

DİNAMİK PROGRAMLAMA VE BİR ECZA DEPOSUNUN ŞEHİR İÇİ İLAÇ DAĞITIMINA ALTERNATİFLİ BİR ÇÖZÜM ÖNERİSİ

Sait PATIR^(*)

Özet: Günümüz rekabet ortamında, bir dizi kararlar olarak ürün veya hizmeti iyileştirme çabaları kaçınılmazdır. Hizmetin veya verilen siparişin en kısa sürede ve en iyi kalitede karşılanması iş dünyasının temel prensibidir. En iyi kararın alınmasına yardım eden birçok yöntem sayılabilir. Bu tekniklerden biri de, Dinamik programlama tekniğidir. Dinamik programlama, sorunu parçalara ayırarak, her parça veya durum için en iyi kararı almaya ve nihayetinde de optimal karara ulaşmayı hedefleyen bir optimallik tekniğidir. Bu nedenle, bir ilaç deposunun, dağıtım sorunu ele alınmıştır. Dinamik programlama ile en kısa yol bulunmuştur. Elde edilen bulgularla, şehir içi eczaneler dokuz bölgeye ayrılmış ve gelen siparişlerin en kısa zamanda karşılanması için alternatif bir yol belirlenerek, işletmeye önerilmiştir.

Anahtar Kelimeler: Rekabet, Karar Verme, Dinamik Programlama, En Kısa Yol, Sipariş Verme, İlaç Depolama

Abstract: In the competition world today, it is inevitable to take decisions in order to improve quality of goods and services. In today's business worlds the based principle is to delivery the service or placed orders in shortest time period and at best quality. There are a number of methods helping to take the best decision. One of these techniques is The Dynamic Programming. Dynamic programming is an optimization technical that divides the problem into parts and aims finally to reach to the optimal decision in every part and every situation. For his reasons a delivery problem of a medical store is chosen as a case study. The shortest way is found by using Dynamic Programming. With the data reached pharmacies in the city center are divided into nine regions and a new and alternative distribution way is found and offered in order to deliver the incoming orders in the best and the shortest way possible.

Key Words: Competition, Decision Making, Dynamic Programming, the Shortest Way, Ordering, Medical Storage

I. Giriş

Günümüz rekabet dünyasında, müşteri memnuniyeti ön planda tutulmakta ve bir defalık satın alım değil, süreklilik arz eden bir satın alma alışkanlığının oluşturulması arzu edilmektedir. Sürekli satın alım, müşterinin istedikleri ve arzu ettiklerinin mal ve hizmette bulunması halinde mümkün olabilecektir. İşletmeler, mal ve hizmetlerini müşterinin istediği yer, zaman ve miktarda bulundukları sürece başarılı sayılabilirler. Sağlık dünyasında bu daha da önem arz etmektedir. Bir eczaneden sipariş edilen, önemli bir ilaç için beklenen her saniyenin telafisi olmayan bir sonucu olabilir. Bu gibi durumlarda zamanlama ve siparişin istenilen sürede hasta yakınına ulaştırılması hayati önem taşımaktadır. İster sağlık sektörü isterse de iş dünyasında

^(*) Yrd.Doç.Dr. İnönü Üniversitesi İİBF İşletme Bölümü

kaybedilen ve memnun edilmeyen müşterinin telafisi çoğu zaman olamamaktadır

Çalışmada, sağlık sektöründe hizmet veren bir ilaç deposunun, eczanelerin ilaç siparişlerini zamanında ve en düşük maliyetle karşılama sorununa çözüm üretilmeye çalışılmıştır. Bu nedenle, Dinamik Programlama tekniği ele alınarak değerlendirilmiş ve eczaneler için bölgenin bir uçtan diğerine ulaşan en kısa yol bulunmuştur. Siparişlerin karşılanmasında bu yol takip edilerek zaman ve maliyet tasarrufu sağlanabileceği, önerilmektedir.

II. Dinamik Programlama

A.Tanımı Ve İlkeleri

Dinamik programlama, Richard Ernest Bellman tarafından 1950 yılında geliştirtmiş ve isimlendirilmiştir(Bellman,1957). Bu isim çok aşamalı karar süreçlerini ifade etmek adına da kullanılmaktadır(Dreyfus,2002,48-51). Dinamik programlama, tipik olarak problemi aşamalar halinde çözer ve her bir aşamada en uygun bir çözümü bulmaya çalışır. Farklı aşamalardaki hesaplamalar, en nihayetinde optimal bir çözümün elde edildiği yinelenen hesaplamalarla ilişkin dirilerek elde edilir (Taha,1987, 344).

Dinamik programlama çok yaygın bir kullanım alanı olan yineleme denklemleri ile en uygun şekilde sokma tekniğidir. Yineleme denklemleri ile optimizasyon yöntemi deyimi optimizasyonun bir önceki aşamanın içerdiği bilgilerden yararlanarak aşama aşama yapıldığını anlatır. Diğer matematik programlama yöntemlerinde de en iyi çözüme aşamalarla gidilmektedir, ancak yaklaşım yineleme değil iteratiftir. Örneğin doğrusal programlama iteratiftir. Dinamik programlama ise her aşama daha önceki aşamalar ile sırasal olarak ilişki içerisindedir. Her bir aşamadaki çözüm kendi başına problemin bir çözümü değil, ancak optimal bir çözümün bir parçasını belirleyen bilgiyi içermektedir(Shamplig –Stevens,1967,365).

Doğrusal programlamanın aksine dinamik programlama mevcut problemi birbiri ardından işlem gören parçalara ayırmıştır doğrusal programlamada n sayıda bilinmeyen arasında (m+1) tane bağıntı kurulur. Bu bağıntılardan m tanesi doğrusal şartları tanımlayan kısıtlar ve bir tanesi de optimalliği araştırılan amaç fonksiyonudur. Dinamik programlamada doğrusallık söz konusu değildir ve (mxn) büyüklüğündeki problemi bir bütün (sistem) olarak düşünür. Sadece tüm sistem parçaları ardışık işlemler görür ve ardışık iki işlem arasında fonksiyonel bir bağıntı kurulur(Karayalçın,1993,243). Dinamik programlamanın diğer tekniklere göre üstünlüğü, zaman faktörünü değerlendirmesi yanında, çok boyutlu problemlerin çözümünde etkin hesaplama kolaylığı sağlamasıdır(Doğan,1985).

