Politeknik Dergisi
Cilt: 6 Sayı: 3 s. 585-596, 2003

Journal of Polytechnic
Vol: 6 No: 3 pp. 585-596, 2003

KAYNAKLARIN KISITLI OLDUĞU KOŞULLARDA BİR İNŞAAT PROJESİNİN SEZGİSEL YÖNTEMLE PLANLANMASI

Recep KANIT
Gazi Üniversitesi Teknik Eğitim Fakültesi Yapı Eğitimi Bölümü
06500 Teknikokullar, ANKARA

ÖZET

Yaşadığımız son yüzyılda kaynakların uygun kullanımı ekonomik hayatın öncelikli hedefi haline gelmiştir. Bu bakımdan son yıllarda inşaat sektöründe kaynak kullanımı önem kazanmıştır. İnşaat üretiminde çok sayıda farklı imalat ve bu imalatlar içinde çok sayıda farklı kaynak gerekmektedir (iş gücü, makine gücü, malzeme, para vb.). İnşaat projelerinin boyutları oldukça büyük ve karmaşıktır. Bu projelerin gerçekleşmesi sürecinde, kaynakların öngörüldüğü zamanda ve öngörüldüğü miktarda bulunabileceği düşünülmemelidir. Bu bakımdan kaynak kullanımının da planlanması gerekmektedir.

Bu çalışmada kaynakların kısıtlı olduğu koşullarda programlama yöntemleri genel olarak incelenmiş ve sezgisel yöntem ile çözüm yapabilen 2 (iki) algoritmanın uygulanması açıklanmıştır. Uygulanan algoritmalar öncelik kuralı prensibine göre çalışmaktadır. İlk algoritmanın birinci önceliği, projenin başlangıcından bitişine kadar, faaliyetlerin sürelerinin büyükten küçüğe doğru sıralanması; ikinci önceliği süresi eşit faaliyetlerinden bolluk süresi az olana öncelik vermesi ve üçüncü önceliği de süresi az olan faaliyete öncelik vermesidir.

İkinci algoritmayla da; faaliyetlerden bolluğu az olana birinci öncelik, süresi fazla olana ikinci öncelik ve süresi az olana da üçüncü öncelik verilmektedir.

Bu algoritmalar bir köprü inşaatı projesine uygulanmış ve performansları incelenmiştir.Buna göre kaynakların sınırsız olduğu koşullarda 46 gün içerisinde tamamlanan proje; günlük kaynak 1 adet kazı makinesi, 80 m^2 kalıp, 10 ton demir, 16 m^3 beton, 15 işçi olarak alındığında; her iki algoritmayla yapılan programlamaya göre de 58 günde tamamlanmaktadır.

Anahtar Kelimeler: Proje programlama, sınırlı kaynak, sezgisel metotlar

PLANNING OF A CONTRUCTION PROJECT WITH INTUITIVE APPROACH UNDER LIMITED RESOURCES CONDITIONS

ABSTRACT

Adequate use of resources has become a target of high priority for the economics of the last century. Parallel to the general tendency, national use of resources in the construction industry has become more and more important the field of construction comprises of various sorts of production processes and each process require a number of different resources (labour, machinery and equipment, materials, finance, etc). Accordingly, a construction project, in many cases, may prove to be complicated and huge in terms of contents. When undertaking these projects, difficulties in providing the required amount of a specific resource at the required time are likely to arise. Consequently, a strict policy of planning for the use of resources is essential.

In this work various methods of programming have been worked out with a special emphasis on the cases of limited resources and the application of the algorithms employed, work according to the principle of priorities rule. The prime priority of the fist algorithm is putting all the activities, from the start to the end, into an order of decreasing lengths of period duration. The one with a shorter time of resource abundance takes the urgency. The activities with shortest duration are given the third place in terms of urgency.

In the second algorithm, the resource with the least abundance, was given the first priority whereas the are with a longer duration was considered to be of a lesser priority and the activity of the shortest run was third.

The both algorithms were applied on the construction works of a bridge and their performances were investigated. Under the conditions of unlimited resource, the construction was to be completed in 46 days whereas, one excavator, 80.0 of concrete template, 10 tons of iron, 10.0 m^3 of concrete and 15 workers, on the daily basic, a delay of 12 days was inevitable with the both algorithms.

Key Words: Project Planning, limited resource, Intuitive methods

1. GİRİŞ

Proje programlama, bir projedeki bütün faaliyetlerin (imalatların) öncelik ilişkileri ve bütün diğer etkenler gözetilerek sıralanmasının sağlanmasını içeren çalışmalar bütünüdür (1) şeklinde veya işlemlerin "başlama zamanlarının tespiti" ve yapılan "kaynak tahsisi" üzerinde verilen kararlarla çözüme ulaştıran çok amaçlı bir "karar problemi" şeklinde tanımlanabilir (2). Yöneylem araştırmasında ise "bir seçenekler kümesinden en az bir amaç ölçütüne göre en iyi sonucu veren çö-

züm seçeneğinin belirlenmesi" olarak da isimlendirilen bu programların yapısını aşağıdaki öğeler oluşturur.

- Karar değişkenleri; Çözüm sürecinde kesinleşen değerleriyle problemin bir çözüm seçeneğini belirleyen tüm değişkenler, karar değişkeni olarak nitelenir.
- Kısıtlar; Kısıtlar karar değişkenlerinin alabileceği değerleri sınırlayarak yalnız olurlu çözüm seçeneklerinin belirlenmesini sağlarlar. Örnek olarak bağıntı kısıtları, kaynak kısıtları, zaman kısıtları, vb.
- Amaçlar; Bir karar probleminin çözüm seçeneklerinin hangi ilkelere göre değerlendirileceğinin belirlenmesidir (3).

Günümüzde çeşitli proje planlama ve kontrol yöntemleri geliştirilmiştir. Bu yöntemleri genel olarak iki bölüme ayırabiliriz. Bunlardan ilki Gantt Şeması yöntemi ki, basit projelerin planlanması ve kontrolu amacıyla en yaygın kullanılan ve çubuk diyagramlardan oluşan bir planlama tekniğidir. Diğeri ise şebeke analizi yöntemidir. Bu yöntem birbiri ile bağlantılı bir dizi faaliyetten oluşan bir projenin planlanması amacıyla kullanılır.

Proje programlama, akış yapısı belirlenerek şebekesi oluşturulan bir projenin, sınırlı miktardaki kaynaklar kullanılarak en düşük toplam maliyetle gerçekleştirilmesi amacıyla, her faaliyet için başlama zamanının, faaliyete tahsis edilen kaynak türlerinin, kaynak tahsisi miktarının, kapasite zorlamadaki çalışma hız ve sürelerinin belirlenmesiyle çözüme ulaştırılan bir problemi ortaya koymaktadır. Şebeke analizi programlama teknikleri dört ana gurupta toplanır (4).

