

DİNAMİK PROGRAMLAMAMANIN İŞÇİLİK MALİYETLERİNİN MİNİMİZASYONUNDA UYGULANMASI

THE APPLICATION OF DYNAMIC PROGRAMMING FOR MINIMIZATION OF THE COST OF LABOURS

Yrd.Doç.Dr. Mustafa YÜCEL*
Alptekin ULUTAŞ**

ÖZET

Artan rekabet şartları içerisinde karlılıklarını korumak ve devamlılıklarını sağlamak isteyen firmalar için maliyetlerin en aza indirilmesi kaçınılmaz bir zorunluluktur. Bu çalışmanın temel amacı inşaat sektöründe çalışan işçilerin maliyetlerinin optimizasyonudur. Çalışmada, işçilik maliyetini optimize etmek için dinamik programlama metodu kullanılıyor. Dinamik programlama özellikle üretim planlaması, envanter kontrolü, yenileme, yatırım planlaması, kaynakların dağıtım, ulaştırma, proje yönetimi, yükleme problemleri gibi sorunlarda çok geniş ve yaygın bir uygulama alanına sahiptir. Çalışma dört aşamadan oluşmaktadır. Birinci aşamada İşçilik Maliyetleri Minimizasyonu. İkinci aşamada, dinamik programlama hakkında genel bilgi verilmiştir. Üçüncü aşamada tablolar yardımı ile çalışmanın uygulaması yapılmıştır. Son olarak, sonuç ve öneriler verilmiştir.

ABSTRACT

It is crucial for the firms, which aim to sustain their profitability and to survive, to reduce their operational costs within an environment where competition kept rising. The basic aim of this study is the optimization of the cost of labours who work at construction sector. In this study, dynamic programming which is method was used to optimize the cost of labours. Dynamic programming has a very broad and common practice in solving such problems as especially production planning, inventory control, renovation, investment planning, resources distribution, transportation, project management, and installation problems. The study consists of four stages. First stage, Labor Costs Minimization. Second stage, general information about dynamic programming is provided. Third stage, the

* İnönü Üniversitesi, İ.İ.B.F, İşletme Bölümü, Sayısal Yöntemler Anabilim Dalı

** İnönü Üniversitesi, İ.İ.B.F, İşletme Bölümü, Üretim Yönetimi ve Pazarlama Anabilim Dalı

application of study was acquired by means of tables. Finally, result and recommendations was procured.

Dinamik Programlama, İşçilik Maliyetleri Minimizasyonu, Yöneylem Araştırması, Kantitatif Karar Verme Teknikleri, İşgücü Planlama
Dynamic Programming, Labor Costs Minimization, Operations Research, Quantitative Decision Making Techniques, Workforce Planning.

GİRİŞ

İşletmeler mal ve hizmet üretebilmek için çeşitli girdiler kullanmak zorundadırlar. Üretim faktörleri olarak isimlendirilen bu girdiler üç ana grupta toplanabilir:

- Emek
- Tabiat
- Sermaye

Bu çalışmada, emek faktörü dikkate alınmaktadır. Emek faktörü, üretimin en önemli beşeri bir faktörüdür. Emek faktörünün payı olmadan hiçbir üretim şekli hemen hemen düşünülemez. Bu nedenle, emek girdisi üretim maliyetleri için önemli bir yer tutmaktadır. Literatürde de bu maliyetler genellikle “iş gücü (workforce) maliyetleri” olarak ifade edilmektedir.

Mal ve hizmet üretiminde kullanılan işgücü girdisi önemli bir maliyet unsuru olduğundan, bu maliyetlerin minimizasyonu ile toplam üretim maliyetinde çok önemli indirimler yapılabilmekte ve böylece işletmelerin karı önemli ölçüde artırabilmektedir.

İşgücü maliyetlerinin minimizasyonu ile; belli miktardaki mal ve hizmetin üretilebilmesi için gerekli en az işgücü miktarı ve bu amaçla kullanılacak işçilerin optimum çalışma planı araştırılır. Bu konuda yapılan çalışmalar iki gruba toplanabilir.

1. İşgücü planlaması (manpower planning) ile ilgili çalışmalar.
2. Optimum vardiya planlaması (shift scheduling) ve tur planlaması (tour scheduling) ile ilgili çalışmalar.(Güngör,2005;1)

Her iki grupta yer alan çalışmalarda kullanılan çeşitli kantitatif yöntemler vardır bunlar; Markov modelleri, Doğrusal programlama modelleri, Dinamik programlama modelleri gibi. Çalışmada, işçilik maliyetini optimize etmek için dinamik programlama metodu kullanılıyor.

Dinamik programlama ilk olarak 1957 yılında Richard Bellman tarafından ortaya konulmuş bir en iyi sonucu bulmak için uygulanan bir karar verme yöntemidir. Bu yöntem, 1940'larda karar teorisinde kullanılmakta olan Ardışık ilişkiye dayanmaktadır. Günümüzde yöneylem araştırması problemleri çözümünde de dinamik programlama yöntemi kullanılmaktadır.

Dinamik programlama çok karmaşık problemlerde kullanılmaması tercih edilen bir karar verme yöntemidir. Bunun en büyük nedeni karmaşık problemlerin formüle edilmesinin zorluğundan kaynaklandığı söylenebilir. Diğer yöneylem araştırması yöntemlerine göre aşama süreçlerinin uzun olması ile birlikte adımlardaki işlem sayısının az olması bu karar verme yönteminin iyi bir planlama ile çözüme kolay ulaşılabilir. Dinamik programlamanın diğer tekniklere göre üstünlüğü, zaman faktörünü değerlendirmesi yanında; çok boyutlu problemlerin çözümünde etkin hesaplama kolaylığı sağlamasıdır. (Dogan,1985; sf: 122).

Diğer tekniklerden farklı olarak dinamik programlama yaklaşımında, karar problemine belli bir anda veya aşamada çözüm aranmaz. Bu yöntemle karşılaşılan problemi, önce alt bölümlere (aşamalara veya küçük problemlere) ayırarak her alt problem veya aşama için çözümü aranır. Daha sonra, her bir alt problemin optimum çözümüyle yetinilmeyip tüm problem için optimum çözüm bulmaya çalışır. (Doğan,1985; sf: 122 -123).

Dinamik programlama yaklaşımında amaç, ardışık ve birbirlerini etkileyen alt problemler için ayrı ayrı optimum kararlar bulup bulunan sonuçlar toplayarak problemi sonuçlandırmak değil, asıl problem için optimum sonucu verecek şekilde alt problemler için en iyi karar bulmaktır. Yani bazı durumlarda alt problemlerde en iyi olmayan bir karar gerçek problem için optimum sonucu sağlayan karar olabilmektedir.

Bu çalışmada dinamik programlama yöntemi inşaat işçilerinin sayısının optimize edilmesine yönelik olarak kullanılmıştır. İnşaat işçilerinin sayısı optimize edilerek inşaat şirketinin işçilik maliyetleri azaltılmaya çalışılmıştır.

1. İŞGÜCÜ MALİYETLERİNİN MİNİMİZASYONU

İşgücü maliyetlerinin minimizasyonu çalışması ile; işletmelerin sahip oldukları teknolojik sistem içinde, belli miktarda mal veya hizmeti, en az işgücü kullanarak üretebilmesi için gerekli planların nasıl yapılması gerektiği araştırılmaktadır.