En büyük dezavantajı bütün problemler için uygun gelen bir modelin ifade edilememesidir. Dolayısıyla her problem kendi içinde formüle edilme zorluğunu taşır.

Dinamik programlamanın en fazla uygulandığı işletme sorunları aşağıdaki gibi sıralanabilir(Tütek- Gümüšoğlu,1994,334).

- Yeniden sipariş kurallarının belirlenmesinde zaman ve nicelik değişkenlerinin saptanması,
- Değişen talep koşullarında üretim programlaması ve istihdam düzeyi,
- Pahalı araç ve gerecin etkin biçimde kullanımını garantilemek üzere yedek parça düzeyinin belirlenmesi,
- Yeni alanlara kaynak dağılımını yapan sermaye bütçelemesi,
- Yapılanları halka geniş ölçüde tanıtabilmek için reklam araçlarının seçimi,
- Kıymetli bir kaynağın bulunmasında sistematik bir aramanın yapılması,
- Karmaşık makinelerin bakım ve onarımlarının programlanması,
- Eskiyen makine ve donanımların yenilenmesi için uzun dönem stratejilerin belirlenmesi,
- Çeşitli makineler için (genellikle rulo türü kağıt, boru, vb) kesme kalıplarının belirlenmesi.

Buna göre en fazla uygulama alanları olarak; döküm ve stok planlama, üretim planlaması, kaynak dağılımı, ulaştırma, yatırım, yenileme gibi sorunlar sayılabilir.

Sorun çözülürken, her bir bölümde dikkatlice durulmalı ve düşünülenin olup olmadığını kontrol edilerek modelin her safhasında aşağıdaki sorulara cevap aranmalıdır(Wagner,1969,331) .

- Politika ya da karar değişkenleri nelerdir?
- Optimal politikayı belirleyen amaç fonksiyonu ya da kriter nedir?
- Sorun, aşama evresinde nasıl karakterize edilebilir ve daha sonra nasıl analiz edilebilir?
- Her aşamadaki durumlar nasıl elde edilebilir?
- Kısıtlar sorunun durumlarını ve politika değişkenlerin uygun değerini nasıl etkiler?

Dinamik programlama problemleri aşağıdaki ilkeler uygulanarak çözüme ulaştırılır(Shamplin ve Diğerleri,age,372);

Çok aşamalı (veya kademeli) problem alt parçalara, adımlara veya tek tek aşamalara bölünür. Buna ayrıştırma veya dekompozisyon denir.

- Her aşamada, her seferinde bir kez olmak üzere(veya yineleme ile) belirli bir en uygun şekilde sokma amacına bağlı kalarak karar verilir.
- Problemin tümüne ait çözümü elde etmek için aşama sonuçları birleştirilir. Birleştirme sonucu politika olarak adlandırılan kararlar dizisi elde edilir.

B. Dinamik Programlamanın Özellikleri

Dinamik programlama problemlerinde kullanılan kavramlar problemin yapısı ne olursa olsun; aşama(stage),durum(state), politika kararı, optimal politika ve yineleme ilişkisi (recursive relationship) olarak ifade edilebilir.

- **Aşama(Stage):** Dinamik Programlama problemlerinde, hesaplamaların problemin alt problemlere parçalanmasıyla aşamalar haline getirilmesidir(Taha,age,346).Aşama, çok aşamalı problemde karar verilmesi gereken noktalar olarak da tanımlanır. Karar probleminin bir parçası olan aşama, kararların verilmesinde ve verilen kararların düzenlenmesinde kullanılmaktadır. Aşama sayısı sürecin uzunluğuyla alakalıdır. Aşama yapısını belirleyen önemli bir özellik sürecin sürekli ya da kesikli olmasıdır. Tam sayılarla ifade edilebilen bir kesikli süreç söz konusu olduğunda aşama değişkeni de kesiklidir(Sezen,2007,202).
- **Durum(State):** Her bir aşamada sistemin veya değişkenin alabileceği değerdir. Başka bir ifadeyle durum, bir aşama ve onu izleyen aşamalara dağıtılan kaynaklardır. Herhangi bir aşamadaki sürecin koşulu olarak da tanımlanan durum kavramı içerdiği tüm bilgiler ve sınırlamaları anlatır. Sistemin verilen bir duruma bağlı olarak ortaya çıkan ilgili aşamadaki eylem seçeneklerine durum değişkenleri adı verilir(Sezen,age,204). Durum kavramı mutlak bir kavram olmayıp, analizin özelliğine bağlıdır. Herhangi bir stok problemi için üretim düzeyi vs. herhangi bir aşamanın durumunu gösterebilir. Bunun yanı sıra herhangi bir aşamadaki durum, daha önceki aşamalarda verilen kararların bir sonucudur. Durum değişkeni kesikli veya sürekli olabilir. Yine bazı hallerde, durum betimlemesi ölçülebilir(cardinally) olabileceği gibi nitelik belirten türden (ordinally) veya karışık(cardinally-ordinally) de olabilir(White,age,23).
- **Politika Kararı(Policy Decision):**Her bir aşamada en uygun seçeneğin tercih edildiği davranış şekline politika kararı denir. Bir sonraki aşamada alınacak kararda da bu alınan karar kullanılacaktır.
- **Optimal Politika(Optimal Policy).** Bütün problem için geçerli olacak bir davranış şeklidir. Örneğin, problemin her bir aşamasında bütün durumlar için optimal politika kararların alınabilmesini içeren bir yol haritasıdır(Frederick ve Diğerleri,1986,337). Çözüm bir aşamadan diğerine sıra önceliğine göre gidilerek elde edilir ve son aşamaya erişildikten sonra her parametre için değerler belirlenerek işlem tamamlanır. Böylece en uygun politika oluşturulmuş olunur. Eğer durum betimlemesi yeterliyse, verilmiş bir durumda başlayan optimal düzenleme, o duruma nasıl gelindiğine değil, sadece o andaki duruma bağlıdır(Sezen, age,205).