1. Kritik Yol Analizi: Diğer tekniklere göre daha basit olup kaynak kısıtlarını dikkate almadan en kısa proje bitiş süresini verir (5).

2. Zaman-Maliyet Dengeleme: Zamandan sağlanacak tasarrufların , maliyetleri ne ölçüde etkileyeceğini inceler (6). Kısaca kapasite üst sınırını dikkate almadan, kaynak artırımı yoluyla sürenin azaltılabileceğini varsayar.

3. Kaynak Dengeleme: Kapasite yüklenmesindeki değişimlerin minimizasyonunu ve bu değişimler ile ortaya çıkan maliyetlerin optimizasyonunu içerir (7).

4. Kaynak Kısıtı Altında Proje Programlama: Kaynakların sınırlı olduğu durumlarda ise, kritik yol ve bolluk gibi terimler gündeme gelir. Kritik yol üzerindeki bazı faaliyetler kaynakların yetersizliği durumunda ertelenebilirler (8). Proje süresinin kısaltılması ile ilave kaynak tahsisi gerekir.

2. SINIRLI KAYNAK KULLANIMI VE PROJE PROGRAMLAMA

Kaynak kısıtlı proje programlama farklı kombinasyonlara sahiptir. Bu programlamada en uygun veya en kesin çözümü bulmak, tüm kombinasyonların en iyisine ulaşmak ile mümkündür. Kaynak kısıtı altında bulunan projelerde optimum çözüm proje süresinin minimum yapılması ile mümkündür (9), (5). Bir projedeki faaliyetlerin büyük bir bölümü farklı kaynaklara ihtiyaç duyar ve kaynakların büyük bir bölümü de kısıtlıdır. Bu koşullardaki çözüm bekleyen proje programlama problemler, iki farklı ana yöntem ile çözümlenebilmektedir (10), (11), (12), (13).

2.1. Matematiksel Yöntemler

Kombinasyonlara sahip proje programlama, matematiksel ve sayılama teknikleri ile en uygun çözümün bulunması esasına dayanır. Olası tüm çözüm seçenekleri denenmeden en uygun çözümün bulunmasını amaçlar (14). Bu amaçla geliştirilen çok sayıda tam sayılı programlama, numaralama, dal-sınır ve bunların alt guruplarını oluşturan yöntemler bulunmaktadır (15). Bu yöntemler özellikle kısıtlı kaynak koşullarında, projenin müsaade edilen zaman içerisinde tamamlanacak şekilde çözülmesi esasına dayanır. Matematiksel yöntemler küçük ölçekli test problemlerinin çözümünde yeterli olmakla birlikte, karmaşık problemlerin çözümünde yetersiz kalmaktadır (16). Bu nedenle inşaat sektörü gibi kendine özgü koşulları ve faaliyet sayısı çok fazla olan projelerde uygulanmamaktadır.

2.1.1. Lineer ve tamsayılı programlama

Lineer programlama genel olarak matematiksel programları çözebilen güçlü bir teknik olmasına rağmen, sadece bazı özel durumlarda ve küçük projelerde uygulanabilmektedir (12), (4). Bu yöntem yenilenebilen kaynakların tam sayı birimlerle ifade edildiği, seçilen zaman biriminin tam sayı değişkeni olmadığı, yalnız bitiş-başlangıç bağıntılarının kullanıldığı ve bağıntı sürelerinin sıfır kabul edildiği çözüm metodu olarak kabul edilebilir (5), (17).

2.1.2. Dal-Sınır tekniği,

Dal-sınır tekniği ile, dallanma planı kullanılarak etkin programlama yapılabilir. Bu teknik ana programın kısmi dallara ayrılması ve her kısmi dalların alt sınırlara göre programlanması ile oluşturulur (18). Bu yöntem ile ilgili son yıllarda yoğun çalışmalar yapılmakta ve performansları incelenmektedir (5), (9). Yapılan çalışmalarda dal-sınır tekniği ile elde edilen sonuçlara herhangi bir sezgisel yöntemle ulaşılamayacağı görülmektedir. Ancak test edilen problemlerin 25 faaliyeti geçmesi halinde kesin kanaatin verilemeyeceği görülmüştür (19).

Tablo 1. Öncelik kuralları performansları

Sonuçlar	Sezgisel Öncelikler							
	MINSLK	RSM	LFT	RAN	GRU	GRD	SIO	MJP
Optimum süreyi aşma oranı (%)	5,6	6,8	6,7	11,4	13,1	13,1	15,3	16
Optimum süreyi veren problem sayısı	24	12	17	4	2	11	1	2

2.1.3. Sayılama teknikleri

Bu teknik 0-1 değişkenleri yardımı ve örtülü sayılama yöntemi ile çözüm kümesinin aranması esasına dayanır. Sezgisel algoritma ile belirlenen bir proje süresi üst sınır kabul edilir ve tek dal üzerinde oluşturulan düğümlerdeki alt sınır değerleri hesaplanır. Bu tekniğin, kaynakların ve faaliyetlerin az olduğu problemlerde yeterli performansta olduğu söylenebilir (20), (21).

2.2. Sezgisel Yöntemler

Kaynak kısıtı altındaki projelerin programlanmasında problemlerin boyutu ve şebekenin karakteristiklerine bağlı olarak, bazen matematiksel metotlar ile çözüme ulaşılamamaktadır. Bu durumda etkin ve güvenirliği yüksek, basit programlama kuralları kullanan sezgisel algoritmalar kullanılmaktadır. Sezgisel algoritmalar proje zamanı ve proje maliyetini minimize etmek üzerine yoğunlaşır (22). Araştırmacılar bu iki ana konuyu minimize edecek çözüm algoritmaları üretmişlerdir (23), (24), (25). Sezgisel algoritmalar büyük olasılıklar içerisinde en iyiye yakın veya önceden tespit edilmiş değerlendirme ölçütlerine göre, yeterli sayılabilecek çözümleri sağlayan yöntemlerdir(8). Sezgisel yöntemler, faaliyet bazında belirli önceliklere göre programlama yapmaktadır. Kaynakların yetersiz kaldığı durumlarda öncelik verilen faaliyet önce programlanmaktadır. Bu öncelikleri şu şekilde sıralayabiliriz (26):

- En kısa süreli faaliyet önce,
- En uzun faaliyet önce
- Bolluk miktarı az önce,
- Bolluk miktarı çok önce,
- Kendinden sonraki faaliyet sayısı az olan önce,
- Kendinden sonraki faaliyet sayısı çok önce,
- İlk gelen önce.