İşgücü, işletmelerin mal ve hizmet üretimi için kullandıkları önemli girdilerden biri olması nedeniyle, toplam üretim maliyetleri içinde önemli bir yer tutmaktadır. Bu nedenle, işgücü maliyetlerinde yapılacak önemli miktardaki bir indirim, işletmenin karını da o ölçüde artırmış olacaktır.

İşgücü maliyetlerinin azalması amacıyla çok sayıda çalışma yapıldığı görülmektedir. Bu çalışmalar iki grupta toplanabilir:

1. İnsan gücü planlaması (manpower planning) ile ilgili çalışmalar.
2. Optimum vardiya planlaması (shift scheduling) ve tur planlaması (tour scheduling) ile ilgili çalışmalar. (Güngör,2005;5)

1.1.İnsan Gücü Planlaması

Planlama; herhangi bir şeyin olmadan önce belirlenmesidir. Planlama, gelecekteki ekonomik eylemler için temel veriler sağlayan karar verme sürecidir. Planlama yönetimi ise gerçekte düzenli bir düşünme planlaması işletme yönetiminin temel taslarından biri olmaktadır. Personel planlaması kaynaklarda işgücü planlaması, insan kaynakları planlaması, işgören planlaması ve beşeri kaynaklar planlaması olarak da geçmektedir. Personel planlaması, işletmenin bütünsel amacını gerçekleştirme doğrultusunda, gerekli sayıda, istenen yetenek ve beceri düzeyindeki personelin doğru zamanlarda ihtiyaç duyulan yerlere yerleştirilmesidir.

Personel planlama; İşletmenin genel amaçları çerçevesinde mevcut ve gelecekteki insan gücü gereksinimlerinin belirlendiği, insan gücünün sağlandığı, insan gücünün geliştirilmesi ve yetiştirilmesi ile ilgili planlamaların yapıldığı, insan gücünün çıkarılmasının planladığı evredir. Personel planlamasının öncelikleri verimlilik, işletme içinde personel etkinliğinin artırılması, is etüdü ve endüstri psikolojisi alanlarındaki çalışmaları kapsamıştır. Personel planlaması, insan kaynaklarının etkili kullanımını ve geliştirilmesini amaçlar. Sürekli olarak işgücü arz ve talebini analiz eder. Dengesizliklere neden olan faktörleri tespit ederek, bunların Bir başka kaynakta ise personel planlama su şekilde tanımlanmaktadır. (Küçüksille ve Güngör,2009;95)

Bir başka kaynakta ise personel planlama şu şekilde tanımlanmaktadır: “İşletmenin genel amaçları çerçevesinde mevcut ve gelecekteki insangücü gereksinimlerinin belirlendiği, insangücünün sağlandığı, insangücünün geliştirilmesi ve yetiştirilmesi ile ilgili planlamaların yapıldığı, insangücünün çıkarılmasının planladığı evredir”. (Tsinlly, 2003: 7).

İnsangücü planlamasını, makro ve mikro insangücü planlaması olarak ikiye ayırmak mümkündür. Makro açıdan insangücü planlaması, ülke ekonomisinin ihtiyaç duyduğu işgücünü nicelik ve nitelik açısından belirlemek ve belirlenen hedeflere ulaşmak için yapılmaktadır. Mikro açıdan insan gücü planlaması ise, kurum ve kuruluşun gayelerine ve görevlerine uygun olarak personel istihdamını gerçekleştirmek ve personelden diğer kaynaklara orantılı hatta onlardan daha fazla verim elde etmeyi amaçlamaktadır. (Canman, Öktem,1992;32)

İşletmelerde insangücü planlaması, işletmenin devamlılığını sağlamak ve daha verimli hale getirme amacına yöneliktir. İşletmelerde insangücü planlaması yapılırken, diğer kaynaklarında göz önünde tutulması gerekir. Bu konuda alınan kararlar, diğer konularda alınan kararlarla uyum içinde olmalıdır. İnsangücü maliyet sorunlarına ilişkin etkin bir çözüm, ancak iyi bir insangücü planlaması ile yapılabilir.(Güngör,2005;6)

1.2. Vardiya Planlaması

İşgücü sayısı ve niteliğine bağlı olarak düzenlenen vardiya planlaması işletme verimliliğini doğrudan etkileyen faktörlerdendir. Küresel

finans krizi sonucunda reel sektörde yaşanan talep daralması işletmelerin kısa çalışma süresi, üretime ara verme ve işçi çıkarma gibi farklı önlemler almalarına neden olmaktadır. Vardiyalı çalışma sistemi iş kanunlarından kaynaklanan birtakım kurallara ve sınırlamalara tabiidir. Bu kural ve sınırlamalar; haftalık ve günlük çalışma süreleri, dinlenme zamanları, fazla çalışma esasları, hafta tatili ve genel tatil vb. uygulamalarını düzenleyen genel nitelikte ama vardiya düzeninin organizasyonunda dikkate alınması gereken hukuksal düzenlemeleri kapsamaktadır. Bir mamul ya da hizmet üretim sisteminde işgücü azaltılması sistemin toplam personel maliyetini düşürebilir, buna karşın mamul veya hizmet kalitesinin ise yetersiz kalmasına neden olabilir. Tersi durumda ise işgücü fazlalığı ise personel maliyetlerini yükseltirken gereksiz kaynak kullanımına neden olmaktadır. Bu aşamada iyi bir işgücü planlaması ve iş çizelgesi ile işlemlerin zamanlaması ve sıralanması gibi dengeler kurulmalıdır. (Seçkiner,2005;161)

Vardiya planlama/çizelgeleme probleminin çözüm önerileri ile ilgili oldukça zengin bir literatür mevcuttur. Genellikle bir veya daha fazla vardiyalı çalışma düzenlerinin, çalışanların ve işletmenin performans göstergeleri açısından değerlendirilmesi ya da iş yükünün en küçüklenmesi gibi amaçlara yönelik gerçekleştirildiği görülmektedir. Basit vardiya planlama problemi ilk kez Dantzig tarafından 1954 yılında tanımlanmış ve araştırmada doğrusal bir programlama modeli önerilmiştir. Vardiya planlama problemi NP-Zor (deterministik polinom zamanlı çözümü olmayan) problem sınıfına girdiğinden konuya olan ilgi azalmamıştır. (Azmat,2004;148)

Optimum vardiya planlaması ve tur planlaması ile ilgili çalışmalar; insan gücü planlaması ile belirlenen “insangücü gereksinimlerini” yerine getirmek için, o iş yerinde çalıştırılması gereken en az işçi sayısının ve bu işçilerce bir dönme boyunca (genellikle bir hafta) uygulanacak çalışma çizelgesinin nasıl belirlenebileceği araştırmaktadır. (Güngör,2005;7)

Düzenli iş çizelgesinde haftalık toplam çalışma süresi günlere eşit olarak dağıtılır. Düzenli vardiya planı olarak genellikle haftada 5 günlük çalışma yapıldığı kabul edilir. Sıkıştırılmış veya hızlandırılmış iş çizelgesi haftalık çalışma saatinin daha az sayıda güne yayılmasından oluşur. Haftada 3 ya da 4 gün çalışma durumu bu şekilde düşünülebilir. Hiyerarşik çalışma düzeninde belirli bir işi belirli bir personelin dışında başka personelin yapabilmesi ile iş rotasyonu sağlanır. Kısmi zamanlı iş çizelgesinde haftalık çalışma saatleri aynı olmamasına rağmen yıllık çalışma saati toplamda değişmeyecek şekilde düzenlenir.(Korusu, 2010;471)

2. DİNAMİK PROGRAMLAMAMANIN TANIMI

Dinamik Programlama, 1920’den bu yana önemini arttırarak gelişmiştir. Başlangıçta yalnızca bir ekonomik sistemin zaman içindeki durumunun incelenmesinde kullanılmış olan yöntem, günümüzde zamanla ilgili olanların yanı sıra, daha değişik nitelikteki süreçlerin incelenmesinde de yaygınca kullanılmaktadır. (Sezer,2007,201)

Dinamik programlama yaklaşımında, ilk çözüme problemin son aşamasından başlanıp her defasında bir önceki aşamaya dönülerek “Geriye Doğru En İyileme” biçiminde bir çözüm yolu benimsenebileceği gibi, problemin ilk aşaması birinci aşama olarak ele alınıp, her defasında bir sonraki aşamaya geçilerek “İleriye Doğru En İyilime” biçiminde bir çözüm yolu da seçilebilir. (Doğan, 1995;94).

