

2226

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ İNŞAAT FAKÜLTESİ

BİR DEMİRYOLU AĞINDA BOŞ YÜK ÜNİTELERİ ARZ VE TALEP TRAFİĞİ AKIMININ DENGELENMESİ

İstanbul Teknik Üniversitesi İnşaat Fakültesince
«Doktor»
ünvanının verilmesi için kabul edilen tezdır.

Yük. Müh. Mehmet Ayhan SAYRAÇ

Tezin Dekanlığa Verildiği Tarih : 3/6/1977
Tezin Savunulduğu Tarih : 8/9/1977

Doktorayı Yöneten Profesör : Prof. Enver BERKMEN
Diğer Jüri Üyeleri : Prof. Faruk UMAR
Prof. Kemal KUTLU

İSTANBUL TEKNİK ÜNİVERSİTESİ
MÜH. - MİM. FAKÜLTESİ MATBAASI
1 9 7 7

T. C.
Yükseköğretim Kurulu
Dokümantasyon Merkezi



Bu alıřmanın yapılmasında büyük desteęini ve yakın ilgilerini esirgemeyen Prof.Enver BERKMEN'e minnet ve teřekkürlerimi sunarım.

İ Ç İ N D E K İ L E R

Sayfa

ÖZET	IX
SUMMARY	XI
BÖLÜM I	1
1. GİRİŞ VE AMAÇ	1
1.1. GENEL	1
1.2. DEMİRYOLU ÜNİTESİNİN TARIFI	1
1.3. DAĞITILACAK ÜNİTELERİN TİPİ	2
1.4. ÜNİTELERİN YAPTIKLARI HAREKETLER	2
1.5. BOŞ ÜNİTELERİN DAĞITIMI İÇİN HALEN KULLANILMAKTA OLAN METOD	3
1.6. ÇALIŞMANIN AMACI	4
BÖLÜM II	5
2. BOŞ ÜNİTELERİN DAĞITIMINDA KULLANILAN MATEMATİK TEKNİĞİ	5
2.1. LİNEER PROGRAMLAMA	5
2.1.1. Taşın Problemi	5
2.1.1.1. Genel	5
2.1.1.2. Problemin Formülasyonu	5
2.1.1.3. Maliyet Matrisinin Transformasyonu	9
2.1.1.4. İlk Çözüm	12
2.1.1.5. Çözümün İslahı	17
2.1.1.6. Gölge Maliyetler Metodu	21
2.1.1.7. Birden Fazla Optimal Çözümün Olması Hali	26
2.1.2. Taşın Probleminin Özel Halleri	27
2.1.2.1. Sanal Kaynaklar Ve Hedefler	27
2.1.2.2. Kapasite Kısıtlı Taşın Problemi	32

	Sayfa
2.2. ULAŞTIRMA AĞLARINDA AKIMLAR	42
2.2.1. Statik Maksimal Akım	42
2.2.1.1. Genel	42
2.2.1.2. Ağlar	42
2.2.1.3. Ağlarda Akım	43
2.2.1.4. Kesimler	45
2.2.1.5. Maksimal Akım	46
2.2.1.6. Etiketleme Metodu	47
2.2.2. Dinamik Maksimal Akım	51
2.2.2.1. Genel	51
2.2.2.2. Problemin Formülasyonu	55
2.2.2.3. Dinamik Ağdan Statik Ağ Geçiş	56
2.2.3. Out-of-Kilter Algoritması	58
BÖLÜM III	61
3. BOŞ ÜNİTELERİN DAĞITIMI İÇİN ÖNERİLEN METOD	61
3.1. GENEL	61
3.2. ÜNİTELERİN HAREKETLERİNİN TANIMI	61
3.3. TARİFELİ KATARLARIN İŞLEMELERİ HALİNDE YENİ BİR DİYAGRAM ŞEKLİ	66
3.4. OKLAR ARASINDAKİ BAĞINTILAR	68
3.5. AĞIN BAŞLICA ÖZELLİKLERİ	69
3.6. AKIM KURALLARI	70
3.7. BOŞ ÜNİTELERİN DAĞITIMI İÇİN BİR ALGORİTMA	71
3.7.1. Genel	71
3.7.2. Ağın ve Problemin Tanımı	72
3.7.3. Problemin Formülasyonu	73
3.7.4. Algoritmanın Çözüm Şekli	75
3.7.5. Düğümlerin Sınıflandırılması ve Seçimi	75
3.7.6. Açıcıların Özellikleri	76
3.7.7. Algoritmanın Bir Açıcı Çözüm Şekli	77
3.7.8. Ağın Düğümlerinin Potansiyelleri	78
3.7.9. Potansiyellerin Getirdikleri Kısıtlama	78

VII

Sayfa

3.7.10. Bir Demiryolu Ağındaki Boş Ünitelerin
Dağıtımında Kullanılan Algoritmanın
Hesap Adımları

80

BÖLÜM IV

82

4. SAYISAL ÖRNEK VE SONUÇLAR

82

4.1. SAYISAL ÖRNEK

82

4.2. SONUÇLAR

94

NOTASYONLAR

96

BİBLİYOGRAFYA

97

BİYOGRAFİ

104

ÖZET

Bugün, toplumun karşılaştığı önemli problemlerin arasında, ulaştırma problemleri çok geniş yer tutmaktadır. Bu problemlerin çözümü ve gelecekteki ulaştırma ihtiyaçlarının pre-vizyonu için çeşitli matematik modeller yapılmış ve çok sayıda algoritma geliştirilmiştir. Bu matematik modellerden ağ-akım tipinden olanlar ise son yıllarda hızla gelişen yöney-lem araştırmasının önemli bir bölümünü oluşturmaktadır. Nite-kim, çeşitli mühendislik dallarındaki uygulamalara bakıldı-ğında ağları kullanan birçok matematik modellerin geliştiril-miş olduğu görülmektedir.

Burada yapılan çalışmada da ağ kavramından geniş ölçüde yararlanılmış ve önerilen silindirik biçimindeki yeni diyagram-da, problemin çözümüne ağ-akım cinsinden bir algoritmayla gidilmiştir.

Çalışmanın esası, demiryolu ağlarındaki boş yük ünite-lerine ait trafiğin dengelenmesidir. Diğer bir deyimle, de-miryolu ağındaki boş yük ünitelerini optimal bir şekilde da-ğıtımıştır. Zaten çalışmada boş yük ünitesi yerine boş ünite veya yalnızca ünite denmekte ve yine boş yük üniteleri tra-fiğinin dengelenmesine de, boş ünitelerin dağıtımını ifadesi kullanılmaktadır.

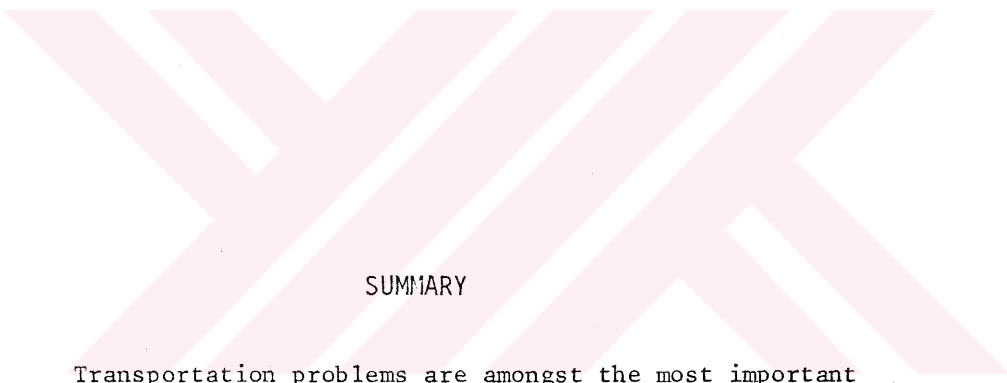
Birinci bölümde boş yük ünitelerinin neler olabileceği, nasıl hareket edecekleri ve kaç tip olacakları kısaca açık-lanmış ve amacın, bunların optimum dağıtımının yapılması için öncelik metodu yerine daha sıhhatli ve kesin neticeler veren bir algoritmanın bulunması olduğu belirtilmiştir.

İkinci bölümün birinci kısmında, Lineer Programlamanın Taşın Problemi üzerinde durulmuştur. Bir demiryolu ağındaki boş ünitelerin dağıtımını problemi taşın problemi aracılığı ile çözülmüş ve bu problemin özel halleri örnekler verilerek incelenmiştir. Burada incelenen özel haller tamamen demiryo-lu sistemlerin getirdiği kısıtlamalardan oluşmaktadır ve bu çalışmada bunların ayrı ayrı çözümleri yapılmıştır.

İkinci bölümün ikinci kısmında ise ağlar ve ağ-akım bağıntıları etraflıca incelenmiş ve bunlara ait teorem ve algoritmalar açıklanmıştır. Bu kısımda statik maksimal akım, dinamik maksimal akım ve Out-of-Kilter algoritmasının özelliklerinden bahsedilmiştir. Ayrıca Out-of-Kilter algoritmasının da kullandığı, dinamik ağdan statik ağa geçiş de anlatılmıştır.

Üçüncü bölüm, boş üniteler trafiğinin dengelenmesinde kullanılan dağıtım algoritmasının ve bu algoritmanın kullanıldığı ağın tanıtılmasından oluşmuştur. Burada, bir demiryolu ağındaki ünitelerin hareketleri silindirik biçimindeki yeni bir diyagram üzerinde gösterilmiş ve bu yeni diyagrama göre problemin formülasyonu yapılmıştır. Yine bu bölümde, yeni diyagrama göre ünitelerin hareketlerinin, dağıtım ağının tarifi yapılmış, akım kuralları konmuş ve daha sonra formülasyonu yapılan problemi çözecek algoritma etraflıca açıklanmıştır.

Dördüncü bölüm iki kısımdan oluşmuştur. Birinci kısımda 4 istasyon arasında işleyen 8 katarın belli bir süre içindeki hareketleri ele alınmış ve 24 düğümlü bir ağda boş ünitelerin dağıtımı üçüncü bölümdeki algoritma yardımı ile yapılmıştır. İkinci kısım ise çalışmanın sonuçlarını kapsamaktadır.



SUMMARY

Transportation problems are amongst the most important problems faced by today's society. Although the railway transport is one of the oldest of the modern modes of transport, there are still many problems to be solved. Among these, there is the balancing of the supply demand traffic flow of empty freight units in a railway network. This is commonly known as the distribution of the empties in a railway network.

In this study this problem is taken and it has been tried to make an engineering approach to its solution.

Instead of freight wagons, the term of railway freight units is used throughout the work. The reason of using unit instead of wagon is that today on the railways there is not only the wagon but the container as well. Therefore the problem is to optimize the movements of different railway units.

The use of network models has been increasingly widespread in the recent development of operational research. Because of the basic simplicity and generality of network concepts and because of the amenability of network calculations to digital computation, networks have proved ideally suited for mathematical modeling in a variety of scientific and engineering applications. In this study different kinds of solutions of the problem are explained and a network flow type algorithm is chosen to be the best method for its solution.

The study consists of four parts.

The first part is an introduction. Before explaining the method of solution, it is thought that it would be rather useful to clarify the meaning of the problem by explaining it very briefly and giving the earlier and the actual distribution methods used by the railway dispatchers.

One of the important problems of the railways is the great cost of the unit fleet and its low level of productivity. In order to improve the productivity of the unit fleet a central unit authority should be established and the duty of this authority should be the organization of the distribution of empty units. In the earlier days some degree of central control existed. Each district officer arranged distribution within his own area and then reported the overall position to regional headquarters where adjustments between districts were arranged.

In some countries the central unit authority exists and there is a direct liaison between the authority and the districts for the submission of the daily rolling stock reports and the issue of distribution orders. This liaison helps very much the dispatchers to make the optimum distribution of the empty freight units.

The aims of the distribution of the freight units are:

- To so arrange empty unit supply that when and wherever acceptable traffic is offered, the right number of units to move that traffic are then and there available
- In arranging that availability to utilise the least number of resources

XIII

-To measure and demonstrate the efficiency of the performance of these two tasks.

By the proper distribution of the empties the number of the units is decreased and the reliability of the service provided to the customer is increased.

At the end of the introduction section it is mentioned that the aim of the study is to find a solution to this optimisation problem with the aid of linear programming techniques.

The second part of the study consists of two sections.

In the first section the classical transportation problem of the linear programming is examined in details.

The reason of examining the transportation problem is that it is used by many scientists as a solution to the problem of the distribution of the empty freight units in a railway network.

The transportation problem is a special case of linear programming. The methods for its solution are simpler than the general methods designed to solve any linear programming problem. However, many practical problems are of transportation type, and many others can be transformed into transportation problems.

The general method for solving the linear programming problems is Dantzig's Simplex Method, but for the solution of the transportation problem the Stepping-stone method is used.

This section consists of two parts.

In the first part a railway network with nine stations is taken and it is assumed that certain units have to be transported from four stations to five different stations. The first four stations are called sources and the last five are called destinations. In this example the supply of units is accepted to be equal to the demand.

The formulation of the problem is made and the method of shadow costs for the improvement of the solution is explained.

XIV

In the second part special cases of transportation problem is studied. Some restrictions of the railway systems are included in the problem and two special cases are obtained.

The first case is obtained by assuming that the total demand is not equal to the total supply and an imaginary destination is invented because the demand was less than the supply, if the demand were more than the supply an imaginary source would have been invented.

The second case is obtained by eliminating a link between two stations of the system. More than one link can be eliminated or a capacity restriction of the train running between two stations can be introduced as well. In this case a very large cost element is placed instead of the element of the eliminated link.

In the second part of the study the flows in transportation networks are studied. This part consists of three sections.

In the first section the static maximal flow is examined.

Before the direct examination of the static maximal flow, the networks and the flows in networks are explained and then the theorems concerning the maximal flow are stated. The most important theorem is the max-flow min-cut theorem of Ford and Fulkerson. At the end of the section a network with four nodes is taken and the famous labelling method is explained. The labelling method is known as Ford and Fulkerson algorithm as well. The labelling method gives the static maximal flow in a network.

In the second section the dynamic maximal flow is examined.

In this section the same network with four nodes is taken as above and the dynamic flow is illustrated on it. Then the formulation of the dynamic flow is done and the conditions of the maximal dynamic flow are stated. The last part of the section is the construction of a static network from a dynamic network.

In the third section the famous Out-of-Kilter algorithm of Ford and Fulkerson is briefly examined. The theorem itself is stated, and a comparison between this algorithm and the simplex method is briefly done as well.

The third part of the study consists of the method chosen for the solution of the distribution of the empties in a railway network.

As it was mentioned before, the method of solution explained in this section is a network flow type algorithm.

The movements of the railway freight units are described as arrows of the network which will be solved by the algorithm. It is worth to mention that there are two kinds of arrows, those are temporal arrows and physical arrows. Physical arrows represent the movement of units on a train between two different stations. Temporal arrows represent the waiting period of units at the same station between the arrival and/or the departure of a train. The nodes of the network represent events. Here, an event is a departure or an arrival of a train running on the railway system.

The relations between arrows of the network and the particularities of the network are described in detail, and the flow rules are given.

After the description of nodes, arrows and events the movements of the units are drawn on a cylinder and a new cylindrical diagram is obtained. On this new cylindrical diagram the beginning and the end of a distribution period are adjacent. Thus, if the distribution is started in the middle of one period it will be finished without any cut in the middle of the next period.

The algorithm used here is a special form of the Out-of-Kilter algorithm, therefore it is a linear programming algorithm. It can be seen from the formulation of the problem as well, because the problem itself is a linear programming problem.

The algorithm uses an inductive method of solution. Instead of examining the whole network, only a part of it is solved at any one time. Once a subnetwork is solved, another subnetwork is taken.

At any stage of the algorithm the nodes can be marked nodes or unmarked nodes. All of the marked nodes are examined by the algorithm, the rest of the nodes are unmarked or markable nodes. Some of the arrows are distinguished.

by the algorithm and a flow can be sent only over distinguished arrows of the network.

With each node of the network a dual variable or potential is associated. The potential can also be called the node price and it represents the marginal cost of getting an additional car to the node which is examined at present, from the source node.

At the end of the third part the computation of the algorithm is given. It consists of initial step, step 1, step 2 and step 3.

The fourth part of the study consists of two sections.

The first section is a numerical example. The example taken to illustrate the algorithm is a twenty-four node problem. These twenty four nodes are formed by the movements of eight trains between four stations. Every stage of the algorithm is shown on the figures.

The second section contains the results of the study.

B Ö L Ö M I

1. GİRİŞ VE AMAÇ

1.1. GENEL

Demiryolu işletmecilerinin uzun yıllardır çözümüne yaklaşmak için büyük çabalar harcadıkları önemli problemlerden birisi "*Bir demiryolu ağındaki boş vagonların dağıtımı*"dır. Halen bir çok ülkenin demiryolu işletmecileri bu problem ile yakından ilgilenmektedirler.

Son yıllarda ulaştırmacılığa yeni bir boyut kazandıran konteynırlar demiryollarında da kullanılmaya başlanmıştır. Ancak, konteynırlar kendileriyle birlikte bazı problemleri de getirmişlerdir. Bazı memleketlerde mevcut demiryolu ağı üzerinde, demiryolu idaresinden tamamen bağımsız konteynır ulaştırma sistemleri kurulmuştur. Böyle bir sistemin en belli başlı problemlerinden birisi, yukarıda vagonlar için söylenenin bir eşleniği olan, boş konteynırların dağıtımıdır.

Vagon ve konteynırlardan oluşan katarlardan başka, bazı endüstri kesimlerinin kendilerine ait ürünlerin kolayca taşınabilmesi için özel imal ettirmiş oldukları ünitelerden oluşmuş katarlar da vardır. Bunlara örnek olarak, petrol rafinerileri ve depoları arasında işleyen katarların elemanları olan petrol taşıma vagonları ve çeşitli demir çelik fabrikaları arasında işleyen katarların vagonları verilebilmektedir. Bu taşıma ünitelerinin de kendi sistemleri içinde bir dağıtılma problemi vardır.

1.2. DEMİRYOLU ÜNİTESİNİN TARİFİ

Buraya kadar bir demiryolu ağında ne gibi ulaştırma sistemlerinin var olabileceği açıklanmıştır. Bu sistemlerin elemanlarını vagon, konteynır olarak ayırmaktansa, genelleştirmeye gidilip bunlara demiryolu ünitesi veya yalnız ünite adı verilmesi uygun görülmektedir. Böylece boş vagon veya boş konteynır dağıtımı yerine "*boş ünitelerin dağıtımı problemi*" incelenecektir.

1.3. DAĞITILACAK ÜNİTELERİN TİPİ

Çalışmada tek tip Ünite dağıtımı ele alınmaktadır. Bunun en büyük sebeplerinden birisi ünitenin konteynır olması halinde ve herhangi bir konteynır sistemi incelendiğinde, işletmecilerin tek tip konteynıra doğru bir eğilimlerinin olduğu kolaylıkla görülmektedir. Tek tip konteynır işletmeciliği çok ekonomik olmaktadır, çünkü değişik yükleme-boşaltma makinalarına veya değişik bakım elemanlarına ihtiyaç duyulmamaktadır.

1.4. ÜNİTELERİN YAPTIKLARI HAREKETLER

Bir demiryolu ağındaki konteynırlar Ünite olarak ele alındığında, bunların yaptıkları hareketler şöyle olmaktadır: bir konteynır bir müşteri tarafından doldurulmakta veya boşaltılmaktadır. Sonra bu konteynır bir çekiciye yüklenmekte ve demiryolu konteynır terminaline yollanmaktadır. Bu terminale gelen konteynır, ya orada hazır bekleyen katarı yüklenmekte veya daha sonra yollanmak üzere depo sahasında tutulmaktadır. Birinci halde, katarın varış istasyonunda, konteynır ya hemen müşteriye yollanmakta ya da daha sonra yollanmak üzere depo sahasına konmaktadır. Müşteriden depo sahasına ve oradan katarı kadar kat edilen yol, kara yolu ve yükleme araçları aracılığı ile olmaktadır. Aynı durum depo sahasından veya katarıdan müşteriye gidiş için de geçerlidir. Burada incelenecek problem bir demiryolu problemi olduğuna göre müşteri ile katar arasında katedilen yollar gözönüne alınmayacaktır. Yani dağıtım yapılacak ünitelerin kesinlikle demiryolu üzerinde hareket ettikleri kabul edilmektedir. Bu konuya çözüm metodunda da değinilecek ve hareketler grafiklerinde açıklanacaktır.

1.5. BOŞ ÜNİTELERİN DAĞITIMI İÇİN HALEN KULLANILMAKTA OLAN METOD

Demiryollarında boş ünitelerin dağıtımını dispeçerler (dağıtıcı elemanlar) tarafından yapılmaktadır. Müşterilerin ihtiyacını karşılamak için işletmeciler talep ettikleri üniteleri dispeçerlere bildirmekte ve bunlar da talep edilen miktar veya bu miktara yakın üniteyi ihtiyaç olan yerlere tahsis etmektedirler. Dispeçerlerin gelecekteki ihtiyaçları da gözönüne alarak yaptıkları bu dağıtım, olağan üstü bir durum olmaması halinde gayet iyi bir şekilde işlemektedir.

Dispeçerler, ünitelerin kullanılma oranlarını esas alarak onları tahsis etmektedirler. Bu kullanılma oranları bazı kısıtlamaları ortaya çıkarmaktadır. Bu kısıtlamalar, bazı istasyonlara belli sayıdaki ünitenin üstünde ünite yollanmamasını gerektirdiği gibi, diğer istasyonlara da belli sayıda ünitenin altında ünite yollanmamasını gerektirmekte ve sonuncu tip istasyonlara bir öncelik tanınmaktadır. Bu tanınan öncelikten dolayı bazı işletmelerde bu şekilde yapılan dağıtım metoduna *Öncelik Metodu* adı verilmektedir.

Yukarıda bahsedilen kullanılma oranlarında dalgalanma olması halinde bu dağıtım sistemi hemen hemen hiç işlememektedir. Maalesef kullanılma oranlarındaki dalgalanmalar da çok sık olarak ortaya çıkmaktadır. Her zaman için ünite sağlayan arz sahalarından birisinin ünite sağlayamaması halinde sistemde bir bozukluk olmaktadır. Şayet bu bozukluk bir çok ihtiyaç sahibi için sık sık ortaya çıkarsa, bu dağıtım metodu güvenilir olmaktan tamamiyle uzaklaşmaktadır, çünkü olaylar çok karışık bir hale gelmektedir. İstedikleri ünite sayısında sık sık eksiklikler meydana gelmesi halinde müşteriler hiç memnun olmamaktadırlar. Ayrıca fazla ünite yollama veya gereksiz ünite hareketlerinden dolayı ünitelerin kullanılma prodüktivitesi çok düşmektedir. İstasyon işletmecileri, devamlı olarak ihtiyaçlara cevap verebilmek için, her sefer gereğinden fazla boş ünite talep etmekte veya istasyonlarında, bir emniyet faktörü olarak, gereğinden fazla boş ünite bulundurmaktadırlar. Böyle bir tutum müşterilerin hoşnutsuzluğunu gidermekte fakat ünitelerin kullanılma oranlamasını azaltmaktadır. Bu durumda demiryolu sisteminde arz edilen ünite miktarı küçülmekte ve böylece yeni ünite satın alma veya kiralama ihtiyacı ortaya çıkmaktadır. Buradan

anlařılacağı gibi, daha elverişli dağıtım metodlarına ihtiyaç duyulduđu aşıkarak ortaya çıkmaktadır.

1.6. ÇALIřMANIN AMACI

Bir demiryolu ağındaki boş ünitelerin ihtiyaç fazlası yerlerden, ihtiyaç olan yerlere tahsis edildiklerinde yapacakları hareketlerin maliyetlerinin minimum olması her işletmecisi tarafından istenen bir olaydır. Bu durumda boş ünitelerin dağıtım problemi bir *optimizasyon problemi* olmakta ve çalışmanın amacı da bu problemin çözümünü bulmaktır.

Optimizasyon problemlerinin çözümü için kullanılan değişik matematik teknikler vardır. Boş ünite dağıtımının optimizasyonu için yararlanılan teknikler *lineer programlama* ve *Graflar teorisidir*. Nitekim lineer programlamaya özel bir hali olan taşıın problemi Bölüm II de boş ünite dağıtımına uygulanmakta ve yine aynı bölümde, ağlardaki akımlar için olan algoritmalara değinilmektedir. Burada en uygun algoritma aranmakta ve daha sonra Ford ve Fulkerson'un bulduđu ve White'in geliřtirdiđi Out-of-kilter algoritmasının bir adaptasyonunun yapılmasıyla boş ünitelerin optimum dağıtılması metodu ortaya konmaktadır.

B Ö L Ü M II

2. BOŞ ÜNİTELERİN DAĞITIMINDA KULLANILAN MATEMATİK
TEKNİĞİ

2.1. LINEER PROGRAMLAMA

Herhangi bir sistemdeki ünitelerin dağıtımı için kullanılan yaklaşım metodlarının çoğunun kökeni lineer programlama dayanmaktadır. Burada yapılan çalışmanın konusu olan problemin çözümüne, yardımcı olabileceğine inanılan programlama metodları, örnekler üzerinde incelemeler yapılarak izah edilecektir. Ancak bu izahat yapılırken gerekli karşılaştırmalara ve eleştirilere de yer verilecektir.

2.1.1. Taşın Problemi

2.1.1.1. Genel

Taşın problemi lineer programlamanın özel bir halidir. Bu problemin çözümünde kullanılan metodlar, herhangi bir lineer programlama probleminin çözümünde kullanılan genel metodlardan daha basittir. Bir çok problemler taşın problemi tipindedir ve diğer bir çokları da taşın problemi tipine dönüştürülebilmektedir.