- Yineleme İlişkisi(Recursive Relationship):Her bir aşamada tekrarlanan ve çözümü en uygun karara götüren yineleme süreci ve fonksiyonudur. Başka bir ifadeyle, her aşamanın bulunabilecek durumlarda verilebilecek karara göre, bu aşamayı izleyen veya daha önceki aşamanın hangi duruma gelebileceğini belirleyen ilişkilere geçiş fonksiyonları da denmektedir(Kara,1986,202).

-Optimallik Kuramı

İlk kez Richard Bellman tarafından kullanıldıktan sonra isimlendirilen Bellman eşitliği(optimallik eşitliği veya dinamik programlama eşitliği olarak bilinir) Dinamik Programlamada (matematiksel optimizasyon metodu) optimallığe ulaşmak için gerekli bir şarttır(Bellman,2007). Başka bir ifadeyle, bir optimal politikanın özelliği, başlangıç durumu ve başlangıç kararları ne olursa olsun geri kalan kararlar, ilk verilen kararların sonucuna göre optimal bir politika oluşturulur(Telchrow,1975). Dinamik programlamada süreç için optimal politika, alt optimal politikalarından meydana gelir. Başka bir deyişle, optimal bir politikanın her türlü alt politikası da optimaldir(Dreyfus,1961,216).

Başlangıç koşulu aşağıdaki gibi bir problem düşünelim:

$$x_0 \in X, \quad (1)$$

$$\max_{\{x_{t+1}\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t F(x_t, x_{t+1}) = V(x_0), \quad (2)$$

Kısıtlar

$$\begin{aligned} x_{t+1} &\in \Gamma(x_t), \quad t = 0, 1, 2, \dots \\ x_0 &\in X. \end{aligned} \quad (3)$$

X bir durum vektörü veya değişkenleri, t aşamayı, $0 \leq \beta \leq 1$ ise indirgenmiş faktörü temsil eder. Bu problemin Bellman yinelenme eşitliği ile çözümünde;

$$V(x) = \max_{y \in \Gamma(x)} [F(x, y) + \beta V(y)], \forall x \in X \quad (4)$$

Burada,

$$y \in \Gamma(x) \quad (5)$$

Formüldeki;

V(x): Bellman eşitsizliğiyle çözülen V, x durum değişkenlerinin bir fonksiyonu olarak problemin optimala ermesine yardım eden değer fonksiyonudur.

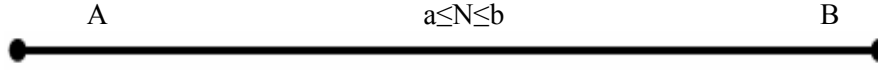
$Y(x)$: Her x durumunda optimal olarak seçildiği düşünülen politika kararıdır (Edward ve diğerleri,1989,67–77).

C. Dinamik Programlama Çeşitleri

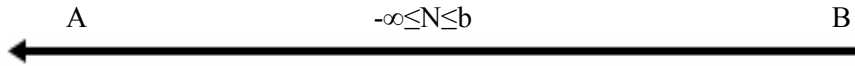
Dinamik programlama problemi, rassallığın bulunduğu durum ve bulunmadığı durum olmak üzere iki şekilde sınıflandırılır. Bu sürecin özelliğini belirleyen şey dönüşüm fonksiyonudur. Süreçle ilgili geçiş fonksiyonunda bir rassallık varsa, stokastik yapılı bir problem değilse, deterministik yapılı bir problemden söz edilir. Dinamik yapılı problemlerde sonlu ve sonsuz durumlar da gözlenebilir. Bir süreçte aşamalar ve durum değerleri sonlu ise çok aşamalı karar süreci de sonludur. Yani, seçeneklerin önceden bilinen sonlu değerleri vardır. Süreç sonsuz uzunlukta ya da pratik olarak çok geniş ise sürecin sonsuz olduğu söylenebilir. Sürecin hakkında hiç bilgi edinemiyor ya da çok az bilgi edinilebiliyor ise Dinamik Programlama problemini ifade etmek imkânsızdır. Sonlu süreçlerin karar değerleri önceden biliniyorsa, bu tür süreçler sonlu bilinen, değilse sonsuz bilinmeyen olarak adlandırılır (Sezen, age,208)

Deterministik ve stokastik dinamik programlama problemleri sonlu ve sonsuz durumlara göre dört şekilde sınıflandırılır (Sezen, age,209). N ; aşama sayısını, a , ve b sınır değerlerini göstermek üzere;

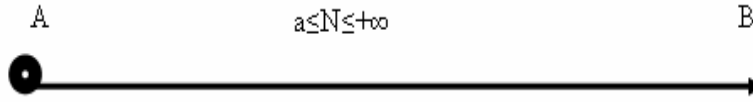
1) Her iki tarafı kapalı durum;



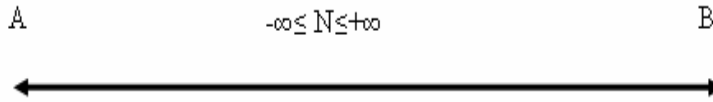
2. Sol taraf açık, sağ tarafı kapalı durum



3. Sağ tarafı açık, sol tarafı kapalı durum.



4. Her iki tarafı açık durum



D. Dinamik Programlamanın Çözüm Yöntemleri

Dinamik programlama problemlerinin çözümüne, uygun bir modelin kurulması ile başlanır. Farklı türdeki modellerin çözümlerinde Tablosal ve

Analitik olmak üzere farklı yöntemler kullanılmaktadır. Tablosal yöntemde her hangi bir süreçle ilgili tüm aşamalar için bütün durumlar göz önünde bulundurulur tüm seçenekler belirlenir her aşama ile ilgili seçenekler arasından en iyileri seçilerek bir tabloya yerleştirilmesi ve bu tablodan hareketle optimal politikanın belirlendiği yöntem olarak ifade edilir. Analitik yöntem ise, verilen dönüşüm denkleminin her bir aşamada türevleri alınarak bu aşamalar için optimal değerlerin bulunmaya çalışıldığı bir yöntemdir. Tablosal yöntem için kesikli verilerin olması ve analitik yöntem için sürekli verilerin kullanıldığı problemlerde kullanılır (Sezen, age, 212)

Dinamik programlama problemleri deterministik ve probabilistik yapıları olmalarına göre iki grupta toplanabilmektedir. Her iki problem türünde de çözüm şekilleri olan ileriye doğru hesaplama ve geriye doğru hesaplama (backward induction method) şekli uygulanabilmektedir.

