 + (0.25 \times 3790) + (0.65 \times 3350) + (0.1 \times 1760) = 7091$

t = opt (t), k=1 → $2600 + 0 + (0.4 \times 3350) + (0.6 \times 1760) = 4996$

t = opt (t), k=2 → $3350 + (0.2 \times 4620) + (0.5 \times 3350) + (0.3 \times 1760) = 6477$

t > opt (t), k=1 → $1220 + 0 + (0.2 \times 3350) + (0.8 \times 1760) = 3298$

t > opt (t), k=2 → $1760 + 0 + (0.4 \times 3350) + (0.6 \times 1760) = 4156$

Çizelge 5.16 İkinci aşamada elde edilen getiri

Aşama 2				
i	k=1	k=2	$f_2(i)$	k^*
t < opt (t)	7779	7091	7779	1
t = opt (t)	4996	6477	6477	2
t > opt (t)	3298	4156	4156	2

$v_i^k + p_{i1}^k f_2(1) + p_{i2}^k f_2(2) + p_{i3}^k f_2(3)$ denklemi ile elde edilen değerler 1'inci aşama Çizelgeye yerleştirildiğinde;

t < opt (t), k=1 → $4620 + (0.1 \times 7779) + (0.7 \times 6477) + (0.2 \times 4156) = 10763$

t < opt (t),k=2 → $3790 + (0.25 \times 7779) + (0.65 \times 6477) + (0.1 \times 4156) = 10360.4$

t = opt (t), k=1 → $2600 + 0 + (0.4 \times 6477) + (0.6 \times 4156) = 7684.4$

t = opt (t), k=2 → $3350 + (0.2 \times 7779) + (0.5 \times 6477) + (0.3 \times 4156) = 9391.1$

t > opt (t), k=1 → $1220 + 0 + (0.2 \times 6477) + (0.8 \times 4156) = 5840.2$

t > opt (t), k=2 → $1760 + 0 + (0.4 \times 6477) + (0.6 \times 4156) = 6844.4$

Çizelge 5.17 Birinci aşamada elde edilen getiri

Aşama 1				
i	k=1	k=2	$f_1(i)$	k^*
t < opt (t)	10763	10360.4	10763	1
t = opt (t)	7684.4	9391.1	9391.1	2
t > opt (t)	5840.2	6844.4	6844.4	2

Markov sürecine uygun sistemde dinamik programlama çözüm yöntemi ile elde edilen sonuca göre, iş makinesi bakımının yapılması durumu işin tamamlanma süresine ve getiriye etki etmektedir. Her üç yıl içinde hesaplanan özet tablo değerlerine göre eğer iş optimum süreden kısa tamamlanıyorsa bakım yapılmaması, optimum sürede ya da daha uzun sürede tamamlanıyorsa bakım yapılması öngörülmektedir. Üç yılın sonunda beklenen toplam getiri ise birinci yılda optimum süreden kısa sürede ise 10763 TL, optimum sürede ise 9391.1 TL, optimum süreden uzun ise 6844.4 TL'dir. Ortalama süreden daha kısa sürede işin tamamlanmasının sağladığı getiri sonraki yıl için bakıma gidilmesine, bakımın sebep olacağı ek maliyete katlanmaya gerek olmadığını göstermektedir. Aksi durumlarda ise bakım yapılmaması durumlarında sonraki yılda durum daha iyiye gitme olasılığı da düşük olması nedeniyle bakımı gerekli kılmaktadır.

Farklı bölümlerden ve birçok alt iş bölümünden oluşan fabrikada, yapılan her bir işin tamamlanma süresi toplam süreye etki etmektedir. Burada işgücü ve teçhizat en önemli iki etken olduğu görülmektedir. Bu iki etkenin maliyet unsuru göz önünde bulundurularak strateji belirlenmesi yapılan çalışmalarda ortak yönlendiricidir. Uygulamanın başlangıcında yükleme fabrikasındaki toplam işin tamamlanma süresi

önceki senelerden elde edilen bilgiye göre sabit olduğu durumda, işçi sayılarına maliyet değişkeni göz önünde bulundurularak alternatif sunulurken, yükleme işinin gerçekleştirildiği taşıyıcı makinenin bakım faaliyetinin yapılma durumu da maliyet unsuru dahilindeki getiri ile değerlendirilmektedir. Dinamik programlama yöntemi ile markov sürecine uygun sistemde, mevcut yıldaki durum (birim işin daha kısa sürede tamamlanma ya da tamamlanmaması) sonraki yıl için bakım yapılıp yapılmamasına karar vermeyi sağlamaktadır.