Lineer programlama problemleri için genel çözüm metodu Dantzig'in Simpleks metodudur. Ancak taşın probleminin çözümünde *Stepping-stone* adı verilen bir çözüm metodu kullanılmaktadır.

Problemin formülasyonuna geçmeden önce, burada incelenecek olan problemin bir demiryolu taşıma problemi olduğunu, yani ulaştırma ağının rijid olduğunu belirtmek gerekmektedir. Ancak formülasyondan sonra bu rijid ağın ortaya çıkartabileceği kısıtlanmalı hallerinde incelenmesi gerekmektedir.

2.1.1.2. Problemin Formülasyonu

Bu problemi en iyi şekilde inceleme yolu bir örneğin üzerinde çalışmalar yapmaktır, böylece problemin formülasyonu da yapılmış olmaktadır.

Bir demiryolu ağında dokuz ($m + n$) istasyon vardır ve bunların arasında konteynır trafiğı yapılmaktadır. Bu istasyonların dördünde ($m = 4$) konteynır fazlası yani arz (a_i , $i = 1, 2, \dots, m$), beşinde de konteynır isteğı yani talep (b_j , $j = 1, 2, \dots, n$) vardır. Ancak burada arzın talebe eşit olduğu

$$\left(\sum_{i=1}^m a_i = \sum_{j=1}^n b_j = 200, \quad a_i > 0, \quad b_j > 0 \right)$$

kabul edilmektedir. (m) istasyonlarına kaynak, (n) istasyonlarına ise hedef adı verilmektedir. Herhangi bir i kaynağı ile herhangi bir j hedefi arasındaki birim taşın maliyeti

(c_{ij} $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$) dir ve daima $c_{ij} > 0$ dir.

c_{ij} ler arz ve taleplerle birlikte tablo (2.1) de verilmektedir. i kaynağından j hedefine taşınacak konteynır sayısı

(x_{ij} , $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$) dir ve problemin çözümü x_{ij} lerin oluşturduğu taşın şemasının bulunmasıdır.

Şimdiye kadar verilenler aşağıdaki dört denklem ile ifade edilebilmektedir.

$$\min C = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad x_{ij} > 0 \quad (2.1)$$

$$\sum_{j=1}^n x_{ij} = a_i \quad i = 1, 2, \dots, m \quad (2.2)$$

$$\sum_{i=1}^m x_{ij} = b_j \quad j = 1, 2, \dots, n \quad (2.3)$$

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad (2.4)$$

(2.1) denkleminde de anlaşılabacağı gibi x_{ij} lerin toplam taşın maliyetini (C) minimum yapacak şekilde saptanması gerekmektedir.

TABLO (2.1)

HEDEFLER KAYNAKLAR	1	2	3	4	5	ARZ
1	6	2	8	7	5	40
2	4	3	7	5	9	70
3	2	1	3	6	4	60
4	5	6	4	8	3	30
TALEP	30	60	50	40	20	200

Tablo (2.1) deki son sütun 4 kaynaktan yollanacak konteynır sayısını, son satır ise 5 hedeften istenen konteynır sayısını göstermektedir.

Çözümüne şöyle başlanmaktadır, tablo (2.1) den son satır ve son sütun kaldırılırsa şu elde edilir.

6	2	8	7	5
4	3	7	5	9
2	1	3	6	4
5	6	4	8	3

(2.5)

Bu bir konteynır taşın maliyetlerini gösteren maliyet matrisidir. Birinci satır, birinci kaynaktan değişik hedeflere bir konteynır yollama maliyetlerini 6, 2, 8, 7 ve 5 para birimleri şeklinde göstermektedir.

Kaynaklardan hedeflere yollanacak konteynır sayıları ise aşağıdaki gibidir:

$$\begin{array}{ccccc}
 x_{11} & x_{12} & x_{13} & x_{14} & x_{15} \\
 x_{21} & x_{22} & x_{23} & x_{24} & x_{25} \\
 x_{31} & x_{32} & x_{33} & x_{34} & x_{35} \\
 x_{41} & x_{42} & x_{43} & x_{44} & x_{45}
 \end{array} \quad (2.6)$$

Burada x_{45} sembolü 4 kaynağından 5 hedefine yollanacak konteynır sayısını göstermektedir. Sayet bu yerler arasında hiç bir konteynır yollanmayacak ise $x_{45} = 0$ olur, aksi halde x_{45} daima pozitiftir.

(2.5) ve (2.6) deki elemanların çarpımlarının toplamlarından C elde edilir:

$$\begin{aligned}
 C = & 6x_{11} + 2x_{12} + 8x_{13} + 7x_{14} + 5x_{15} + 4x_{21} + 3x_{22} \\
 & + 7x_{23} + 5x_{24} + 9x_{25} + 2x_{31} + 1x_{32} + 3x_{33} + 6x_{34} \\
 & + 4x_{35} + 5x_{41} + 6x_{42} + 4x_{43} + 8x_{44} + 3x_{45} \quad (2.7)
 \end{aligned}$$

Burada bulunan x_{ij} sembolleri aşağıdaki şartları sağlayan ve daima pozitif (veya sıfır) olan değerler taşıyabilirler:

$$\begin{aligned}
 x_{11} + x_{12} + x_{13} + x_{14} + x_{15} &= 40 \\
 x_{21} + x_{22} + x_{23} + x_{24} + x_{25} &= 70 \\
 x_{31} + x_{32} + x_{33} + x_{34} + x_{35} &= 60 \\
 x_{41} + x_{42} + x_{43} + x_{44} + x_{45} &= 30 \\
 x_{11} + x_{21} + x_{31} + x_{41} &= 30
 \end{aligned}$$

$$x_{12} + x_{22} + x_{32} + x_{42} = 60$$

$$x_{13} + x_{23} + x_{33} + x_{43} = 50$$

$$x_{14} + x_{24} + x_{34} + x_{44} = 40$$

$$x_{15} + x_{25} + x_{35} + x_{45} = 20 \quad (2.8)$$

Bu şartlar 20 bilinmeyenli 9 denklemlili homojen olmayan bir sistem meydana getirirler. İlk dört denklem kaynaklardan kullanılacak konteynır sayısının arza, geri kalan denklemler ise hedeflerin alacağı konteynır sayısının talebe eşit olduğunu göstermektedir.

(2.8) denklemler sisteminin negatif olmayanı çözümleri mümkün olan bütün tarifeleri meydana getirir, bunların arasından en az tasın maliyetli olanı seçilmelidir. Diğer bir deyimle (2.7) fonksiyonunun minimumunun saptanması gerekmektedir.

Bu da hedef fonksiyonu (2.7) fonksiyonu olan bir lineer programlama probleminin çözümünü bulmak demektir.

Burada karşılaşılan problem, simpleks metodunun özel bir hali olan *stepping-stone* metodunun vardımı ile çözülebilmektedir.

Bu metod burada bulunan örnek üzerinde bir çalışma yapılarak açıklanmaktadır.

2.1.1.3. Maliyet Matrisinin Transformasyonu

Esas programlamaya başlamadan önce uygun bir şekilde maliyet matrisinin değişmesi gerekmektedir. Maliyetler arasındaki farkların, optimal çözümün bulunmasındaki oynadıkları karar verdirici rol bu transformasyonun esasını oluşturmaktadır.

Bu transformasyonun açıklanması için yalnız iki kaynak ve iki hedefin bulunduğu bir örneğin incelenmesi yararlı ve yeterlidir. Bu örneğin her kaynağında bir konteynır vardır ve her hedefinde de yalnız bir konteynır talep edilmektedir.

Maliyet matrisinin

$$\begin{array}{cc} 5 & 3 \\ 4 & 6 \end{array}$$

olduğu kabul edilmektedir. Burada mümkün olan çözümlerin sayısı ikidir. İlk çözümde elde bulunan konteynırlar birinci kaynaktan birinci hedefe, ve ikinci kaynaktan ikinci hedefe yollanmaktadır. İkinci çözümde ise bir konteynır birinci kaynaktan ikinci hedefe, diğer bir konteynır da ikinci kaynaktan birinci hedefe yollanmaktadır. Birinci ve ikinci çözümlerin maliyetleri C_1 ve C_2 olduğundan, verilen bilgilere göre bunların değerleri:

$$C_1 = 5 + 6 = 11$$

$$C_2 = 3 + 4 = 7$$

olmaktadır.

İkinci çözüm birinciye göre

$$C_1 - C_2 = 11 - 7 = 4$$

parasal birim kadar daha iyidir. Maliyet matrisinin birinci sütunundan 3 çıkartıldığında, matris

$$\begin{array}{cc} 2 & 3 \\ 1 & 6 \end{array}$$

sekline dönüşmekte ve

$$C_1 = 2 + 6 = 8$$

$$C_2 = 1 + 3 = 4$$

olmaktadırlar. Bu durumda da ikinci çözüm birinciye göre yine

$$C_1 - C_2 = 8 - 4 = 4$$

parasal birim kadar daha iyidir.

Bu küçük örnek şu teoremi ortaya koymaktadır:

Maliyet matrisinin herhanqi bir satır veya sütundaki elemanlar aynı miktarda arttırılır veya eksiltirilirse optimal bakımından taşın probleminde herhanqi bir değişiklik olmaz.

Yeni maliyet matrisi esas alınarak hesaplanan toplam maliyetler ise doğal olarak aynı kalmaktadır. Bunlar da değiştirilen satır ve sütunun değerine bağlı olarak değişirler.

Bu teoreme dayanarak her satır veya sütundaki en küçük eleman sıfır olacak şekilde maliyet matrisi transforme edilebilmektedir. Transformasyon iki etapta gerçekleştirilmektedir.

a-Her sütundaki en küçük eleman seçilmekte ve bu kendi sütunundaki diğer elemanlardan çıkartılmaktadır.

Burada verilen örnekte sütunların elemanları sırasıyla 2, 1, 3, 5 ve 3 kadar eksiltilmektedir. Böylece ilk matris şu şekli alır:

$$\begin{pmatrix} 4 & 1 & 5 & 2 & 2 \\ 2 & 2 & 4 & 0 & 6 \\ 0 & 0 & 0 & 1 & 1 \\ 3 & 5 & 1 & 3 & 0 \end{pmatrix} \quad (2.9)$$

b-Yeni matrisin her satırındaki en küçük eleman seçilmekte ve bu kendi satırındaki diğer elemanlardan çıkartılmaktadır.

Böylece her satır sırasıyla 1, 0, 0 ve 0 kadar eksilmekte ve aşağıdaki matris elde edilmektedir.

$$\begin{pmatrix} 3 & 0 & 4 & 1 & 1 \\ 2 & 2 & 4 & 0 & 6 \\ 0 & 0 & 0 & 1 & 1 \\ 3 & 5 & 1 & 3 & 0 \end{pmatrix} \quad (2.10)$$

Bu matrisde her sütun ve satırdaki en küçük eleman sıfırdır.

Bu çıkartma işlemine sütun yerine satırdan da başlanabilmekte ve netice yine aynı olmaktadır.

Neticede problemi aşağıdaki tabloyla göstermek mümkündür:

TABLO (2.2)

3	0	4	1	1	40
2	2	4	0	6	70
0	0	0	1	1	60
3	5	1	3	0	30
30	60	50	40	20	200

Bu tabloda kaynakların ve hedeflerin numaraları yazılmamıştır.

Yollanacak miktarlar düşünüldüğünde tablo (2.2) nin tablo (2.1) ile aynı optimal çözüme sahip olduğu görülmektedir. Bu husus taşın maliyetleri için geçerli değildir, fakat maliyetlerindeki fark, a ve b etaplarında çıkartılan miktarlar yollanacak miktarlar ile çarpılıp toplanmak suretiyle elde edilmektedir. Örnekte bu toplamı:

$$[(2 \times 30) + (1 \times 60) + (3 \times 50) + (5 \times 40) + (3 \times 20)] + [(1 \times 40) + (0 \times 70) + (0 \times 60) + (0 \times 30)] = 570$$

olmaktadır.

Bundan sonra yapılacak hesaplarda tablo (2.2) esas alınmaktadır. Bu esaslar neticede optimal çözümü vermektedir. Ancak toplam maliyete biraz önce hesaplanan miktar da ilave edilmelidir.

2.1.1.4. İlk Çözüm

Stepping-stone metodunun esas fikri hemen mümkün bir çözüm bulmaktır. Daha sonra bu çözüm, optimal çözüm bulunana kadar adım adım islah edilmektedir.

Toplam maliyetleri küçük olan ilk çözümün aranmasına şu şekilde başlanmaktadır: Maliyet matrisinin en küçük elemanı

seçilmekte ve buna mümkün olan en büyük miktar tahsis edilmektedir, yani minimal maliyet elemanına, maksimal madde miktarı tahsis edilmektedir. Burada madde olarak konteynır alınmaktadır. Genellikle minimal eleman bir tek olarak tesbit edilememektedir.

İncelenen örnekte minimal eleman sıfırdır. Her satır ve sütunda da en azından bir sıfır eleman vardır. Bu durumda mümkün olduğu takdirde kendi satır ve sütununda yalnız kendisinin sıfır olduğu bir sıfır elemanının seçilmesi uygundur. Böyle bir sıfır elemanı yoksa kendi satır ve sütununda en az sıfır bulunan bir sıfır elemanının seçilmesi gerekmektedir.

Böylelikle tablo (2.2) de birinci satırın ikinci elemanı, yani $c_{12} = 0$ seçilmektedir. Bu elemana ait olan x_{12} değişkeni kaynak 1 den hedef 2 ye kaç adet konteynır yollanacağını ifade etmektedir. Kaynak 1 de 40 konteynır bulunduğuna ve hedef 2 de ise 60 konteynır talep edildiğine göre, tahsis edilebilecek konteynır sayısının 40 olduğu aşıkardır, yani $x_{12} = 40$ dır. Bu durum tablo (2.3) de şu şekilde gösterilmektedir.

TABLO (2.3)

3	0 ⁴⁰	4	1	1	40
2	2	4	0	6	70
0	0	0	1	1	60
3	5	1	3	0	30
30	60	50	40	20	200

Bundan sonraki adımda tahsis edilen miktar problemden çıkartılmakta ve işlemlere indirgenmiş tablo (2.4) ile devam edilmektedir.

TABLO (2.4)

.	.	.	.	.	.
2	2	4	0	6	70
0	0	0	1	1	60
3	5	1	3	0	30
30	20	50	40	20	160

Burada kaynak 1 tamamen çıkartılmış ve hedef 2 deki talep de 20 konteynır'a indirgenmiştir.

Yeni tablodan c_{24} maliveti seçilmektedir. Burada arz edilen konteynır sayısı talepten daha büyüktür, dolayısıyla talep edilen miktar kadar konteynır tahsis edilebilmektedir.

Böylece $x_{24} = 40$ olmakta ve tablo (2.5) elde edilmektedir.

TABLO (2.5)

.	.	.	.	.	.
2	2	4	.	6	30
0	0	0	.	1	60
3	5	1	.	0	30
30	20	50	.	20	120

Bundan sonra sırasıyla c_{45} , c_{31} , c_{32} , c_{33} , c_{34} , c_{23} minimum malivetleri seçildiğinde ve bunlara ait işlemler yapıldığında tablo (2.6) ya varılmaktadır.

TABLO (2.6)

3	0 ⁴⁰	4	1	1	40
2	2	4 ³⁰	0 ⁴⁰	6	70
0 ³⁰	0 ²⁰	0 ¹⁰	1	1	60
3	5	1 ¹⁰	3	0 ²⁰	30
30	60	50	40	20	200

Bu tablo mümkün bir çözümü içermektedir. Bu çözümde değişkenler aşağıdaki pozitif değerleri almaktadır:

$$x_{12} = 40$$

$$x_{32} = 20$$

$$x_{23} = 30$$

$$x_{33} = 10$$

$$x_{24} = 40$$

$$x_{43} = 10$$

$$x_{31} = 30$$

$$x_{45} = 20$$

Diğer değişkenler sıfır değerini almaktadır.

x_{ij} pozitif değişkenine sahip bir c_{ij} maliyetine işgal edilmiş eleman, diğer elemanlara ise serbest elemanlar denmektedir.

Tablo (2.6) daki çözümde 8 işgal edilmiş ve 12 serbest eleman bulunmaktadır. İşgal edilmiş elemanların sayısı kaynak ve hedeflerin toplam sayısından bir eksiktir,

$$4 + 5 - 1 = 8$$

Bu sayıya problemin kritik sayısı adı verilmektedir. Buradaki örnekte yukarıda hesaplandığı gibi kritik sayı 8 dir. Bu çözüm metodunda işgal edilmiş elemanların sayısının daima kritik sayıya eşit olduğu kabul edilmektedir.

Bazen bir eleman için arz, talebe eşit olmakta ve bu durumda ortaya bir güçlük çıkmaktadır. Aşağıdaki problem ele alındığında

3	1	0	2	30
0	2	1	4	20
2	0	5	0	50
10	25	30	35	100

ve ilk maliyet olarak $c_{13} = 0$ seçildiğinde $x_{13} = 30$ olmaktadır. Tablodan hem birinci satır hem de üçüncü sütun beraberce çıkartılamamaktadır, çünkü bu işlem yapıldığı takdirde kritik sayı şartı bozulmaktadır. Bu durumda uygun olmayan satır veya sütun çıkartılmaktadır. Örnekte birinci sıra çıkartıldığında tablo aşağıdaki şekli almaktadır:

.	.	.	.	.
0	2	1	4	20
2	0	5	0	50
10	25	0	35	100

Böylece hedef 3'ün talebi sıfıra eşit olmaktadır. Bu gerçek bir talep değildir, fakat pozitif bir talep gibi düşünülüp işlemlere devam edilmektedir. Neticede aşağıdaki tabloya ulaşılmaktadır:

3	1	$\boxed{0}^{30}$	2	30
$\boxed{0}^{10}$	$\boxed{2}^{10}$	$\boxed{1}^0$	4	20
2	$\boxed{0}^{15}$	5	$\boxed{0}^{35}$	50
10	25	30	35	100

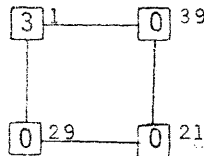
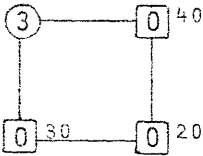
Burada yine kritik sayı şartını sağlayan bir çözüm elde edilmiştir. Fakat c_{23} işgal edilmiş elemana ait olan $x_{23} = 0$ değişkeni pozitif değildir, böyle bir çözüme *dejenere olmuş çözüm* adı verilmektedir. Herhangi bir çözümde x_{ij} pozitif değişkenlerinin sayısı kritik sayıdan az ise buna dejenere olmuş çözüm denmektedir.

Yukarıdaki şartı sağlayan bir ilk çözümü oluşturmak için yapılması gerekli işlemler tarif edildikten sonra, adı geçen çözümün islahatında yapılması gerekmektedir.

2.1.1.5. Çözümün İslahı

Bütün işgal edilmiş elemanların sıfır olması halinde, optimal bir çözüm elde edilmiş demektir, çünkü bu durumda taşın maliyetlerini daha da azaltmak için herhangi bir olanak kalmamıştır. Genellikle bu durum ortaya çıkmamaktadır, dolayısıyla çözümün islah edilmesi gerekmektedir.

İslah metodunu incelemek için tablo (2.6) ya dönmek ve çözüme nasıl bir serbest elemanın sokulabileceğini düşünmek gerekmektedir. Birinci satırın birinci elemanı ele alındığında ve kaynak 1'den hedef 1'e bir konteynır yollanmak istendiğinde, hedef 2'ye tahsis edilen miktar (40 konteynır) bir konteynır kadar azaltılmalıdır. Bu durumda da hedef 2'ye kullanılan miktar 39 konteynıra inmiş olmaktadır. Hedef 2'de eksik olan konteynır kaynak 3'den yollanarak bu durum düzeltilebilmektedir, ancak bu konteynır hedef 1'e gidecek olanlardan bir tanesi olmalıdır. Böylece birinci adımda hedef 1'e giden fazla konteynır gitmemiş olmakta ve devre tamamlanmaktadır. Yapılan işlemler, grafik olarak aşağıdaki devrelerle gösterilmektedir. İlk devre bir adet konteynır yollanmadan önceki, ikinci devre ise sonraki durumu göstermektedir.



Böylece birinci sıranın, birinci elemanına ait bir devre kurulmuş olmaktadır. Bu serbest eleman çözümün içine sokulmak istendiğinde, bu devre için olan orijinal çözümün değiştirilmesi gerekmektedir, fakat bu değişikliğin çözümü islah edip etmediğini araştırıp bulmak gereği de vardır. Bu değişikliğin toplam taşın maliyetini 3 parasal birim kadar arttırdığını grafikten kolayca görmek mümkündür. Buna $\Delta_{11} = 3$ denmektedir.

Burada Δ nın 11 endisi bu miktarın birinci satırın ilk maliyetine eşit olduğunu göstermektedir. Bu eleman çözümün içine sokulmamaktadır, çünkü $\Delta_{11} > 0$ dır, bu da maliyetlerin azalmadığını, aksine çoğaldığını göstermektedir. Δ_{11} miktarı devrenin maliyetlerini toplamak suretiyle bulunmaktadır. Bu maliyetler serbest elemandan başlamak üzere sırasıyla artı ve eksi işaretli olmakta ve sonunda daima pozitif bir netice vermektedir. Δ_{11} de şöyle hesaplanmıştır:

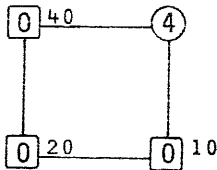
$$\Delta_{11} = 3 - 0 + 0 - 0 = 3$$

Yapılan işlemler şu şekilde özetlenebilmektedir:

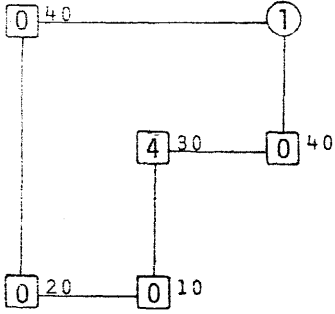
Herhangi bir serbest elemandan başlandığında, yatay ve düşey hareket ederek ve işgal edilmiş elemanlarda yön değiştirerek yine aynı elemana dönmek daima mümkün olmaktadır.

Bu şekilde elde edilen yola devre, yön değiştirilen elemanlara da devrenin köşeleri denmektedir. Bu köşelerden bir tanesi daima serbest eleman, diğerlerinin hepsi ise işgal edilmiş eleman olmaktadır. Köşelerin sayısı daima çift ve ikiden büyük olmaktadır. Yukardaki devrede köşelerin sayısı 4 dür. Serbest eleman ve ondan iki sonraki köşeler pozitif, diğer köşeler ise negatif değerlidir.

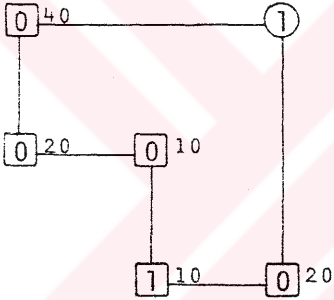
Burada incelenen örneğin birinci sırasının serbest elemanlarına ait devreler aşağıdadır:



$$\Delta_{13} = 4 - 0 + 0 - 0 = 4$$



$$\Delta_{14} = 1 - 0 + 0 - 0 + 4 - 0 = 5$$



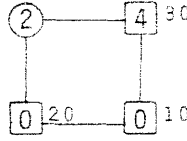
$$\Delta_{15} = 1 - 0 + 0 - 0 + 1 - 0 = 2$$

Hesaplar her bir serbest eleman için yapıldığında aşağıdaki Δ_{ij} ler tablosu elde edilmektedir.

TABLO (2.7)

3	☐	4	5	2
-2	-2	☐	☐	3
☐	☐	☐	5	2
2	4	☐	6	☐

Bu tabloda işgal edilmiş elemanlar karelerle gösterilmiştir. Çözümün iki yerde islah edileceğini görmek gayet kolaydır, çünkü ikinci sırada iki negatif miktar ortaya çıkmıştır. Bunlardan c_{22} ve ait devre aşağıda gösterilmektedir.

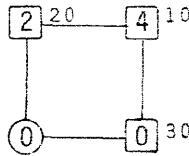


$$\Delta_{22} = 2 - 4 + 0 - 0 = -2$$

olduğuna göre, devrede bir konteynır hareket ettirildiğinde, maliyetler 2 birim kadar azalmaktadırlar. Bu şekilde kaç adet konteynırın hareket ettirilmesi gerektiği sorusuna cevap şudur : mümkün olduğu kadar çok konteynır hareket ettirilmelidir.

Devrede hareket ettirilmesi gereken birimlerin maksimum sayısı, devrenin negatif bir köşesine ait en küçük x_{ij} ile saptanmaktadır.

Burada en küçük miktar 20 konteynırdır. Bunlar yukarıdaki devrede hareket ettirildiğinde devre şu şekli almaktadır.



Hangi serbest elemanın çözüme dahil edileceğine karar verileceği zaman şu şekilde hareket edilmektedir. :

- a-Serbest elemana ait olan devre saptanmakta,
- b-Negatif köşelere ait x_{ij} miktarlarının minimumu bulunmakta,
- c-Pozitif köşelere ait x_{ij} değişkenlerinin değerleri bu minimum kadar arttırılmakta ve negatif köşelere ait değişkenlerin değerleride bu minimum kadar azaltılmaktadır.

Bu işlem yapıldıktan sonra, serbest eleman işgal edilmiş eleman olmakta, fakat minimum miktara ait önceki işgal edilmiş eleman da serbest eleman olmaktadır. Böylece işgal edilmiş elemanlar için olan şart da geçerliliğini kaybetmez.