- İleriye Doğru Hesaplama: İleriye doğru hesaplama, 1. aşamadan başlanarak n. aşamaya doğru hesaplanarak en iyi değerlerin elde edilmesidir. Buna göre en iyi toplam değer; $F_1 + F_2 + F_3 + \dots + F_{n-1} + F_n$ şeklinde olacaktır. ($n = 1, 2, 3, \dots$) Her aşamada karşılaşılan durumundan hareketle n. aşama sonundaki D_n durumunun genel ifadesi ve problemin eniyi durumu; $D_n = G_n(X_n, D_n)$ kısıtları altında

$$\text{Max/Min} = \{F_1(X_1, D_1) + F_2(X_2, D_2) + \dots + F_{n-1}(X_{n-1}, D_{n-1}) + F_n$$

(X_n, D_n) şeklinde oluşacaktır.

- Geriye Doğru Hesaplama: Birbirleriyle ilişkili alt problemlere ayrılmış ardışık karar problemleri için, problemi eniyileyecek (kâr maksimizasyonu/maliyet minimizasyonu) değerlerin n aşamalı bir problem için en son aşamadan (n. aşamadan) başlanarak ilk aşamaya (1. aşama) doğru ilerlenerek bulunması geriye doğru çözüm yaklaşımıdır. F_n n. aşamanın katkısı (getirisi/sonucu) olmak üzere geriye doğru çözüm $F_n + F_{n-1} + \dots + F_3 + F_2 + F_1$ şeklindedir. ($n = 1, 2, 3, \dots$). Her aşamada belirlenecek durumların diğer aşamalarda verilecek kararları etkilediği göz önüne alınırsa n aşamada D_n durumuyla başlayan problemlerin eniyi; $D_{n-1} = G_n(X_n, D_n)$ kısıtları altında $\text{Max/Min} = \{F_n(X_n, D_n) + F_{n-1}(X_{n-1}, D_{n-1}) + \dots + F_1(X_1, D_1)$

karar modeli elde edilebilir.

-Deterministik Dinamik Programlama İle En Kısa Yolun Hesaplanması

Dinamik programlama, n değişkenli bir problemin optimum çözümünü, problemi n aşamaya ayırarak ve her aşamada tek değişkenli bir alt problemi çözerek belirler. Hesaplamalar yinelenerek yapıldığı için bir alt problemin optimum çözümü bir sonraki problemin girdisini oluşturur. Son alt problem çözüldüğünde optimum çözüme ulaşırlar. Algoritmada önemli olan konu problemin nasıl parçalara ayrıştırılacağıdır.

Dinamik programlamanın yinelenen hesaplamaları matematik olarak şu şekilde ifade edilebilir:

$f_i(x_i)$ -----→ i. aşamada x_i . Dügüme en kısa uzaklık,
 $d(x_{i-1}, x_i)$ -----→ x_{i-1} . düğümden x_i . düğüme olan uzaklık olarak tanımlanırsa f_i
aşağıdaki eşitlik yardımıyla f_{i-1} den hesaplanır:

$$f_i(x_i) = \min (d(x_{i-1}, x_i) + f_{i-1}(x_{i-1}))$$

tüm uygun
 (x_{i-1}, x_i) yolları
 $i=1,2,3,4$

Başlangıçta $i=1$ ve $f_0(x_0) = 0$ olarak alınır. Dinamik programlama terminolojisinde x_i den i. aşamada sistemin durumu olarak söz edilir. i. aşamada sistemin durumu, aşamaları birbirine bağlayan bir bilgidir. Durumun uygun bir biçimde tanımlanması, her aşamayı ayrı hesaplamayı sağlar ve çözümün tüm aşamalar için uygun olmasını garantiler.

III. Uygulama

A. Malatya'da Faaliyetlerini Sürdüren Selçuk Ecza Deposunun Şehir İçi İlaç Dağıtımında Dinamik Programlama Uygulaması

Selçuk Ecza Deposu, Malatya'da 30 Mart 1998 yılında 2400 m² 'lik bir alan üzerinde faaliyetlerine başlamıştır. İşletme, 11'i motor kurye olmak üzere 45 araçlık filosu ve 120 personeli ile gelen siparişlere cevap vermektedir. İşletme il dışı; Diyarbakır, Batman, Siirt, Şırnak, Mardin, Elazığ, Bingöl, Tunceli ve Adıyaman illerine hizmet götürmektedir(Denge,2006,6-7).

Şehir içi siparişler 165 eczaneden bilgisayar ortamında alınmakta, 9 bölgeye ayrılan eczanelerin buldukları bölgeler aynı hat üzerindeki siparişler birleştirilerek en kısa sürede cevaplanılmaktadır. Günlük ortalama 900 sipariş verildiği belirlenen eczanelere, her birisine 5,45 kez gidildiği tahmin edilmektedir. Yani işletme her bir eczaneye günde yaklaşık olarak 5,5 kez gitmek durumundadır. İşletme bu siparişleri karşılama sorununu en az maliyet ve en kısa sürede cevaplamak zorundadır. İşletmenin gelen siparişleri karşılamadaki bir gecikme müşteri kaybı ve ek maliyete neden olmaktadır. Bunun için problemin yapısı gereği, bölgeler arasında, en kısa yolun bulunması sorununu, Dinamik Programlama yöntemi ile çözülmeye çalışılmıştır.