Dinamik programlama, n değişkenli bir problemin optimum çözümünü problemi n aşamaya ayrıştırarak ve her aşamada tek değişkenli bir alt problemi çözerek sonuca ulaşır. Bunu hesaplama avantajı olarak, n değişkenli alt problemler yerine tek değişkenli alt problemleri optimum hale getirilmesi olarak gösterilebilir. Dinamik programlamanın asıl katkısı, problemleri aşamalara ayrıştırmasının çerçevesini oluşturan uygun çözümlülük ilkesidir(Taha,2000;403).

Ardışık karar sorunlarını çözmede etkili bir yöntem olan dinamik programlama, doğrusal programlama ile karşılaştırılacak olursa her iki yöntemin de amacı kaynakların verimli dağılımıdır; ancak doğrusal programlama varsayımları modeli oldukça kısıtlar. Buna karşın dinamik programlamayla çok daha çeşitli sorunlar ve ilişkileri modellenebilir. Ancak dinamik programlama sorunun çözümünde genelde kabul edilmiş bir algoritmanın olmaması modelin çözümünde zorluklar ortaya koyabilir. Ele alınan her sorun için sorunun yapısına uygun eşitlikler geliştirtmelidir. Bu yüzden yöntem belirli düzeyde yaratıcılık ve görüş yeteneği gerektirmektedir.(Tütek ve Gümüšoğlu, 1994;334)

Her bir aşamada bulunan çözüm kendi başına problemin çözümünü oluşturmamakla birlikte optimal çözümün bir parçasının belirleyen bilgiyi içermektedir(Halaç,1995;157).

Dinamik programlamanın en fazla uygulandığı işletme sorunları aşağıdaki biçimde sıralanabilir(Tütek,Gümüšoğlu,1994;334):

- Sipariş kurallarının belirlenmesinde zaman ve nicelik değişkenlerinin saptanması,
- Değişen istem koşullarında üretim programlaması ve işgücü düzenlemesi,
- Pahalı araç ve gerecin etkin bir biçimde kullanılmasını sağlamak üzere yedek parça düzeyinin belirlenmesi,
- Yeni alanlara kaynak dağıtımını yapan sermaye bütçelemesi,
- Ürünleri halka geniş ölçüde tanıtılabilmek için reklam araçlarının seçimi,
- Değerli bir kaynağın bulunmasında sistematik aramanın yapılması,
- Karmaşık makinelerin bakım onarımının programlanması,
- Eskiye donanım ve makinelerin yenilenmesi için uzun dönem stratejilerinin Saptanması,
- Çeşitli malzemeler için kesme kalıplarının belirlenmesi.

2.1 Dinamik Programlama Türleri

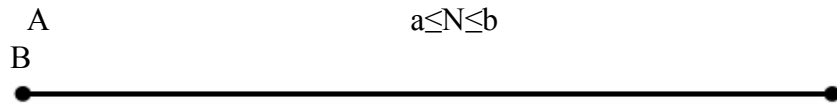
Dinamik programlama problemi, rassallığın bulunduğu durum ve bulunmadığı durum olmak üzere iki şekilde sınıflandırılır. Bu sürecin özelliğini belirleyen şey dönüşüm fonksiyonudur. Süreçle ilgili geçiş fonksiyonunda bir rassallık varsa, stokastik yapı bir problem değilse, deterministik yapı bir problemden söz edilir. Dinamik yapı 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 edinmiyor ya da çok az bilgi edinilebiliyor ise Dinamik Programlama problemi ifade etmek imkânsızdır. Sonlu süreçlerin karar değerleri önceden biliniyorsa, bu tür süreçler sonlu bilinen, değilse sonsuz bilinmeyen olarak adlandırılır. (Sezen, age,208)

- **Deterministik (Belirlenim) Dinamik Programlama;** Deterministik DP'da gelecek aşamadaki durumun ne olacağı şu anki durum ve politika tarafından tamamen belirlenir. Diğer bir ifadeyle, çok aşamalı bir karar sürecinde, karara etki eden tüm dışsal etmenler tam olarak bilinirse süreç deterministiktir. (Hillier,1995;433)

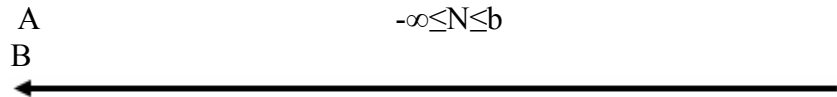
- **Stokastik (Olasılıklı) Dinamik Programlama;** Olasılıklı DP deterministik DP'dan farklıdır. Burada gelecek aşamadaki durum, şu anki durum ve karar tarafından tamamen belirlenemez. Daha doğrusu gelecek durumun ne olacağı ile ilgili olasılık dağılımı bellidir. Bununla birlikte bu olasılık dağılımı da şu anki durum ve politik karar tarafından tamamen belirlenir. Çok aşamalı bir karar probleminde iki farklı şekilde rassal değişken söz konusu olabilir. Bunlardan ilki, bir aşamanın içinde bulunulan duruma göre verilen kararlar, izleyen aşamanın gelinecek durumunun belirli, ancak içinde bulunulan aşamada elde edilecek katkının olasılıklı olmasıdır. (Hillier,age,453)

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;

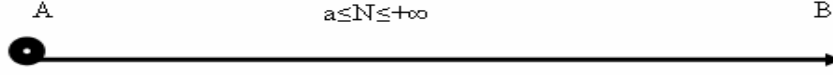
1) Her iki tarafı kapalı durum;



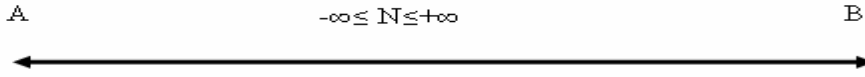
2. Sol taraf açık, sağ tarafı kapalı durum