Yeni çözümün aldığı şekil tablo (2.8) de gösterilmektedir.

TABLO (2.8)

3	$\boxed{0}^{40}$	4	1	1	40
2	$\boxed{2}^{20}$	$\boxed{4}^{10}$	$\boxed{0}^{40}$	6	70
$\boxed{0}^{30}$	0	$\boxed{0}^{30}$	1	1	60
3	5	$\boxed{1}^{10}$	3	$\boxed{0}^{20}$	30
30	60	50	40	20	200

Bu yeni tabloda serbest elemanların yeniden gözden geçirilmesi ve bütün işlemlerin gerektiği kadar tekrarlanması gerekmektedir.

2.1.1.6. Gölge Maliyetler Metodu

Daha önce incelenen çözümü islah etme metodunun, büyük problemler için gayet sıkıcı ve ağır olduğu aşikâr olarak görülmektedir. İşlemleri basitleştirmek için *Gölge Maliyetler Metodunun* kullanılması gereklidir.

Bu methoda maliyet matrisinin her satır ve sütununa bir gölge maliyet verilmektedir. Satırlar için olan gölge maliyetlere u_1, u_2, u_3, u_4 , sütunlar için olanlara ise v_1, v_2, v_3, v_4, v_5 denmektedir. Maliyet matrisinin her elemanının biri satır diğeri sütun olmak üzere iki gölge maliyeti vardır. Bu gölge maliyetler şu şekilde hesaplanmaktadır:

İşgal edilmiş bir elemana ait gölge maliyetlerin toplamı bu elemanın kendisine esittir.

Bu ifade tablo (2.9) da açık olarak gösterilmektedir. Bu tabloda yalnız işgal edilmiş elemanlar ve gölge maliyetler verilmektedir.

TABLO (2.9)

v_j u_i	0	0	0	-4	-1
0		0			
4			4	0	
0	0	0	0		
1			1		0

Tablonun birinci sırasında sütun gölge maliyetleri, birinci sütununda ise satır gölge maliyetleri bulunmaktadır. Burada gölge maliyetlerinin nasıl bulunacağı sorusunun cevabı şudur : İlk gölge maliyetin büyüklüğü keyfi olarak seçilmekte ve buna bağlı olarak diğer gölge maliyetlerin değerleri saptanmaktadır.

Tablo (2.9) da ilk değer olarak $u_1 = 0$ seçilmiştir, buna bağlı olarak ikinci sütunun gölge maliyeti

$$0 = 0 + v_2$$

denklemini sağlamaya mecbur olmaktadır. Buradan $v_2 = 0$ elde edilmektedir. Buna benzer şekilde

$$0 = u_3 + 0$$

dan $u_3 = 0$ elde edilmektedir. Böylece adım adım bütün gölge maliyetler hesaplanmaktadır. Her adımda bir gölge maliyeti bilinen bir eleman alınmakta ve bunun diğer gölge maliyeti hesaplanmaktadır.

Gölge maliyetler metodunun avantajı serbest elemanlara ait olan Δ_{ij} miktarlarının kolaylıkla hesaplanabilmesidir. Ele alınan serbest elemandan kendisine ait gölge maliyetlerin toplamı çıkartılmakta ve Δ_{ij} elde edilmektedir.

Tablo (2.9) serbest maliyet elemanları ile tamamlandığında aşağıdaki şekli alır.

TABLO (2.10)

$v_j \backslash u_i$	0	0	0	-4	-1
0	3	0	4	1	1
4	2	2	4	0	6
0	0	0	0	1	1
1	3	5	1	3	0

Bu tablodaki her elemandan kendisine ait gölge maliyetlerin toplamı çıkartıldığında tablo (2.11) elde edilmektedir.

TABLO (2.11)

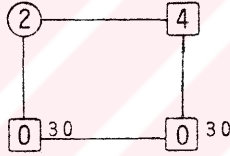
3	0	4	5	2
-2	-2	0	0	3
0	0	0	5	2
2	4	0	6	0

Tablo (2.11), tablo (2.7) nin aynısıdır. Böylece görüldüğü gibi, gölge maliyetler metoduyla çözümün islahına yaklaşmak daha kolay olmaktadır. Ancak çözümün islahı daha önce den izah edildiği şekilde yapılmaktadır. Çözüme tablo (2.8) den itibaren devam edilmektedir. Bu tablonun gölge maliyetleri hesaplandığında ve pozitif farklar için yalnız + işareti konduğunda tablo (2.12) elde edilmektedir.

TABLO (2.12)

$v_j \backslash u_i$	2	0	2	-2	1
0	+	0	+	+	0
2	-2	0	0	0	+
-2	0	+	0	+	+
-1	+	+	0	+	0

Bu son tablodan görüldüğü gibi, çözümün bir yerde daha
 islah edilmesi gerekmektedir, çünkü $\Delta_{21} = -2$ dir. Bu ele-
 manlara ait devre aşağıdaki gibidir :



Negatif köşelerdeki miktarların minimumu 10 dur, böy-
 lece aşağıdaki yeni çözüm elde edilmektedir.

TABLO (2.13)

3	0 ⁴⁰	4	1	1	40
2 ¹⁰	2 ²⁰	4	0 ⁴⁰	6	70
0 ²⁰	0	0 ⁴⁰	1	1	60
3	5	1 ¹⁰	3	0 ²⁰	30
30	60	50	40	20	200

İşlemlere devam edildiğinde tablo (2.12) gibi bir tablo elde edilmektedir.

TABLO (2.14)

$v_j \backslash u_i$	0	0	0	-2	-1
0	+	0	+	+	0
2	0	0	+	0	+
0	0	0	0	+	+
1	+	+	0	+	0

Tablo (2.14) de negatif Δ_{ij} bulunmadığına göre tablo (2.13) deki çözüm optimal olmaktadır. Bu tablo ulaştırılacak konteynırların sayılarını vermekte fakat toplam maliyeti doğrudan doğruya vermemektedir. Buradaki çözümün maliyeti şöyle hesaplanmaktadır.

$$(40 \times 0) + (10 \times 2) + (20 \times 2) + (40 \times 0) + (20 \times 0) \\ + (40 \times 0) + (10 \times 1) + (20 \times 0) = 70$$

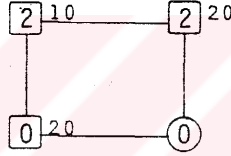
Bu miktara, daha önceden hesaplanan 570 ilave edilerek orijinal problemin toplam maliyeti olan 640 bulunmaktadır.

Ulaştırılacak konteynır miktarları orijinal maliyet tablosunda yerlerine konup toplam maliyet hesaplandığında da aynı sonuç elde edilmektedir.

$$(40 \times 2) + (10 \times 4) + (20 \times 3) + (40 \times 5) + (20 \times 2) \\ + (40 \times 3) + (10 \times 4) + (20 \times 3) = 640$$

2.1.1.7. Birden Fazla Optimal Çözümün Olması Hali

Bir optimal çözümde serbest elemanların Δ_{ij} leri istisnasız olarak pozitif ise problemin yalnız bir optimal çözümü var demektir. Fakat herhangi bir serbest elemanın Δ_{ij} sinin sıfır olması halinde yalnız bir optimal çözüm vardır denememektedir. Böyle bir durumda adı geçen eleman da çözümün içine sokulmakta ve elde edilen tablo optimal çözüm vermektedir. Tablo (2.14) de bu durumla karşılaşılmaktadır, çünkü $\Delta_{32} = 0$ dır. Gereken işlemler yapıldığında aşağıdaki devre ve tablo (2.15) bulunmaktadır.



TABLO (2.15)

3	0 ⁴⁰	4	1	1	40
2 ³⁰	2	4	0 ⁴⁰	6	70
0 ⁰	0 ²⁰	0 ⁴⁰	1	1	60
3	5	1 ¹⁰	3	0 ²⁰	30
30	60	50	40	20	200

Tablo (2.15) de optimal bir çözüm vermektedir. Burada kritik sayı şartını sağlamak için c_{31} eleman işgal edilmiş bir eleman olarak bırakılmaktadır. Tablo (2.15) den hesaplanan maliyetin daha önceki minimum maliyete eşit olduğu görülmektedir.

Yukarıdaki devre ile bir optimal çözümden başka bir optimal çözüm elde edildiği için bu devreye *optimal devre* adı verilmektedir. Bu devrede maksimal olmayan bir miktar hareket ettirildiğinde, çözüm yine optimal olmaktadır. Yukarıdaki devrede 10 yerine 8 konteynırın hareket ettirilmesi halinde aşağıdaki tablo elde edilir :

TABLO (2.16)

3	$\boxed{0}^{40}$	4	1	1	40
$\boxed{2}^{18}$	$\boxed{2}^{12}$	4	$\boxed{0}^{40}$	6	70
$\boxed{0}^{12}$	$\boxed{0}^8$	$\boxed{0}^{40}$	1	1	60
3	5	$\boxed{1}^{10}$	3	$\boxed{0}^{20}$	30
30	60	50	40	20	200

Burada elde edilen çözüm de optimaldir, ancak kritik sayı şartı sağlanamamaktadır.

2.1.2. Taşın Probleminin Özel Halleri

Şimdiye kadar yapılan çalışmada kaynaklarda bulunan ulaştırılacak miktarların, hedeflerden talep olunan miktarlara eşit oldukları veya kaynaklarla hedeflerin hepsinin arasında birer demiryolu bağlantısının var olduğu kabul edilmekteydi. Bu durumlar her zaman için geçerli olmamaktadırlar, dolayısıyla bunlarında incelenmesi gerekmektedir.

2.1.2.1. Sanal Kaynaklar ve Hedefler

Kaynaklardaki arzın, hedeflerdeki talebe eşit olmaması halinde, sanal kaynaklar ve/veya hedefleri problemin çözümü ne sokmak gerekmektedir.

Bu hal tablo (2.1) de bir değişiklik yapılmak suretiyle incelenecektir. Bu tabloda 2 hedefinin, 60 yerine sadece 45 konteynır talep ettiği farz olunmakta, diğer bütün

bilgilerin ise aynı kaldığı kabul olunmaktadır. Elde bulunan konteynırlar talep edilenlerden fazla olduğuna göre, sanal bir hedefin var olduğu ve burada da talebin 15 konteynır kadar olduğu kabul edilmektedir. Ayrıca sanal bir hedefin talebini karşılamak maddeten imkansız olduğundan, bu hedefe ait maliyet elemanlarının birbirine eşit olduğu kabul edilmektedir. İşleri basitleştirmek için de bu elemanların hepsi sıfır alınmaktadır. Ortaya çıkan bu yeni problemi tablo (2.17) ile ifade etmek mümkündür:

TABLO (2.17)

6	2	8	7	5	0	40
4	3	7	5	9	0	70
2	1	3	6	4	0	60
5	6	4	8	3	0	30
30	45	50	40	20	15	200

Daha önceki işlemler bu tabloya uygulandığında, bu problemin ilk çözümü elde edilmektedir. Tablo (2.18) de ilk çözüm ile birlikte gölge maliyetler verilmektedir:

TABLO (2.18)

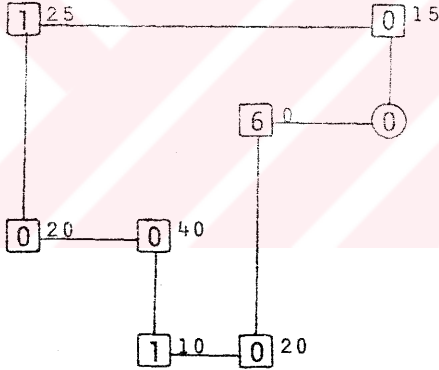
$u_i \backslash v_j$	-4	1	1	-6	0	0	
0	4	$\boxed{1}^{25}$	5	2	2	$\boxed{0}^{15}$	40
6	$\boxed{2}^{30}$	2	4	$\boxed{0}^{40}$	6^0	0	70
-1	0	$\boxed{0}^{20}$	$\boxed{0}^{40}$	1	1	0	60
0	3	5	$\boxed{1}^{10}$	3	$\boxed{0}^{20}$	0	30
	30	45	50	40	20	15	200

Δ_{ij} ler tablosu aşağıdaki gibidir.

TABLO (2.19)

+	0	+	+	+	0
0	-5	-3	0	0	-6
+	0	0	+	+	+
+	+	0	+	0	0

Burada en küçük eleman Δ_{26} dir ve -6 ya eşittir. Bu elemanlara ait devre ise aşağıdaki gibidir.

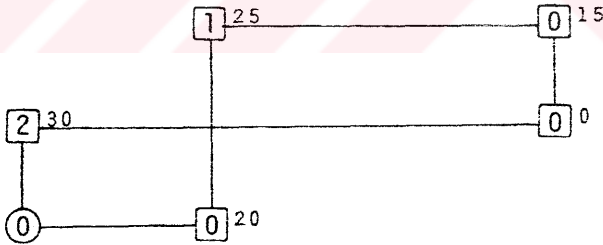


Bu devre kullanılarak yapılan islah etme işlemi bir sonuç vermemekte, ancak burada ulaştırılacak konteynır sayılarından 0_u , başka bir noktaya konarak yeni bir deneme yapılması gerekmektedir. Yeni gölge maliyetler ile tablo (2.20) elde edilmektedir.

TABLO (2.20)

$u_i \backslash v_j$	-2	1	1	0	0	0	
0	4	$\boxed{1}^{25}$	5	2	2	$\boxed{0}^{15}$	40
0	$\boxed{2}^{30}$	2	4	$\boxed{0}^{40}$	6	$\boxed{0}^0$	70
-1	0	$\boxed{0}^{20}$	$\boxed{0}^{40}$	1	1	0	60
0	3	5	$\boxed{1}^{10}$	3	$\boxed{0}^{20}$	0	30
	30	45	50	40	20	15	200

Bu tablodan yalnız bir adet negatif Δ_{ij} bulunabilmektedir. $\Delta_{31} = -1$ 'e ait devre ise aşağıdaki gibidir.



Bu devre ise optimal çözümü vermektedir.

TABLO (2.21)

4	$\boxed{1}^{40}$	5	2	2	0	40
$\boxed{2}^{15}$	2	4	$\boxed{0}^{40}$	6	$\boxed{0}^{15}$	70
$\boxed{0}^{15}$	$\boxed{0}^5$	$\boxed{0}^{40}$	1	1	0	60
3	5	$\boxed{1}^{10}$	3	$\boxed{0}^{20}$	0	30
30	45	50	40	20	15	200

Kaynak 2 den yalnız 55 konteynır yollanmıştır, 15 konteynır ise hiç kullanılmamıştır. Taşın maliyeti tablo (2.21) e göre şöyle hesaplanmaktadır:

$$(40 \times 1) + (15 \times 2) + (40 \times 0) + (15 \times 0) + (15 \times 0) + (5 \times 0) + (40 \times 0) + (10 \times 1) + (20 \times 0) = 80$$

Ancak bu miktara tablo (2.17) den tablo (2.18) e geçerken yapılan değişikliğin oluşturduğu

$$(30 \times 2) + (45 \times 1) + (50 \times 3) + (40 \times 5) + (20 \times 3) = 515$$

farkının da ilave edilmesi gerekmektedir. böylece toplam maliyet

$$515 + 80 = 595$$

olmaktadır.

Talebin arzdan fazla olması halinde ise, sanal bir kaynağın var olduğu kabulü yapılmaktadır.

2.1.2.2. Kapasite Kısıtlamalı Taşın Problemi

Bir istasyondan diğerine ulaştırılacak konteynır miktarlarının bazen kısıtlamalara maruz kaldıkları görülen olaylar arasındadır. Bu kısıtlamalar alt, üst limitler veya kapasite sınırlarıdır. Bu problemin çözümü, sınırların sıfır olması halinde, yani hiç bir konteynırın, herhangi bir istasyondan diğerine ulaştırılamaması halinde gayet basit olmaktadır. Ancak sınırın pozitif olması halinde, çözüm zorlaşmaktadır, çünkü bu durum bir kapasite kısıtlaması olmaktadır.

İlk olarak basit olan hal incelenecektir. Daha Önceden incelenen örnek yeniden ele alındığında ve kaynak 2 den hedef 4'e hiç bir konteynırın ulaştırılamayacağı farz olduğunda, kapasite kısıtlamalı bir problem elde edilmektedir. Bu problemin yalnız $x_{24} = 0$ olduğu zaman çözümü vardır. Bu durumu elde etmek için değeri Önceden 5 olan c_{24} maliyet elemanının yerine çok büyük bir sayı olan S konmaktadır. İşlemlerin esas gayesi minimum maliyetleri tesbit etmek olduğuna göre bu yeni S elemanının daima çözüm dışı bırakılmasına gayret edilmektedir. S 'e sayısal bir değer verilmekte, ancak sonsuz denebilecek kadar büyük bir değerinin olduğu kabul edilmektedir. Bu durumda bu elemanı bir miktar arttırmak veya eksiltmekle değerinin değişmediği de kabul edilmektedir. Tablo (2.1) e S de sokulunca tablo (2.22) elde edilmektedir.

TABLO (2.22)

6	2	8	7	5	40
4	3	7	S	9	70
2	1	3	6	4	60
5	6	4	8	3	30
30	60	50	40	20	200

Daha Önceden yapıldığı gibi *Stepping-stone* metodu kullanılarak optimal çözüm elde edilmektedir. Bu çözüm tablo (2.23) de gösterilmektedir.

TABLO (2.23)

6	2	8	$\boxed{7}^{40}$	5	40
$\boxed{4}^{10}$	$\boxed{3}^{60}$	7	S	9	70
$\boxed{2}^{20}$	1	$\boxed{3}^{40}$	$\boxed{6}^0$	4	60
5	6	$\boxed{4}^{10}$	8	$\boxed{3}^{20}$	30
30	60	50	40	20	200

Bu tablo vasıtasıyla taşın maliyeti bilindiği gibi şu şekilde hesaplanır :

$$(40 \times 7) + (10 \times 4) + (60 \times 3) + (20 \times 2) \\ + (40 \times 3) + (0 \times 6) + (10 \times 4) + (20 \times 3) = 760$$

Burada elde edilen maliyet daha önce hesaplanandan doğal olarak daha büyüktür. $760 - 640 = 120$ farkı kısıtlanmanın maliyetini belirtmektedir.

Bir çok maliyet elemanının yerine S koymak mümkündür, ancak bu durumda S'i her zaman çözüm dışı bırakmak mümkün olmamaktadır. Böyle bir durumla karşılaşıldığında istenen şartın gerçekleştirilemeyeceği, yani problemin bir çözümü-
nün olmadığı ortaya çıkmaktadır.

İkinci olarak yine aynı örnekte kaynak 2 den hedef 4'e yalnız 30 konteynır yollanabilmesi hali ele alınmaktadır. Bu durumda $x_{24} \leq 30$ kısıtlaması var demektir, bir önceki örnekte $x_{24} = 0$ idi.

Burada hemen c_{24} değiştirilerek problem çözülememektedir. Tablo (2.1) den elde edilen tablo (2.2) eörüne alınmakra ve ilk çözüme başlanmaktadır. Ancak bu sefer bazı değişiklikler yapılmaktadır. İlk olarak c_{24} elemanından işe başlanmakta ve buna $x_{24} = 30$ üst sınırı verilmektedir. $x_{24} = 30$ koymakla buna ait maliyet elemanı doymuş duruma getirilmektedir. Bundan sonra probleme, bir önceki örnekte olduğu gibi değiştirilerek devam edilmektedir. Problemin

hesaplarına aşağıdaki tablo (2.24) den hareket ederek başlanmaktadır.

TABLO (2.24)

3	0	4	1	1	40
2	2	4	S	6	40
0	0	0	1	1	60
3	5	1	3	0	30
30	60	50	10	20	170

Bu tabloda göze çarpan iki özellikten birincisi herhangi bir satır veya sütunun eksilmemesi, ikincisi de c_{24} ün yerine S in almasıdır. Burada c_{24} ün yerine S koymak gerekmektedir, çünkü c_{24} ün tamamı kullanılmaktadır, diğer bir deyimle doymuş hale gelmektedir. $x_{24} > 0$ olduğuna göre c_{24} işgal edilmiş bir elemandır, ancak ne bir kaynak ne de bir hedef c_{24} 'e bağlı olarak problemde çikartılmadığından c_{24} temel bir eleman olamamaktadır. Buradan hareket ederek yeni bir kavram olan temel eleman şöyle tarif edilebilmektedir, bir işgal edilmiş elemanın herhangi bir satır veya sütunun iptali ile ilgisi olduğu takdirde bu elemana temel eleman adı verilmektedir.

Gereken işlemlerin probleme uygulanmasıyla optimal çözüm elde edilmektedir. Bu çözüm tablo (2.25) de gösterilmektedir.

TABLO (2.25)

3	$\boxed{0}^{30}$	4	1	1	40
$\boxed{2}^{10}$	$\boxed{2}^{30}$	4	S	6	40
$\boxed{0}^{20}$	0	$\boxed{0}^{40}$	1	1	60
3	5	$\boxed{1}^{10}$	3	$\boxed{0}^{20}$	30
30	60	50	10	20	170

Bu tablodan hareket ederek bir optimal çözüm elde edilmektedir. Bu çözüm ise tablo (2.26) da gösterilmektedir.

TABLO (2.26)

3	$\boxed{0}^{30}$	4	$\boxed{1}^{10}$	1	40
$\boxed{2}^{10}$	$\boxed{2}^{30}$	4	$\boxed{0}^{30}$	6	70
$\boxed{0}^{20}$	0	$\boxed{0}^{40}$	1	1	60
3	5	$\boxed{1}^{10}$	3	$\boxed{0}^{20}$	30
30	60	50	40	20	200

Burada c_{24} için özel bir notasyon kullanılmaktadır, buna sebep c_{24} ün işgal edilmiş fakat temel olmayan bir eleman olduğunu göstermektir.

Bu çözümün islah edilemeyeceği, gölge maliyetler yardımıyla anlaşılmaktadır. Burada gölge maliyetler yalnız temel elemanlar gözönüne alınarak hesaplanmaktadır. c_{24} temel eleman olmadığı için bu hesaplar yapılırken işin içine sokulmamaktadır. $u_1 = 0$ alınarak gölge maliyetler hesaplanmakta ve tablo (2.27) elde edilmektedir.

TABLO (2.27)

$v_j \backslash u_i$	0	0	0	1	-1	
0	3	$\boxed{0}^{30}$	4	$\boxed{1}^{10}$	1	40
2	$\boxed{2}^{10}$	$\boxed{2}^{20}$	4	$\boxed{0}^{30}$	6	70
0	$\boxed{0}^{20}$	0	$\boxed{0}^{40}$	1	1	60
1	3	5	$\boxed{1}^{10}$	3	$\boxed{0}^{10}$	30
	30	60	50	40	20	200

TABLO (2.29)

0	0	0	0	S	25
1	0	4	2	S	25
1	0	0	3	S	50
S	S	S	S	0	
15	20	30	35		

Bu genişletme gereklidir, çünkü daha önceden problemin çözümünün olup olmadığı belli değildir. Nereye yerleştirileceği belli olmayan miktarlar için sanal kaynak ve hedefler ilk çözüme dahil edilmektedir. Bundan sonra işlemlerin varlığına geçilerek tablo (2.30) elde edilmektedir.

TABLO (2.30)

$\boxed{0}^{15}$	$\boxed{0}^{10}$	$\boxed{0}^0$	0	S	25
1	$\boxed{0}^{10}$	4	$\boxed{2}^{15}$	S	25
1	0	$\boxed{0}^{22}$	$\boxed{3}^{20}$	$\boxed{S}^8$	50
S	S	$\boxed{S}^8$	S	0	8
15	20	30	35	8	

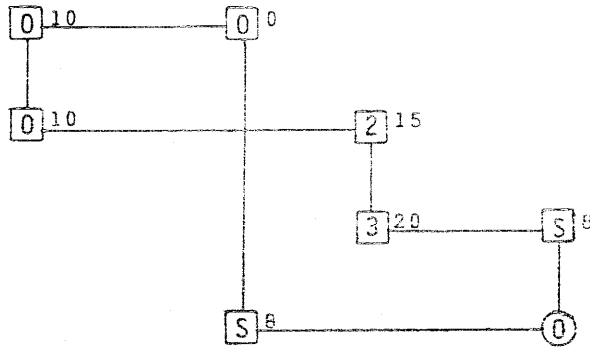
Bu tabloda ilave edilen 4. satır ve 5. sütun, görüldüğü gibi 8 ünite için kullanılmaktadır. c_{33} elemanı için $x_{33} = 22$ bulunmaktadır, yani bu eleman doymuş hale gelmekte ve bir temel eleman olamamaktadır. Şimdi S yüksek maliyet elemanlarına tahsis edilen sanal miktarların ortadan kaldırılması gerekmektedir. Bu iş yalnız x_{45} e 8 ünitenin tahsis edilmesiyle gerçekleşmektedir, aksi takdirde kısıtlamalar orijinal problemin bir çözümünün olmasını engellemektedir. Tablo (2.30) a gölge maliyetler ilave edildiğinde

tablo (2.31) elde edilmektedir.

TABLO (2.31)

$v_j \backslash u_i$	0	0	0	2	S-1	
0	$\boxed{0}^{15}$	$\boxed{0}^{10}$	$\boxed{0}^0$	0	S	25
0	1	$\boxed{0}^{10}$	4	$\boxed{2}^{15}$	S	25
1	1	0	$\boxed{0}^{22}$	$\boxed{3}^{20}$	$\boxed{S}^8$	50
S	S	S	$\boxed{S}^8$	S	0	8
	15	20	30	35	8	

$c_{45} = 0$. $u_4 + v_5$ den çok küçük olduğundan c_{45} ait devrenin kurulması gerekmektedir.