B. Problem Cümlesi

Selçuk Ecza Deposu ile, eczaneler buldukları mevkilerin uzaklıklarına ve birbirlerinin yakınlığına göre şehri dokuz ayrı bölgeye ayırmıştır. Deponun bu bölgelere uzaklıkları tablo 1'de metre olarak verilmiştir(Malatya Belediyesi Şehir Planlaması).

Tablo 1: Selçuk Ecza Deposunun Bölgelere Uzaklığı

Bölgeler	Selçuk Ecza Deposu (Metre):(A)
B. Bölge: Atatürk Caddesi (Kışla Caddesi)	1648
C. Bölge: Devlet Hastanesi & Çöşnük	1510
D. Bölge: Beydağı Devlet Hastanesi& Eski Ssk Cıvarı	4770
E. Bölge: Cengiz Topel Paşaköşkü	2394
F.Bölge: Sıtmapınarı İnönü-Emeksiz	4000
G. Akpınar	1639
H.Bölge: Milli Eğemenlik- Emeksiz	2510
I.Bölge: Dabakhane- Ptt	2530
J.Bölge. Yeşilyurt	11000

Bu bölgelerin birbirlerine olan uzaklıkları aşağıdaki tablo 2' de gösterilmiştir.

Tablo 2: Bölgelerin Birbirlerine Uzaklıkları

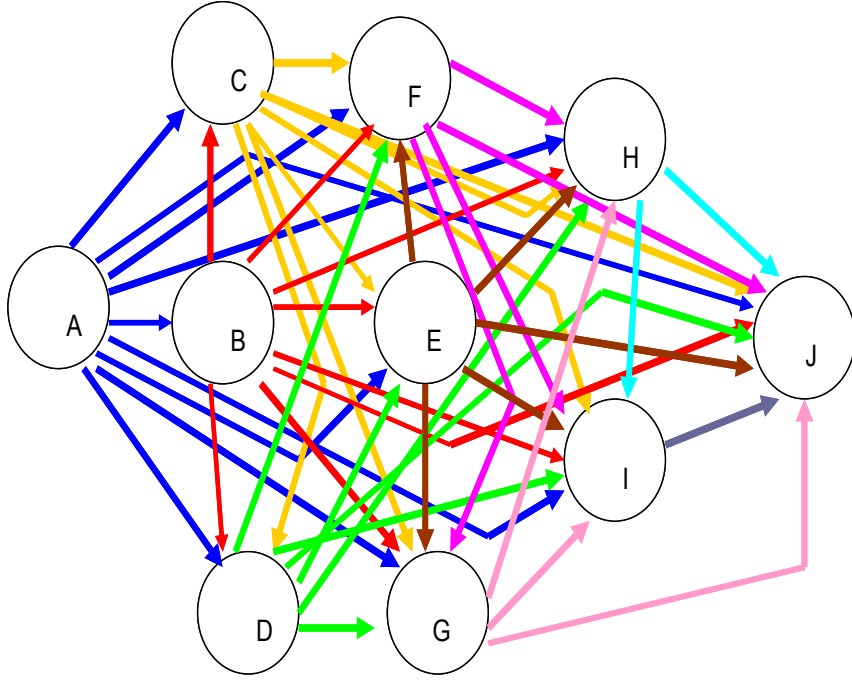
	B	C	D	E	F	G	H	I	J
B	0	1339	3546	1364	1891	291	706	2600	10269
C	1339	0	4532	1434	2898	2898	1326	1471	11400
D	3547	4532	0	4791	1342	3235	2478	2560	8420
E	1364	1434	4791	0	1576	1430	747	858	8973
F	1891	2598	1342	1576	0	1836	1188	1346	7785
G	291	1418	3235	1430	1836	0	654	784	9139
H	706	1326	2478	747	1188	654	0	176	8068
I	2600	1471	2560	858	1346	784	176	0	8168
J	10269	11400	8420	8973	7785	9139	8068	8168	0

C. Tablosal Yöntemle Problemin Çözümü

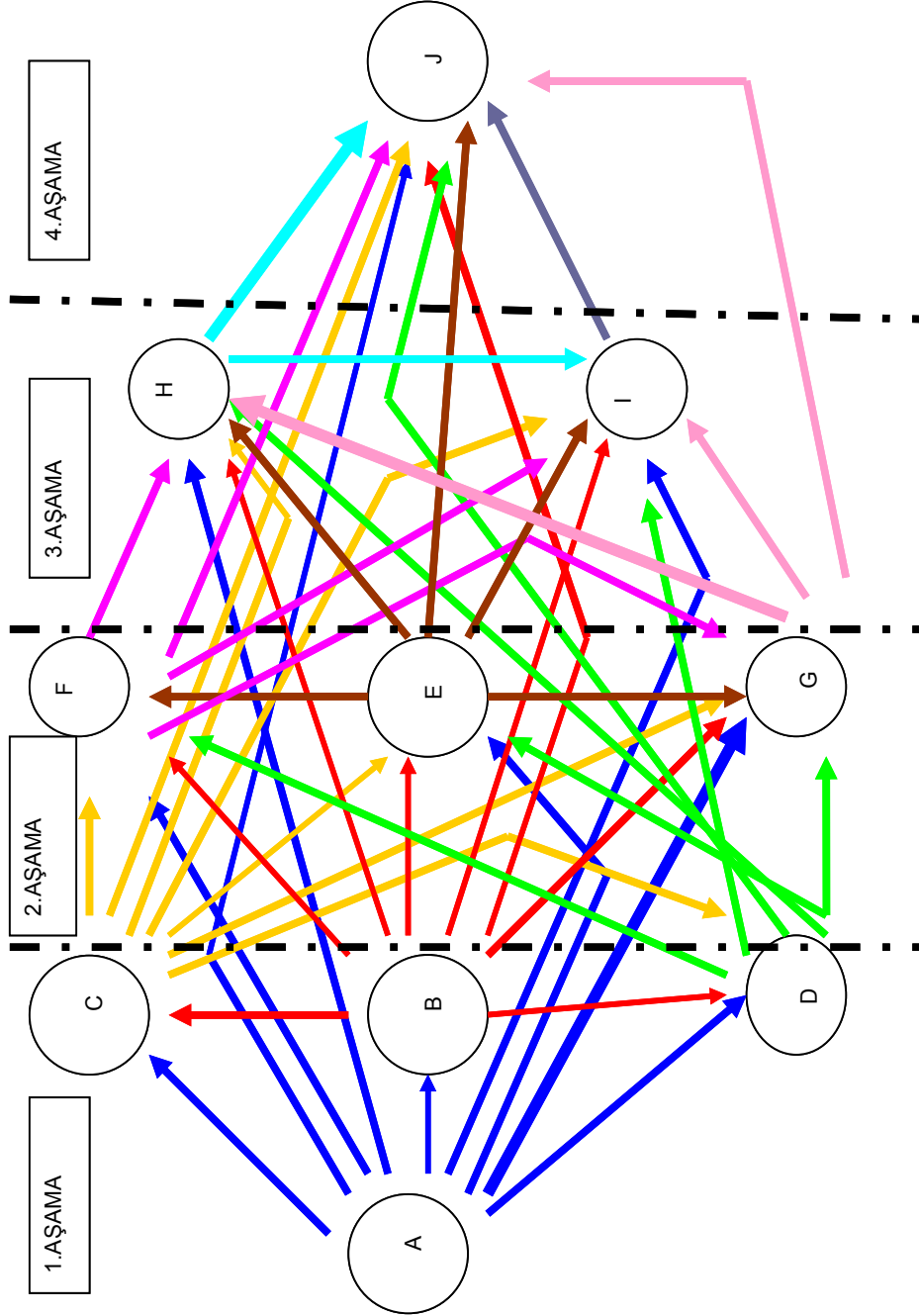
Tablosal Yöntem, her bir aşama için geçerli durum değerlerinin ortaya çıkardığı seçeneklerden ayrı ayrı değerlendirilip bu seçeneklerden en iyisinin seçilmesi biçiminde uygulanmaktadır. Gerek seçeneklerin gözden geçirilmesi, gerekse her bir duruma ilişkin kararların tablolarla taşınıyor olması yönüne bu