BÖLÜM 6

SONUÇ ve ÖNERİLER

Dinamik programlamanın kullanılabildiği oldukça geniş çerçevedeki örneklerine de yer verilen bu çalışmada, uygulama alanı olarak seçilen bir tersanede proje başlangıcında belirlenen işçi alımı stratejisi değerlendirilmiştir.

Yapılan tablosal gösterim analizinde, fabrikanın proje başlangıcında belirlediği strateji değerlendirilmiştir. Dinamik programlama yöntemi ile alternatif sunulmuştur. Uygulama sahası fabrikada, hazırlık aşamasının ardından diğer fabrikalarda çalışma dönemlerinin yoğunlaşması durumlarına göre işleyiş hızlanmaktadır. Yeni bir işçi alımının ve fazla işçi bulundurmanın belirli maliyeti getirmesi dikkate alınarak önerilen gerekli işçi sayılarına göre, ilk 1.5 yıllık dönemde 5 işçi alımına ihtiyaç duyulmaktadır. 19'uncu aydan itibaren ise dönemsel artışlarda alım iki katına çıkarken, değişikliğe gidilmemesinin maliyet azaltıcı etken olduğu durumlarla da karşılaşıldığı söylenebilir. Bu durumda, ilk 18 aylık dönemde 5 işçi alarak işe başlanılması, ikinci dönemde 5 işçi daha alınması ve üçüncü dönemde 3 işçi alınması fakat dördüncü dönemde 4 işçi çıkarmak yerine 2 işçi çıkarıp, beşinci dönemde değişikliğe gidilmemesi değerlendirilmektedir. Altıncı dönemde 5 işçi alınması yine yedinci dönemde 2 işçi çıkarılarak sekizinci dönemde değişiklik yapılmaması durumu önerilmektedir. Dinamik programlama uygulaması sonucu alınan değerlerde, fabrika stratejisine karşılık işçi alımı faaliyetine maliyet azaltıcı bir teklif olarak görülmektedir.

Rakip tersane niteliğindeki fabrikalara bakıldığında, toplam sürenin bir yıl kadar daha az üstelik dönem bazında çalışan işçi sayısının daha az olduğu dikkat çekmektedir.Toplam sürede azalış ihtimali değerlendirilmek üzere faaliyet adımları ele alınmıştır. Öncül faaliyetler ve faaliyetlerin tahmini tamamlanma süreleri bilgileri ile

proje ağı şeması çizilmiştir. Tamamlanma süreleri ile kritik faaliyetlerden oluşan yol çıkarılmıştır. İşçi alımı, işçi eğitimi-stajı tamamlanması, yükleme fabrikasında hazırlık çalışmaları yapılması, kaynak fabrikasında çalışmalara devam edilmesi, yükleme fabrikasında işin tamamlanması aşamalarından oluşan süreç kritik yolu oluşturmaktadır. 60 aylık sürede tamamlanan işin 54 ayda tamamlanma olasılığı ise PERT analizi sonucuna göre % 6 ile oldukça düşüktür. Her aşamada faaliyetler daha uzun sürmektedir.

İşgücü verimliliği düşük olmasının başlıca neden olmakla birlikte teknolojik donanımın da etkin bir sebeptir. İşçi alımı politikalarında değişikliğe gidilmesi, kriterlerin yeniden düzenlenebilmesi ve ek maliyete sebep olmasına rağmen tamamlanma süresindeki muhtemel azalma hedefi doğrultusunda işçi eğitimine ayrılan bütçenin yeniden değerlendirilmesi önerilmektedir. Teknolojik donanım kapsamında ise teçhizat bakımının birim işin süresine etkisi markov sürecine uygun dinamik programlama modeli ile analiz edilmiştir. Bakımın neden olduğu ek maliyet göz önünde bulundurularak hesaplanan getiri değerleri, mevcut yılda ortalama süreden kısa sürede tamamlanan iş olması durumunda sonraki yıl için bakım yapılmamasını diğer hallerde teçhizat bakımının ihmal edilmemesini gerekli kılmaktadır.