Kapasite kısıtlamalarının olmaması halinde, bu devre etrafında 8 ünitenin hareket ettirilmesi mümkündür, ancak

$x_{34} \leq 22$ kısıtlamasından dolayı yalnız 2 birimin hareket ettirilmesi mümkün olmaktadır. Böylelikle c_{34} elemanı doygun, c_{45} elemanı ise temel olmaktadır. Gölge maliyetleri ile beraber yeni tablo aşağıdaki gibidir :

TABLO (2.32)

$u_i \backslash v_j$	0	0	0	2	-S	
0	$\boxed{0}^{15}$	$\boxed{0}^8$	$\boxed{0}^2$	0	S	25
0	1	$\boxed{0}^{12}$	4	$\boxed{2}^{13}$	S	25
25	1	0	$\boxed{\emptyset}^{22}$	$\boxed{2}^{22}$	$\boxed{S}^6$	50
S	S	S	$\boxed{S}^6$	S	$\boxed{0}^2$	8
	15	20	30	35	8	

TABLO (2.33)

$u_i \backslash v_j$	0	0	0	2	
0	$\boxed{0}^{15}$	$\boxed{0}^2$	$\boxed{0}^6$	0	25
0	1	$\boxed{0}^{12}$	4	$\boxed{2}^{13}$	25
0	1	$\boxed{0}^6$	$\boxed{0}^{22}$	$\boxed{2}^{22}$	50
	15	20	30	35	100

Bu tablo, ilk önce c_{14} e daha sonra c_{21} e pozitif degerler vererek islah edilmekte ve tablo (2.34) e gegilmektedir.

TABLO (2.34)

$u_i \backslash v_j$	0	-1	0	0	
0	$\boxed{0}^4$	0	$\boxed{0}^6$	$\boxed{0}^{13}$	25
1	$\boxed{1}^{11}$	$\boxed{0}^{14}$	4	2	25
1	1	$\boxed{0}^6$	$\boxed{0}^{22}$	$\boxed{3}^{22}$	50
	15	20	30	35	100

Herhangi bir serbest eleman bu çözümü islah edememektedir, ancak $u_3 + v_4 < c_{34}$ olduğu için, c_{34} doygun elemanı islah için kullanılmaktadır. Bu elemana ait devre kurulduğunda 4 ünitenin hareket ettirilebileceği görülmekte ve aşağıdaki tablo (2.35) elde edilmektedir.

TABLO (2.35)

$u_i \backslash v_j$	-2	-3	0	0	
0	0	0	$\boxed{0}^8$	$\boxed{0}^{17}$	25
3	$\boxed{1}^{15}$	$\boxed{0}^{10}$	4	2	25
3	1	$\boxed{0}^{10}$	$\boxed{0}^{22}$	$\boxed{3}^{18}$	50
	15	20	30	35	100

Islah işlemine devam edildiğinde optimal çözüme tablo (2.36) da varılmaktadır.

TABLO (2.36)

0	0	$\boxed{0}^8$	$\boxed{0}^{17}$	25
$\boxed{1}^7$	0	4	$\boxed{2}^{18}$	25
$\boxed{1}^8$	$\boxed{0}^{20}$	$\boxed{0}^{22}$	3	50
15	20	30	35	100

2.2. ULAŞTIRMA AĞLARINDA AKIMLAR

2.2.1. Statik Maksimal Akım

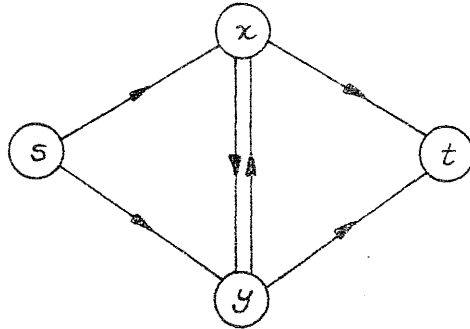
2.2.1.1. Genel

Ulaştırma ağlarındaki akımların incelenmesine ve kullanılmasına geçilmeden önce, bunların genel tariflerinin yapılması ve nasıl kullanılacaklarının belirtilmesi gerekmektedir.

2.2.1.2. Ağlar

Yönlendirilmiş bir ağ veya yönlendirilmiş bir lineer graf $G = [N; A]$ ile gösterilmektedir. Burada N , $x, y, \dots$ elemanlarından, A ise N in x, y elemanlarının çiftler alınmasından doğan (x, y) elemanlarından oluşmaktadır. N in elemanlarına, *düğüm*, *birleşme noktası* veya *nokta*; A nın elemanlarına ise, *ok*, *bağ*, *dal*, *kenar* adları verilmektedir. Bu çalışmada N in elemanlarına *düğüm*, A nın elemanlarına da yönlendirilmiş oldukları takdirde *ok*, aksi halde *bağ* adı verilmektedir. Buraya kadar yapılan açıklamaları küçük bir örnek üzerinde göstermek gerekmektedir.

Şekil (2.1) gösterilen ağın s, x, y, t gibi 4 düğümü ve (s, x) , (s, y) , (x, y) , (y, x) , (x, t) ve (y, t) gibi 6 oku vardır.



Şekil (2.1)

Bu örnekte gösterilen ağ, yönlendirilmiş bir ağdır, çünkü her bağın bir yönü vardır.

$x_1, x_2, \dots, x_n$, ($n \geq 2$), bir ağın farklı düğümlerinin dizisi ve (x_i, x_{i+1}) , $i = 1, \dots, n - 1$ ise okları olduğu taktirde, düğüm ve oklardan oluşan $x_1, (x_1, x_2), x_2, \dots, (x_{n-1}, x_n), x_n$ dizisine de *zincir* adı verilmektedir. $x_n = x_1$ olduğu taktirde adı geçen diziye *yönlendirilmiş devre* adı verilmektedir.

Şekil (2.1) deki ağda $s, (s, x), x, (x, t), t$ bir zincir, $x, (x, y), y, (y, x), x$ ise bir yönlendirilmiş devre olmaktadır.

s den t ye bağların yönlerine önem vermeden gidildiğinde, elde edilen bağ ve düğüm grubuna *yol* adı verilmektedir. Verilen örnekte $s, (s, y), y, (x, y), x, (x, t), t$ düğüm ve oklarıyla s den t ye gidildiğinde bir yol elde edilmektedir. Yolun zincirden farkı, yoldaki okların aksi yönlerinden de gidilebilinmesidir. Zincir için olan notasyon, basitleştirilip $x_1, x_2, \dots, x_n$ olarak kullanılabilir. *devre*.

2.2.1.3. Ağlarda Akım

Bir $G[N; A]$ ağında, her $(x, y) \in A$ oku negatif olmayan gerçek bir $k(x, y)$ sayısına sahip olabilmektedir. Bu $k(x, y)$ sayısına da (x, y) okunun *kapasitesi* adı verilmektedir.

Böylece kapasite de tarif edildikten sonra statik maksimal akımın tarifine geçilmektedir. s den t ye giden ve v değerine sahip statik bir f akımının aşağıdaki denklem ve eşitsizlikleri yani kısıtlamaları sağlaması gerekmektedir.

$$\sum_{y \in S(x)} f(, y) - \sum_{y \in E(x)} f(y, x) = \begin{cases} v & x = s \\ 0 & x \neq s, t \\ -v & x = t \end{cases}$$

(2.11)

$$f(x, y) \leq k(x, y) \quad (x, y) \in A \quad (2.12)$$

Burada s kaynak, t hedef ve diğer düğümler ara düğümler olmaktadır. Böylece s den çıkan net akım v , t den çıkan net akım $-v$ (veya t ye giren net akım v) ve ara düğümlerden çıkan net akım sıfır olmaktadır. Herhangi bir x ara düğümünden çıkan net akım denklemi

$$\sum_{y \in S(x)} f(x, y) - \sum_{y \in E(x)} f(y, x) = 0 \quad (2.13)$$

şeklinde ve bu gibi denklemlere *korunum denklemi* adı verilmektedir.

Statik maksimal akım problemi (2.11) ve (2.12) akım kısıtlamalarına tabi olan v değişkeninin maksimize edilmesidir.

Şimdi burada kullanılan $S(x)$ ve $E(x)$ notasyonlarının tanımlanması gerekmektedir. $(x, y) \in A$ ve $y \in N$ ile tanımlanan bütün y lerin oluşturduğu cümlelerin $S(x)$, yani x den sonra olduğu ve buna benzer olarak $(y, x) \in A$ ve $y \in N$ ile tanımlanan bütün y lerin oluşturduğu cümlelerin ise $E(x)$, yani x den evvel olduğu kabul edilmektedir. Bu söylenenleri aşağıdaki gibi ifade etmek mümkündür.

$$S(x) = \{ y \in N \mid (x, y) \in A \} \quad (2.14)$$

$$E(x) = \{ y \in N \mid (y, x) \in A \}$$

Şekil (2.1) deki ağda

$$S(x) = \{x, y\}$$

$$E(x) = \emptyset \quad (\text{boş cümle})$$

olmaktadır.

2.2.1.4. Kesimler

Maksimal akım probleminin çözümüne doğru ilerlerken bir ağdaki kesimleri de açıklamak gereklidir. Ancak kesimlere geçmeden önce notasyon kolaşlaştırıcı bir işlem yapılmaktadır. X ve Y nin, N in birer cümlecikleri oldukları ve $x \in X$ den $y \in Y$ ye giden bütün oklar cümlesinin (X, Y) olarak gösterildiği kabul olunmaktadır. Böylece herhangi bir g fonksiyonunun notasyonu aşağıdaki şekilde basitleşir.

$$\sum_{(x, y) \in (X, Y)} g(x, y) = g(X, Y)$$

Ayrıca,

$$(E(x), x) = (N, x)$$

$$(x, S(x)) = (x, N)$$

notasyonları kullanılmaktadır.

$[N; A]$ içinde s yi t den ayıran bir C kesimi $(X, \bar{X})$ okları cümlesidir. Burada $s \in X$ ve $t \in \bar{X}$ dir. $(X, \bar{X})$ kesiminin kapasitesi $k(X, \bar{X})$ olmaktadır. Kesime örnek olarak Şekil (2.1) deki

$$C = \{(s, y), (x, y), (x, t)\}$$

oklar cümlesi

$$X = \{s, x\}$$

olmak üzere alınmaktadır.

Şimdi şu önemli açıklamanın yapılması gerekmektedir. Bir $[N; A]$ ağında s den t ye giden f akımının değeri v ve s den t yi ayıran kesimin $(X, \bar{X})$ olduğunda

$$v = f(X, \bar{X}) - f(\bar{X}, X) \leq k(X, \bar{X}) \quad (2.15)$$

olmaktadır.

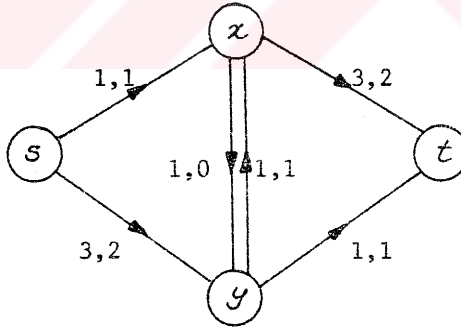
2.2.1.5. Maksimal Akım

Maksimal akımı elde etmek için Ford ve Fulkerson'un *maks-akım min-kesim* teoreminden faydalanılmaktadır. Yine bu iki araştırmacının adı geçen teoreme dayanarak kurdukları *etiketleme metodu* ile maksimal akım bulunabilmektedir.

Burada şu hatırlatmayı yapmak faydalı olacaktır. Bazı literatürde maks-akım min kesim teoremine Ford-Fulkerson teoremi, ve etiketleme metoduna da çoğunlukla Ford-Fulkerson algoritması denmektedir.

Maks-akım min-kesim teoremi : Herhangi bir ağda s den t ye giden bir maksimal akımın değeri, s ve t yi birbirinden ayıran bütün kesimlerin içinde, minimal kesim kapasiteli olanının değerine eşittir.

Maks-akım min-kesim teoremine, Şekil (2.2) örnek olarak verilmektedir. Burada okların üzerinde yazılı olan sayılardan ilki kapasiteyi, $k(x, y)$, ikincisi ise akım, $f(x, y)$, göstermektedir.



Şekil (2.2)

Bu şekilde kolaylıkla görüldüğü gibi maksimum akımın değeri 3 ve (s, x) , (y, x) , (y, t) oklarından oluşan minimum kesimin değeri de 3'dür. Bu da maks-akımın min-kesimine eşit olduğunu göstermektedir.

2.2.1.6. Etiketleme Metodu

Maks-akım min-kesim teoremi, bir ağda maksimal akımın kolayca bulunabilmesi için çok elverişli bir algoritmanın ortaya çıkmasına sebep olmuştur. Bu metoda *Ford-Fulkerson algoritması* dendiği kadar *İşaretleme Metodu* da denmektedir.

Bu algoritmada, hesaplama iki adımda gerçekleştirilmektedir. 1. adımda etiketlemeler yapılmakta, 2. adımda ise bu yapılanlar, ya daha yüksek değerli bir akım vermekte veya şimdiki akımın maksimal olduğunu göstererek sonucu vermektedir.

Bu algoritmada sonucu garantilemek için kapasite fonksiyonu k nın değerinin tam sayı olduğu kabul olunmaktadır. Bu çok zor bir iş değildir, çünkü hesaplamaya başlamadan önce kesirli sayıları tam sayı haline getirmek yeterli olmaktadır.

Bu algoritmada, hesaplamaya başlamadan önce ele alınan ağda, tam sayı değerli bir ilk f akımının var olduğu kabul edilmekte ve bundan sonra etiketleme işlemine girilmektedir.

Etiketleme Metodu :

Adım 1 (etiketleme) : İlk önce s kaynağı $(-, \epsilon(s) = \infty)$ etiketini almaktadır. Şimdi s etiketlenmiş ve incelenmemiş, diğer düğümler etiketlenmemiş durumdadır. Bundan sonra herhangi bir etiketlenmiş, incelenmemiş x düğümü seçilmekte ve bunun etiketini $(z^+, \epsilon(x))$ olduğu kabul edilmektedir. Etiketlenmemiş ve $f(x, y) < k(x, y)$ olan y düğümlerine $(x^+, \epsilon(y))$ etiketi verilmektedir. Burada,

$$\epsilon(y) = \min [\epsilon(x), k(x, y) - f(x, y)] \quad (2.16)$$

formülüyle hesaplanmaktadır. Şimdi y etiketlenmiş ve incelenmemiş durumdadır. Etiketlenmemiş ve $f(y, x) > 0$ olan y düğümlerine ise $(x^-, \epsilon(y))$ etiketi verilmektedir. Burada,

$$\epsilon(y) = \min [\epsilon(x), f(y, x)] \quad (2.17)$$

formülüyle hesaplanmaktadır. Şimdi y etiketlenmiş ve incelenmemiş, x de etiketlenmiş ve incelenmiş durumda olmaktadır. Bu ana kadar yapılan işlemlere, t hedefi etiketlenip, incelenmemiş olana kadar veya hiç bir düğüm etiketlenemez duruma gelene kadar devam edilmektedir. Birinci halde adım 2 ye geçilmekte, ikinci halde ise işlemler durdurulmaktadır, çünkü hesaplama bitmiştir.

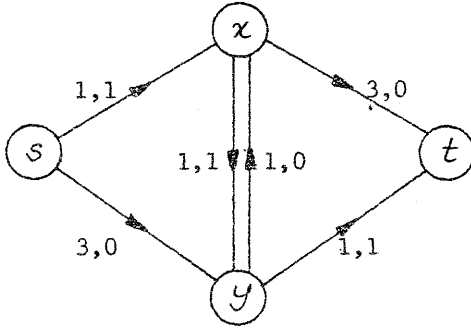
Adım 2 (akım değiştirme) : t hedefi $(y^+, e(t))$ ile etiketlenmiş durumdadır. t nin $(y^+, e(t))$ ile etiketlenmiş olması halinde $f(y, t)$ nin yerine $f(y, t) + e(t)$ konmaktadır. t nin $(y^-, e(t))$ ile etiketlenmiş olması halinde ise $f(t, y)$ nin yerine $f(t, y) - e(t)$ konmaktadır. Her iki halde de, bundan sonra y düğümüne geçilmektedir. y düğümünün $(x^+, e(y))$ ile etiketlenmiş olması halinde $f(x, y)$ nin yerine $f(x, y) + e(t)$ konmaktadır. Aynı düğümün $(x^-, e(y))$ ile etiketlenmiş olması halinde ise $f(y, x)$ in yerine $f(y, x) - e(t)$ konmaktadır ve x düğümüne geçilmektedir. s kaynağına erişilince akım değiştirilmesi durdurulmakta, eski etiketler kaldırılmakta ve adım 1 e geri dönülmektedir.

Bu metod izah edilmeden önce ağdaki ilk f akımının değirinin tam sayı olduğu belirtilmişti. Böyle bir durumda sonuç olarak elde edilen maksimal akımın değeri de tam sayı olmaktadır. Buna ait aşağıdaki teorem de vardır :

Kapasite fonksiyonu k nın tam sayı değerli olması halinde, yine tam sayı değerli bir f maksimal akımı da vardır.

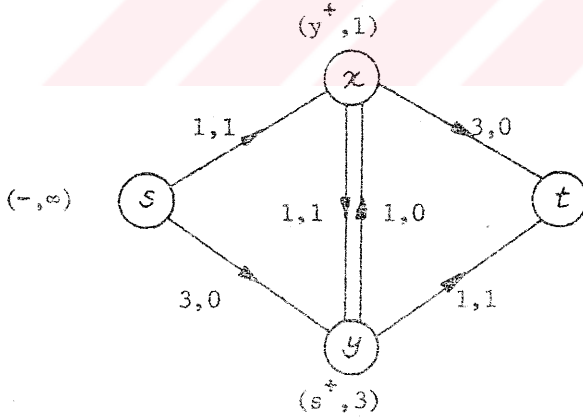
Şimdi ele alınacak örnek, etiketleme metodunu ve bu metod aracılığı ile maksimal akımın bulunmasını kolaylıkla göstermektedir.

Şekil (2.3) de ele alınan ağda, okların üzerinde kapasiteler ve ilk akımın değeri bulunmaktadır. Bunlar herhangi bir (x, y) akımın yanında $k(x, y)$ ve $f(x, y)$ olarak yapılmışlardır.



Şekil (2.3)

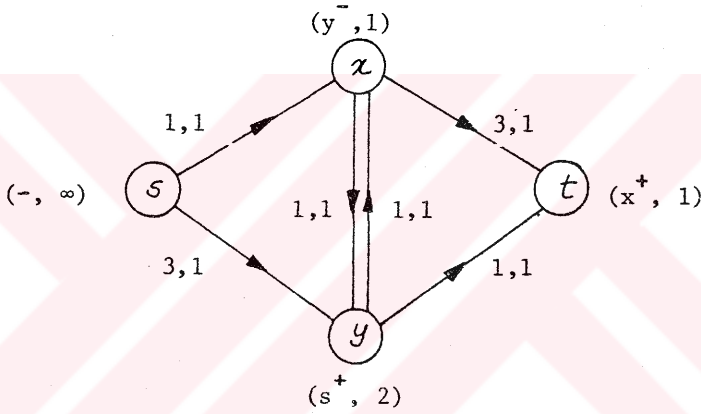
Adım 1'i uygulayarak s düğümünden işe başlanmaktadır. Verilen etiketler Şekil (2.4) de gösterilmektedir. Burada s'e $(-, \infty)$ etiketi verilmektedir.



Şekil (2.4)

s den hareket ederek y'ye $(s^+, \min(3, \infty) = s^+, 3)$ etiketi verilmektedir. x in etiketi $(y^+, 1)$ veya $(y^-, 1)$ olabilir, burada $(y^+, 1)$ alınmaktadır. Daha sonrada t ye $(x^+, 1)$ etiketi verilmektedir.

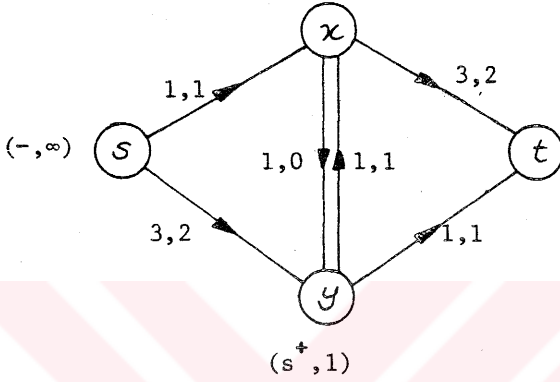
Şimdi Adım 2 uygulanarak akım değiştirilmesine geçilmekte ve $\epsilon(t) = 1$ alınmaktadır. Burada takip edilen yol s, y, x, t zinciridir. 2 değerine sahip yeni akım Şekil (2.5) de gösterilmektedir.



Şekil (2.5)

Bu akım elde edildikten sonra, eski etiketler iptal edilip ve yeniden etiketleme yapıldığında Şekil (2.5)deki etiketler elde edilmektedir. $\epsilon(t) = 1$ konarak, Şekil (2.6) da gösterildiği gibi yeni bir akım elde edilmektedir. Bu yeni akım için takip edilen yol s, (s, y), y, (x, y), x, (x, t), t den oluşmaktadır.

Tekrar adım 1 in uygulanması bir sonuç vermemektedir. Bu uygulamaya ait etiketler Şekil (2.6) de gösterilmektedir. Bu durumda maksimal akım elde edilmiş demektir ve minimal kesim (s, x), (y, x) ve (y, t) oklarından oluşmaktadır.



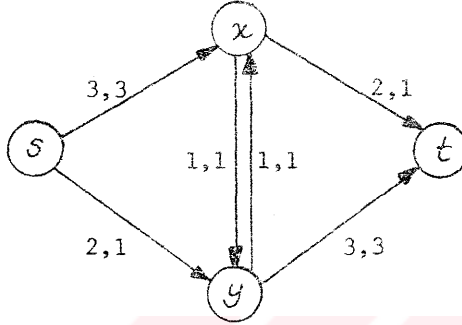
Şekil (2.6)

2.2.2. Dinamik Maksimal Akım

2.2.2.1. Genel

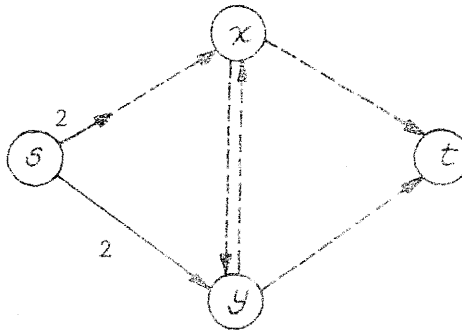
Kaynağı s , hedefi t olan bir $G[N; A]$ ağı ele alınmakta ve bunun her akımın bir kapasitesi olduğu gibi, bir de geçiliş süresi olduğu kabul edilmektedir. G nin düğümlerindeki üniteler, ya hemen ulaştırılabilmekte veya daha sonra ulaştırılmak üzere düğümlerde tutulabilmektedir. Bu durumda, belirli sayıdaki zaman birimi içerisinde kaynaktan hedefe gidecek ünite (akım) miktarının belirlenmesi problemi ile karşı karşıya gelinmektedir. Bu problemin çözümü ise maksimal dinamik akımı vermektedir.

Bu problemi, aşağıdaki örneği inceleyerek açıklamak faydalı olacaktır. Şekil (2.7) de ki ağın oklarının üzerindeki birinci sayı, birim zamanda ünite akımı cinsinden kapasiteyi ikinci sayı ise okun geçiliş süresini göstermektedir.

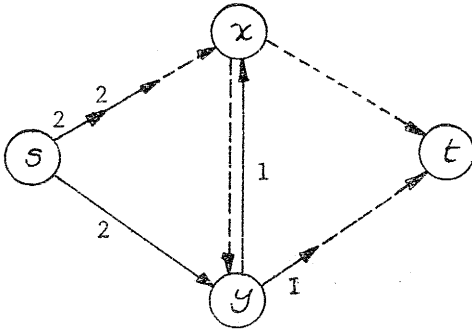


Şekil (2.7)

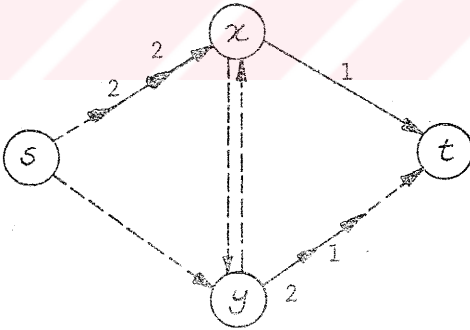
Burada 5 birim zaman içerisinde s den t ye ne kadar ünite ulaştırılabileceği ve bunu gerçekleştirmek için nasıl bir çizelge takip edileceğinin incelenmesi gerekmektedir.