adın verilmesine neden olmaktadır. Tablosal yöntemde, optimal politika sonuç tablosunun analizi yoluyla elde edilmektedir (Sezen, age, 220). Uygulamada Tablosal Yöntemle kullanılmıştır, şehir içi dağıtım problemini önce ileriye doğru hesaplama, sonra geriye doğru hesaplama ve son olarak da geçiş fonksiyonu yardımıyla ayrı ayrı çözümler en kısa yol bulunmuştur.

Başlangıç noktasının Ecza deposu alınarak, dokuz bölgenin şebeke diyagramı olarak gösterimi şekil 1'dedir. Bu bölgelerin ileriye doğru yineleme süreciyle elde edilmesi için oluşturulan aşamalar şekil 2'de aşağıdaki gibi elde edilmiştir. Problemin çözümünde tablosal yöntem kullanılmıştır.



Şekil 1: İlk merkezden Dokuz Bölgeye Ulaşımın Şebeke Diyagramı Üzerinde Gösterilmesi



Şekil 2: Bölgelerin Dört Aşamaya Ayrılarak Şebeke Diyagramının Elde Edilmesi

1. İleri Doğru Hesaplama

Problem dört aşama ayrılarak çözümü yapılmaya çalışılacaktır. İleriye doğru hesaplama için; birinci aşamadan başlanarak, dördüncü aşamaya kadar en kısa yollar bulunarak en nihayetinde de optimal değer bulunacaktır.

Birinci Aşama

B.Düğümüne en kısa uzaklık; 1648 m (A,B).

C.Düğümüne en kısa uzaklık; iki yol bulunmaktadır, bunlar içerisinde en kısa olanı seçilir. (A,C), (B,C)

$$\left[\begin{array}{c} \text{C.düğümüne en} \\ \text{kısa uzaklık} \end{array} \right] = \left[\begin{array}{c} \text{Minimum} \\ i=A,B \end{array} \right] \left[\begin{array}{c} \text{i.düğümüne en} \\ \text{kısa uzaklık} \end{array} \right] + \left[\begin{array}{c} \text{i.düğümünden} \\ \text{C.düğümüne olan} \\ \text{uzaklık} \end{array} \right]$$

$$\left[\begin{array}{c} 1648 + 1339 = \\ 2987 \\ 1510 \end{array} \right] \left[\begin{array}{c} \text{min} \\ i=A,B \end{array} \right] = \left[\begin{array}{c} 1510 \text{ (A.düğümünden)} \end{array} \right]$$

D. Düğümüne en kısa uzaklık, iki yol bulunmaktadır. (A,D), (B,D),(CD)

$$\left[\begin{array}{c} \text{D.düğümüne en} \\ \text{kısa uzaklık} \end{array} \right] = \left[\begin{array}{c} \text{Minimum} \\ i=A,B,C \end{array} \right] \left[\begin{array}{c} \text{i.düğümüne en} \\ \text{kısa uzaklık} \end{array} \right] + \left[\begin{array}{c} \text{i.düğümünden} \\ \text{C.düğümüne olan} \\ \text{uzaklık} \end{array} \right]$$

$$\left[\begin{array}{c} 4770 \\ 1648+3546=5194 \\ 1510+4532=6042 \end{array} \right] \left[\begin{array}{c} \text{min} \\ i=A,B,C \end{array} \right] = \left[\begin{array}{c} 4770 \text{ (A.düğümünden)} \end{array} \right]$$

İkinci Aşama

E. Düğümüne en kısa uzaklık; dört yol bulunmaktadır; (A,E),(B,E),(C,E),(D,E) tablo 3’de verilmiştir.

Tablo 3: E Düğümüne En Kısa Uzaklık Seçenekleri

	E. Düğüm.	En Kısa yol
(A,E)	2394	2349 (A,E)
(B,E)	1648+1364=3012	*
(C,E)	1510+1434=2944	*
(D,E)	4770+4791=9561	*

F Düğümüne en kısa uzaklık; beş yol bulunmaktadır. (A,F), (B,F), (C,F), (D,F), (E,F) tablo 4’de verilmiştir.