Sonuç olarak, dinamik programlama formülasyonu kullanılarak elde edilen sonuç değerleri ile işçi alımı-çıkarımı faaliyetlerine maliyeti azaltıcı alternatif sunulmuş, mevcut durum analiz edilmiştir. Yaygın örneklerde istatistiksel uygulamalar denenmiştir. Dinamik programlama yönteminde, her uygulama konusu için farklı modeller geliştirmek gerekmektedir. Bütünü ayrıştırma yöntemini gerektirdiğinden karmaşık görünen problemler de dahil olmak üzere süreci daha açık göstererek en iyi sonuca ulaşmayı sağlamaktadır. İlk adımdan başlanarak elde edilen optimum sonuçla son adıma kadar ilerlemektedir. Hesaplamada kolaylık sağlandığı, yöneylem araştırma felsefesini her uygulamada göz önüne çıkardığı, adım adım ilerleyerek her aşamada en iyi kararı almaya yönlendirdiği yapılan bütün çalışmalardaki ortak nitelik olarak söylenebilir.

KAYNAKLAR

[1] Öztürk A., (2013), "Yöneylem Araştırmasının Tarihi Gelişimi ve Özellikleri", Alphanumeric Journal, 1, 1:11.

[2] Aygüneş, H., Binay, S., Çetin, A., ve Oral, H. (2001). Yöneylem Araştırması Ders Kitabı, Kara Harp Okulu Matbaası, Ankara.

[3] Genel Kurmay Başkanlığı, (2000). Savunma Planlama Sürecinde Yöneylem Araştırması, Ankara.

[4] Bellman, R.E., (1957). Dynamic Programming, Princeton University Press., New Jersey.

[5] Demir M.H., (1974). 'Dinamik Programlama', İstanbul Üniversitesi İşletme Fakültesi Dergisi, 3:296-304,

[6] Dreyfus, S., (1961). Dynamic Programming, Sons Inc, New York.

[7] Taha, H.A., (2010). Yöneylem Araştırması, 6.baskı, Literatür Yayınevi, İstanbul.

[8] Shampling, J.E., Stevens, G.T., (1974). Operations Research; A Fundemental Approach, Mc Graw Hill Book Co., New York.

[9] Karayalçın, İ.İ., (1993). Yöneylem (Hareket) Araştırması, 3. Baskı, Menteş Kitabevi, İstanbul.

[10] Render, B.,Stair, Jr. R. M. ve Hama, M.E., (2009). Quantitative Analysis for Management, 10.Baskı, Prentice Hall, United States.

[11] Dabak Ö., Yöneylem Araştırması-2 Dinamik Programlama, www.itu.edu.tr>pdf>dinamik_slayt, Mart 2016.

[12] Winston, L.W., (1994). Operations Research: Applications and Algorithms, 3.Baskı, Duxbury Press, Boston.

[13] Sezen, H.K., (2007). Yöneylem Araştırması, 2. Baskı, Ekin Yayınevi, Bursa.

[14] ÇİÇEK, D.S, (2000). Otel İşletmelerinde Dinamik Programlamaya Dayalı Oda Fiyatının Belirlenmesi, Yüksek Lisans Tezi, Anadolu Üniversitesi Sosyal Bilimler Enstitüsü, Eskişehir.

[15] Hillier, F.S.,Lieberman, G.J. (2009). Introduction to Operation Research, 9.Baskı, Mc. Graw-Hill, New York.

[16] Patır S., (2009). Dinamik Programlama ve Bir Ecza Deposunun Şehir İçi İlaç Dağıtımına Alternatifli Bir Çözüm Önerisi, Atatürk Üniversitesi İktisadi ve İdari Bilimler Dergisi, 23(2) : 63-79.

[17] Yücel M. ve Ulutaş A., 2010. Dinamik Programlamanın İşçilik Maliyetlerinin Minimizasyonunda Uygulanması, Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, C.15, (3) : 271-290.