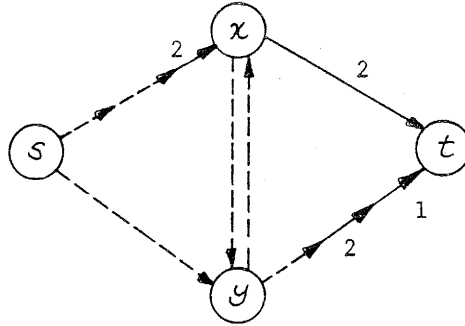
Şekil (2.8)



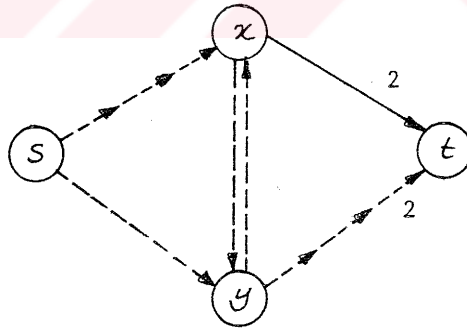
Şekil (2.9)



Şekil (2.10)



Şekil (2.11)



Şekil (2.12)

Şekil (2.8) - (2.12) de 5 birim zamanda ulaştırılan üniteler şematik olarak gösterilmektedir. Şekil (2.8) şu şekilde açıklanabilmektedir. 0 zamanında s den, ikisi x yönünde, ikisi y yönünde olmak üzere dört ünite hareket etmektedir. 1 zamanında x e giden iki ünite (s, x) okunun 1/3 ünü katetmiş, y'ye giden iki ünite ise (s, y) okunun tamamını katetip y'ye varmış olmaktadır. Bu durumda Şekil (2.8), 0-1 zaman aralığını, diğer şekiller de adı geçen zamanı takip eden aralıkları göstermektedir. 3 zamanında 1 ünite, 4 zamanında 3 ünite ve 5 zamanında da 4 ünite t ye varmaktadır. Böylece 0-1, 1-2, 2-3, 3-4, 4-5 zaman aralıklarında s den t ye toplam olarak 8 ünite varmış olmaktadır.

2.2.2.2. Problemin Formülasyonu

Önce problemin formülasyonunda kullanılacak elemanların açıklanmasının yapılması gerekmektedir. Burada ele alınan herhangi bir (x, y) okunun kapasitesi $k(x, y)$, geçiliş süresi ise $a(x, y)$ olmaktadır. Bunların her ikisinde pozitif tam sayılardır. τ zamanında x den hareket edip, (x, y) üzerinde giden ve $\tau + a(x, y)$ zamanında y'ye varan akımın değeri $f(x, y; \tau)$ olmakta, τ dan $\tau - 1$ 'e kadar geçen sürede x de bekleyen akımın değeri ise $f(x, x; \tau)$ olmaktadır. 0-1, 1-2, ..., (p-1)-p aralıklarında s den çıkıp t ye giren net akımın $v(p)$ olduğu kabul edilmektedir. Bu açıklamalardan sonra problem bir lineer programlama problemi şeklinde aşağıdaki denklemlerle ifade edilebilmektedir.

$$\text{Maks } v(p) \quad (2.18)$$

$$\sum_{\tau=0}^p \sum_{y \in N} |f(s, y; \tau) - f(y, s; \tau - a(y, s))| - v(p) = 0 \quad (2.19)$$

$$\sum_{y \in N} |f(x, y; \tau) - f(y, x; \tau - a(y, x))| = 0 \quad (2.20)$$

$$\begin{aligned} x &\neq s, t \\ \tau &= 0, 1, \dots, p \end{aligned} \quad (2.20) \text{ için}$$

$$\sum_{\tau=0}^p \sum_{y \in N} |f(t, y; \tau) - f(y, t; \tau - a(y, t))| + v(p) = 0 \quad (2.21)$$

$$0 \leq f(x, y; \tau) \leq k(x, y) \quad (2.22)$$

Burada (2.18) denklemini hedef fonksiyonunu, (2.19) - (2.22) denklemleri ise kısıtlamaları ifade etmektedir. (2.20) kısıtlaması, bir x ara düğümüne τ zamanında giren akımın, aynı zamanda çıkan akıma eşit olduğunu, (2.19) kısıtlaması p sürelerinde s den çıkan akımın $v(p)$, (2.21) kısıtlaması da bu zaman aralığı içerisinde t ye gelen akımın $v(p)$ olduğunu göstermektedir.

$f(x, y; \tau)$ ve $v(p)$ nin (2.19) - (2.22) şartlarını sağlamaları halinde s den t ye, p süreleri içinde, giden bir f dinamik akımın varlığı ortaya çıkmaktadır. $v(p)$ nin maksimal olması halinde de, f maksimal dinamik akım olmaktadır.

x düğümünde bekleyen bir akımın var olması halinde, $a(x, x) = 1$ ve $k(x, x) = \infty$ kabulleri yapılmaktadır. Ayrıca $\tau < 0$ olduğunda, $f(x, y; \tau)$ değişkeni problemten dışartılmakta yani (x, y) oku G nin elemanlarından sayılmamaktadır.

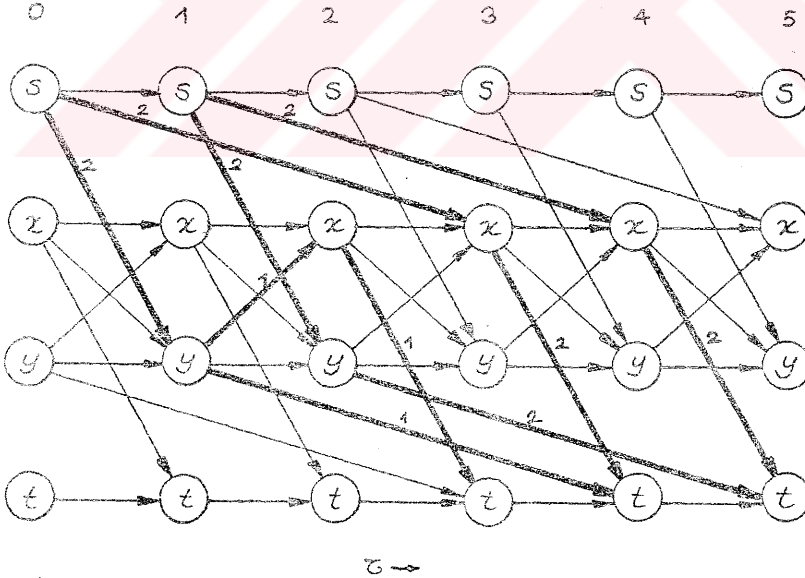
Dinamik akımı tarif eden kısıtlamalar karışık gibi görünüyorlar ise de statik akım için olanlardan daha zor değildirler. Nitekim G ağından geçen p süreli bir dinamik akım, G nin zamana göre genişletilmiş hali olan $G(p)$ ağından geçen bir statik akıma karşı gelmektedir.

2.2.2.3. Dinamik Ağdan Statik Ağa Geçiş

G esas alınarak kurulan $G(p)$ ağı, aşağıda açıklanan şekilde elde edilmektedir. G nin x düğümüne karşılık $G(p)$ nin $(p + 1)$ adet $x(\tau)$ düğümü vardır; burada,

$\tau = 0, 1, \dots, p$ değerlerini almaktadır. G nin (x, y) okuna karşılık $G(p)$ nin $|x(\tau), y(\tau + a(x, y))|$ oku vardır; burada, $0 \leq \tau \leq p - a(x, y)$ değerleri arasında bulunmaktadır. Ayrıca x düğümünde bekleyen akımı göstermek için $|x(\tau), x(\tau + 1)|$ okları ilave edilmektedir; burada $0 \leq \tau \leq p - 1$ değerleri arasındadır. (x, y) okunun karşılığı olan $|x(\tau), y(\tau + a(x, y))|$ okunun kapasitesi $k(x, y)$ olarak alınmaktadır. G nin s ve t sine karşılık $G(p)$ de $s(0), s(1), \dots, s(p)$ ve $t(0), t(1), \dots, t(p)$ ler, kaynak ve hedefler olarak alınmaktadır. Böylece, G de s den t ye giden p süreli dinamik akım için olanı kısıtlamalar, $G(p)$ de s lerden t lere giden statik akım için de geçerli olmaktadır.

Şekil (2.13) de, şekil (2.7) de verilen ağın 5 süreli dinamik hali ve yine aynı ağın üzerinde bulunan ve (2.8)-(2.12) şekilleriyle gösterilen, statik akım görülmektedir.



Şekil (2.13)

Maks-akım min-kesim teoremi kullanılarak, şekil (2.13) deki akımın maksimal olduğunu kontrol etmek çok kolay olmaktadır. $C = (X, \bar{X})$ minimum kesiminde X , aşağıdaki düğümlerden oluşmaktadır.

$s(\tau)$	$\tau = 0, 1, 2, 3, 4, 5$
$x(\tau)$	$\tau = 3, 4, 5$
$y(\tau)$	$\tau = 3, 4, 5$

Daha evvelden de bilindiği gibi $C = 8$ dir.

Herhangi bir ağ yukarıda açıklanan şekilde geliştirilerek maksimal dinamik akım problemini, her zaman için bir maksimal statik akım problemi gibi çözmek mümkün olmaktadır.

2.2.3. Out-Of-Kilter Algoritması

Out-of-kilter algoritması, bir ağdaki minimum maliyetli akımların hesaplanması için olan metodların en genelidir. Bu algoritma Ford ve Fulkerson isimli iki araştırmacının eseri-
dir. Aşağıda, hedef fonksiyonlu ve kısıtlamalı bir problem ele alınarak *Out-of-kilter algoritması* ile neler yapılabileceği açıklanacaktır. Buradaki problemde, akımların sirkülasyon şeklinde oldukları kabul edilmektedir. Sirkülasyonlar, kaynak ve hedeflere bağlı olmayan ve okların alt ve üst sınır şartlarını sağlayan akımlardır.

Ele alınan problemin formülasyonu şöyle olmaktadır.

$$\min \sum_A a(x, y) f(x, y) \quad (2.23)$$

$$f(x, N) - f(N, x) = 0 \quad x \in N \quad (2.24)$$

$$l(x, y) \leq f(x, y) \leq k(x, y) \quad (x, y) \in N \quad (2.25)$$

Burada, $l(x, y)$, (x, y) okunun alt limitini $k(x, y)$ ise aynı okun üst limitini göstermektedir. Üst limit olarak okun kapasitesi alınmaktadır. Problemdeki bütün değerlerin tam sayı oldukları kabul edilmektedir.

Düğüm sayıları veya *düğüm fiyatları* olarak π fiyat-
lama vektöründe işin içine sokarak (2.26) denklemi kurul-
maktadır.

$$\bar{a}(x, y) = a(x, y) + \pi(x) - \pi(y) \quad (2.26)$$

Verilen bir f sirkülasyonu ve π için (x, y) oku-
nun aşağıdaki halleri ortaya çıkmaktadır.

(α)	$\bar{a}(x, y) > 0,$	$f(x, y) = l(x, y)$
(β)	$\bar{a}(x, y) = 0,$	$l(x, y) \leq f(x, y) \leq k(x, y)$
(γ)	$\bar{a}(x, y) \leq 0,$	$f(x, y) = k(x, y)$
(α_1)	$\bar{a}(x, y) > 0,$	$f(x, y) < l(x, y)$
(β_1)	$\bar{a}(x, y) = 0,$	$f(x, y) < l(x, y)$
(γ_1)	$\bar{a}(x, y) < 0,$	$f(x, y) < k(x, y)$
(α_2)	$\bar{a}(x, y) > 0,$	$f(x, y) > l(x, y)$
(β_2)	$\bar{a}(x, y) = 0,$	$f(x, y) > k(x, y)$
(γ_2)	$\bar{a}(x, y) < 0,$	$f(x, y) > k(x, y)$

Bir $\bar{a}(x, y)$ oku, α, β, γ hallerinin bir tane-
sinin içinde ise, bu ok *kilter içindedir* (in kilter), ak-
si takdirde *kilter dışındadır* (out of kilter) demek-
tedir.

Algoritmada bir kilter dışı ok ele alınmakta ve bu kil-
ter içine sokulmaya çalışılmaktadır. Bu iş yapılırken kil-
ter içinde ki oklar yine kilter içinde kalmakta, ancak kilter
dışındaki oklar kiltere yaklaşmakta veya durumlarını koru-
maktadırlar. Algoritmanın neye yaradığını ve neler yaptığı-
nı bir teorem şeklinde vermek faydalı olmaktadır.

Teorem : Out-of-kilter algoritması, (2.23), (2.24),
(2.25) ile ifade edilen problemi, etiketleme işleminin son-
lu sayıda uygulanmasıyla çözmekte veya mümkün sirkülasyonların
var olmadığı sonucunu vererek çözümü durdurmaktadır. He-
sap süresince okların kilter sayıları monoton olmakta ve
artmamaktadır. Ayrıca algoritmaya mümkün bir sirkülasyonla

başlandıđı takdirde, her etiketlemede en az bir okun kilter sayısı azalmaktadır.

Problemin formülasyonundan da anlaşıyacağı gibi out-of-kilter algoritması, lineer programlama problemlerinin çözümünde kullanılmaktadır.

Lineer programlamada en fazla kullanılan Dantzig'in Simpleks algoritmasıyla Ford ve Fulkerson'un Out-of-kilter algoritmasının arasındaki farkları belirterek kısa bir karşılaştırılmalarının yapılması faydalı olacaktır. Simpleks metodunun iki safhası vardır. Birinci safhada mümkün bir sirkülasyonun araştırması yapılmakta, ikinci safhada da optimal sirkülasyonun bulunmasına çalışılmaktadır. Her iki safhada da temel çözümlerle çalışılmaktadır, halbuki out-of-kilter metodunda temel çözümün hiçbir rolü yoktur. Out-of-kilter algoritmasında yukarıda adı geçen iki safha birleştirilmiştir.

İki metod arasındaki en büyük fark, Simpleks metodunda kilter sayılarının monoton olmayışındır. Yeni Simpleks metoduyla yapılan hesaplamaların herhangi bir adımında kilter içinde bulunan oklar, daha sonraki adımlarda kilter dışına çıkabilmektedirler. Bilindiđi gibi bu iş, out-of-kilter metodunda olmamaktadır.

B Ö L Ü M III

3. BOŞ ÜNİTELERİN DAĞITIMI İÇİN ÖNERİLEN METOD

3.1. GENEL

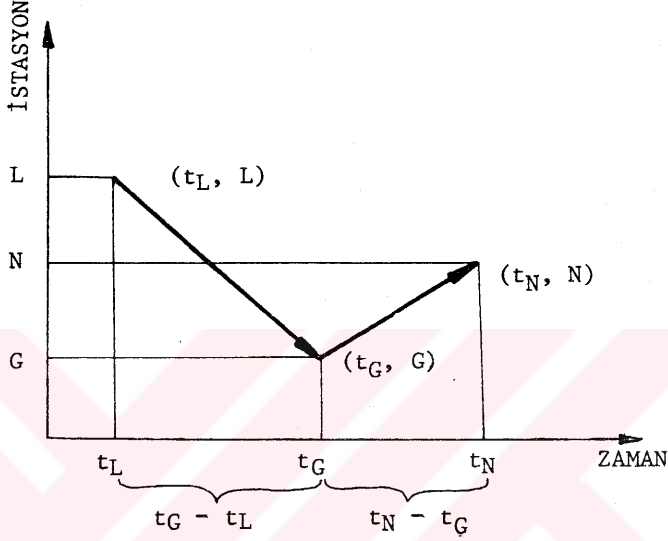
Boş ünitelerin dağıtımında, taşın problemi algoritmasının nasıl kullanıldığı Bölüm II de bir örnek üzerinde incelenmişti. Ayrıca, lineer programlama problemlerinin çözümü için simpleks algoritmasının varlığından da söz edilmişti. Burada geliştirilen modelde, yukarıda bahsedilen klasik çözüm metodlarının yerine ağlarda akım teorisi kullanılmaktadır.

Burada kullanılan algoritmanın en büyük özelliklerinden birisi, arz ve talep istasyonları arasındaki ara durakların da hesaba katılmasıdır. Böylelikle ünitelerin katedecekleri yol kesinlikle belli olmaktadır.

3.2. ÜNİTELERİN HAREKETLERİNİN TANIMI

Ünitelerin hareketleri yersel ve zamansal olmaktadır. Ünitelerin meydana getirdiği, hareket eden bir bütüne katar adı verildiği bilinmektedir. Bir katar L istasyonundan t_L zamanında hareket edip, G istasyonuna t_G zamanında ve N istasyonuna da t_N zamanında vardığında, bu katarın bir kısmını oluşturan boş ünitelerin hareketleri de aynı yol üzerinde gerçekleşmektedir. G istasyonundan N istasyonuna gitmek için gerekli zaman $t_N - t_G$ olmaktadır. Bu hareketler aşağıda anlatılan şekilde diyagram haline getirilmektedir. Bir x, y koordinat takımı alındığında, x ekseninde zaman, y ekseninde de istasyonlar gösterilmektedir ve bu eksenlere zaman ve istasyon eksenleri adları verilmektedir.

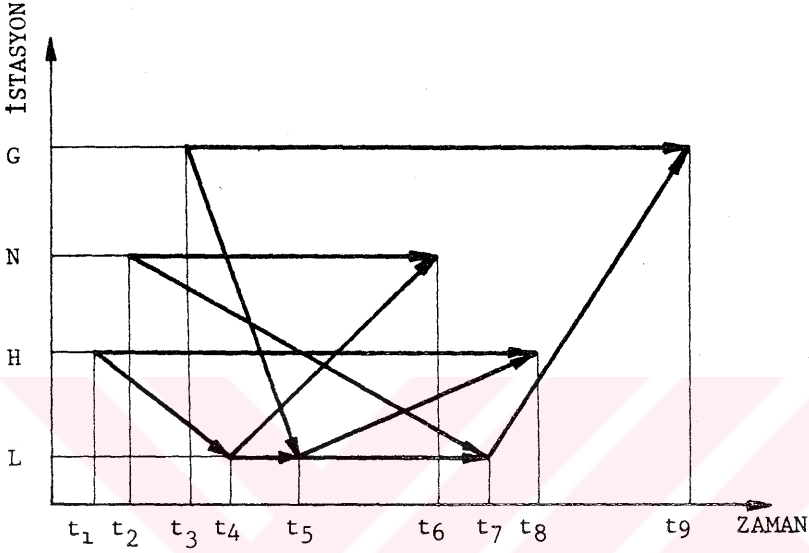
L istasyonundan N istasyonuna gidiş iki ok ile gösterilmektedir. Birinci ok hareketin ilk kısmını yani L ile G arasını, ikinci ok ise hareketin ikinci kısmını yani G ile N arasını göstermektedir. Birinci okun başlangıç koordinatları (t_L, L) , bitiş koordinatları (t_G, G) ; ikinci okun ise başlangıç koordinatları (t_G, G) , bitiş koordinatları (t_N, N) dir. Bu hareketin diyagramı şekil (3.1) de gösterilmekte ve bu çeşit diyagramlara, yer-zaman diyagramı adı verilmektedir.



Şekil (3.1)

Her katarın için de bulunan boş ünitelerin hareketleri yukarıda açıklandığı gibi bir diyagram ile gösterilebilmektedir. Ünitelerin hareketlerini gösteren okların başlangıç ve bitiş noktaları arasındaki yatay uzaklık, bir ünitenin o iki nokta arasını katetmesi için gerekli olan zamanı göstermektedir.

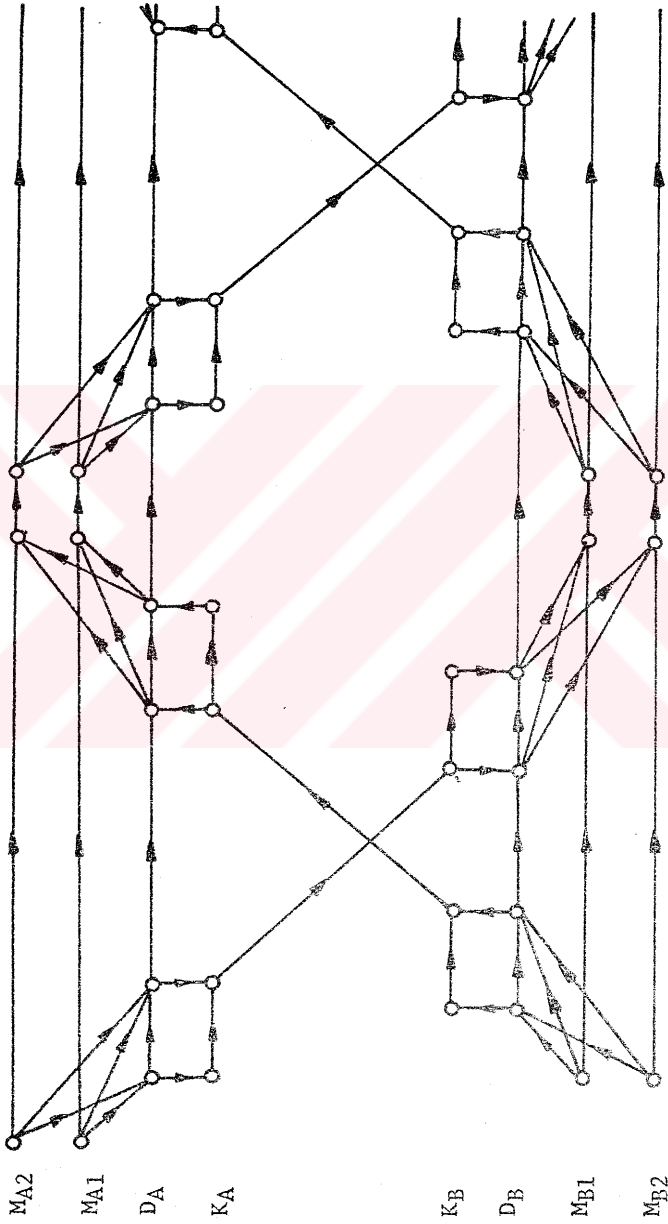
Şekil (3.2) de 4 istasyona uğrayan 3 katarın yaptıkları hareketler görülmektedir. Diyagramdan da kolayca görüleceği gibi, M istasyonundan G istasyonuna giden bir katar yoktur. Ancak, boş üniteleri M den G ye yollamak aşağıda açıklandığı gibi mümkün olmaktadır. Boş üniteler, ilk önce M den L ye t_4 zamanında gelmekte, burada G den t_5 ve N den t_7 zamanlarında gelen katarları beklemekte ve t_7 zamanında G ye hareket eden katarla katılıp t_9 zamanında G ye varmaktadır. t_4 ile t_7 zamanları arasında üniteler L istasyonunda beklemekte veya diğer bir deyimle zamansal bir hareket yapılmaktadır. Şekil (3.2) de bu zamansal hareketler de oklar ile gösterilmektedir. Böylelikle oklar fiziksel ve zamansal olarak ikiye ayrılabilir.



Şekil (3.2)

Örneğin M den G ye giden bir ünitenin hareketi iki fiziksel ve iki zamansal oktan meydana gelmektedir.

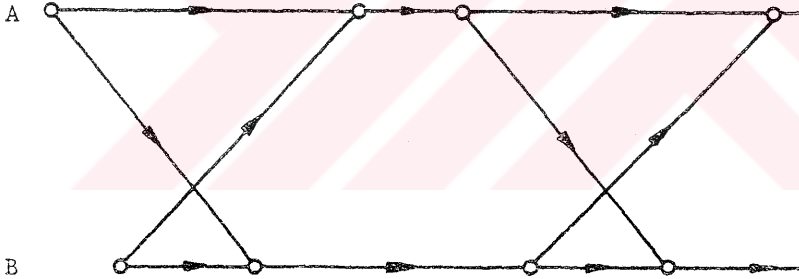
Ünitelerin konteynır olması halinde, iki istasyon arasında işleyen konteynır katarlarının ve konteynırların hareketleri şekil (3.3) görülmektedir. Bu diyagramdaki hareketlerin veya bu hareketleri temsil eden okların açıklanması şu şekilde yapılmaktadır: A istasyonu ele alındığında, bu istasyonun D_A depo sahası, burada duran katarın K_A platformu ve bu istasyonla iş yapan M_{A1} ve M_{A2} gibi müşterileri vardır. M_{A1} veya M_{A2} ile D_A arasındaki oklar, konteynırların müşterilerden depoya veya depodan müşterilere gidip gelmelerini; D_A ile K_A arasındaki oklar konteynırların katara yüklenmelerini veya kataradan boşaltılmalarını göstermektedir ve bunlar fiziksel oklardır. M_{A1} , M_{A2} , D_A , K_A boyunca olan yatay oklar, konteynırların buralarda bekledikleri süreleri göstermektedir, yani bunlar zamansal oklardır. K_A ile K_B arasındaki oklar ise fiziksel olup



Şekil (3.3)

bu istasyonlar arasında işleyen katarların ve dolayısıyla ünitelerin hareketlerini göstermektedir.

Konteynırların hareketlerini gösteren bu diyagram bakıma giden vagonların hareketleri için de geçerli olmaktadır, fakat bu durumda $D_A - K_A$ okları diyagramdan tamamen çıkarılmaktadır. Yine konteynırlar gözönüne alındığında şekil (3.3) ün her hareketi gösterdiği iddiasında bulunulmamaktadır. $D_A - K_A$ okları düşeydir, yani yükleme ve boşaltma süreleri ihmal edilmektedir. Ayrıca müşterilerle depo sahası arasındaki gidiş gelişlerin hepsi toplanarak birer ok ile gösterilmiştir. Bu gidiş gelişler karayolu çekicileri ile yapıldıklarından ve burada çok sayıda düğüm bulunabileceğinden M, D ve K düğümleri bir tek düğüm halinde ele alınacaktır. Böylelikle, şekil (3.3) ün indirgenmiş hali şekil (3.4) de görülmektedir.