Sezgisel algoritmaların kullanmış oldukları öncelik kurallarının performansları Tablo 1'de verilmektedir (12).

MINSLK:En az bolluğu olan en önce
GRD :En fazla kaynak talebi
RSM :Kaynak sıralama yöntemi
SIO :En kısa süreli iş
LFT :Minimum geç başlama zamanı
MJP :Mümkün olan fazla iş
GRU :En fazla kaynak kullanan
RAN :Rassal faaliyet Seçimi

Tablo 1'e göre bolluğu az olan faaliyete öncelik verildiğinde, projenin performansı artmakta ve optimum süre için projeyi programlayabilmektedir. Sezgisel algoritmalar genel olarak seri yöntem ve paralel yöntem olarak ikiye ayrılmaktadır.

2.2.1. Seri yöntem

Bu yöntemde; proje için gerekli kaynakların tamamının bir arada, kaynak havuzunda bulunduğu, kaynağın görevini bitirdiğinde kaynak havuzuna döndüğü ve başka bir faaliyet için bu kaynağa ihtiyaç duyduğunda kaynak miktarının havuz içerisindeki değerinin esas alınacağı kabulü yapılır. Diğer bir yaklaşım ise bir faaliyete ayrılan kaynağın faaliyet bitene kadar o kaynağa tahsisi olduğu kabulüdür (27).

2.2.2. Paralel yöntem

Proje süresince her süre birimi üzerinde durulup, faaliyetlerin önceliklerinin yeniden gözden geçirildiği ve sadece o an ki süre birimi içerisinde öncelikli olan faaliyetin programlandığı yöntemdir (28), (29).

2.3. İleri Sezgisel Yöntemler

2.3.1. Genetik algoritma

Genetik algoritma, doğadaki evrim mekanizmasını örnek alan bir arama metodudur ve bir veri grubundan özel bir veriyi bulmak için kullanılır. Bu algoritmayı diğer algoritmalardan ayıran en önemli özellik, çözümün uygunluğunun rasgele seçilmesidir. Ancak bu rasgele seçimde, seçilme olasılıkları çözümlerin uygunluğunu belirlemektedir (30), (31).

Genetik algoritmalar çözüm kodları ile çalışır, çözüm kümelerini araştırır ve sadece tek çözüm vermez. Türetilmiş veya diğer yardımcı bilgiler yerine değişim bilgilerini kullanır ve ihtimal hesaplarını kullanarak çözüm üretir (32).

2.3.2. Tabu arama (Tabu search)

Tabu arama yöntemi üzerinde ilk defa 1989 yıllarında Glover ve Laguna çalışmaya başlamıştır. Bu teknik daha önce incelenmiş belli sayıda çözümü, tabu listesi olarak adlandırılan listede tutarak, o çözümlere geri dönmeyi bir zaman yasaklayan ve böylece aramayı çözüm uzayının daha iyi noktalarına yönlendiren bir arama yöntemidir (33).

3. MATERYAL VE METOT

İnşaat yatırımları gibi faaliyet sayısı çok fazla olan projelerin programlanmasında, matematiksel metotların performanslarının yeterli olmadığı yapılan bazı çalışmalarda vurgulanmakta ve faaliyet sayısı fazla olan proje programlama problemlerinde, sezgisel yöntemin kullanılması gerektiği belirtilmektedir. Bu bağlamda, inşaat yatırımlarında kullanılabilecek iki sezgisel algoritmanın aşamaları bu çalışmada ele alınmış ve bu algoritmaların performansı bir örnek üzerinde test edilmiştir. Kullanılan algoritma bir çok literatürde de yer alan Brook's algoritmasıdır. Sınırlı kaynakların göz önüne alındığı sezgisel programlama modellerinde, projenin programlanması yapılırken öncelik kuralı geçerli olmaktadır. Çalışmamızda sezgisel yaklaşım ile çözüm yapabilen iki algoritmanın uygulanması açıklanmıştır. Kullanılan algoritmalar öncelik kuralı prensibine göre çalışmaktadır. Birinci algoritmanın birinci önceliği, projenin başlangıçından bitişine kadar olan sürede faaliyet süreçlerinin büyükten küçüğe doğru planlanmasını sağlamaktadır. Bu sürelerinin eşit olması durumunda ikinci öncelik, bolluk süresi az olan ve üçüncül öncelik de proje süresi kısa olan faaliyete verilmektedir. Diğer algoritma da ise; faaliyetlerden bolluğu az olana birinci öncelik, süresi fazla olana ikinci öncelik ve süresi az olana üçüncü öncelik verilmektedir.

Bu çalışmada, bir köprü projesine ait şebeke oluşturulmuş, bu şebeke sınırlı kaynak koşullarında açıklanan algoritmalar ile çözülmüş ve projenin programlanmış şekli çubuk diyagram üzerinde gösterilmiştir.

3.1. Algoritma ve Evreleri

Projenin programlanmasında kullanılan algoritmanın 1.ve 3. adımlar arası sınırsız proje programlama modelleri için kullanılmakta ve kaynakların kısıtlı olduğu koşullarda algoritma devam ettirilmektedir. Algoritma, kaynakların yetersiz kaldığı durumlarda iki farklı öncelik kuralı ile çözüm yapmaktadır. Birinci algoritmanın öncelik kuralına göre; faaliyetlerin başlangıcından proje sonuna kadar ki zamanların büyükten küçüğe doğru sıralanması yapılarak, en büyük olan zamana sahip faaliyet önce, sürelerin eşit olması durumunda ikinci öncelik kuralı bolluğu az olan önce ve üçüncü olarak da işlem süresi az olan önce programlanmaktadır. İkinci algoritmada ise birinci olarak bolluğu az olan faaliyet, ikinci olarak projenin başlangıcından proje sonuna kadar faaliyet süreleri büyükten küçüğe doğru sıralanmakta ve süresi en büyük olan faaliyet önce, üçüncü olarak da süreleri az olan faaliyet önce programlanmaktadır (3).

Algoritmanın çalışma prensibi aşağıda açıklanmıştır.