Tablo 4: F Düğümüne En Kısa uzaklık Seçenekleri

	F. Düğüm	En Kısa Yol
(A,F)	4000	*
(B,F)	1648+1891=3539	3539 (B,F)
(C,F)	1510+2898=4408	*
(D,F)	4770+1342=6112	*
(E,F)	2349+1576=3925	*

G. Düğümüne en kısa uzaklık; altı Yol bulunmaktadır. (0,6), (1,6),(2,6), (3,6),(4,6),(5,6) tablo 5’de verilmiştir.

Tablo 5: G Düğümüne En Kısa Uzaklık Seçenekleri

	G. Düğüm	En Kısa Yol
(A,G)	1639	1639 (A,G)
(B,G)	1648+291 = 1939	*
(C,G)	1510+2898 = 4408	*
(D,G)	4770+3235 = 8005	*
(E,G)	2349+1630 = 3979	*
(F,G)	3539+1836= 5375	*

Üçüncü Aşama

H Düğümüne en kısa uzaklık; yedi yol bulunmaktadır. (A,H), (B,H), (C,H), (D,H), (E,H), (F,H), (G,H) tablo 6’de verilmiştir.

Tablo 6: H. Düğüme En Kısa Uzaklık Seçenekleri

	H Düğüm	En Kısa Yol
(A,H)	2510	*
(B,H)	1648+706= 2354	*
(C,H)	1510+1326= 2836	*
(D,H)	4770+2478= 7248	*
(E,H)	2349+747= 3096	*
(F,H)	3539+1188= 4727	*
(G,H)	1639+654= 2293	2293 (G,H)

I. Dügüme en kısa uzaklık; sekiz yol bulunmaktadır. (A,I), (B,I), (C,I), (D,I),(E,I),(F,I),(G,I) ,(H,I) tablo 7’de verilmiştir.

Tablo 7. I Dügüme En Kısa Uzaklık Seçenekleri

	I. Dügümü	En Kısa Yol
(A,I)	2530	*
(B,I)	1648+2600=4248	*
(C,I)	1510+1471=2981	*
(D,I)	4770+2560=7330	*
(E,I)	2349+858= 3207	*
(F,I)	3539+1346=4885	*
(G,I)	1639+784= 2432	2432 (G,I)
(H,I)	2293+176=2469	*

Dördüncü Aşama

J. Dügüme en kısa uzaklık, dokuz yol bulunmaktadır. (A,J), (B,J), (C,J), (D,J), (E,J), (F,J) ,(G,J) ,(H,J),(I,J) tablo 8’da verilmiştir.

Tablo 8: J Dügümüne En Kısa Uzaklık Seçenekleri

	J. Dügümü	En Kısa Yol
(A,J)	11,000	*
(B,J)	1648+10259=11907	*
(C,J)	1520+11400=12920	*
(D,J)	4770+8420=13190	*
(E,J)	2349+8973=11322	*
(F,J)	3539+7785=11324	*
(G,J)	1639+9139=10778	*
(H,J)	2293+8068=10361	10,361 (H,J)
(I,J)	2432+8168=10600	*

En Kısa Yol: A → G → H → J; 10,361 metre olarak elde edilmiştir.

2. Geriye Doğru Hesaplama

En son aşamadan başlanarak başlangıç noktasına doğru ilerlenecektir.

E,F,G noktasının geriye doğru en kısa uzaklık seçenekleri tablo 9’da verilmiştir.

Tablo 9: E,F,G Noktalarının En Kısa Yol Seçenekleri

Karar	İlk Uzaklık	En Kısa Uzaklık	Toplam Uzaklık	En İyi Karar
E-H	747	8068	8815*	E-H
E-I	858	8168	9026	*
E-G	1430	9139	10569	*
E-F	1576	7785	9361	*
E-J	8973	0	8973	*
F-I	1346	8168	8514	*
F-H	1186	8068	9254	*
F-G	1836	9139	10975	*
F-J	7785	7785	7785*	F-J
G-H	654	8068	8722*	G-H
G-I	784	8168	8952	
G-J	9139	0	9139	

B,C,D noktalarının geriye doğru en kısa uzaklık seçenekleri tablo 10’de verilmiştir.

Tablo 10: B,C,D Noktalarının En Kısa Uzaklık Seçenekleri

Karar	İlk Uzaklık	En Kısa Uzaklık	Toplam Uzaklık	En İyi Karar
B-E	1364	8815	10179	*
B-F	1891	7785	9676	*
B-D	3546	8420	11966	*
B-C	1339	9394	10733	*
B-G	291	8722	9813	*
B-I	2600	8168	10768	*
B-H	706	8722	9428*	B-H
B-J	10269	0	10268	*
C-F	2898	7785	10683	*
C-D	4532	8420	12952	*
C-E	1432	8815	10249	*
C-H	1326	8068	9394*	C-H
C-I	1471	8168	9639	*
C-G	2898	8722	11620	*
C-J	11400	0	11400	*
D-F	1342	7785	9127	*
D-E	4791	8815	13606	*
D-G	3235	8722	11957	*
D-H	2478	8068	10546	*
D-I	2560	8168	10728	*
D-J	8420	0	8420*	D-J

Başlangıç noktası A için en kısa uzaklık seçenekleri, tablo11’de verilmiştir.

Tablo 11: *A Noktasına En Kısa Uzaklık Seçenekleri*

Karar	İlk Uzaklık	En Kısa Uzaklık	Toplam Uzaklık	En İyi Karar
A-B	1648	9394	11042	*
A-C	1510	9394	10904	*
A-D	4770	8420	13190	*
A-E	2390	8815	11205	*
A-F	4000	7785	11785	*
A-G	1639	8722	10361*	A-G
A-H	2510	8068	10578	*
A-I	2530	8168	10698	*
A-J	11000	0	11000	*

En Kısa Yol: A-G-H-J; 10361 metre olarak elde edilmiştir.