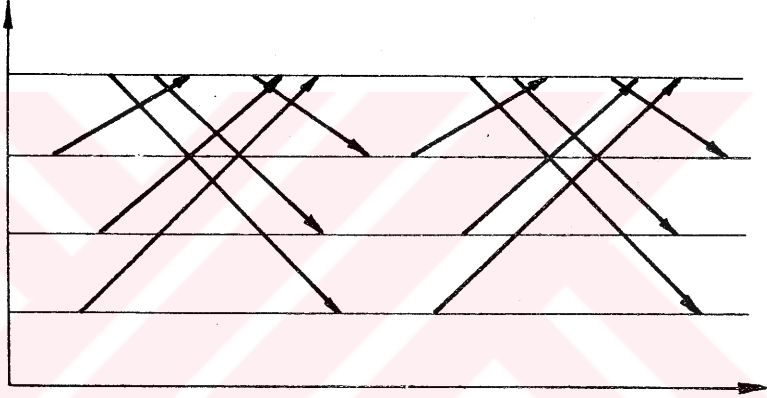


Şekil (3.4)

3.3. TARİFELİ KATARLARIN İŞLEMELERİ HALİNDE YENİ BİR DİYAGRAM ŞEKLİ

Şekil (3.5) de, 4 istasyon arasında işleyen tarifeli katarların yaptıkları hareketler görülmektedir.

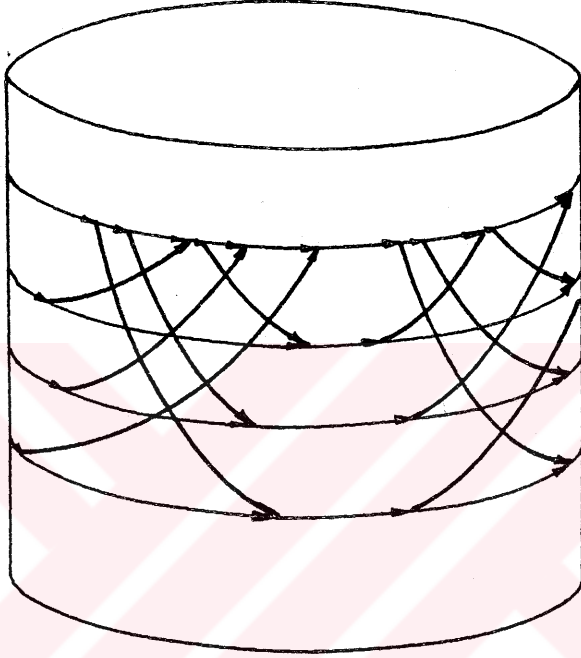
Aşağıdaki şekilde katarların iki günlük hareketleri görülmektedir. Katarların bir haftalık hareketlerini göstermek için bu şeklin daha büyüğünün çizilmesi gerekmektedir.



Şekil (3.5)

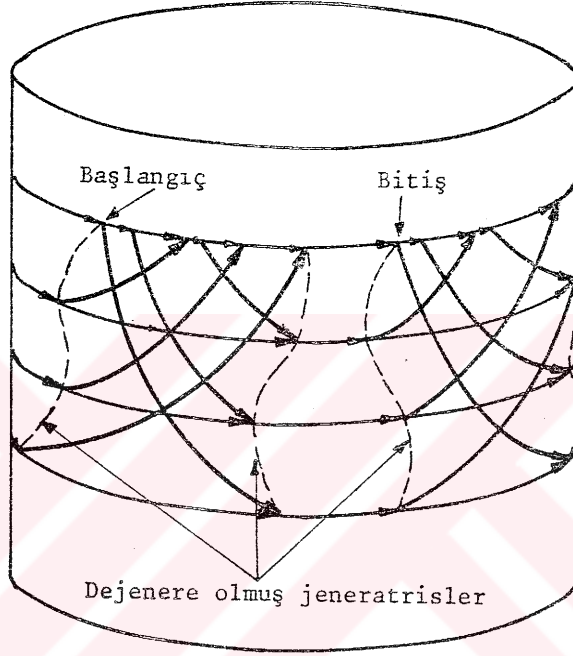
Haftada 5 çalışma gününün olması halinde, bir aylık diyagram burada çizilenin 10 katı kadar büyümektedir. Bu diyagramda tarifeli katarların hareketlerinin başlangıcı diyagramın başlanış noktasıyla belirlenmektedir.

Bu diyagramın yerine, katarların ve dolayısıyla ünitelerin hareketleri bir silindir üzerine çizilmekte ve silindirik bir diyagram elde edilmektedir. Bu silindirik diyagramlar haftalık veya aylık olabilmektedir. Dağıtım süresi bir hafta ise silindirik diyagram haftalık ve benzer olarak süre bir ay ise diyagram aylık olmaktadır. Bu silindirik diyagramda haftanın son gününün bitimi, haftanın ilk gününün başlangıcıyla birleşmektedir. Yeni diyagram Şekil (3.6) da görülmektedir.



Şekil (3.6)

Silindirik diyagramın herhangi bir jeneratrisinin yakınlardaki düğümler bir diyagram süresinin başlangıcını tayin edebilmektedir. Böylelikle dağıtım haftanın birinci günü başlanmayıp da üçüncü günü başlansa bile yeni tam bir diyagram süresince dağıtım yapılabilir. Bu durumda başlangıç haftanın üçüncü günü, bitiş de gelecek haftanın ikinci günü olmaktadır. Silindirin jeneratrisleri şekil (3.7) de görüleceği gibi dejenere olabiliyor birer eğri haline gelebilmektedir. Ayrıca pratikte ünitelerin hareketlerinin silindir üzerine çizilmesi ve bu silindirin üzerine şeffaf bir kılıf geçirilip arz ve talep miktarlarında bu şeffaf kılıfın üzerine yazılması düşünülmektedir.



Şekil (3.7)

3.4. OKLAR ARASINDAKİ BAĞINTILAR

Yer-zaman diyagramının özelliklerinden birisini oklar arasındaki bağıntılar oluşturmaktadır. Bir istasyona bir katarın vardığı veya ortadan kalktığı zamanlarda, yani bir olayın olduğu zamanlarda, bir ok diğer oklarla birleşmekte veya onlardan ayrılmaktadır. Örneğin şekil (3.2) de, L istasyonuna, G den gelip M ye giden katarın varması ve buradan tekrar kalkması bir olaydır. Bu olayın ve zaman olduğunu belirleyen bir de t_5 zamanı vardır. Olayların hangisinin daha önce, hangisinin daha sonra olduğunu yer-zaman diyagramının okları belirtmektedir. Bu diyagramda düğümler olayları göstermektedir. Böylece her ok iki düğümü birbirine

bağlanmaktadır. Her düğümün yanına yerini ve zamanını belirten koordinatları yazmak gereksiz olmaktadır, çünkü bunların yerine düğümlere birer numara verilmektedir. Ayrıca, bu numaraların hangi düğümleri gösterdiklerini belirleyen bir liste yapmak da mümkündür. Düğümler numaralandıktan sonra ünitelerin hareketlerini gösteren okların üzerine ünitelerin miktarları, yani akımlar, yazılmakta ve böylece bir ağ elde edilmektedir.

3.5. AĞIN BAŞLICA ÖZELLİKLERİ

Yukarıda anlatılan şekilde elde edilen ağın başlıca üç özelliği vardır :

1- Bir düğümler cümlesi aynı istasyonu gösterdiği takdirde, bu cümlelerin her düğümü bir diğerine bitişiktir, yani bu cümlelerin düğümlerinden birinden diğerine gitmek için, cümlelerin düğümlerini bağlayan okları takip etmek yeterli olmakta, cümlelerin dışındaki düğüm ve oklara ihtiyaç duyulmamaktadır. Bu özellik katarın durduğu istasyonların meydana getirdiği düğümler cümlesi için de geçerli olmaktadır. Nitekim, olaylar istasyonlardan çok katarlar tarafından belirlenmektedir.

2- Her düğüme en fazla iki ok gelebilmekte ve iki ok gidebilmektedir. Herhangi bir düğüme gelen veya giden bu dört oktan ikisi, bunlardan birisi gelen, diğeri giden ok olmak üzere; fiziksel, diğeri ikisi ise zamansal olmaktadır. Zamansal okların da birisi gelen, diğeri giden ok şeklindedir. Buradan anlaşılacağı gibi bir düğüm etrafındaki okların ikisi katarın seyrini, diğeri ikisi ise istasyonda beklemesini göstermektedir.

3- Düğümler numaralanırken, önceki düğümlere küçük numaralar, sonraki düğümlere de büyük numaralar verilmektedir. Bu şekilde numaralama yapılarak düğümler sıralanmaktadır. Böylece, ağdaki küçük numaralı bir düğümden hareket edip tekrar aynı düğüme varmak imkansız olmaktadır. Bu özellikten dolayı ağda yönlendirilmiş halkalar veya devreler olmamakta, yani ağ devresiz olmaktadır. Böylelikle, ağın herhangi bir kısmında bir boş üniteler hareketi (akım) var olduğunda, bu üniteler hiç bir zaman bir önceki düğüme dönmemekte, bunun yerine bir sonraki düğüme gitmektedirler.

3.6. AKIM KURALLARI

Ağın okları üzerinden geçen ünite akımlarının bazı kurallara uymaları gerekmektedir. Şimdi bu kurallar verilmekte ve herbiri teker teker incelenmektedir.

1- *Akım bir yerden geliyor olmalıdır.* : Ünitelerin sisteme herhangi bir şekilde girmeleri gerekmektedir, aksi takdirde herhangi bir akımın oluşmasına imkan yoktur. Ünitelerin sisteme girişleri ağın başında veya ortalarında olmaktadır. Ağın başlangıcında bazı boş üniteler vardır ve bunlar ilerdeki hareketleri oluşturacaklardır. Bu ünitelerin miktarları, bulundukları istasyonların ilk düğümlerinin yanlarına yazılmaktadır. Boş ünitelerin, ağın ortasında bir yerde meydana çıkmaları halinde ise, meydana çıktıkları istasyonun o andaki ilk düğümünün yanına miktarları yazılmaktadır. Boş ünitelerin ağın ortasında bir yerde çıkmaları şu şekilde olmaktadır: müşteriler ellerinde bulunan üniteleri boşaltıp sisteme yollamakta veya başka bir hizmet sahasından boş üniteler gelmektedir. Ağın ortasında meydana çıkan bu yeni boş üniteler de, ağın boş ünite ihtiyacı olan düğümlerine gitmekte ve ünite akımlarını meydana getirmektedirler.

2- *Akım bir yere gidiyor olmalıdır.* : Bu kural da birinci kural gibidir, yani sistemdeki ünitelere, ya ağın bir yerinde ihtiyaç hasıl olmakta ve bu üniteler oraya gitmektedir, ya da sonunda kullanılmamış olarak kalmaktadır. Ağın bir silindir üzerine çizilmesi halinde, hesap yapılacağı zaman silindirin bir jeneratrisinin yakınlarındaki düğümlerin ağın başlangıcını, diğer bir jeneratrisinin yakınlarındaki düğümlerin de ağın bitişini göstermek üzere saptanmaları gerekmektedir. Birinci kuralda olduğu gibi, ağın ünitelere ihtiyacı olan düğümlerinin yanlarına ihtiyaç miktarları yazılmaktadır. Talep edilen üniteler bu düğümlere daha önceki düğümlerden yollanmaktadır. Düğümlerin yanına yazılan ünite miktarları arz veya talebi göstermektedir, yani bir düğümden 10 ünite arz ediliyor ve 7 ünitenin de talep ediliyor ise bu düğüm $10 - 7 = 3$ ünite arz eden bir düğüm olmaktadır ve bunun yanına sadece 3 yazılmaktadır. Böylece bir düğümden, ya fazla boş ünite vardır, yani bu düğüm bir kaynaktır, veya boş üniteye ihtiyaç vardır, yani bu düğüm bir hedeftir. Ağ boyunca kullanılmayan boş üniteler ise ağın son düğümlerine yollanmakta ve orada elde kalmış fazla üniteler durumuna gelmektedir.

Hiç bir ilk düğümden önce zamansal okun bulunmadığı gibi; hiç bir son düğümden sonra da zamansal ok yoktur.

3- *Akım korunmuş olmalıdır.* : Bu kural herhangi bir düğüm için *akımın korunması prensibini* belirtmektedir. Bir düğüme gelen akımın, giden akıma eşit olması gerekmektedir. Bir istasyona gelen boş üniteler ile orada boşalan ünitelerin toplamı, aynı istasyondan giden boş üniteler ile orada ihtiyaç duyulan boş ünitelerin toplamına eşit olmaktadır.

4- *Akım okların gösterdiği yönde olmalıdır.* : Bu kural da aşikârdır, aksi takdirde ünitelerin zamana göre geriye gidiyor olmaları gerekmektedir. Bu da bilindiği gibi imkânsızdır.

Akım kuralları böylece saptandıktan sonra, bir demiryolu ağında dağıtılması gerekli boş ünitelerin bulunduğu kabul edilmektedir. Burada *tek cins üniteler* kullanılmaktadır, yani ünitelerin herhangi birisi bir diğeri yerine kullanılabilir. Boş ünitelerin hareketleri sistemde bir çok şekilde olabilmektedir, esas zorluk bu ünitelerin hareketlerinin içinden en uygun olanlarının seçilmesidir. İşte bu seçim boş ünitelerin dağıtım problemi'nin esasını oluşturmaktadır. Bu uygun dağıtımı elde etmek için de bir algoritmaya ihtiyaç vardır.

3.7. BOŞ ÜNİTELERİN DAĞITIMI İÇİN BİR ALGORİTMA

3.7.1. Genel

Bu kısma kadar, boş ünite hareketlerinin oluşturduğu yer-zaman diyagramının özelliklerinden bazıları incelenmiş ve adı geçen diyagram bir ulaştırma ağı olarak kabul edilmişti. Bu ağın üzerindeki akımların optimizasyonu yapılacağına ve bu işlem de yapılırken bir algoritma kullanılacağına göre, ağın her özelliğini saptamak ve problemi kısıtlamalarıyla belirtmek gerekmektedir.

3.7.2. Ağın ve Problemin Tanımı

Ağın düğümleri sırasıyla 1 den n e kadar numaralanmaktadır. Herhangi iki düğüm i ve j olduğu takdirde, bu iki düğümü bağlayan ok (i, j) olmaktadır. Burada $1 \leq i \leq n$ ve $1 \leq j \leq n$ dir. Her ok fiziksel yani iki istasyonu birbirine bağlayan bir katar, veya zamansal, yani ünitelerin bir istasyondaki bekleme süresi, olabilmektedir.

Ağın her düğümün arz veya talep edilen ünitelerin miktarları bilinmektedir. i düğümünde arz edilen ünitelerin miktarı b_i dir, $b_i < 0$ olduğu takdirde, bu düğümde $-b_i$ kadar ünite talebi var demektir.

Ünitelerin hareketleri x ile gösterilmektedir, bu x bilinmemektedir ve algoritma yardımı ile bulunacaktır. i düğümünden j düğümüne (i, j) oku üzerinden giden ünitelerin sayısı x_{ij} olmaktadır. Ağın her oku için yalnız bir x değişkeni vardır. Ünitelerin en uygun hareketlerini saptama problemi, akım kurallarını sağlayan x ler için uygun bir değer grubunun bulunması işi olmaktadır. En mühim iş ise, daha evvelden de ehemmiyetle belirtildiği gibi mümkün uygun değerler arasından bir seçimin yapılmasıdır.

Her fiziksel (i, j) okunun negatif olmayan bir c_{ij} maliyeti vardır. Bu c_{ij} maliyeti bir ünitenin i düğümünden j düğümüne taşındığında ortaya çıkan taşın maliyetidir. Zamansal okların maliyeti sıfır kabul edilmektedir, yani üniteler istasyonlarda bekledikleri sürece işletmeciye herhangi bir masraf çıkartmamaktadır. Bu durumda c_{ij} taşın maliyetlerinin, ünitelerin istasyonlardaki yükleme, boşaltma ve öteleme maliyetlerini de içerdikleri kabul edilmektedir. c_{ij} nin biriminin parasal olduğu, yapılan kabullerden birisidir. c_{ij} maliyetleri de işin içine girdikten sonra problem, toplam maliyeti minimize eden x lerin değerlerinin bulunması şekline dönüşmektedir. Diğer bir deyimle, akım kurallarına uyan ve diğer mümkün ünite hareketlerinin toplam maliyetlerinden daha büyük bir toplam maliyeti olmayan uygun ünite hareketleri grubunun bulunması bu problemin çözümünü oluşturmaktadır. Bu, ünitelerin bir istasyondan diğerine en ucuz şekilde taşınmalarının araştırılması ve saptanması işidir. Bu kriterden anlaşılacağı gibi, şayet talepleri yerine getirmek için bir ünitenin bir istasyondan diğerine gitmesi icap etmiyorsa, o ünite bulunduğu istasyondan başka bir yere

yollanmamaktadır. Burada hemen şu önemli noktaya değinmekte fayda vardır : fazla ünitelerin, çözümün sonunda ağın bitiş düğümlerine geldikleri daha önceden de söylenmişti. İşte bu bitiş düğümlerindeki üniteler, silindirik çizimle verilen bir ağda, bir sonraki ağın başlangıç düğümlerinde arz üniteleri olarak kullanılmaktadır.

Yukarıda da söylendiği gibi ünitelerin istasyonlarda bekleme maliyetlerinin herhangi bir maliyeti yoktur. Bekleme maliyetleri işin içine girdiğinde burada açıklanan algoritmadan daha değişik bir algoritma gerekmektedir.

3.7.3. Problemin Formülasyonu

x_{ij} değişkenlerinin, aşağıdaki bağıntıların hepsini sağlayan, değerlerinin bulunması problemin çözümü olmaktadır. Problemin hedef fonksiyonu ve kısıtlamalar şöyledir :

$$\text{minimum} \quad \sum_i \sum_j c_{ij} x_{ij} \quad (3.1)$$

$$\sum_j x_{ij} - \sum_k x_{ki} \leq b_i \quad (3.2)$$

Burada i son düğümdür.

$$\sum_j x_{ij} - \sum_k x_{ki} = b_i \quad (3.3)$$

Burada i son düğümden başka bir düğümdür.

$$x_{ij} \geq 0 \quad (3.4)$$

ve tam sayıdır.

(3.1), (3.2), (3.3) ve (3.4) bağıntılarının fiziksel anlamlarının açıklanması yararlı olacaktır. Birinci ifade, bütün ünite hareketlerinin toplam maliyetinin minimize edilmesi icap ettiğini göstermektedir. Bu ifade de (i, j) oku, fiziksel olduğu takdirde $c_{ij} > 0$, zamansal olduğu takdirde ise $c_{ij} = 0$ olmaktadır.

İkinci ve üçüncü bağıntılar, akımın korunması kanununu ifade etmektedir. İkinci bağıntı, tahsis edilmemiş, elde bulunan boş ünitelerin bir istasyonun son düğüm noktasına gideceğini göstermektedir. Daha önce de belirtildiği gibi, elde bulunan fazla üniteler hiç bir zaman gereksiz olarak başka istasyonlara yollanmamakta, bulundukları istasyonda bekletilmektedir, dolayısıyla bunlar ağına adı geçen istasyonunun son düğümüne gitmektedir. Üçüncü bağıntı, son düğümlerin dışındaki kalan düğümlerden çıkan net ünite sayısının, bu düğümlere dış kaynaklardan gelen net ünite sayısına eşit olması icap ettiğini ifade etmektedir, yani bir düğüme gelen boş ünite sayısı, bu düğümden giden boş ünite sayısına eşit olmaktadır. Dördüncü bağıntı, akımın okun gösterdiği yönde olması icap ettiğini ifade etmektedir. Bu durum daha evvelden de söylendiği gibi aşıkârdır, aksi halde negatif ünitelerin var olması icap etmektedir, bu da fiziksel olarak imkansız olmaktadır. Ayrıca sonuçlar tam sayı olmalıdır, çünkü kesirli bir ünitenin elde edilmesinin hiç bir pratik yararı olmamaktadır. Zaten kesirli bir boş ünitenin var olması, bunu kullanmak veya herhangi bir yere yollamak imkansızdır. Formülasyonu yapılan problem, kolayca görüldüğü gibi bir lineer programlama problemidir ve lineer programlama teknikleri kullanılarak çözüme gidilmektedir. Burada uygulanan çözüm metodu, minimum maliyetli akım tekniklerinin içine girmektedir ve Ford ve Fulkerson'un Out-of-Kilter algoritmasının bir adaptasyonu yapılmaktadır. Bunun en büyük sebeplerinden birisi, problemin bir ağ - akım problemi şeklinde takdim edilmesi ve bunun için de en uygun çözümü yukarıda adı geçen algoritmanın vermesidir. Burada adaptasyonu yapılarak kullanılan algoritmanın en büyük özelliklerinden birisi de, daha önceden de bilindiği gibi, işe tam sayılarla başladığı takdirde sonuçların da tam sayı olarak elde edilebilmesidir. Boş ünite miktarları veri olarak hiç bir zaman kesirli olmayacaklarına göre, sonuçların da daima tam sayı olarak çıkmaları garantisi vardır.

3.7.4. Algoritmanın Çözüm Şekli

Algoritma problemin çözümüne adım adım yaklaşılmasını sağlamaktadır. Bütün ağ bir seferde incelenmemekte, belli bir anda ağın bir kısmı incelenmektedir. Bu incelenen ağ parçasına ise *ağcık* adı verilmektedir. İncelenen bu ağcıkta iş bitince başka bir ağcığın incelenmesine geçilmektedir. Bu çözümüne geçilen ikinci ağcık, birinci ağcığa bir düğüm ve bu düğümü ilk ağcığa bağlayan bütün oklar eklenerek elde edilmektedir. Bu yeni eklenen düğümden bir önceki bütün düğümler daha evvelden incelenen ağcığın içinde bulunmaktadırlar. Dolayısıyla yeni düğüme gelen herhangi bir akımın geliş yeri, bir evvelki düğümlerden birisi olmaktadır.

Ağın devresiz olması nedeniyle, bir ağcıktan diğer ağcığa geçerek çözümü ilerletmek her zaman için mümkün olmaktadır. Düğümlerin hepsi daha evvelden numaralanmıştır ve en küçük numaralı düğüm ele alınarak ve bu düğümün kendisinin bir ağcık olduğu kabul edilerek çözüme başlanmaktadır. İkinci ağcık, yukarıda açıklanan şekilde birinci düğüme ikinci bir düğüm ve bu iki düğümü birbirine bağlayan oklar eklenerek elde edilmektedir. Bu usule, bütün ağ elde edilinceye kadar devam edilmektedir.

3.7.5. Düğümlerin Sınıflandırılması ve Seçimi

Algoritmanın herhangi bir adımında incelenen ağcığın düğümleri *işaretlenmiş düğümler*, geri kalan bütün düğümler ise *işaretlenmemiş düğümler* olmaktadır. İşaretlenmemiş düğümlerin hepsi çözüm boyunca işaretlenecek düğümlerdir, bunlara *işaretlenebilir düğümler* adı verilmektedir. İşaretlenmemiş bir düğümden önceki bütün düğümler işaretlenmiş durumdadır. Buradan düğümler için şu önemli sonuca varılmaktadır: bir düğümün işaretlenebilir olması için kendisinden evvel bütün düğümlerin işaretlenmiş olmaları gerekmektedir.

Verilen bir ağcıktan hareket ederek diğer bir ağcığı kurmak için işaretlenebilir bir düğüm seçilmekte, bu seçilen düğüm işaretlenmekte, her iki düğüm arasındaki bütün oklar işin içine sokulmakta ve böylece yeni bir ağcık elde edilmektedir. Bir ağcıktan diğerine geçerken, problemin formülasyonunda belirtilen bağıntılardan (3.2), (3.3) bağıntısına dönüşebilmektedir. Bu durumda probleme yeni bir (3.2) kısıtlaması eklenmektedir.

3.7.6. Ağcıkların Özellikleri

Birer düğüm eklenerek birbirlerinden türetilen ağcıkların herbiri problemin bir kısmını içermektedir. Bir ağcık kendinden önce bütün ağcıkları içerdigi gibi, kendinden sonra olan bütün ağcıklar tarafından da içerilmektedir. Herhangi bir ağcıktan sonra olan ağcıkların problemleri, önceki problemlerden daha fazla kısıtlayıcı olmaktadır, çünkü bunlara yeni kısıtlamalar eklenmekte ve var olan kısıtlamalar da olasılıkla (3.2) şeklinden (3.3) şekline dönüşerek daha da kuvvetlenmektedir. (3.1) den (3.4) e kadar olan bağıntıların hepsini sağlayan bir çözüme *optimal çözüm*, (3.2) den (3.4) e kadar olanları sağlayan *mümkün çözüm* ve (3.2) den (3.4) e kadar olanlardan herhangi bir tanesini sağlamayana da *mümkün olmayan çözüm* denmektedir. Buradan şu iki özellik meydana çıkmaktadır.

- 1- Herhangi bir ağcığa ait olan bir problemin optimal çözümü, daha önceki ağcıklara ait problemler için mümkün çözüm olmaktadır.
- 2- Herhangi bir ağcığa ait olan bir problem için mümkün çözüm bulunmadığı takdirde, bu ağcıktan sonraki ağcıklara ait problemler için de mümkün çözüm bulma olanağı olmamaktadır.

Bu özelliklerin ne şekilde kullanıldıklarının açıklanması şöyledir : verilen bir ağcık için bir optimal çözümün elde edilmiş olduğu kabul edilmektedir. Bu ağcıktan hareket ederek daha evvelce izah edildiği yeni bir ağcık kurulmaktadır. Daha önceki çözüm, yeni problem için mümkün çözüm olabilmekte veya olmayabilmektedir; mümkün çözümün olması halinde, bu çözüm yeni problem için optimal çözüm haline getirilmektedir. Daha önceki çözümün yeni problem için mümkün çözüm olmamasının sebebi, yeni eklenen kısıtlamadan dolayıdır ve çözümde bu kısıtlama sağlanamamaktadır. İşte bu durumda, algoritma diğer bağıntıların icap ettirdiklerini bozmadan adım adım yeni kısıtlamanın icap ettirdiklerini yerine getirmeye çalışmaktadır. Kısıtlamanın sağlanamaması halinde problemin bir çözümü yok demektir. Zaten bu durumda yeni bir ağcık kurmak imkansız olmaktadır. Optimal çözüm elde edildiği takdirde ise yeni ağcıklar kurmaya devam edilmekte ve böylece probleme ya bir optimal çözüm bulunmakta, ya da hiç bir çözüm bulunamamaktadır.