EK-A

AYNI PROBLEMİN LINGO PROGRAMI İLE ÇÖZÜMÜ

```
!10 düğümlü en kısa yol problemi;
sets:
        dugum / 1..10 /;
        akis (dugum,dugum) : m,e;
endsets

min=@sum(akis:m*e);
@sum(dugum(j)|m(1,j) #ne# 0:e(1,j))=1;
@sum(dugum(i)|m(i,10) #ne# 0:e(i,10))=1;
@for(dugum(k)|k #ne# 1 #AND#k #NE#10:
 @SUM(dugum(i)|m(i,k)#ne#0:e(i,k))=@SUM(dugum(j)|m(k,j) #ne#
0:e(k,j)));
@for(akis|m#ne#0:@BIN(e));

data:
m= 0,2,3,5,0,0,0,0,0,0,
   0,0,0,0,8,7,4,0,0,0,
   0,0,0,0,6,9,8,0,0,0,
   0,0,0,0,9,7,3,0,0,0,
   0,0,0,0,0,0,0,4,6,0,
   0,0,0,0,0,0,0,11,9,0,
   0,0,0,0,0,0,0,15,13,0,
   0,0,0,0,0,0,0,0,0,6,
   0,0,0,0,0,0,0,0,0,5,
   0,0,0,0,0,0,0,0,0,0;

enddata
end
```

```
 Global optimal solution found.
 Objective value:                              19.00000
 Objective bound:                              19.00000
 Infeasibilities:                              0.000000
 Extended solver steps:                               0
 Total solver iterations:                             0
 Elapsed runtime seconds:                          0.28

 Model Class:                                      MILP

 Total variables:                   100
 Nonlinear variables:                 0
 Integer variables:                  20

 Total constraints:                  11
 Nonlinear constraints:               0

 Total nonzeros:                     60
 Nonlinear nonzeros:                  0


                      Variable           Value        Reduced Cost
                      M( 1, 1)       0.000000            0.000000
                      M( 1, 2)       2.000000            0.000000
                      M( 1, 3)       3.000000            0.000000
                      M( 1, 4)       5.000000            0.000000
                      M( 1, 5)       0.000000            0.000000
                      M( 1, 6)       0.000000            0.000000
                      M( 1, 7)       0.000000            0.000000
                      M( 1, 8)       0.000000            0.000000
                      M( 1, 9)       0.000000            0.000000
                     M( 1, 10)       0.000000            0.000000
                      M( 2, 1)       0.000000            0.000000
                      M( 2, 2)       0.000000            0.000000
                      M( 2, 3)       0.000000            0.000000
                      M( 2, 4)       0.000000            0.000000
                      M( 2, 5)       8.000000            0.000000
                      M( 2, 6)       7.000000            0.000000
                      M( 2, 7)       4.000000            0.000000
                      M( 2, 8)       0.000000            0.000000
```

M(2, 9)	0.000000	0.000000
M(2, 10)	0.000000	0.000000
M(3, 1)	0.000000	0.000000
M(3, 2)	0.000000	0.000000
M(3, 3)	0.000000	0.000000
M(3, 4)	0.000000	0.000000
M(3, 5)	6.000000	0.000000
M(3, 6)	9.000000	0.000000
M(3, 7)	8.000000	0.000000
M(3, 8)	0.000000	0.000000
M(3, 9)	0.000000	0.000000
M(3, 10)	0.000000	0.000000
M(4, 1)	0.000000	0.000000
M(4, 2)	0.000000	0.000000
M(4, 3)	0.000000	0.000000
M(4, 4)	0.000000	0.000000
M(4, 5)	9.000000	0.000000
M(4, 6)	7.000000	0.000000
M(4, 7)	3.000000	0.000000
M(4, 8)	0.000000	0.000000
M(4, 9)	0.000000	0.000000
M(4, 10)	0.000000	0.000000
M(5, 1)	0.000000	0.000000
M(5, 2)	0.000000	0.000000
M(5, 3)	0.000000	0.000000
M(5, 4)	0.000000	0.000000
M(5, 5)	0.000000	0.000000
M(5, 6)	0.000000	0.000000
M(5, 7)	0.000000	0.000000
M(5, 8)	4.000000	0.000000
M(5, 9)	6.000000	0.000000
M(5, 10)	0.000000	0.000000
M(6, 1)	0.000000	0.000000
M(6, 2)	0.000000	0.000000
M(6, 3)	0.000000	0.000000
M(6, 4)	0.000000	0.000000
M(6, 5)	0.000000	0.000000
M(6, 6)	0.000000	0.000000
M(6, 7)	0.000000	0.000000
M(6, 8)	11.00000	0.000000
M(6, 9)	9.000000	0.000000