3. Sağ tarafı açık, sol tarafı kapalı durum.



4. Her iki tarafı açık durum



2.2. Dinamik Programlama Problemlerinin Özellikleri

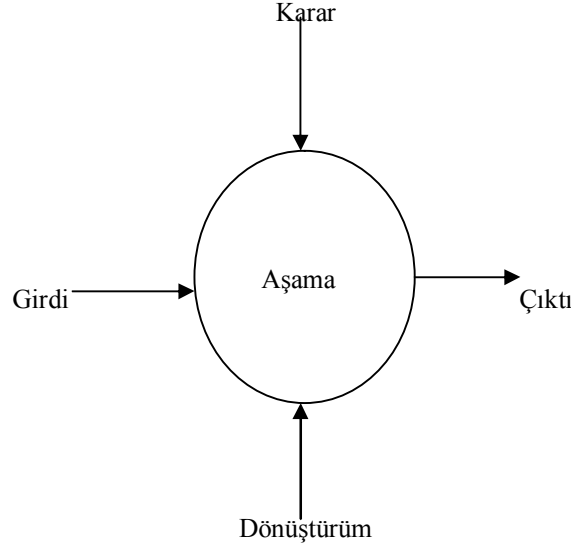
Bütün dinamik programlama problemlerini içeren ve bunların tek bir metotla çözümünü belirleyen bir yol yoktur. Her bir problem teker teker ele alınır ve bunların fonksiyonel denklemi kurulur. Bu denklemin çözümünü aramak amacı bütün problemlerde aynıdır. Sadece tanım ve uygulanan çözüm yöntemleri farklılıklar gösterir. (Karayalçın, 1993; 244)

Temelde, dinamik programlama, problem çözümüne, problemin veya problemin bir kısmının parçalara bölünmesi ve bu parçaların çözülerek, bu çözümlerin depolanması şeklinde bir problem çözüm yaklaşımı sunmaktadır. Dinamik programlama, sistem analizi alanında yaygın olarak kullanılan bir yöntemdir ve çok aşamalı karar verme problemlerinde optimal bir silsileye karar vermede kullanılabilir. Dinamik programlama, özellikle karar aşamasının zaman periyodunda silsile halinde olan problemlere çok uygundur. Periyotlar birbirine öyle bir bağla bağlıdır ki bir zaman döneminde alınan kararlar sonraki karar verme aşamalarını etkilemektedir.

Problem, alt problemlere bölünür ve her bir alt problem için optimal bir çözüm bulunur, n sayıda karar verme aşamalarına sahip bir problem, n sayıda ve her biri tek bir karar değişkenine sahip, problemlere bölünür. Hesaplama süresi, bir problem içindeki değişkenler sayısınca eksponensiyel olarak büyürken, alt problemler sayısınca doğrusal olarak büyür. Bir problemin tümü sistem ve alt problemler de basamak olarak düşünülebilir. Dinamik programlamada basamaklar, genellikle, bir zaman aralığını temsil eder. Bir sistemin her bir basamağında, problemin çözüm şanlarına karşılık gelen birden fazla durum vardır. Durumlar, tamamlanmamış çözümleri karakterize eder. Karar verici, her bir basamakta, o basamak için en iyi sonucu veren karar vermelidir. Bir karar, sistemi bir durumdan diğerine taşır. Bir sistemi bir durumdan diğerine taşıyan her bir aşamaya basamak denir. Dinamik programlama genellikle geriye doğru, yani son durumdan ilk duruma doğru, bir işlemler silsilesi şeklinde uygulanır. (www.maden.org.tr)

Bir dinamik programlama problemini, doğrusal programlamada olduğu gibi, problemin girdisi olan parametre değerlerinin bilindiği ve en iyileme sorunun bu değerlerin kullanımıyla çözülmek istendiğini düşünelim. Bir başka değişle, sistemin rastgele değişken içermediğini yani, deterministlik olduğunu varsayalım. Dinamik programlamada karar verme

durumunun ortaya çıktığı her bir noktaya “aşama”, sistemin durumunu yansıtan girdi parametrelerine “durum” denir. Dinamik programlamada kararı etkileyen bir çeşit kural veya bir çeşit eşitlik tanımlaması zorunludur. Tanımlanan bu kural veya eşitliğe “dönüştürüm” denir. Dinamik programlamanın bu önemli bileşenleri ve bunlar arasındaki ilişkiler aşağıda şekilde gösterilmiştir.(Cinemre,1997;390)



Şekil-1: Dinamik Programlama Bileşenleri

Dinamik programlama uygulamasının en önemli noktası aşmalarının belirlenmesidir. Her bir problemin kendine has bir geçiş noktası olduğundan ve durumun belirlenmesi amacıyla yapılan seçim problemlerinden probleme değiştiğinden, dinamik programlamanın bilimden çok bir sanat olduğu söylenebilir. Bu konuda ustalık kazanabilmenin tek yolu, çok sayıda dinamik programlama problemi çözmektir. Dinamik programlama çok aşamalı süreçlerin en iyilenmesiyle ilgili matematiksel bir tekniktir. Dinamik programlamanın ana fikri en iyileme kuralında saklıdır. Bu kural şu şekilde açıklanabilir. Çok aşamalı bir karar süreci probleminde kararların en iyi kümesi başlangıç aşaması, durumu ve kararlar ne olursa olsun birinci aşamanın kararlarından sonra ortaya çıkan aşama ve durum için geriye kalan kararlar kalan problem için en iyi kararlar dizisini oluşturur. (Cinemre,1997;394)

Dinamik programlama problemlerinin özellikleri aşağıda belirtilmiştir(İnce,2006;8):

- Problemler, karar vericinin uygun gördüğü aşama sayısına ve kısıtlara ayrılabilir. (Hiller, Lieberman,1996;336).
- Her aşama kendinden önce ve sonra gelen aşama ile bütünlük oluşturmayabilir. Her aşama kendine göre bir bağımsız çözüme sahip olmalıdır.

- Dinamik programlama ile şebeke, akış problemleri çözülebilir.
- Problemlerin çözüm süreci bütün aşamaları optimal edecek şekilde belirlenir.
- Aşama n+1 için optimal politika belli ise aşama n için optimal politikayı belirten tekrarlı yapıya ulaşılabilmektedir. Aşama n'de ve s durumunda başlanması halinde optimal politika kararının verilmesi X_n 'in değerinin bulunmasını gerektirmektedir. Bu model şu şekilde olabilir(Hiller,Lieberman,1996;336).

$$f_n(S_n) = \text{enb/enk}_{x_n}[f_n(S_n, X_n)]$$

S_n = Girdi durumu

X_n = Karar Değişkeni

$f_n(S_n, X_n)$ = Belirli bir durum (S_n) için, n aşama boyunca hesaplanan birikmiş toplam sonuç

$f_n(S_n)$ = Belirli bir durum (S_n) için, birikmiş toplam en iyi sonuç

2.3. Dinamik Programlamanın Temel Kavramları

Dinamik programlamada kullanılan temel kavramlar 6 başlık altında toplanabilir. Şimdi bunları kısaca açıklamaya çalışacak olursak;(Sezen,1998,17):

- **Aşama:** Aşama büyük bir problem çözümlenmek üzere küçük alt problemlere ayrıştırıldığında, her alt problem çözümlenmesi gereken bir karar problemi olarak düşünülürse, karar verilmesi gereken her nokta safha olarak tanımlanır. Bundan dolayı her alt problem bir safhaya karşılık gelmektedir. Safhaların sayısı problemde yer alan düğüm sayısına ya da alt probleme göre değişebilir. (Cook,Russell,1981;294).

Aşama sayısı sürecin uzunluğuna bağlıdır. Safha yapısını belirleyen önemli bir özellik, sürecin sürekli ya da kesikli olmasıdır. Tamsayılarla ifade edilen kesikli bir süreç söz konusu olduğunda safha değişkeni de kesiklidir. Sürekli bir süreçte safha tanımlaması yapmak, uygulama açısından bazı sorunlar ortaya çıkarmaktadır.(Sezen,2007;204).Safhalar birbirleriyle bağlantılı bir sürecin bölmeleri olup, safhalardan birinin yanlış çözüme ulaştırılması bütün problemi yanlış çözüme iter(Top,1994;207).