1. Şebeke başlangıçtan sonra doğru takip edilerek faaliyetlerin en erken başlama ve bitiş, olayların en erken oluşma zamanları hesaplanır.

a. Projenin başlangıcının en erken başlama zamanı sıfır (0) alınır. Bu şekilde faaliyetlerin en erken başlama zamanları (ES_{ij}) belirlenir.

b. En erken bitiş zamanları ($EF_{ij}= ES_{ij}+t_{ij}$) hesaplanır.

c. Her faaliyet için, kendisine bağlanan bütün faaliyetlerin en erken bitiş zamanları belirlenmişse, en erken oluşma zamanı bulunur (T_i^E = max. EF_{ij})

d. En erken oluşma zamanları belirlenen bütün faaliyetlerin en erken başlama zamanları ($ES_{ij}=T_i^E$) eşitlenir. Bu şekilde bütün faaliyetlerin

erken başlama, bitiş zamanları ve en erken oluşma zamanları belirlenir.

2. Şebeke bu sefer de sondan başa doğru takip edilerek faaliyetlerin en geç başlama, bitiş ve faaliyetlerin en geç oluşma zamanları belirlenir.

a. Projenin bitiş faaliyetlerinin en geç oluşma zamanı en erken oluşma zamanına eşitlenir ($Tp^L = Tp^E$). Böylece bu faaliyete bağlanan bütün faaliyetlerin en geç bitiş zamanları da ($LF_{ij}=T_p^L$) bulunur.

b. En geç bitiş zamanları belirlenen bütün faaliyetlerin en geç başlama zamanları ($LS_{ij}=LF_{ij} - t_{ij}$) hesaplanır.

c. Her faaliyet için, eğer kendine bağımlı olan bütün faaliyetlerin en geç başlama zamanları hesaplanmamışsa, en geç oluşma zamanı (T_i^L= min LS_{ij}) hesaplanır.

d. En geç oluşma zamanı hesaplanmış bütün faaliyetlerin en geç bitiş zamanları ($LF_{ji}=T_i^L$) eşitlenir. Bu şekilde en geç başlama, bitiş ve oluşma zamanları belirlenir.

3. İlk iki adımda elde edilen sonuçlar tabloya dökülür. Proje bitiş süresi T_p^L olarak belirlenir. Her faaliyet bolluğu ($S_{ij}=LS_{ij} - ES_{ij}$) hesaplanır. Bolluğu sıfır (0) olan faaliyetler kritik faaliyet olarak adlandırılır.

4. Bütün faaliyetlerin başlangıçtan bitimine kadar geçen süre hesaplanır ($ABPBS_{ij}=T_p^E - T_i^E - S_{ij}$) ve faaliyetler değerlerine göre küçükten büyüğe doğru sıralanır.

5. Her faaliyet için, sözkonusu faaliyetin bir birim zaman içindeki kaynak ihtiyacının o kaynağın bir birim zaman içinde temin edilebilme miktarının üzerinde olmaması gereği kontrol edilir. ($R_{ij}^n <= R_{max}^n$)

6. Başlangıç hesaplamaları olarak zaman sıfır'a (0) eşitlenir. Bütün faaliyetler programlanmamış kabul edilir. Her zaman birimi için her kaynağın miktarı o kaynağın max temin edilebilen miktarına eşitlenir. Bütün faaliyetlerin başlama zamanları en erken başlama zamanına eşitlenir.

z=0 (zaman),

RT_q^n = n Kaynaktan (q-1,q) birim zaman içinde elde bulunan miktar.

R^n_{max} = n Kaynağın bir birim zaman içinde max. temin miktar.

B_{ij} = Başlama zamanı(faaliyet için)

$RT_q^n = R^n_{max}$

$B_{ij} = ES_{ij}$

7. Faaliyetlerin ABPBS'lerine göre büyükten küçüğe doğru sırlanmış listenin en başına gidilir.

8. Sıradaki ilk programlanmamış ve başlama zamanı bulunan zamana eşit veya küçük olan faaliyet ele alınır. Eğer o faaliyet için eldeki kaynak miktarları yeterli ise faaliyet programlanır. Eğer faaliyetin başlama zamanı bulunan zamandan küçükse, başlama zamanı bulunan zamana B_{uw}=z eşitlenir ve sonra 9. adıma geçilir. Aksi ise 8. adımın başına dönülür. Eğer listenin başına gelinmişse bulunan zaman için herhangi bir kaynak alokasyonu yapılamayacağı anlaşıldığından zaman bir birim arttırılır. z=z+1 ve 7. adıma geri gidilir.

9. Söz konusu faaliyetin kullanıldığı bütün kaynaklar için, bulunduğu zamandan itibaren faaliyetinin süresi kadar faaliyete kaynak alokasyonu yapılır ve her kaynağın ilgili zaman dilimleri için mevcut miktarı aşağıdaki şekilde azaltılır.

$RT_q^n = RT_q^n - R^n_{uw}$ n=1.....x, q=z+1.....z+t, uw= Kaynak alokasyonu yapılan faaliyet

10.Söz konusu faaliyetin bitiş zamanı, $F_{uw} = B_{uw} + t_{uw}$ olarak hesaplanır.

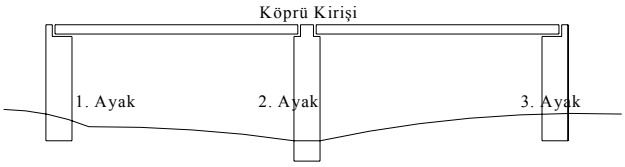
11.Kaynak alokasyonu yapılan faaliyete bağımlı bütün faaliyetlerin başlangıç zamanları, eğer başlangıç zamanları alokasyonu yapılmış faaliyetin bitiş zamanından küçükse, söz konusu faaliyetin bitiş zamanına eşitlenir. $B_{wj} = F_{uw}$

12.Eğer bütün faaliyetler programlanmış ise 13. adıma geçilir. Aksi durumda 8. adıma gedilir.