3.7.7. Algoritmanın Bir Ağcığı Çözüş Şekli

Herhangi bir ağcığın çözümünün algoritma tarafından nasıl yapıldığının izah edilmesinde yarar vardır. Çözümünün yapılması için uğraşılan ağcığın bazı oklarını algoritma ayırmaktadır. Ağcığın bir düğümünden diğerine yalnız ayrılmış oklar üzerinden gidildiğinde, bir ayrılmış oklar grubu elde edilmektedir.

Daha önceden de bilindiği gibi bu oklar grubundaki yolları takip ederek geriye veya başlangıç noktasına dönmek imkansız olmaktadır. Ağcıkta bulunan ayrılmamış okların hepsi çıkartıldığında, geriye kalan okların oluşturduğu gruba orman adı verilmektedir. Bu orman ise birbirinden ayrı bir çok ağaçlardan oluşmaktadır. Her ağaçta bir düğüm, diğerine yalnız ayrılmış oklar tarafından ve yine yalnız bir yol ile bağlanmaktadır. Böylece, herhangi bir ağaçta n kadar düğüm varsa $n-1$ kadar da ok var demektir. Her ağacın bir düğümü, bu ağacın kökünü oluşturmaktadır. Bir ağacın mutlaka oklara sahip olması gerekmemekte, ağaç bir tek düğümünden yani kökünden de oluşabilmektedir. Orman ve ağacın tarifinden de anlaşıldığı gibi, her düğümünün ait olduğu bir ağaç vardır. Ormanın kendisi de bir tek ağaçtan oluşabilmektedir.

Bir ağcıkta, bir çok ağacın var olması mümkündür, yani ayrılmış okları seçmenin çeşitli yolları vardır. Algoritma ilerledikçe orman yeni kurulan ağcıklarla büyümekte ve yeniden oluşmaktadır. Ancak, bu iş yapılırken şu prensibe uyulmak gerekmektedir: Kök olarak alınacak düğümlerin, verilen ağcıkta bir istasyonun son düğümüne karşı gelmeleri gerekmektedir. Her ağacın bir kökü olabileceğine göre, ormandaki ağaçların sayısı da ancak istasyon sayısı kadar olabilmektedir. Ayrıca, kök olarak alınabilecek düğümler, fazla ünitelerin, yani bir ünite arzının var olduğu düğümler olmalıdır.

Algoritmada, orman ve onun ağaçları çok önemli bir yer tutmaktadır, çünkü boş ünitelerin akımı ancak ayrılmış oklar üzerinden olabilmektedir. Çözümünün herhangi bir adımında, üzerinde çalışılan ağcığın fazla üniteli düğümleri yeni bir ağcığın kökü olma durumundadır. Fazla üniteler bu düğümlerin yanlarına yazılmaktadır ve boş ünitelerin sayıları değişken olmaktadır.

3.7.8. Ağın Düzümlerinin Potansiyelleri

Ağcığın her i düğümünün bir düal değişkeni veya π_i potansiyeli vardır. Bu potansiyele, düğüm fiyatı adı da verilmektedir. Bir i düğümünün potansiyeli, i düğümünün içinde bulunduğu ağacın kökünden düğümün kendisine bir fazla boş ünitenin getirilmesinin marjinal maliyetini göstermektedir. Herhangi bir ağacın kök düğümünün potansiyeli sıfır olarak tarif edilmektedir. Bu ağacın düğümlerinin potansiyelleri aşağıda açıklanan kurala göre hesaplanmaktadır.

i ve j düğümlerinin her ikisinde aynı ağacın düğümleri olmaları, c_{ij} maliyetli (i, j) okunun ayrılmış olması ve π_i veya π_j potansiyellerinden birisinin bilinmesi halinde diğer potansiyel

$$\pi_i + c_{ij} = \pi_j \quad (3.5)$$

bağıntısı kullanılarak bulunmaktadır.

Potansiyellerin bu şekilde hesaplanabilmeleri, ağaçların daha evvelce yapılmış olan tariflerinden çıkarılmaktadır. Bütün kök düğümlerinin potansiyelleri sıfır olduğuna göre ve okların maliyetlerinin de daima pozitif veya sıfır olduklarına göre, (3.5) bağıntısı ile hesaplanan potansiyeller hiç bir zaman negatif değerlere sahip olmamaktadır.

3.7.9. Potansiyellerin Getirdikleri Kısıtlama

Ağaçların oluşması ve oluşan bu ağaçların köklerinin yalnız istasyonların son düğümlerinin olabileceği daha evvelden gösterilmişti. Şimdi probleme yeni bir kısıtlama girmektedir. Ağın, bir (i, j) okuyla bağlanmış herhangi iki i ve j düğümünün potansiyellerinin

$$\pi_i + c_{ij} > \pi_j \quad (3.6)$$

bağıntısını sağlamaları gerekmektedir. (3.6) eşitsizliğinin sağlandığı aşikâr değildir. Bu gibi eşitsizlikler ve kök düğümlerinin potansiyellerinin sıfır olmasını gerektiren kısıtlamalar, bir düal lineer programın kısıtlamalarını meydana

na getirmekte ve bunların, problemin herhangi bir optimal çözümü tarafından sağlanmaları gerekmektedir. Algoritmanın uygulanması sırasında, bu kısıtlamalar sağlanacak şekilde oklar ayrılmakta veya ayrılmış oklar tekrar ayrılmamış yapılmaktadır.

Bir potansiyelin, marjinal maliyet olarak gayet gerçek bir manası vardır. Herhangi bir i düğümü için π_i , bu düğüme, mensup olduğu ağacın kök düğümünde bulunan ünitelerin sayısını bir eksilterek, bir ünitenin getirilmesinin maliyetini göstermektedir. Bu ünite, kök düğümünden i düğümüne gelirken, üzerinden geçtiği ayrılmış okların akımları yeni duruma uygun bir şekilde düzeltilmektedir. Bu geçilen yolun üzerindeki ayrılmış okların hepsinin yönlerinin kök düğümünden i düğümüne doğru olmaları halinde, kök düğümünden i düğümüne bir ünitenin getirilmesi, π_i maliyetli bir ünitenin yollanması demektir. Öte yandan, belli bir ok üzerinden daha önceden yollanması gereken bir ünitenin yollanmaması halinde, akımın düzeltilmesi negatif yönde olmaktadır. Bazı ünitelerin, i düğümünden yanındaki j düğümüne, (i, j) ayrılmış oku üzerinden gidecekleri kabul edilmekte fakat i den j ye gitmesi icap eden bir ünite i de tutularak burada bir fazla ünite elde edilmektedir. Böylelikle i deki ünite sayısı bir artmakta, j deki ünite sayısı ise bir eksilmektedir. Bu iş j den i ye bir ünitenin ters yönde gönderilerek doğru yönde gönderileni iptal etmesi şeklinde olmaktadır. Tabii, okların üzerinde iptal edilecek en az bir ünitenin bulunması halinde bu anlatılanlar bir mana ifade etmektedir. Bu durumda, i düğümüne kök düğümünden yollanan ek bir ünitenin maliyeti π_i olmaktadır. Bu maliyet, kök düğümünden i düğümüne giderken yol üzerindeki okların akımlarının, bir ünite artanlarının maliyetleri toplanarak, bir ünite azalanlarının maliyetleri de çıkarılarak elde edilmektedir.

3.7.10. Bir Demiryolu Ağındaki Boş Ünitelerin Dağıtımında Kullanılan Algoritmanın Hesap Adımları

Bu kısma kadar, algoritmada kullanılan elemanların hepsinin özellikleri anlatıldığına göre şimdi sıra algoritmanın adımlarına gelmiştir. Aşağıdaki adımlarla hesap yapılarak boş ünitelerin dağıtımı yapılmaktadır.

BAŞLANGIÇ ADIMI

Kendilerinden önce herhangi bir okun bulunmadığı bütün düğümler işaretlenebilir düğümlerdir. Başlangıçta hiçbir düğüm işaretlenmemiştir.

ADIM 1 :

n_0 , işaretlenmemiş, işaretlenebilir bir düğümdür. Böyle bir düğüm yoksa burada durulmaktadır; bütün problem için optimal bir çözüm bulunmuş demektir. Aksi takdirde $\pi_{n_0} = 0$ konmakta ve n_0 düğümü işaretlenmektedir. n_0 in, bir istasyonun ilk durağı olması halinde ADIM 2 ye geçilmektedir. Aksi takdirde, yerini şimdi n_0 in almış olduğu ve daha önceden bir istasyonun son durağı olan n_1 düğümü ele alınmaktadır. n_1 düğümünün kök düğümü olmaması halinde ADIM 2 ye geçilmektedir. Aksi takdirde (n_1, n_0) oku ayırılmakta ve n_1 de ki fazla akım bu ok üzerinden n_0 a taşınmakta, bu iş bittikten sonra n_1 artık kök düğümü olarak kabul edilmemekte ve ADIM 3 e geçilmektedir.

ADIM 2 :

n_0 düğümünde herhangi bir isteğin yerine getirilmesi istenmediği takdirde n_0 düğümü kök olarak kabul edilmekte ve ADIM 1 e dönlülmektedir. Aksi takdirde ADIM 3 e geçilmektedir.

ADIM 3 :

n_2 düğümü, n_0 düğümü ile aynı ağaçtadır; n_3 düğümü işaretlenmiştir, fakat n_0 ile aynı ağaçta değildir. n_3 ve n_2 düğümlerini birbirine bağlayan (n_3, n_2) okunun marjinal maliyeti $M(n_3, n_2)$ aşağıdaki formülle hesaplanmaktadır.

$$M(n_3, n_2) = \pi_{n_3} + c_{n_3 n_2} - \pi_{n_2} \quad (3.7)$$

$M(i, j)$ lerin arasından minimum değerli olanının seçilmesi gerekmektedir. (i, j) okunun var olmamasından dolayı M in bulunamaması halinde durulmaktadır; problemin çözümü yoktur. Aksi takdirde, n_0 in ağacında bulunan i düğümlerinin π_i leri M kadar arttırılmaktadır ve M değerini veren (n_3, n_2) oku ayrılmaktadır. Bu durumda, n_{n_0} düğümünü içeren ağaç, kökü olan başka bir ağaca bağlanmış n_{n_0} olmaktadır. Şimdi bu genişletilmiş ağacın kökünden, ayrılmış oklar üzerinden, n_0 a aşağıdaki a veya b hallerinden birisi sağlanıncaya kadar akım yollanmaktadır.

a- n_0 da ki talep yerine getirilmiştir.

b- n_0 da ki talep yerine getirilememiştir, fakat artık herhangi bir akım da yollanmamaktadır.

a halinde ADIM 1 e dönlmektedir. b halinin ortaya çıkması ise iki şekilde olmaktadır :

- 1- Kök düğümünden n_0 a gelen akım, yol üzerindeki bir okta akımın iptali nedeniyle sıfıra inmiştir.
- 2- Birinci hal meydana çıkmamıştır, fakat ağacın kökündeki fazla akım, n_0 düğümündeki talepten daha az olduğu için tükenmiştir.

2 halinde, ağacın kökü artık kök sayılmamakta ve ADIM 3 e dönlmektedir. Aksi takdirde, üzerinde ki akımın sıfıra inmiş olduğu ayrılmış ok, ayrılmamış yapılmaktadır. Bu durumda birden fazla okun bulunması halinde, içlerinden bir tanesi keyfi olarak seçilmektedir. n_0 ı içeren ağaç, şimdi iki ağaca bölünmüştür; bir ağaçta n_0 , diğer ağaçta ise kök vardır. Bu durumda ADIM 3 e dönlmektedir.

B Ö L Ü M İ V

4. SAYISAL ÖRNEK VE SONUÇLAR

4.1. SAYISAL ÖRNEK

G, N, M, L şehirleri arasında tarifeli yük katarları işlemektedir. Bu katarlar dolu yük üniteleri taşıdıkları gibi boş yük üniteleri de taşımaktadırlar. Bu katarların ve dolayısıyla boş ünitelerin yaptıkları hareketler silindirik bir diyagram şeklinde çizilmektedir. Bu silindirik diyagramın üzerinde t ve $t - 1$ zamanları arasında kalan diyagram parçasındaki boş üniteler trafiğinin dengelenmesi istenmektedir. Diğer bir deyimle bu diyagram parçasındaki boş ünitelerin dağıtılması gerekmektedir. Silindirik diyagramlardan alınan ve bir düzlem üzerine çizilen diyagram şekil (4.1) de görülmektedir. Bu diyagramlardan da görüleceği gibi G, N, M, L şehirleri arasında 8 katar işlemektedir. Bu katarlar, hareket saatleri esas alınarak sıralandığında, şu hareketleri yapmakta oldukları görülmektedir.

1. katar	$G \rightarrow M \rightarrow L$
2. katar	$M \rightarrow N \rightarrow G$
3. katar	$L \rightarrow M \rightarrow N$
4. katar	$G \rightarrow N \rightarrow L$
5. katar	$M \rightarrow N \rightarrow G$
6. katar	$G \rightarrow M \rightarrow N$
7. katar	$L \rightarrow N \rightarrow G$
8. katar	$N \rightarrow M \rightarrow L$

Burada yapılan sıralamadan 4. katar ele alındığında, bunun G şehrinden hareket edip, N şehrine uğrayıp L şehrine vardığı görülmektedir. Şekil (4.1) de görüldüğü gibi bu 8 katarın sebep olduğu boş ünitelerin hareketleri 24 düğüm ve 36 ok ile gösterilmektedir. Aynı yatay çizgi üzerinde bulunan düğümler aynı istasyona ait olmaktadır. Diyagramdaki 1, 2, 3, 4, 5, 6 düğümleri G şehrindeki istasyona

7, 8, 9, 10, 11, 12 düğümleri N şehrindeki istasyona, 13, 14, 15, 16, 17, 18 düğümleri M şehrindeki istasyona ve 19, 20, 21, 22, 23, 24 düğümleri de L şehrindeki istasyona aittir. Düğümler birer daire ile gösterilmekte ve numaraları bu dairenin içine yazılmaktadır. Bu dairelerin dış kısımlarına ise arz veya talep edilen ünite sayıları yazılmaktadır, arzedilen üniteler pozitif sayılarla, talep edilen üniteler de negatif sayılarla gösterilmektedir. Bunlardan başka düğümlerin yanına potansiyelleri de saptandıkça yazılmaktadır. Bu potansiyelleri ünite sayılarından ayırt edebilmek için altları çizilmektedir. Böylelikle bir düğümün yanındaki iki sayı bulunmaktadır, bunlardan birincisi altı çizili potansiyel diğeri ise arz veya talebi gösteren sayı olmaktadır. İşaretlenmiş her düğümün bir potansiyeli vardır, işaretlenmemiş düğümlerin yanında yalnız ünite sayısı yazılıdır. Düğümlerin yanında olduğu gibi, okların üzerinde de iki sayı yazılıdır, bunlardan altı çizili olan taşın maliyetini, diğeri ise taşınan boş ünite sayısını yani akımı göstermektedir. Algoritmanın yapacağı hesaplardan sonra, kök olan düğümler daire yerine kare ile ve ayrılmış oklar ayrılmamışlara göre daha kalın çizgilerle gösterilmektedir.

Algoritmaya göre boş ünitelerin dağıtılması aşağıda anlatılan şekilde olmaktadır.

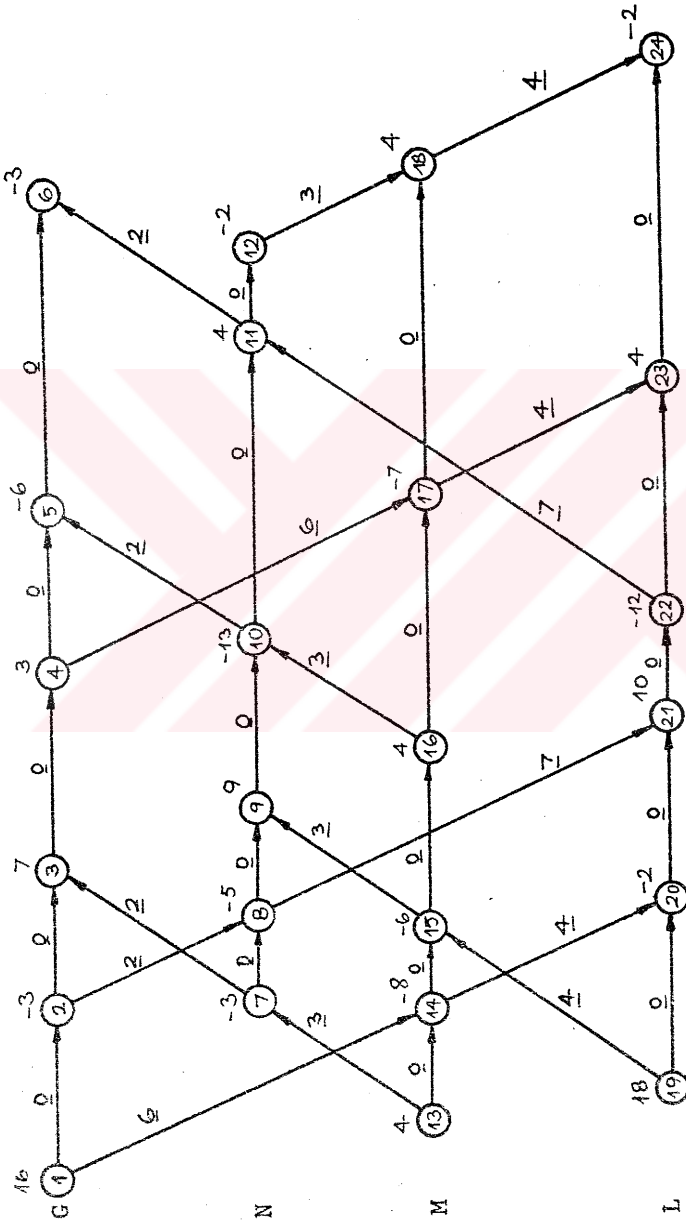
1. *Safha* : 1, 13 ve 19 düğümleri işaretlenebilir düğümlerdir, bunlardan 1 düğümü sıfır potansiyeli verilerek işaretlenmekte ve algoritmaya göre bir kök düğümü olmaktadır. Bu safha Şekil (4.2) de görülmektedir.
2. *Safha* : 2, 13 ve 19 düğümleri sırasıyla ele alınmakta ve ağın durumu Şekil (4.3) de görülmektedir.
3. *Safha* : 2, 13 ve 19 düğümleri işaretlendikten sonra, sıra 7 ve 14 düğümlerine gelmiştir. İlk önce 7 daha sonra da 14 işaretlenmektedir. Bu safhayı Şekil (4.4) de görmek mümkündür. 7 de ki talep 13 den, 14 de ki talep ise sırasıyla 13 ve 2 den ünite getirilerek karşılanmaktadır.
4. *Safha* : Bu safhada 3, 8, 15 ve 20 düğümleri işaretlenmekte ve bu düğümlere ünite yollanmaktadır. Bu düğümlerin işi bittikten sonra ağ Şekil (4.5) de ki gibidir.
5. *Safha* : 4, 9, 16 ve 21 düğümleri ele alındığında, bunların her birinin işaretlendikten sonra kök düğümü oldukları Şekil (4.6) da görülmektedir.

6. *Safha* : 10, 17 ve 22 düğümleri sırasıyla incelenmekte ve işaretlenmektedir. 10 düğümünde ki talep 3 adımda karşılanmaktadır, buraya 9, 4 ve 21 düğümlerinden ünite getirilmektedir. 17 düğümündeki talep ise 2 adım da karşılanmaktadır ve üniteler 16 ve 21 düğümlerinden getirilmektedir. 22 düğümü, bir kök düğümü haline gelmektedir. Bu safhanın sonunda ağın durumu Şekil (4.7) de görülmektedir.

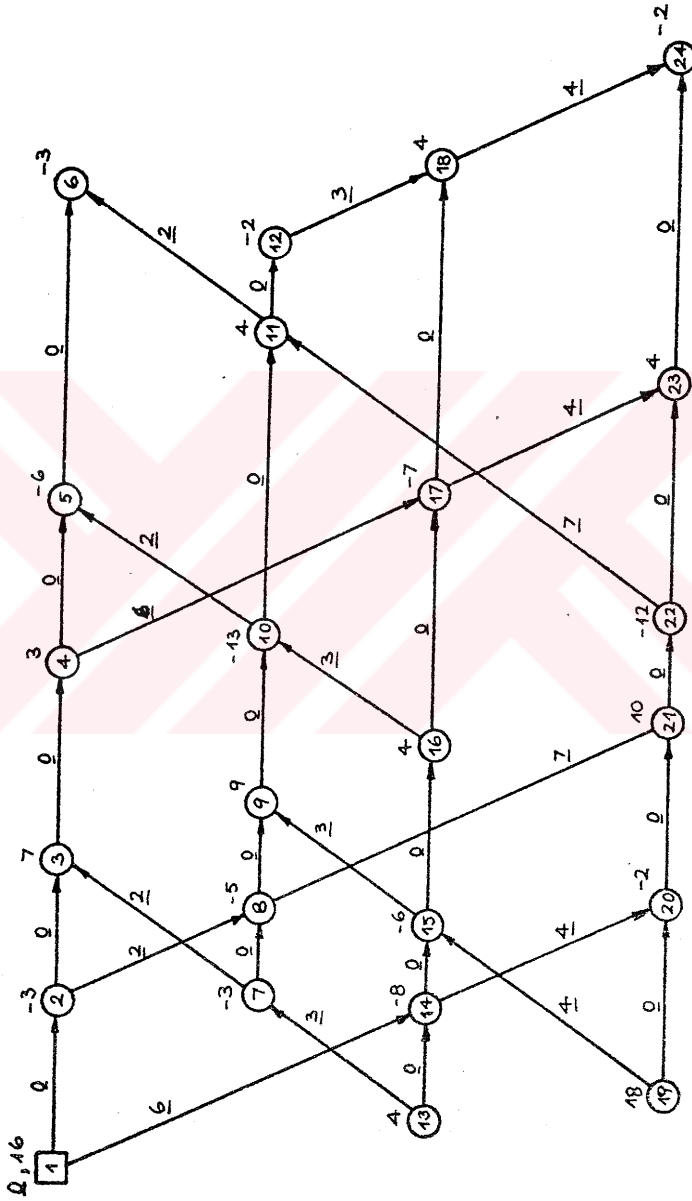
7. *Safha* : 5, 11 ve 23 düğümleri incelenip işaretlendiklerinde, hepsinin kök düğümü oldukları Şekil (4.8) de görülmektedir.

8. *Safha* : İşaretlenmemiş olarak yalnız 6, 12, 18 ve 24 düğümleri kalmıştır; bunlar da işaretlendiğinde hesaplar sona ermektedir. Ağın son durumu Şekil (4.9) da görülmektedir.

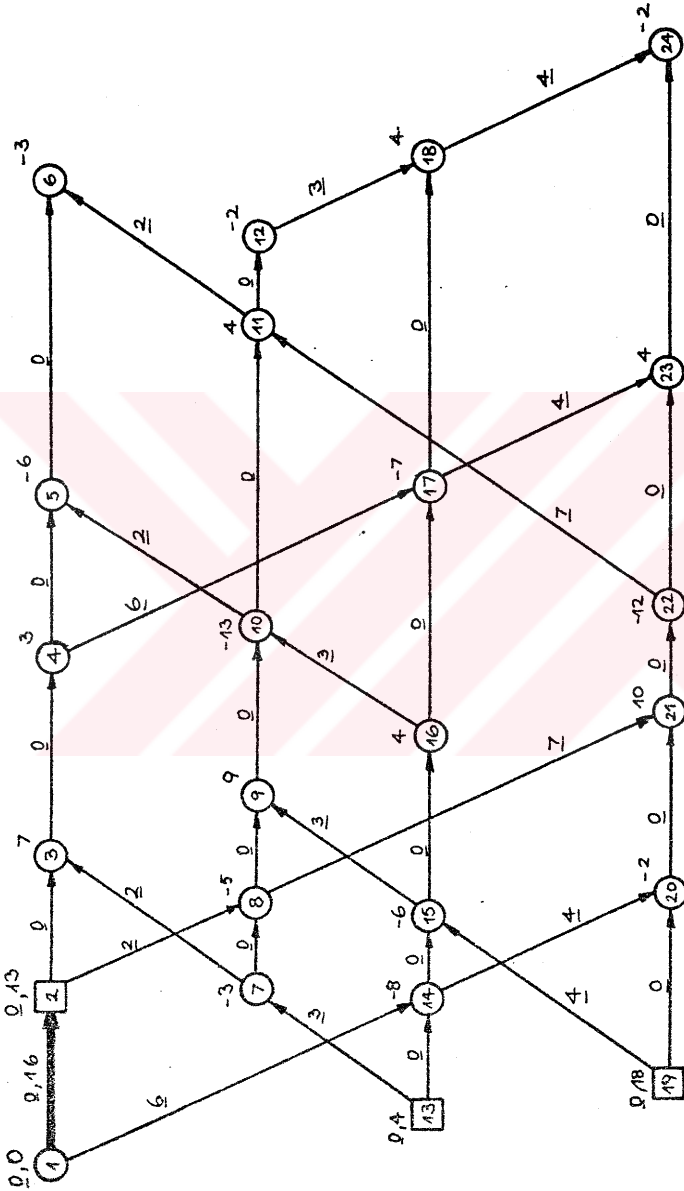
Her ne kadar 24 düğümlü sistem için 24 diyagram yapılması gerekirse de, burada amaç önerilen tekniği belirtmek olduğuna göre, 24 diyagramın hepsi çizilmeyip yalnızca 8 diyagramla yetinilmiştir.