- **Durumun İncelenmesi:** İncelenen problemin bir sonraki geleceği durum doğrudan şu andaki durumuna bağlıdır ve verilen problemin durumuna bağlı olarak ilgili safhadaki faaliyet seçenekleri tespit edildiğinden, bu seçeneklere durum değişkenlikleri denir. Yani, her safhanın başında incelenen problem bir durumdan diğer duruma değişir yeni durum sisteminin bir önceki devredeki durumunun fonksiyonu şeklindedir. Özet olarak, bir safhadaki durum, daha önceki safhada verilen kararların bir

sonucudur (Sezen,1998;8). Durum değeri, sistemin mevcut durumundan başlayarak özel bir yol izlediğinde ortaya çıkan getirilerin toplam bir fonksiyonudur(Winston,1994;1010).

• **Dönüşüm Fonksiyonları(Dönüşüm Denklemleri):**Her safhanın bulunabilecek durumlarında verilebilecek karara göre bu safhayı izleyen veya daha önceki safhanın hangi durumuna gelineceğini belirleyen ilişkilere dönüşüm fonksiyonları denilmektedir. Dönüşüm fonksiyonları çözüm yolunun ileriye veya geriye doğru en iyilime olmasına göre farklılık gösterir(Sevinç,2008;102).

Bu denklemler aşamalar arasında bağlantıyı sağlayan matematik ifadelerdir. Aslında Dinamik Programlama yaklaşımının kullanımını gerektiren ilişkiler de bunlardır. Dinamik programlamada hesaplamalar yenilenecek yapılır, bu bakımdan bir alt problemin uygun değer çözümü bir sonraki alt problemin girdisidir. Son alt problem çözüldüğünde, probleminin tamamının optimum çözümüne ulaşılmış olunur. Yenilenecek hesaplamaların uygulanma biçimi orijinal problemin nasıl ayrıştırıldığına bağlıdır. Özellikle alt problemler bazı ortak kısıtlarla normalde birbirleriyle ilişkilendirilmişlerdir. Bir alt problemden bir sonrakine ilerledikçe bu kısıtların uygunluğuna dikkat etmek zorunludur.(Sezen,2007;206)

• **Ardışık En İyileme:** Dinamik programlama yaklaşımı için gerekli olan, her safha bir karar probleminin çözümüyle biri diğerini izleyen kararlarda uygulanmakta olup, tüm safhalarla birlikte, başlangıç karar probleminin en uygun çözümünü veren fonksiyonel bağlantılar ile bir en iyileme tekniği bulmaktadır(Çiçek,2000;41).

• **Karar:** Dinamik programlama herhangi bir problem için mümkün olan koşullar içinde ne yapılacağına karar vermeyi sağlar. Karar, problemdeki bir safhayı tamamlamak için ilgili alternatifler arasından bir seçim yapılması işlemidir. Belirli bir durum ve safhada verilen bir karar, sürecin hem durumunu hem de safhasını değiştirir. Dolayısıyla her karar, geçerli bir durumdan bir sonraki safhaya bağlı olan duruma geçişi etkilemektedir. Çok safhalı bir karar süreci safhalara ayrıldıktan sonra her safha için bir dönüşüm fonksiyonu oluşturulur. Her safhada karar verme süreci, o safhanın seçeneklerinden birinin seçilmesiyle sonuçlanır. Buna safha kararı denir. Ayrıca her hangi bir safhadaki karar uygun çözüm ise bu karar, problemin en uygun çözümünün bir parçası olarak kullanılabilir(Sevinç,2008;101).

• **Optimal Politika:** Çok safhalı bir karar sürecinin her karara bağlı, maliyet veya kar cinsinden bir getirisi olacaktır. Bu getiri, sürecin safha ve durumu ile birlikte değişir. Optimal politika, sürecin her bir safhası için, verilen kararların bir sırasıdır. Çözüm bir aşamadan diğerine ulaştıktan sonra her parametre için değerler belirlenir ve işlem tamamlanır(Sezen,1998;9).

2.4 Dinamik Programlamanın Üstünlükleri

Dinamik programlama problemlerin çözülmesinde bazı üstünlükler sağlamaktadır. Bu üstünlükler aşağıda sıralanmıştır

- Karmaşık problemler, dinamik programlama yaklaşımı ile birbirleriyle ilişkili alt problemlere ayrılarak daha kolay bir şekilde çözülebilmektedirler. Dinamik programlamanın bu avantajı özellikle üretim düzeltme ve planlama gibi ardışık kararlar içeren problemler için önemlidir.
- Dinamik programlama diğer matematiksel programlama problemlerine uygulanabilen esnek yapısıyla, bir teknikten ziyade bir optimizasyon yaklaşımıdır. Tamsayı programlama problemleri ve doğrusal olmayan tamsayı programlama problemleri dinamik programlama yaklaşımıyla çözülebilmektedir.
- Dinamik programlama deterministik ve stokastik süreçlere uygulanabilmektedir.
- Dinamik programlamanın hesaplama yöntemi, duyarlılık analizi yapmaya elverişli olabilmektedir.
- Dinamik programlama problemlerin çözümünde olası tüm seçeneklerin göz önüne alınması yerine, çok daha az hesaplama yapma kolaylığı sağlamaktadır. (Sevinç,2008;104)
- Bazı problemler yalnızca Dinamik programlama ile çözülebilmektedir. Örneğin kimya prosesleri gibi ardışık tipteki problemler.(Halaç,1983;177)

2.5 Dinamik Programlamanın Yineleme İle Çözüm Yöntemleri

Dinamik programlamada hesaplamalar yinelenerek yapılır, bundan dolayı bir alt problemin optimum çözümü bir sonraki alt problemin girdisidir. Son alt problemi çözüldüğünde, problemin bütünü optimum çözüme ulaştırılmış olunur. Yinelenen hesaplamaların uygulanma biçimi orijinal problemin nasıl ayrıştırıldığına bağlıdır. Özellikle, alt problemler bazı ortak kısıtlarla normalde birbirleriyle ilişkilendirilmişlerdir. Bir alt problemden bir sonraki alt probleme ilerledikçe bu kısıtların uygunluğu göz önünde bulundurulmalıdır(Taha,2000;403).

Dinamik programlama yineleme ile çözüm yöntemleri ikiye ayrılır(Taha,2000;407):

- Geriye Doğru Yineleme İle Çözüm Yöntemi
- İleriye Doğru Yineleme İle Çözüm Yöntemi

Aslında her iki yöntemde aynı sonucu vermektedir. İleriye doğru yineleme ile çözüm yöntemi daha mantıklıdır bununla birlikte, dinamik programlama literatürü değişmez bir biçimde geriye doğru yinelemeyi kullanmaktadır. Bu tercihin nedeni, geriye doğru yinelemenin hesaplama bakımından daha etkili görülmesidir (Taha,2000;407).

2.5.1 Geriye Doğru Yineleme İle Çözüm Yöntemi

Bu çözüm yönteminde, çözüme 1 inci safhadan başlanılarak 2, 3,...,n-1, n ve N inci safha için işlemler gerçekleştirilir, ancak işlem sırası akışın aksi yöndedir. Yani çözüme sondan başlanır. Örneğin 6 safhalı bir problem 6 ncı safhasından başlanarak çözülmeye çalışılır(Sevinç,2008,109).