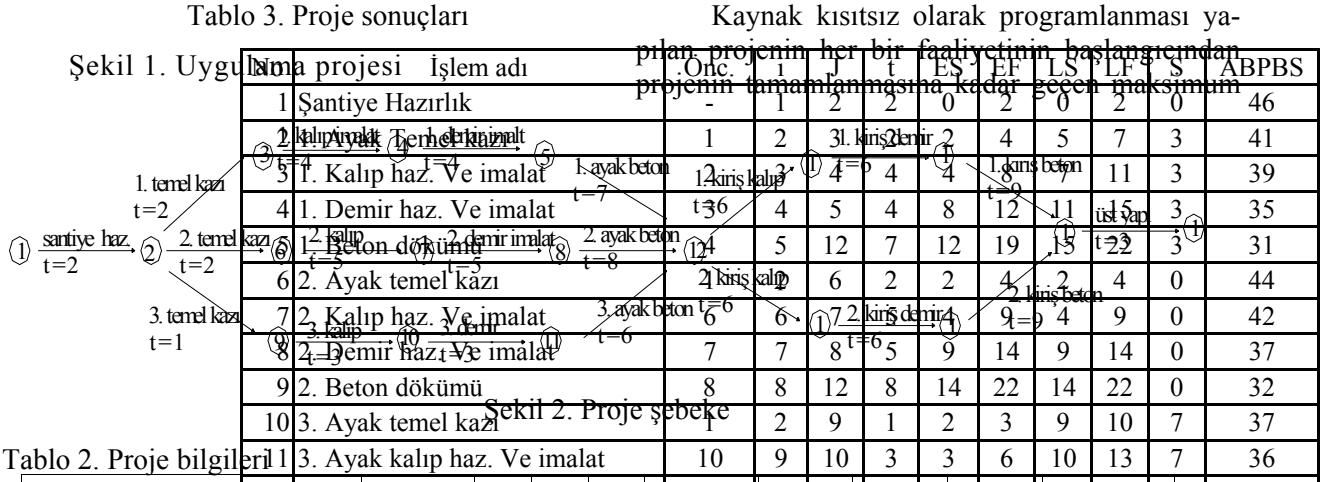
13.Projenin çubuk diyagramı çizilerek kaynakların zaman dilimine göre kullanım tablosu düzenlenir.

3.2. Proje Uygulaması

Projemiz 20 faaliyetten oluşan, beş adet kaynak ihtiyacı olan küçük bir köprü projesi olarak düşünülmüş ve Şekil 1' de verilmiştir. Faaliyetler ve her bir faaliyetin koşulları, zamanları, başlangıç ve bitiş düğümleri Tablo 2'de gösterilmiş ve bu koşullara göre şebeke çizilerek Şekil 2'de verilmiştir. Kaynakların kısıtları her gün için aynı ve kazı makinesi 1 adet, kalıp 80 m^2, demir 10 ton, beton 16 m^3, iş gücü ise 15 adet işçidir.



Şekil 1. Uygulama projesi



Şekil 2. Proje şebeke

Tablo 2. Proje bilgileri

	Faaliyet No									
	Faaliyet adı	Ön işlem sayısı	i	j	t	Kazı mak. (1 adet)	Kalıp (80 m2)	demir (10 ton)	beton (16 m3)	İşgücü (15 işçi)
1	Şantiye Hazırlık		1	2	2	-	-	-	-	-
2	1. Ayak Temel kazı	1	2	3	2	1	-	-	-	-
3	1. Kalıp haz. Ve imalat	2	3	4	4	-	40	-	-	-
4	1. Demir haz. Ve imalat	3	4	5	4	-	-	10	-	-
5	1. Beton dökümü	4	5	12	7	-	-	-	8	-
6	2. Ayak temel kaz	1	2	6	2	1	-	-	-	-
7	2. Kalıp haz. Ve imalat	6	6	7	5	-	40	-	-	-
8	2. Demir haz. Ve imalat	7	7	8	5	-	-	10	-	-
9	2. Beton dökümü	8	8	12	8	-	-	-	8	4
10	3. Ayak temel kazı	1	2	9	1	1	-	-	-	2
11	3. Ayak kalıp haz. Ve imalat	10	9	10	3	-	40			2
12	3. Ayak demir haz. Ve imalat	11	10	11	3	-	-	10	-	5
13	3. Ayak beton dökümü	12	11	12	6	-	-	-	8	4
14	1. Kiriş kalıp	5,9,13	12	13	6	-	40			2
15	1. Kiriş demir	14	13	14	6	-	-	10	-	5
16	1. Kiriş beton	15	14	17	9	-	-	-	8	4
17	2. Kiriş kalıp	5,9,13	12	15	6	-	40			2
18	2. Kiriş demir	17	15	16	6	-	-	10	-	5
19	2. Kiriş beton	18	16	17	9	-	-	-	8	4
20	üst yapı	19	17	18	3	-	-	-	8	4

3.3. Proje Sonuçları

Algoritmada belirtildiği gibi bütün faaliyetlerin gerekli hesaplamaları, kaynak kısıtsız ve kaynak kısıtlı koşullara göre yapılmıştır. Öncelik olarak projenin Erken Başlama (ES$_{ij}$), Erken Bitiş (EF$_{ij}$), Geç Başlangıç (LS$_{ij}$), Geç Bitiş (LF$_{ij}$) ve Toplam Bollukları (S) hesaplanmıştır. Bolukları sıfır (0) olan faaliyetler kritik faaliyet olarak adlandırılmıştır. Yapılan hesaplamalar Tablo 3'de verilmiştir. Daha sonra bu durumda kaynak kısıtsız olarak projenin programlanması yapılmış ve çubuk diyagramı Ek 1'de verilmiştir.

Tablo 3. Proje sonuçları

Nu	İşlem adı	Önc.	i	j	t	ES	EF	LS	LF	S	ABPBS
1	Şantiye Hazırlık	-	1	2	2	0	2	0	2	0	46
2	1. Ayak Temel kazı	1	2	3	2	2	4	5	7	3	41
3	1. Kalıp haz. Ve imalat	2	3	4	4	4	8	7	11	3	39
4	1. Demir haz. Ve imalat	3	4	5	4	8	12	11	15	3	35
5	1. Beton dökümü	4	5	12	7	12	19	15	22	3	31
6	2. Ayak temel kazı	1	2	6	2	2	4	2	4	0	44
7	2. Kalıp haz. Ve imalat	6	6	7	5	4	9	4	9	0	42
8	2. Demir haz. Ve imalat	7	7	8	5	9	14	9	14	0	37
9	2. Beton dökümü	8	8	12	8	14	22	14	22	0	32
10	3. Ayak temel kazı	1	2	9	1	2	3	9	10	7	37
11	3. Ayak kalıp haz. Ve imalat	10	9	10	3	3	6	10	13	7	36
12	3. Ayak demir haz. Ve imalat	11	10	11	3	6	9	13	16	7	33
13	3. Ayak beton dökümü	12	11	12	6	9	15	16	22	7	30
14	1. Kiriş kalıp	5,9,13	12	13	6	22	28	22	28	0	24
15	1. Kiriş demir	14	13	14	6	28	34	28	34	0	18
16	1. Kiriş beton	15	14	17	9	34	43	34	43	0	12
17	2. Kiriş kalıp	5,9,13	12	15	6	22	28	22	28	0	24
18	2. Kiriş demir	17	15	16	6	28	34	28	34	0	18
19	2. Kiriş beton	18	16	17	9	34	43	34	43	0	12
20	üst yapı	19	17	18	3	43	46	43	46	0	3