M(6, 9)	9.000000	0.000000
M(6, 10)	0.000000	0.000000
M(7, 1)	0.000000	0.000000
M(7, 2)	0.000000	0.000000
M(7, 3)	0.000000	0.000000
M(7, 4)	0.000000	0.000000
M(7, 5)	0.000000	0.000000
M(7, 6)	0.000000	0.000000
M(7, 7)	0.000000	0.000000
M(7, 8)	15.00000	0.000000
M(7, 9)	13.00000	0.000000
M(7, 10)	0.000000	0.000000
M(8, 1)	0.000000	0.000000
M(8, 2)	0.000000	0.000000
M(8, 3)	0.000000	0.000000
M(8, 4)	0.000000	0.000000
M(8, 5)	0.000000	0.000000
M(8, 6)	0.000000	0.000000
M(8, 7)	0.000000	0.000000
M(8, 8)	0.000000	0.000000
M(8, 9)	0.000000	0.000000
M(8, 10)	6.000000	0.000000
M(9, 1)	0.000000	0.000000
M(9, 2)	0.000000	0.000000
M(9, 3)	0.000000	0.000000
M(9, 4)	0.000000	0.000000
M(9, 5)	0.000000	0.000000
M(9, 6)	0.000000	0.000000
M(9, 7)	0.000000	0.000000
M(9, 8)	0.000000	0.000000
M(9, 9)	0.000000	0.000000
M(9, 10)	5.000000	0.000000
M(10, 1)	0.000000	0.000000
M(10, 2)	0.000000	0.000000
M(10, 3)	0.000000	0.000000
M(10, 4)	0.000000	0.000000
M(10, 5)	0.000000	0.000000
M(10, 6)	0.000000	0.000000
M(10, 7)	0.000000	0.000000
M(10, 8)	0.000000	0.000000
M(10, 9)	0.000000	0.000000
M(10, 10)	0.000000	0.000000

Doğrusal olmayan kısıtlı modelin LINGO ile çözümü ise;

```
MAX=3*x^2-x^3+5*y^2-y^3;
x+2*y<=4;
END
```

```
Local optimal solution found.
Objective value:                              12.00000
Infeasibilities:                              0.000000
Extended solver steps:                               5
Total solver iterations:                            21
Elapsed runtime seconds:                          1.34

Model Class:                                       NLP

Total variables:                     2
Nonlinear variables:                 2
Integer variables:                   0

Total constraints:                   2
Nonlinear constraints:               1

Total nonzeros:                      4
Nonlinear nonzeros:                  2


                     Variable           Value        Reduced Cost
                            X        0.000000            0.000000
                            Y        2.000000           -8.000000

                          Row    Slack or Surplus      Dual Price
                            1        12.00000            1.000000
                            2        0.000000            0.000000
```

ÖZGEÇMİŞ

KİŞİSEL BİLGİLER

Adı Soyadı :Eylül YILDIRIM

Doğum Tarihi ve Yeri :13.12.1985 Kayseri

Yabancı Dili :İngilizce

E-posta :eyyildirim85@hotmail.com

ÖĞRENİM DURUMU

Derece	Alan	Okul/Üniversite	Mezuniyet Yılı
Lisans	İstatistik	Dokuz Eylül Üniversitesi	2008
Lise	Fen Bilimleri	Kocasinan Anadolu L.	2003

İŞ TECRÜBESİ

Yılı	Firma/Kurum	Görevi
2011-..	Eğitim Merkezi	İstatistikçi
2010	İpek Mobilya	Mühendislik Birimi(Kalite Kontrol)