Geriye doğru yineleme ile çözüm yönteminin formülasyonu aşağıdaki gibidir (Taha,2000;407):

$$f_i(x_i) = \min_{(x_i, x_{i+1})} \{d(x_i, x_{i+1}) + f_{i+1}(x_{i+1})\}, i=1,2,3 \quad (2)$$

Genelde problem yapılarına göre değişen bu çözüm yönteminin genel formülasyonu aşağıdaki gibidir (Sevinç,2008;110):

Çözümüne başlanacak ilk safha için;

$$f_1(s_1) = \min (\max) [r_1 (s_1, x_1)] \quad (3)$$

İkinci safha için;

$$f_2(s_2) = \min (\max) [r_2 (s_2, x_2)] + f_1(s_1) \quad (4)$$

N inci safha için;

$$f_N(s_N) = \min (\max) [r_N (s_N, x_N)] + f_{N-1} (s_{N-1}) \quad (5)$$

Verilen ifadelerde;(Halaç,1994;162)

i =Aşama numarası

X_i = i .aşamada karar değişkeni

S_i = i .aşama sonunda sistemin durumu

r_i = i .aşamada belirli bir durum-karar kombinasyonuna bağlı olarak ortaya çıkan getiri veya sonuç

f_i = i .aşamada kriterin değeri

2.5.2 İleriye Doğru Yineleme İle Çözüm Yöntemi

Bu çözüm yönteminde $n-1$ inci safha ile ilgili bilgiler, n inci safhanın karar girdilerini oluşturacaklardır. Bundan dolayı çözüme 1 inci safhadan başlanılarak 2, 3, $n-1$, n ve N inci safhaya doğru gidileceğinden dönüşüm fonksiyonları da buna uygun olarak formüle edilir. Örneğin 5 safhalı bir problem çözümü yapılırken, çözüme 1 inci safhadan başlanılır. İleriye doğru çözümlenme yapılarak, problem çözülmeye çalışılır(Turban, Meredith,1998;528).

Bu çözüm yönteminin genel formülü aşağıda verilmiştir(Sevinç,2008;109):

Birinci safha için;

$$f_1(s_1) = \min (\max) [r_1 (s_1, x_1)] \quad (6)$$

İkinci safha için;

$$f_2(s_2) = \min (\max) [r_2 (s_2, x_2)] + f_1(s_1) \quad (7)$$

N inci safha için;

$$f_N(s_N) = \min (\max) [r_N (s_N, x_N)] + f_{N-1} (s_{N-1}) \quad (8)$$

3. PROBLEMİN TANIMLANMASI VE UYGULAMAYA DÖNÜK ÇÖZÜM ÖNERİSİ

Malatya’da faaliyet gösteren bir inşaat şirketi 5 bina yapım projesi almış ve projelerde kapsamında çalıştıracağı işçi sayısını hesaplamıştır. İnşaat şirketi proje kapsamında çalıştıracağı işçi sayısını ve işçilik maliyetlerini minimum değerde tutmak istemektedir. Bu çalışma kapsamında inşaat firmasına dinamik programlama yöntemi kullanılarak işçilik maliyetlerini azaltmaya yönelik alternatif bir öneri getirilmiştir.

3.1 Problemin Tanımlanması

İşçilik maliyetlerini azaltmaya yönelik oluşturulan denklem aşağıdaki gibidir.

$$K_j = \min((M_1(x_j - b_j) + M_2 + M_3(x_j - x_{j-1}) + K_{j+1}(x_j)) \quad (9)$$

M_1 = Fazla işçi çalıştırma maliyeti; İnşaat firması tarafından 400 TL olarak hesaplanmıştır.

M_2 = Sabit işçi maliyeti; bir işçi alınsa bile oluşacak maliyettir.

Bu maliyet firma tarafından 600 TL olarak hesaplanmıştır.

M_3 = Bir işçiyi işe alma maliyeti: İnşaat firması tarafından 776 TL olarak hesaplanmıştır.

x_j =j döneminde çalışan işçi sayısı.

b_j =j dönemindeki minimum işçi sayısı.

K_j =j dönemindeki işçilik maliyeti.

Tablo 1 Ayları ve minimum aylık işçi sayılarını göstermektedir.

Tablo 1:Aylar ve Aylık İşçi Sayıları

Aylar	Aylık İşçi Sayısı
Haziran	116
Temmuz	120
Ağustos	110
Eylül	115
Ekim	100
Kasım	102

$b_1=116, b_2=120, b_3=110, b_4=115, b_5=100, b_6=102;$

$M_1(x_j - b_j) = 400 * (x_j - b_j)$

$M_2 = 600$

$M_3(x_j - x_{j-1}) = 776 * (x_j - x_{j-1})$

3.2 Problemin Çözümü

Problemin çözümü 6 safhadan oluşmaktadır. Safha tablolarında x_j değerleri, x_{j-1} değerleri ve K_j değerleri yer alacaktır. Safhalar sondan başa doğru olacaktır. Dolayısı ile projenin bittiği Kasım ayından başlanacaktır.

- **Altıncı Safha:** Altıncı safhadaki ay olan Kasım ayındaki işçi sayısı, x_6 değeri ile gösterilecektir. Ekim ayını x_5 değeri gösterecektir. x_6 değeri bu ayın minimum değeri olan 102 değerini almıştır. x_5 değerinin minimum değeri 100 olduğu için, 102'ye ulaşması için; 100, 101 ve 102 olabilir. Tablo 2'de bu değerler ve maliyetler verilmiştir. Maliyetler şöyle hesaplanmıştır:

$$K_6(x_5) = 400 \cdot (102(x_6) - 102(b_6)) + (600 \text{ veya } 0) + 776 \cdot (102(x_6) - x_5) \quad (10)$$

600 değeri sabit işçi maliyetidir. İşçi alınmırsa çözüme katılır, işçi alınmazsa çözüme katılmaz. Bundan dolayı hesaplamada 600 veya 0 diye belirtilmiştir. x_6^* değeri optimum 6. dönem işçi sayısını belirtmektedir.

Tablo 2: Altıncı Safha Tablosu

x_5	x_6	$K_6(x_5)$	x_6^*
100	2152	2152	102
101	1376	1376	102
102	0	0	102

- **Beşinci Safha:** Beşinci safhadaki ay olan ekim ayındaki işçi sayısı, x_5 değeri ile gösterilecektir. Eylül ayını x_4 değeri gösterecektir. x_5 değeri, altıncı safha tablosunda aldığı değerleri beşinci safha tablosunda koruyacaktır. x_4 burada minimum değerini alacaktır. Bu değer 115 tir. Tablo 3'te bu değerler ve maliyetler verilmiştir. Maliyetlerin hesaplanması şu şekildedir:

$$K_5(x_4) = 400 \cdot (x_5 - 100(b_5)) + (600 \text{ veya } 0) + 776 \cdot (x_5 - 115(x_4)) + K_6(x_5) \quad (11)$$

$x_5 - 115$ değeri negatif çıktığı için hesaplamalarda sıfır değerini alınmıştır. $K_6(x_5)$ değeri altıncı safha tablosunda yer alan değerlerdir. Safhalara sondan başa doğru geldiği için 6. dönem maliyeti 5. döneme aktarılmıştır. x_5^* değeri tabloda beşinci dönem için en düşük maliyete sahip işçi sayısı göstermekte olup, $K_5(x_4)$ değeri ise, beşinci dönemdeki en düşük işçi maliyetini göstermektedir.