Kaynak kısıtsız olarak programlanması yapılan projenin her bir faaliyetinin başlangıcından projenin tamamlanmasına kadar geçen maksimum süreleri (ABPBS) hesaplanmış ve bu sürenin büyükten küçüğe doğru sıralaması yapılmıştır. İkinci algoritmada ise faaliyetlerin bolukları hesaplanmış ve bolukları küçük olan faaliyetler öncelikli olarak programlanmıştır. Bu algoritmaya göre ABPBS'leri hesaplanan faaliyetlerin programlama sırası, bilgileri ve bolluk sayılarının küçükten büyüğe doğru sıralanması ile çözüm yapan algoritmanın işlem sırası da Tablo 4' de verilmiştir. Bu sıralamaya göre programlama yapılmış ve proje başlangıçından itibaren her gün için faaliyetlerin kaynak kullanımı hesaplanmıştır. Kaynakların ye-

Tablo 4. Algoritmaların işlem sıraları

Faaliyetnin başlangıcından proje bitiş süresine göre çözüm sırası		Bolluk miktarına göre programlama çözüm sırası	
Nu	İşlem adı	Nu	İşlem adı
1	Şantiye Hazırlık	1	Şantiye Hazırlık
4	1. Ayak Temel kazı	6	1. Ayak Temel kazı
5	1. Kalıp haz. Ve imalat	7	1. Kalıp haz. Ve imalat
9	1. Demir haz. Ve imalat	8	1. Demir haz. Ve imalat
12	1. Beton dökümü	9	1. Beton dökümü
2	2. Ayak temel kazı	2	2. Ayak temel kazı
3	2. Kalıp haz. Ve imalat	3	2. Kalıp haz. Ve imalat
11	2. Demir haz. Ve imalat	4	2. Demir haz. Ve imalat
7	2. Beton dökümü	5	2. Beton dökümü
8	3. Ayak temel kazı	10	3. Ayak temel kazı
10	3. Ayak kalıp haz. Ve imalat	11	3. Ayak kalıp haz. Ve imalat
13	3. Ayak demir haz. Ve imalat	12	3. Ayak demir haz. Ve imalat
14	3. Ayak beton dökümü	13	3. Ayak beton dökümü
15	1. Kiriş kalıp	14	1. Kiriş kalıp
16	1. Kiriş demir	15	1. Kiriş demir
18	1. Kiriş beton	16	1. Kiriş beton
17	2. Kiriş kalıp	17	2. Kiriş kalıp
19	2. Kiriş demir	18	2. Kiriş demir
20	2. Kiriş beton	19	2. Kiriş beton
	üst yapı	20	üst yapı

tersiz olduğu durumda faaliyetin programlanması bir sonraki güne ötelenmesi yapılmıştır. Sınırlı kaynak koşullarında yapılan programlama sonucunda oluşan birinci algoritmanın çubuk diyagramı Ek-2'de, bolluk sayısına göre programlama yapan algoritmanın çubuk diyagramı ise Ek-3'de verilmiştir.

4. SONUÇ

Çalışmanın örneğini oluşturan köprü inşaatına ait faaliyetler serisi, öncelik kuralı ile çözüm yapabilen sezgisel algoritma ile çözülmüş, proje süresi kaynakların kısıtsız olduğu koşullarda 46 gün; 1 adet kazı makinası, 80 m^2 kalıp, 10 ton demir, 16 m^3 beton, 15 adet işgücü kısıtlı koşullarında ise her iki farklı öncelik kuralı ile çözüm yapabilen algoritmada da 58 gün olarak bulunmuştur. Kaynakların kısıtlı olduğu koşullarda proje süresinin her iki algoritmada da 16 gün geç bittiği görülmüştür.

5. KAYNAKLAR

1. Groth, R., Erbsloch, F.D., Strombach, M.E., 1982, Project Management in Mittelbetrieben, Deutcher Instituts -Verlag, pp. 268-288
2. Baker, K.R.1974, Introduction to Sequencing and Scheduling, John Willey Publishing, New York
3. Moder, J.J., Philips, C.R., 1970, Project Management with CPM and PERT, Reinhold Company, Vol. 21 No 16, 718-722
4. Parker, R.G., 1995, Deterministic Scheduling Theory, Chapman&Hall, Vol. 17, pp. 767-776.
5. Herroelen, W.S., Reyck, B.D., Demeulemeester, E., 1998, Resource-Constrained Project Scheduling-A Survey of recent Developments Computer Operational Research, Vol. 25 No 1, 45-60
6. Goyal, S.K., 1975, A Note on a Simple CPM Time-Cost Trade-off Algorithm, Management Science, Vol. 17 No 4, pp.359-379
7. Carruthers, J.A, Battersby, A.,1966, Advances in Critical Path methods, Operations Research Quarterly, Vol. 13 No 6, 359-377
8. Wiest, J.D., 1967, Heuristic Model for Scheduling Large Projects with limited Resources, Management Science, Vol. 1 No3, 121-130
9. Brucker P, Drexl A, Möhring, R., Neumann, K., Pesch, E., 1999, Resource-constrained project schedulig: Notation, classification, models and methods, European Journal Of Operation Research Society, Vol. 25, pp. 279-302
10. Willis, R.J., 1985, Critical Path Analysis and Resource Constrained Project Scheduling-Theory and Practise, European Journal Of Operation Research Society, Vol. 21 No 8, pp.149-155.
11. Buffa, E.S., 1979, Productions and Inventory Systems Planning and Control, Homewood Illinois, Vol. 5 No 4, pp.297-311.
12. Davis, E.W., 1975, Project Scheduling under Resource Constraints-Historical Review and Categorisation of Procedures, AIIIE Transactions, Vol. 23 No3, pp.255-275
13. Herroelen, W.S., 1972, Resource-Constrained Project Scheduling-The State of the Art, Operations Research Quarterly, Vol. 25 No 4, pp. 279-302
14. Patterson, J.H.,1984, A Comparison of Exact Approaches For Solving The Multiple Constrained Resource Project Scheduling problem, Management Science, Vol. 30 No 7, pp. 552-559
15. Öcal, M. E., 1991, Yapı Projelerinde Kullanılan Kaynakların Planlama Ve Kontrolünde Şebeke Analizi Ve Çubuk Diyagrama Dayalı Karma Bir Model Önerisi Ve Uygulaması, Çukurova Üniversitesi, doktora Tezi, Adana
16. Thesen A.,1976, Heuristic Scheduling of Activities Under Resource and Precedence Restrictions, Management Science, Vol. 23 No 4
17. Oğuz, O., Bala, H., 1994, A Comparative Study Of Computational Procedures For The Resource Constrained Project Scheduling Problem, European Journal Of Operation Research Society, Vol. 72, pp. 406-416
18. Dorndorf, U.,Pesch, E., Phan-Huy, T., 2000, A Branch and Bound Algorithm for The Resource Constrained Project Scheduling Problem, Math. Met. of Operation Research, Vol. 15 No 1, 89-98.
19. Hastings, N.A., Willis R.J.,1976, Project Scheduling with Resources Constraint Using Branch and Bound Method, Operations Research Quarterly, Vol. 27 No 2, 341-349.