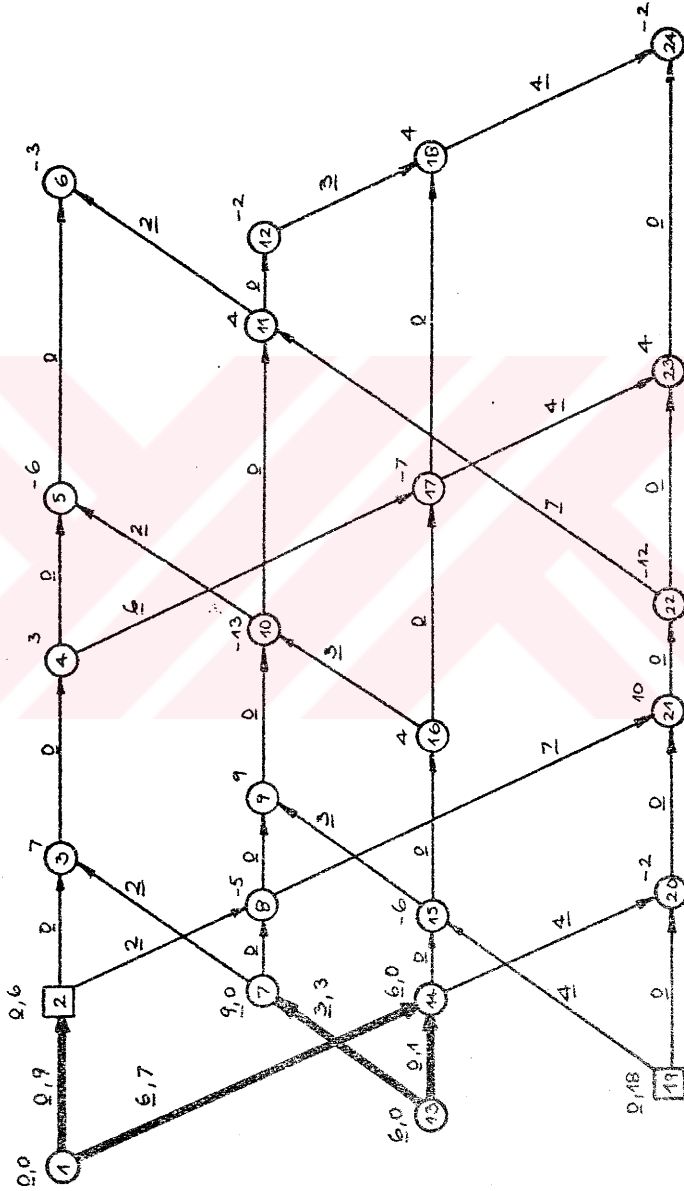
Şekil (4.1)



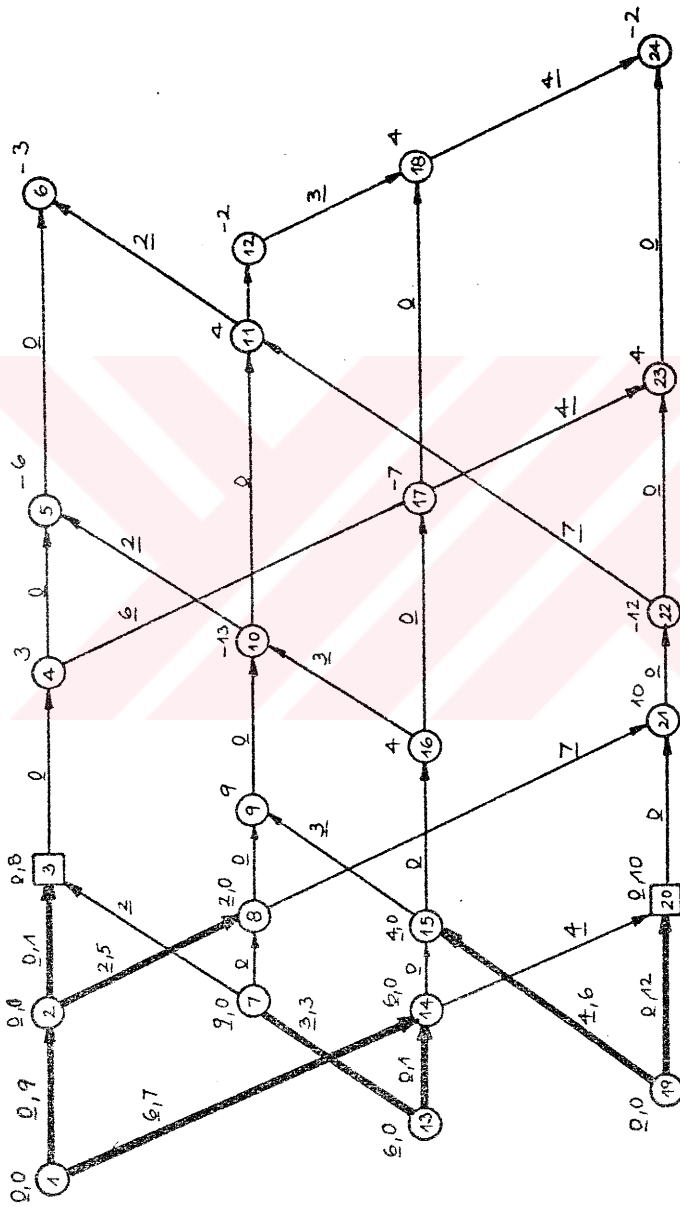
Şekil (4.2)



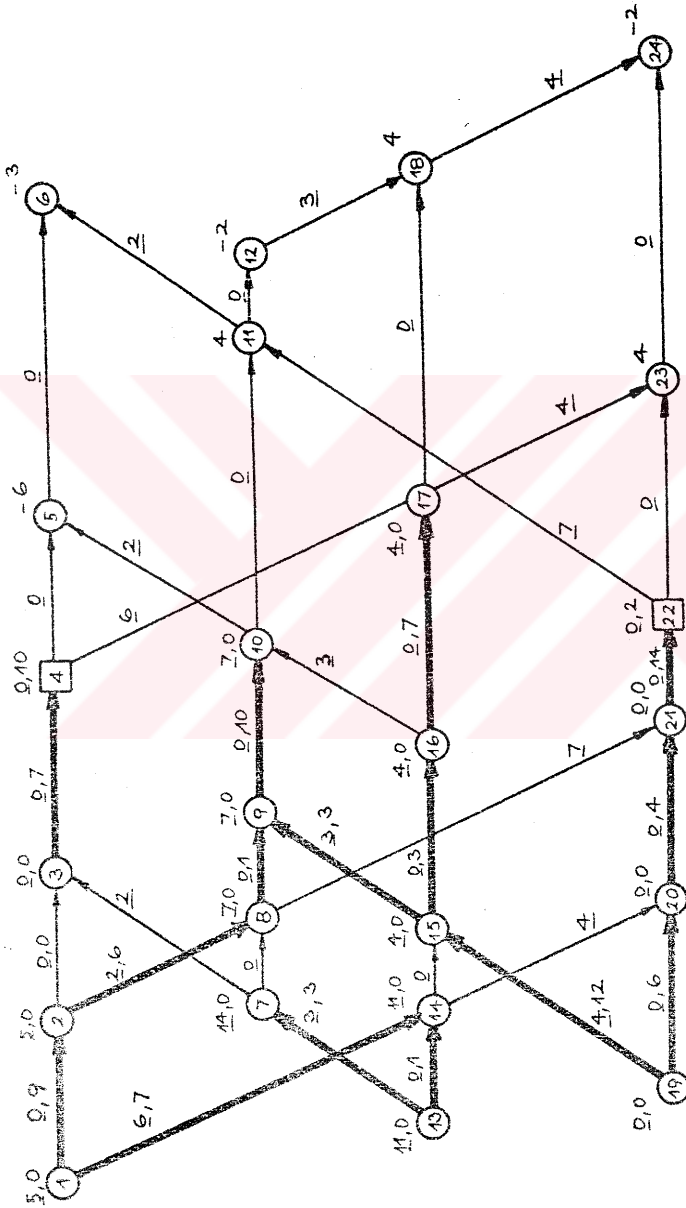
Şekil (4.3)



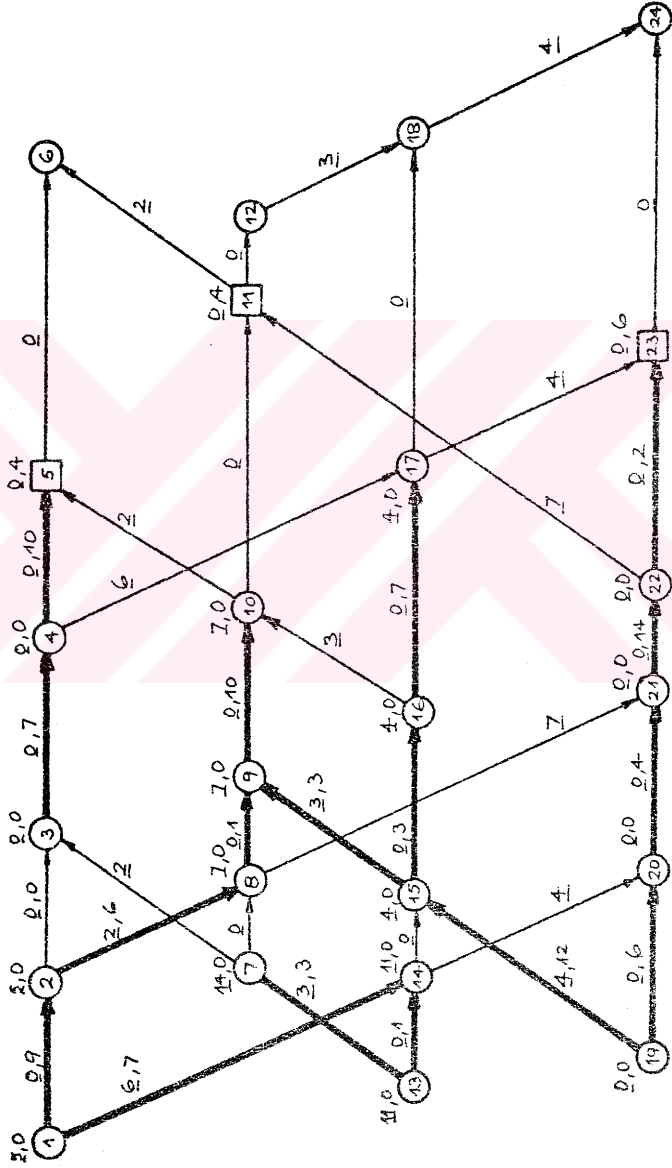
Şekil (4.4)



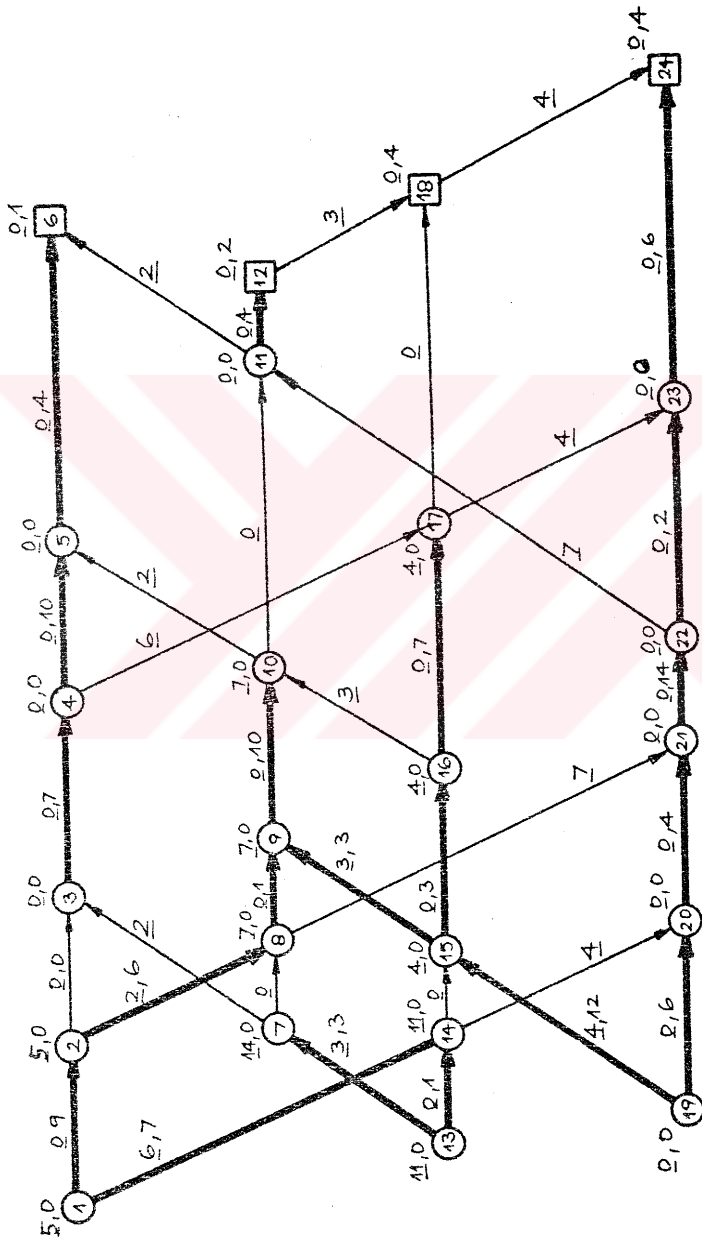
Şekil (4.5)



Şekil (4.7)



Şekil (4.8)



Şekil (4.9)

4.2. SONUÇLAR

1- Bir demiryolu ağında boş yük üniteleri trafiğinin dengelenmesine Yöneylem Araştırması teknikleri kullanarak yaklaşılmaktadır.

2- Bugüne kadar, ünitelerin optimal dağıtımları için lineer programlamanın taşın problemi en fazla kullanılan tekniklerden biri olmuştur. Ancak taşın problemi üniteleri kaynaklardan hedeflere bir kereye mahsus olmak üzere belli bir anda dağıtabilmektedir. Bazı dağıtım problemlerinde halen başarıyla kullanılmasına rağmen burada başka bir çözüm tekniği araştırılmıştır.

3- Bu çalışmada, yine lineer programlamanın kapsamı içine giren bir algoritmayla boş ünitelerin dağıtımı yapılmıştır. Bu algoritma, ağ-akım tipinde bir algoritmadır ve önerilen ağ tipine tamamiyle uymaktadır. Taşın probleminin aksine, statik hale getirilmiş dinamik ağ-akım problemleri bu algoritmayla çözülebilmektedir.

4 Önerilen silindir biçimindeki diyagramın üstünlüğü, dağıtım ağında herhangi bir değişiklik olduğu takdirde bu değişikliğin diyagram üzerine hemen uygulanabilmesi ve diyagramın neresinden başlanırsa başlansın tam diyagram süresince dağıtımın yapılabilmesidir.

5- Silindir biçimindeki diyagramın getirdiği kolaylık ve algoritmanın sonlu adımlarla problemi çözmelerinin yanında, boş ünite dağıtımı enformasyon eksikliği karşısında daima başarısız olmaya mahkumdur. Ünitelerin dağıtılabilmeleri için gerekli olan bilgilerin anında dağıtıcıya ulaştırılmaları gereklidir. Bunun için dağıtım yapılacak ağın her düğümünün birbiriyle her an bağlantısı olması gerekmektedir. Ayrıca ulaştırılan bilgilerin gerek istatistikler ve gerekse previzyonlar bakımından korunmaları da gerekmektedir. Bu haberleşme ve korunma işlemleri bilgi sayarlar yardımı ile yapılabilir.

6- Bu çalışmada ele alınan ağ, bir demiryolu ağı idi. Silindirik çizim şeklinin ve kullanılan algoritmanın aynen veya üzerlerinde yapılacak bazı değişikliklerle diğer ulaşım ağları ve sistemleri için de kullanılabilecekleri düşünülmektedir.

7- Dağıtım problemlerinin çözümü için bu çalışmada açıklanan algoritma ve yeni diyagram, triyaj istasyonları arasında hareket eden ünitelerin dağıtımı için de uygulanabilmektedir. Bu durumda diyagramdaki istasyonlar ekseni triyaj istasyonları eksenine dönüşmektedir.



NOTASYONLAR

C	= minimize edilmesi gerekli toplam maliyet
x_{ij}	= i den j ye taşınacak ünite miktarı
c_{ij}	= i ile j arasındaki birim taşın maliyeti
u_i, v_j	= gölge maliyetler
S	= sonsuza yakın çok büyük maliyet
N	= düğümler cümlesi
A	= oklar cümlesi
s	= kaynak
t	= hedef
x, y	= ara düğümler
$k(x, y)$	= (x, y) okunun kapasitesi
$f(x, y)$	= (x, y) okundaki akım
$a(x, y)$	= (x, y) okunun geçiliş süresi
n	= düğüm sayısı
π_i	= i düğümünün potansiyeli veya düğüm fiyatı
$M(i, j)$	= i, j okunun marjinal maliyeti

BIBLIYOGRAFYA

- |1| ACHER J. : Algèbre linéaire et programmation linéaire.
1965, Dunod, Paris
- |2| ALEXANDER N.J.B. : Computers aid BR's freight planning.
Railway Gazette, 3.April.1970,
s.253, London.
- |3| AVI-ITZHAK B.,
BENN B.A., POWELL B.A.: Car pool systems in Railroad
transportation : mathematical
models.
Management science 13, no.9,
s. 694, May 1967.
- |4| BEECHINGS : The reshaping of British Railways.
1963, H.M.S.O. publication, London
- |5| BERGE C. : Théorie des graphes et ses applications,
1958, Dunod, Paris
- |6| BERGE C. ve
GHOUILLA-HOURI A. : Programmes, jeux et réseaux de
transport
1962, Dunod, Paris.
- |7| BIXLER H.E. : Some thoughts about ways to resolve
the Eternal freight car question.
Traffic world, 1967, January, s.61
- |8| BIXLER H.E. : A bold proposal for utilizing the
car fleet more effectively.
Railway Age.
27.April.1970, New York, s.33

- |9| BUSACKER R.G. ve
SAATY T.L. : Finite graphs and networks.
1965, Mc Graw Hill.
- |10| CHURCHMAN C.W.,
ACKOFF R.L., ARNOFF E.L. : Introduction to Operations
Research,
John Wiley, 1957, New York.
- |11| CULLMANN G. : La recherche opérationnelle.
1970, Masson, Paris.
- |12| DANTZIG G.B. : Linear programming and
Extensions.
1963, Princeton University
Press, Princeton, New Jersey.
- |13| DANZIG G.B.,
FULKERSON D.R., JOHNSON S.: Solution of a large scale
traveling salesman problem
Operations Research, 2, 1954,
s.393
- |14| DANZIG G.B. ve
FULKERSON D.R. : On the max-flow min-cut theorem
of networks
Linear inequalities and related
systems, Annals of Mathematics
study 38, 1956, s. 215,
Princeton University Press
- |15| DESBAZEILLE G. : Exercices et problèmes de
recherche opérationnelle.
1964, Dunod, Paris.
- |16| FORD Jr L.R. ve
FULKERSON D.R. : Maximal flow through a network
Canadian Journal of Mathematics,
8, 1956, s.399.

- |17| FORD Jr, L.R. ve
FULKERSON D.R. : Flows in Networks .
1962, Princeton University
Press, Princeton, New Jersey
- |18| FORD Jr, L.R. ve
FULKERSON D.R. : Constructing maximal dynamic
flows, Operations Research,
6, 1958, s. 419
- |19| FULKERSON D.R. : An out-of-kilter method for
minimal cost flow problems.
Journal on applied mathematics,
9, 1961, s.18
- |20| FORD N. : The car you want when you
want it
Modern Railroads, September
1969, Chicago, s.100
- |21| FORD N. : Task force speeds plans for
car data network
Modern Railroads, February
1972, s. 72, Washington
- |22| GASS S.I. : Linear programming. Methods
and applications.
1958, Mc Graw Hill
- |23| GIGON Jean Marc : The allocation of empty wagons
by means of an electronic
computer.
Bulletin I.R.C.A: cybernetics
and electronics on the railways
July 1967, vol IV, no.7,
Brussels, s.287
- |24| GOTTFRIED G. : The distribution of empty
transport units in the railway
network.
Bulletin I.R.C.A : cybernetics
and electronics on the railways
November 1968, vol V, no.11
Brussels, s.395.

- | 25 | HARARY F. : Graph theory and theoretical physics
1968, Academic Press, New York
- | 26 | HITCHCOCK F.L. : The distribution of a product from several sources to numerous localities
Journal. Math. phys., 20, 1941, s.224
- | 27 | JOHNSON E. : Networks and basic relations
Operations Research 14, no.4, s.619, July-August 1966
- | 28 | JONES R.D.H. : Another nail in the railroad coffin.
Transportation Research, Vol.7, s. 413 - 419, 1973, Pergamon Press, London.
- | 29 | KAUFMANN A. : Methodes et modèles de la Recherche opérationnelle
Tome I et Tome II, 1968, Dunod, Paris
- | 30 | KONYA H. : An algorithm for freight car dispatchings
Quarterly Report : R.T.R.I, 1966, vol 7, no 4, s. 42
- | 31 | LASALA M.J. : La distribution des wagons vides par application de la recherche opérationnelle
Bulletin de l'AICCF, Février 1958 s.163, Bruxelles
- | 32 | LASZKIEWICZ R. : An experiment on the simulation of wagon movements in a railway region
Bulletin I.R.C.A : cybernetics and electronics on the railways
August 1966, vol III, no 8, Brussels, s. 367.

- | 33| LASZKIEWICZ R. : An information base for the
data procassing system for
the Polish Railways
Rail International, January
1970, Brussels, s.19

- | 34| LEDDON C.D., WRATHALL E.: Scheduling empty freight car.
fleets on the Louisville and
Nashville Railroad
Second International symposium
on the use of cybernetics on
the Railways,
Montreal, Canada, October
1-6, 1967

- | 35| MARTENS A. : Mathematical models for the
problem of routeing wagons
between marshalling yards
Bulletin I.R.C.A : cybernetics
and electronics on the railways
April 1967, vol IV, no.4 ,
Brussels, s. 145

- | 36| MERLET J. : Computerised management of
freight traffic.
Modern Railways, April 1971,
Shepperton, s.304

- | 37| MISRA S.C. : Linear Programming of empty
wagon dispassition
Rail International March 1972,
s.151

- | 38| ORDEN A. : The transshipment problem
Management Science, 2 no.3,
s.276, April 1956

- |39| PETROV A.P. : Utilisation des calculateurs électroniques, des transmissions et des procédés de la cybernétique à la gestion du trafic marchandises
Bulletin de l'A.I.C.C.F. Mai 1966, Bruxelles, s.1115

- |40| PLANTUREUX M.J. : La gestion centralisée du trafic marchandises à la S.N.C.F.
Révue générale des chemins de fer, Paris, juillet-Août, 1966, s. 490

- |41| SARGENT W.H. : Real-time computer for wagon control
The railway gazette, February 18, 1966, London, s. 147

- |42| SIMONARD M. : Programmation Linéaire
1962, Dunod, Paris

- |43| TAYLOR R.K. : The application of computers in railways
IRSE Proceeding 1967/68, s.143, London

- |44| TRUSKOLASKI, GRABOWSKI : Les problèmes de la répartition des wagons. Modèle mathématique de la répartition des wagons vides en cas de prévisions à n jours
Bulletin de l'A.I.C.C.F. cybernétique et électronique, Mai 1964, Bruxelles, s.166

- |45| VAJDA S. : The theory of games and linear programming
Methuen, 1956, London

- |46| WHITE W.W., BOMBERAULT A. : A network algorithm for empty freight car allocation.
IBM systems Journal, Vol 8, no.2, 1969, New York

- | 47| WHITE W.W. : Mathematical programming, multi-commodity flows, and communication nets.
Symposium on computer-communications Networks and teletraffic
Polytechnic Institute of Brooklyn, April 4-6, 1972
- | 48| WRATHALL E. : Empty freight car scheduling and extensions
Transportation : a service
New York Academy of Science, 1968, New York

BİYOGRAFI

Mehmet Ayhan Sayraç, 1945 de İstanbul'da doğmuş, ilk öğrenimini 1956 da Sultanahmet İlkokulunda, orta öğrenimini 1963 de Saint Joseph Fransız Lisesinde tamamlamış ve 1964 de Robert College Yüksek Okulunda lisan öğrenimi yapmıştır. 1965 de İ.T.Ü. İnşaat Fakültesine girmiş ve 1970 yılında aynı fakülteyi Yüksek Mühendis olarak bitirmiştir.

İ.T.Ü. den mezun olduktan sonra, bir müddet Kadıncık Hidroelektrik Santrali inşaatında çalışmış ve Mayıs 1971 de İ.T.Ü. İnşaat Fakültesi Demiryollar ve Toprak İşleri Kürsüsüne asistan olarak girmiştir. Aynı yılın Ekim ayında ise İ.T.Ü. Mühendislik-Mimarlık Fakültesi Ulaştırma ve Trafik Kürsüsüne naklen geçmiştir.

1971 de Prof. Enver Berkmen yönetiminde doktora çalışmasına başlamış, 1973-1976 arasında Newcastle Upon Tyne Üniversitesinde çalışmalarına devam etmiştir.

Sayraç, halen Ulaştırma ve Trafik Kürsüsündeki görevine devam etmekte olup, evli ve bir çocukludur.