IV. Sonuç Ve Öneriler

Dinamik programlama ele aldığı problemi parçalara ayırarak çözüm üretmeye çalışan bir tekniktir. Her bir parçanın en iyi sonucunu bularak, en nihayetinde optimal sonuca ulaşmayı hedefler. En kısa yol problemi, gezilecek veya gidilecek yollar içerisinde en kısa yolu bulmak adına geliştirilen bir yöntemdir. Malatya’da faaliyetini sürdüren Selçuk ecza deposu gerek şehir içi ve gerekse şehirlerarası ilaç siparişlerini karşılamaya çalışan bir ilaç deposudur. Şehir içi 165 eczanelerden gelen siparişleri en kısa sürede karşılamak durumundadır. Şayet zamanından daha geç karşılanması halinde sipariş veren müşterinin kaybına ve dolayısı ile anlaşmalı eczanenin rakip ecza deposuna yönelmesine neden olacaktır. Bunun için işletmenin, şehrin bir ucundan diğer ucuna yayılan eczanelere ulaşmada en kısa yolun elde edilmesine ihtiyaç duymaktadır, işletmenin bu sorununa, Dinamik Programlama Tekniği uygulanmıştır. Önce ileriye doğru hesaplama yapılmış, birinci aşamadan dördüncü aşamaya kadar problem aşama aşama çözülmüştür. Dördüncü aşama sonunda en kısa uzaklığın diyagram üzerindeki düğüm değerleri olarak; A-G-H-J hattı olarak bulunmuştur. Daha sonra geriye doğru hesaplama yapılmış ve aynı sonuç elde edilmiştir. Bu yolun mesafesi 10,361 metre olarak elde edilmiştir. İşletme bu yolu kullanarak zamandan ve maliyetten tasarruf sağlayabilecektir.

İlaç deposuna önermeler olarak;

- Yeşilyurt bölgesinden gelecek bir sipariş için en kısa yol ; “Akpınar-Milli Egemenlik-Yeşilyurt olarak elde edilmişti” dolayısı ile Yeşilyurt için bir siparişte, Akpınar ve Milli Egemenlik bölgelerindeki siparişleri ile birleştirilerek cevaplandırılabilir. Bu işletmeye zaman ve maliyet tasarrufu sağlayacaktır.
- Gelen siparişlerin dokuz bölgenin hangisine düştüğü belirlenerek, bölge içindeki eczanelerin ilaçları birleştirilerek aynı anda dağıtılabilir.
- Bölgelere ayrılan eczanelerin bölgeler arası yakınlıklarından faydalanılarak iki bölgenin siparişleri birleştirilerek dağıtılabilir. Örneğin: Akpınar – Devlet Hastanesi, Akpınar-Tabakhane ve Akpınar-

Cengiz Topel hatları birbirlerine çok yakın oldukları için siparişleri birleştirilerek karşılanabilir. Ayrıca, Kışla Caddesi- Devlet Hastanesi, Kışla Caddesi- Cengiz Topel Caddesi ve Kışla Caddesi – Milli Egemenlik Caddesi siparişleri birleştirilerek karşılanabilir.

Kaynaklar

- Bellman, R. E.,” Dynamic Programming”, Princeton University Press,New Jersey, 1957.
- Denge Dergisi; Selçuk Ecza Deposu Kurumsal Yayınları, Maestro Yayıncılık, Sayı: 9/2006–3,İstanbul.
- Dreyfus, Stuart “Richard Belman On The Birth Of The Dynamic Programming”, Operation Research, 2002,Vol,50,No.1,Pp-48-51. <http://www.eng.tau.ac.il/~ami/cd/or50/1526-5463-2002-50-01-0048.pdf>;Erişim Tarihi, 14.06.2008.
- Dreyfus ,Stuart; Dynamic Programming. Der. Russel L.Ackoff, Progress İn Operation Research,No.5,John Wiley,Sons Inc,New York 1961,S.216.
- Doğan,Muammer.,İşletmelerde Karar Verme Teknikleri,Dokuz Eylül Üniversitesi İibf Yayınları,Bilgehan Basımevi,1985,İzmir
- Hillier, Frederick.S.,Lieberman Geralg J.,Introduction To Operations Research,Holden –Day,İnc.1986,Oakland/California,S.337).
- Kara,İmdat.,Yöneylem Araştırması Yöntem Bilimi.,1985, Eskişehir.
- Kara,İmdat Yöneylem Araştırması: Doğrusal Olmayan Modeller,Anadolu Üniversitesi Basımevi,Eskişehir,1986,S.202.
- Karayalçın,İlhami.,Yöneylem “Hareket”Araştırması Operation Research, Menteş Kitabevi, 1993, İstanbul.S. 243.
- Nancy Stokey, and Robert E. Lucas, with Edward Prescott, 1989. *Recursive Methods in Economic Dynamics*. Harvard Univ. Pres. S.67-77
- Shamplin, J.E.,Stevens,G.T.,Operations Research;A Fundamental Approach Mc-Graw Hill Book Co.,New York,1974,S.365)
- Sezen, Hayrettin K. Yöneylem Araştırması, Eken Yayınevi, 2007,S.204. Bursa.
- Telchroew ,Daniel, “Dinamik Programlama”, Çev. Erdal Akan, B.İ.T.İ.A. Dergisi, Cilt Iv,No.1,Mart 1975.
- Tütek,H.H., Şevkinaz G., Sayısal Yöntemler Yönetmel Yaklaşım, Beta Basımevi,1994,İstanbul,S.334.
- Wagner,Harvey M., Principles Of Operations Research With Applications To Managerial Decisions,Prentice Hall,Inc.,Englewood Cliffs,New Jersey,1969,S.331)
- White, D.J. Dynamic Programming, Oliver And Boyd, Edinburg,1969,S.26.
- http://en.wikipedia.org/wiki/Bellman_equation erişim tarihi 3 02. 2007
- <http://www.blackwell-synergy.com/> erişim tarihi 02,03,2007.
- <http://www.cs.berkeley.edu/~vazirani/algorithms/chap6.pdf>, erişim tarihi 12,06,2008