Tablo 3: Beşinci Safha Tablosu

x_4	$x_5=100$	$x_5=101$	$x_5=102$	$K_5(x_4)$	x_5^*
115	2152	1776	800	800	102

- **Dördüncü Safha:** Dördüncü safhadaki ay olan eylül ayındaki işçi sayısı, x_4 değeri ile gösterilecektir. Ağustos ayını x_3 değeri gösterecektir. x_4 değeri, beşinci safha tablosunda aldığı değeri dördüncü safha tablosunda

koruyacaktır. x_3 değeri minimum değeri olan 110 dan x_4 değerine kadar olan bütün değerleri alacaktır. Tablo 4'te bu değerler ve maliyetler verilmiştir. Maliyetlerin hesaplanması şu şekildedir:

$$K_4(x_3) = 400*(115(x_4)-115(b_4))+(600 \text{ veya } 0)+776*(115(x_4)-x_3)+K_5(x_4) \quad (12)$$

$K_5(x_4)$ değeri beşinci safha tablosunda yer alan değerdir. Safhalara sondan başa doğru geldiği için 5. dönem maliyeti 4. döneme aktarılmıştır. x_4^* değeri tabloda dördüncü dönem için en düşük maliyete sahip işçi sayısı göstermekte olup, $K_4(x_3)$ değeri ise, dördüncü dönemdeki en düşük işçi maliyetini göstermektedir.

Tablo 4: Dördüncü Safha Tablosu

x_3	x_4	$K_4(x_3)$	x_4^*
110	5280	5280	115
111	4504	4504	115
112	3728	3728	115
113	2952	2952	115
114	2176	2176	115
115	800	800	115

• **Üçüncü Safha:** Üçüncü safhadaki ay olan ağustos ayındaki işçi sayısı, x_3 değeri ile gösterilecektir. Temmuz ayını x_2 değeri gösterecektir. x_3 değeri, dördüncü safha tablosunda aldığı değerleri üçüncü safha tablosunda koruyacaktır. x_2 değeri minimum değeri olan 120 değerini alacaktır. Tablo 5'te bu değerler ve maliyetler verilmiştir. Maliyetlerin hesaplanması şu şekildedir:

$$K_3(x_2) = 400*(x_3-110(b_3))+(600 \text{ veya } 0)+776*(x_3-120(x_2))+K_4(x_3) \quad (13)$$

x_3^* değeri tabloda üçüncü dönem için en düşük maliyete sahip işçi sayısı göstermekte olup, $K_3(x_2)$ değeri ise, üçüncü dönemdeki en düşük işçi maliyetini göstermektedir.

Tablo 5: Üçüncü Safha Tablosu

x_2	$x_3=110$	$x_3=111$	$x_3=112$	$x_3=113$	$x_3=114$	$x_3=115$	$K_3(x_2)$	x_3^*
120	5280	4904	4528	4152	3776	2800	2800	115

• **İkinci Safha:** İkinci safhadaki ay olan temmuz ayındaki işçi sayısı, x_2 değeri ile gösterilecektir. Haziran ayını x_1 değeri gösterecektir. x_2 değeri, minimum değeri olan 120 değerini alacaktır. x_1 değeri minimum değeri olan 116 dan 120 değerine kadar değerler alacaktır. Tablo 6'da bu değerler ve maliyetler verilmiştir. Maliyetlerin hesaplanması şu şekildedir:

$$K_2(x_1) = 400*(120(x_2)-120(b_2))+(600 \text{ veya } 0)+776*(120(x_2)-x_1)+K_3(x_2) \quad (14)$$

Tablo 6:İkinci Safha Tablosu

x_1	$x_2=120$	$K_2(x_1)$	x_2^*
116	6504	6504	120
117	5728	5728	120
118	4952	4952	120
119	4176	4176	120
120	2800	2800	120

• **Birinci Safha:** Birinci safhadaki ay olan haziran ayındaki işçi sayısı, x_1 değeri ile gösterilecektir. x_0 değeri projeye başlamadan işçi alımını göstermektedir. x_1 değeri, ikinci safha tablosunda aldığı değerleri koruyacaktır. x_0 değeri projenin başlangıcında işçi olmadığı var sayılıp 0 olarak belirlenmiştir. Tablo 7’de bu değerler ve maliyetler verilmiştir. Maliyetlerin hesaplanması şu şekildedir:

$$K_1(x_0) = 400*(x_1 - 116(b_1)) + (600 \text{ veya } 0) + 776*(x_1 - 0(x_0)) + K_2(x_1) \quad (15)$$

Tablo 7: Birinci Safha Tablosu

x_0	$x_1=116$	$x_1=117$	$x_1=118$	$x_1=119$	$x_1=120$	$K_1(x_0)$	x_1^*
0	97120	97520	97920	98320	98120	97120	116

4.SONUÇ VE ÖNERİLER

Yapılan çalışmanın sonucunda işyerine alınması gereken işçi sayısı ve maliyetleriyle ilgili şu sonuçlar elde edilmiştir. İnşaat şirketi Haziran ayında 116 işçi ile inşaat projesine başlamıştır. Temmuz ayında şirket 4 işçiye daha ihtiyaç duymuş ve 4 işçi daha işe alarak işçi sayısını 120 ye çıkarmıştır. Ağustos ayının başında şirketin toplam işçi sayısı 120 olmuş, dinamik programlama ile bu işçi sayısının şirket için fazla olduğu tespit edilmiş ve şirketten 5 işçiyi işten çıkartması önerilmiştir.. Şirket bu öneriye uymuş ve 5 işçiyi işten çıkarmıştır. Eylül ayına gelindiğinde, şirketin işçi sayısı 115’tir. Dinamik programlama ile bu işçi sayısının optimum olduğu tespit edilmiştir. Ekim ayına girildiğinde şirketin optimum miktardan fazla işçi çalıştırdığı dinamik programlama çözümü ile anlaşılmıştır. Bunun üzerine şirkete, şirketin optimum miktarda işçi çalıştırması için 13 işçiyi işten çıkarması önerilmiştir. Böylece ekim ayındaki toplam işçi sayısı 115’ten 102’ye inmiştir. Kasım ayında şirket ekim ayındaki işçi sayısını koruyarak inşaat projesini sonlandırmıştır. Tablo 8’de kararlar yani şirkete verilen öneriler ve sonuçlar özetlenmiştir.

Tablo 8: Sonuçlar ve Kararlar Tablosu

Aylar	Minimum İşçi Sayısı	Hesaplanan İşçi Sayısı	Karar
Haziran	116	116	116 işçi işe al
Temmuz	120	120	4 işçi işe al
Ağustos	110	115	5 işçi işten çıkar
Eylül	115	115	Değişiklik yok
Ekim	100	102	13 işçi işten çıkar
Kasım	102	102	Değişiklik yok