20. Scharage, L., 1970, Solving Resource Constrained Network Problems by Implicit Enumeration- Non-Premptive Case, Operations Research, Vol. 18, pp. 263-278

21. Scharage, L.1972, Solving Resource Constrained Network Problems by Implicit Enumeration- Premptive Case, Operations Research, Vol. 20, pp. 668-677

22. Ulusoy, G., Özdamar L., 1996, A heuristic Scheduling Algorithm for Improving the duration and net present value of a project, Int. Journal of Operation, Vol. 38, pp. 315-319.

23. Bell, C.E., Han, J.,1991, A new heuristic solution method in resource constrained project scheduling problems, Naval Research Logistic Quarterly, Vol. 49, pp.41-44.

24. Boctor, F.,F., 1990, Some Efficient Multi-Heuristic Procedures For Resource Constrained Project Scheduling, European Journal Of Operation Research Society, Vol. 40, pp. 1145-1152

25. Ulusoy, G., Özdamar L.1989, Heuristic Performance And Network / Resource Characteristics In Resource-Constrained Project Scheduling, Journal Of Operational Research Society, Vol. 127, pp. 619-638.

26. Panwalkar, S.S., Iskander, W., 1977, A Survey of Scheduling Rules, Operations Research, Vol. 25 No 1, pp. 45-6

27. Uğur, İ., 1988, Kaynak Kısıtlı Proje Programlama Problemlerinin Çözümü İçin Yeni Yöntem ve Algoritmalar, İTÜ, Fen Bilimleri Enstitüsü Doktora Tezi, İstanbul.

28. Ahuja, H.N.,1976, Construction Performance Control by Network, John Willey Publishing, New York

29. Klein, R.,2000, Bi-directional Planning: Improving Priory Rule-Based Heuristic for Scheduling Resource-Constrained Project, European Journal Of Operation Research Society, 2. Edition, 101-106

30. Leu, S.S., Yang, C., 1999, A Genetic-Algorithm - Based Resource - Constrained Scheduling system, Construction Management and Economics, Vol. 100, pp.134-141

31. Mori, M., Tseng, C.C., 1997, A Genetic Algorithm for Multi-Mode Resource Constrained Project scheduling problem, European Journal Of Operation Research Society, Vol. 45 No 7, 733-750

32. Albrechts, C., Sonke, H.1998, Competitive genetic Algorithm for Resource Constrained project Scheduling, Vol. 112 No 3, pp. 351-361

33. İçmeli, O., Erengüç, S.S., 1994, Tabu search Procedure for the Resource Constrained Project

Ek-1 Kaynak kısıtsız proje programlama

no	İşlem adı	i	j	t	kazı mak	kalıp	demir	beton	iş gücü
1	Şantiye Haz.	1	2	2	-	-	-	-	-
2	1. T. kazı	2	3	2	1	-	-	-	2
3	1. Kalıp	3	4	4	-	40			2
4	1. Demir	4	5	4	-	-	10	-	5
5	1. Beton	5	12	7	-	-	-	8	4
6	2. T. kazı	2	6	2	1	-	-	-	2
7	2. Kalıp	6	7	5	-	40			2
8	2. Demir	7	8	5	-	-	10	-	5
9	2. Beton	8	12	8	-	-	-	8	4
10	3. temel	2	9	1	1	-	-	-	2
11	3. kalıp	9	10	3	-	40			2
12	3. Ayak	10	11	3	-	-	10	-	5
13	3. Ayak	11	12	6	-	-	-	8	4
14	1. K. Kalıp	12	13	6	-	40			2
15	1. K. Demir	13	14	6	-	-	10	-	5
16	1. K. Beton	14	17	9	-	-	-	8	4
17	2. K. kalıp	12	15	6	-	40			2
18	2. K. demir	15	16	6	-	-	10	-	5
19	2. K. beton	16	17	9	-	-	-	8	4
20	üst yapı	17	18	3	-	-	-	8	4

Gün

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
kazı makinası (1 adet)	0	0	3	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
kalıp (80 m2)	0	0	0	40	120	120	80	80	40	0	0	0	0	0	0	0	0	0	0	0	0	0	80	80	80	80	80	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
demir (10 ton)	0	0	0	0	0	0	10	10	20	20	20	20	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	20	20	20	20	20	0	0	0	0	0	0	0	0	0	0	0	0
beton (16 m3)	0	0	0	0	0	0	0	0	0	8	8	8	16	16	24	16	16	16	16	8	8	8	0	0	0	0	0	0	0	0	0	0	0	0	16	16	16	16	16	16	16	16	16	8	8	8
iş gücü (15 işçi)	0	0	6	6	6	6	9	9	12	14	14	14	13	13	12	8	8	8	8	4	4	4	4	4	4	4	4	4	10	10	10	10	10	10	8	8	8	8	8	8	8	8	8	4	4	4

Ek-2 İlk gelen önce, bolluk ve işlem zamanı küçük olan önceliklerine göre programlama