Tabloyu maliyet yönünden incelenirse, şirket proje başlamadan önce hesapladığı işçi sayısı ile çalışmış olsa idi, Ağustos ayında 110 işçi ile çalışsaydı 5280 lira işçilik maliyetine katlanacaktı ama şirket 115 işçi çalıştırdığı için sadece 2800 lira maliyete katlanmıştır. Aslında şirket kendi hesapladığı işçi sayısından daha fazla işçi çalıştırmış olmasına rağmen daha az işçilik maliyetine katlanmıştır. Bunun sebebi Eylül ayı için hesaplanmış işçi sayısıdır. Ağustos ayında şirket 110 işçi ile çalışmış olsaydı Eylül ayına gelindiğinde kendi hesapladığı minimum işçi sayısına göre 5 işçi daha almak zorunda kalacak ve bu da yeni işçi işe alma maliyetini ortaya çıkaracaktı. Fakat şirket 115 işçi sayısını Ağustos ayında koruduğu için yeni işçi işe alma maliyetinden kurtulmuştur. Böylece şirket 2480 (5280-2800) lira fazla işçilik maliyeti ödemekten kurtulmuştur. Aynı şekilde şirket Ekim ayında 100 işçi ile çalışmış olsa idi, 2152 lira işçilik maliyetine katlanmış olacaktı. Fakat şirket 102 işçi ile çalışmış olup sadece 1352 lira işçilik maliyetine katlanmıştır. Bunun sebebi Kasım ayında hesaplanmış olan işçi sayısının 102 olmasıdır. Eğer şirket 100 işçi ile çalışmış olsa idi Kasım ayı için 2 işçi daha alması gerekecekti böylece yeni işçi işe alma maliyeti ile karşılaşacaktı. Fakat şirket 100 işçi yerine 102 işçi ile çalışmış böylece 800 (2152-1352) lira daha az maliyete katlanmıştır.

Sonuç olarak çalışmada; teorik anlamda bir Dinamik Programlama modeli bir inşaat firmasında çalışan işçilerin işçilik maliyetlerinin optimize edilmesi için uygulanmıştır. Çalışmada bulunan firmadaki işçilik maliyetleri minimize edilerek, firmanın yüksek maliyetlere maruz kalması önlenmiştir. Ayrıca işçilik maliyetinin azaltılmasına yönelik yapılan bu dinamik programlama yönteminin daha büyük firmalarda uygulaması yapıldığı takdirde yüksek oranlarda işçilik maliyetlerinden tasarruf edileceği de göz önünde bulundurulmalıdır.

KAYNAKÇA

1. AZMAT, C. S., WIDMER, M., “A Case Study of Single Shift Planning and Scheduling Under Annualized Hours: A Simple Three-Step Approach”, **European Journal of Operational Research**, 153, 148–175, 2004.

2. CANMAN Doğan, ÖKTEM M. Kemal, **Kamu Yönetiminde İnsan Kaynağının Geliştirilmesinde İnsan Gücü Planlaması**, ATO C.25 S.4 Aralık-1992 s.32
3. COOK, T. M. and RUSSELL, R. A., **Introduction To Management Science**, 2. Baskı, Pearson Prentice Hall, New Jersey,1981
4. ÇİÇEK, D.S, “**Otel İşletmelerinde Dinamik Programlamaya Dayalı Oda Fiyatının Belirlenmesi**”, Anadolu Üniversitesi Sosyal Bilimler Enstitüsü Yayınlanmış Yüksek Lisans Tezi, Eskişehir,2000.
5. CİNEMRE, N, **Yöneylem Araştırması**, 1. Baskı, Beta Basım Yayın, İstanbul,1997.
6. DOĞAN, Muammer; **İşletmelerde Karar Verme Teknikleri**; Dokuz Eylül Üniversitesi BF Yayını; Bilgehan Basımevi; 1985 İzmir.
7. DOĞAN, İbrahim, **Yöneylem Araştırması Teknikleri ve İşletme Uygulamaları**, İstanbul: Bilim Teknik Yayınevi, 1995
8. GÜNGÖR, İbrahim, **İşgücü Maliyetlerinin Minimizasyonu VARDİYA PLANLAMASI (Modeller, Algoritma ve Uygulamalar)**, 1. Baskı, Asıl Yayın Dağıtım Ltd.Şti,2005.
9. HALAÇ, O, **Kantitatif Karar Verme Yöntemleri (Yöneylem Araştırması)**, 4. Baskı, Alfa Basım Yayın Dağıtım, İstanbul,1995.
10. HİLLER, F. S. and LİEBERMAN J.G, **Introduction To Operations Research**, McGraw-Hill, USA,1996.
11. İNCE, Ö, “**Dynamic Programming Applications In Investment Analysis**”, İstanbul Teknik Üniversitesi, Yayınlanmış Yüksek Lisans Tezi, İstanbul, 2006.
12. KARAYALÇIN, İ.İlhami, **Kantitatif Planlama ve Karar Verme Yöntemleri (Yöneylem “Hareket” Araştırması)**, Geliştirilmiş 3. Baskı, Mentş Kitabevi, İstanbul,1978.
13. KARAYALÇIN, İlhami, **Endüstri Mühendisliği Ve Üretim Yönetimi El Kitabı**, Çağlayan Kitabevi, İstanbul, 1986.
14. KORUCA HALİL İBRAHİM ;“Simülasyon Destekli Vardiya Planlama Modülü Geliştirilmesi”, **Gazi Üniversitesi, Müh. Mim. Fak. Der.** Cilt 25, No 3, 469-482, 2010
15. KÜÇÜKSİLLE ve GÜNGÖR,”İş Yoğunluğu Tahmini ve İşgücü Planlama: Süpermarket Uygulaması” **Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi**, Y.2009, C.14, S.3 s.91-109.
16. SEVİNÇ, C, “**Lojistik Planlama Çok Ürünlü Çok Kademeli İkmal Zincirindeki Depolarda Bulundurulması Gerekli Minimum Malzeme Miktarının Dinamik Programlama Yöntemiyle Optimizasyonu**”, Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsü, Yayınlanmış Yüksek Lisans Tezi, İzmir, 2008.

17. SEÇKİNER, S. U., KURT, M., “Bütünleşik Tur Rotasyon Çizelgeleme Yaklaşımı İle İşyükü Minimasyonu”, **Gazi Üniversitesi, Müh. Mim. Fak. Dergisi**, Cilt 20, No 3, 161-169, 2005.
18. SEZEN, K, **Dinamik Programlama**,1. Baskı, Ekin Kitapevi, Bursa,1998.
19. SEZEN, K, **Dinamik Programlama**, Genişletilmiş 2. Baskı, Ekin Kitapevi, Bursa,2007..
20. TAHA, H. A, **Yöneylem Araştırması**, 6. Baskı, Çeviren: Ş. Alp Baray, Şakir Esnaf, Literatür Yayıncılık, İstanbul, 2000.
21. TSİNLİ FİSTİKÇIOĞLU, Z., A., “Hastanelerde Hemşire Sayısının Planlanması için Hemşirelik Hizmetlerinde İşyükü Ölçme Sisteminden Yararlanılması ve Bir Model Önerisi”, **Yayımlanmamış Doktora Tezi**, İstanbul Ü_S
22. TOP, A., **Üretim Sistemleri Analiz ve Planlaması**, 1. Baskı, Melisa Matbaacılık, İstanbul,1994.
23. TURBAN E. and MEREDİTH, J,R, **Fundamentals Of Management Science**, 4. Baskı, Business Publications, Texas, 1988.
24. TÜTEK, H. ve GÜMÜŞOĞLU Ş, **Sayısal Yöntemler Yönetmel Yaklaşım**, 1. Baskı, Beta Basım Yayım Dağıtım, İstanbul,1994.
25. WINSTON W. L. **Operation Research Applications And Algorithms**,3. Baskı, Thomson Publishing, USA, 1994.
26. http://www.maden.org.tr/resimler/ekler/f68928ec5b6fae1_ek.pdf, Erişim Tarihi:8.12.2010
27. http://Fatmabeyzabas.files.wordpress.com/2010/07/yoneylem_arastirma_si.docx. Erişim Tarihi:8.12.2010