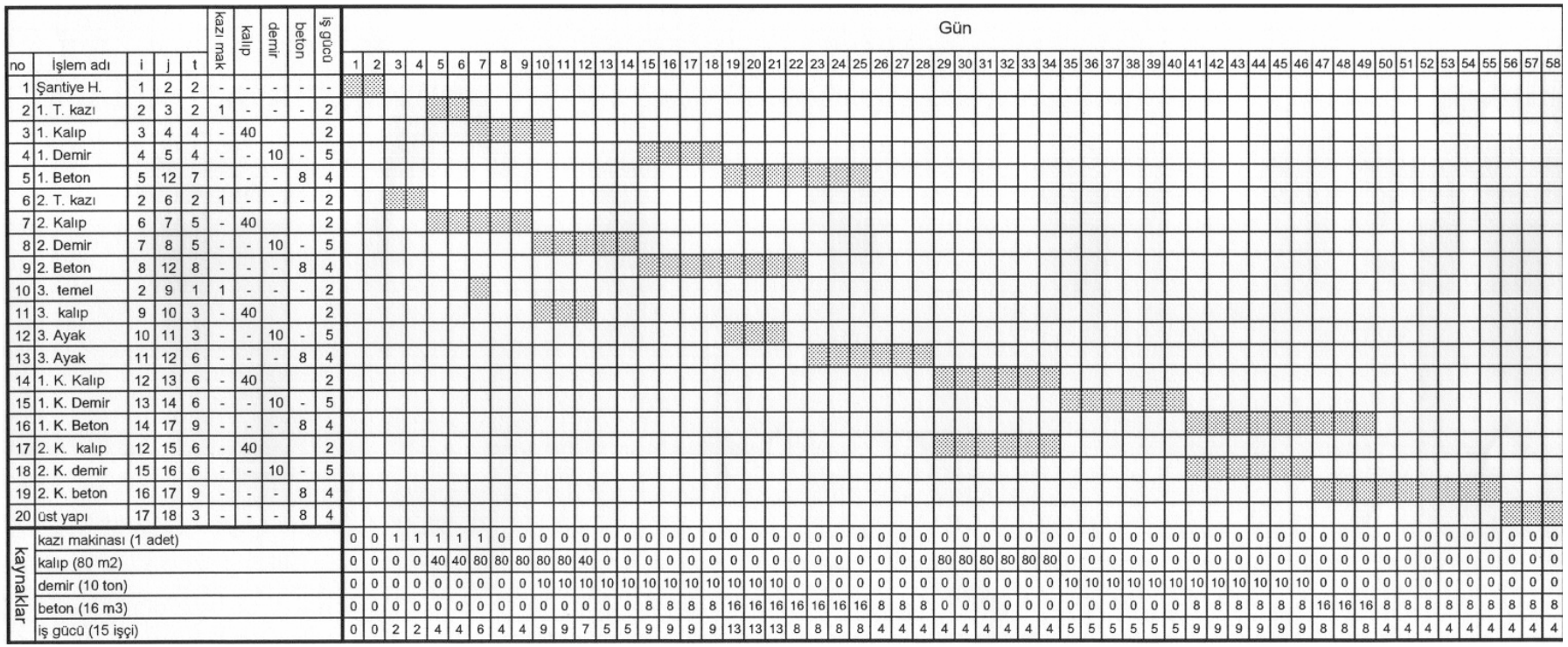
no	İşlem adı	i	j	t	kazı mak	kalıp	demir	beton	iş gücü
1	Şantiye H.	1	2	2	-	-	-	-	-
2	1. T. kazı	2	3	2	1	-	-	-	2
3	1. Kalıp	3	4	4	-	40			2
4	1. Demir	4	5	4	-	-	10	-	5
5	1. Beton	5	12	7	-	-	-	8	4
6	2. T. kazı	2	6	2	1	-	-	-	2
7	2. Kalıp	6	7	5	-	40			2
8	2. Demir	7	8	5	-	-	10	-	5
9	2. Beton	8	12	8	-	-	-	8	4
10	3. temel	2	9	1	1	-	-	-	2
11	3. kalıp	9	10	3	-	40			2
12	3. Ayak	10	11	3	-	-	10	-	5
13	3. Ayak	11	12	6	-	-	-	8	4
14	1. K. Kalıp	12	13	6	-	40			2
15	1. K. Demir	13	14	6	-	-	10	-	5
16	1. K. Beton	14	17	9	-	-	-	8	4
17	2. K. kalıp	12	15	6	-	40			2
18	2. K. demir	15	16	6	-	-	10	-	5
19	2. K. beton	16	17	9	-	-	-	8	4
20	üst yapı	17	18	3	-	-	-	8	4

Gün

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
kaynaklar	kazı makinası (1 adet)	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	kalıp (80 m2)	0	0	0	0	40	40	80	80	80	80	80	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	80	80	80	80	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	demir (10 ton)	0	0	0	0	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0
	beton (16 m3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8	8	8	16	16	16	16	16	16	16	8	8	8	0	0	0	0	0	0	0	0	0	0	0	0	8	8	8	8	8	8	16	16	16	8	8	8	8	8	8	8	8	8
	iş gücü (15 işçi)	0	0	2	2	4	4	6	4	4	9	9	7	5	5	9	9	9	9	13	13	13	8	8	8	8	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	9	9	9	9	9	9	8	8	8	4	4	4	4	4	4	4	4	4

Ek-3 Bolluk az olan, İlk gelen önce ve işlem zamanı küçük olan önceliklerine göre programlama

no	İşlem adı	i	j	t	kazı mak	kalıp	demir	beton	iş gücü
1	Şantiye H.	1	2	2	-	-	-	-	-
2	1. T. kazı	2	3	2	1	-	-	-	2
3	1. Kalıp	3	4	4	-	40			2
4	1. Demir	4	5	4	-	-	10	-	5
5	1. Beton	5	12	7	-	-	-	8	4
6	2. T. kazı	2	6	2	1	-	-	-	2
7	2. Kalıp	6	7	5	-	40			2
8	2. Demir	7	8	5	-	-	10	-	5
9	2. Beton	8	12	8	-	-	-	8	4
10	3. temel	2	9	1	1	-	-	-	2
11	3. kalıp	9	10	3	-	40			2
12	3. Ayak	10	11	3	-	-	10	-	5
13	3. Ayak	11	12	6	-	-	-	8	4
14	1. K. Kalıp	12	13	6	-	40			2
15	1. K. Demir	13	14	6	-	-	10	-	5
16	1. K. Beton	14	17	9	-	-	-	8	4
17	2. K. kalıp	12	15	6	-	40			2
18	2. K. demir	15	16	6	-	-	10	-	5
19	2. K. beton	16	17	9	-	-	-	8	4
20	üst yapı	17	18	3	-	-	-	8	4



kaynaklar	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
kazı makinası (1 adet)	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
kalıp (80 m2)	0	0	0	0	40	40	80	80	80	80	80	40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	80	80	80	80	80	80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
demir (10 ton)	0	0	0	0	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	10	10	10	10	10	10	10	10	10	10	10	10	0	0	0	0	0	0	0	0	0	0	0	0
beton (16 m3)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	8	8	8	16	16	16	16	16	16	16	8	8	8	0	0	0	0	0	0	0	0	0	0	0	0	8	8	8	8	8	8	16	16	16	8	8	8	8	8	8	8	8	8
iş gücü (15 işçi)	0	0	2	2	4	4	6	4	4	9	9	7	5	5	9	9	9	9	13	13	13	8	8	8	8	4	4	4	4	4	4	4	4	4	5	5	5	5	5	5	9	9	9	9	9	9	8	8	8	4	4	4	4	4	4	4	4	4