

FOR REFERENCE

NOT TO BE TAKEN FROM THIS ROOM

TAŞIT GÖZERGÂHI BELİRLEME PROBLEMLERİ
VE
BAZİ YENİ ALGORİTMALAR

Dr. Gündüz Ulusoy
Boğaziçi Üniversitesi
Endüstri Mühendisliği Bölümü

DOÇENTLİK TEZİ

Bogazici University Library



39001100543837

14

Eylül - 1981

İ Ç İ N D E K İ L E R

SAYFA

1. GİRİŞ	1
2. TAŞIT GÜZERGAHI BELİRLEME PROBLEMLERİ: BİR GENEL BAKIŞ	4
2.1. TAŞIT GÜZERGAHI BELİRLEME (TGB) PROBLEMİNDE TANIMLAMA VE SINIFLANDIRMA	4
2.2. TGB PROBLEMLERİNDE GERÇEKÇİ MODELLEMENİN GÜÇLÜKLERİ	7
2.3. TGB PROBLEMLERİNE MODELLEME YAKLAŞIMLARI	10
2.3.1. Çok Ürünlü Akış Problemine Dayandırılmış Model	11
2.3.2. Küme Parçalama Modeli	13
2.3.3. Taşıt Akışına Dayandırılmış Model	15
2.3.4. Ayrıtlara Hizmet Verilen TGB Problemi Formülasyonu (ATGB)	18
2.3.5. Nokta ve Ayrıtlara Hizmet Verilen TGB Problemi (GTGB)	20
2.3.6. TGB Formülasyonlarında Bazı Uzantılar	21
2.3.7. TGB Problemine Temel Teşkil Eden Çekirdek Problemler	23
2.4. TGB PROBLEMLERİNDE İKİ ÖNEMLİ UZANTI	26
2.4.1. Taşıt Sayısı Problemi	26
2.4.2. Taşıt Sayısı ve Kapasiteleri Problemi	28
2.5. ÇÖZÜM YÖNTEMLERİ	28
2.5.1. Giriş	28
2.5.2. Tek Depolu TGB Problemleri İçin Sezgisel Yöntemler	29
2.5.3. Çok Depolu TGB Problemleri İçin SEzgisel Yöntemler	31
2.5.4. Kesin Çözüm Yöntemleri	32
2.5.5. Etkileşimli Yöntemler	33
2.6. BAZI ESKİ VE YENİ PROBLEM ALANLARI	33



3. ASKERİ UÇAKLARIN İNTİKAL PLANLAMASI İÇİN BİR KESİN ÇÖZÜM YÖNTEMİ	36
3.1. GİRİŞ	36
3.2. MATEMATİKSEL PROGRAMLAMA FORMÜLASYONU	39
3.3. MATEMATİKSEL PROGRAMLAMA FORMÜLASYONUNUN DEĞERLENDİRİLMESİ	44
3.4. KESİN ÇÖZÜM YAKLAŞIMI	45
3.4.1. Serim Dönüşümü Uygulaması	45
3.4.2. Ana Hatları ile Bir Çözüm Yöntemi Önerisi	48
3.5. AYRINTILARI İLE ÇÖZÜM YÖNTEMİ	51
3.5.1. Giriş	51
3.5.2. Yalın Yolların Türetilmesi	51
3.5.3. Özel Küme Parçalama Algoritması	55
3.5.4. Yalın Yollar Kümesinin Daraltılması	60
3.6. ÖRNEK PROBLEM	62
4. ASKERİ UÇAKLARIN İNTİKAL PLANLAMASI İÇİN BİR SEZGİSEL ALGORİTMA	72
4.1. GİRİŞ	72
4.2. SEZGİSEL YAKLAŞIM	72
4.2.1. Sezgisel Yaklaşımın Nedenleri	72
4.2.2. Temel Sezgisel Yöntem Yaklaşımları	73
4.2.3. Önerilen Sezgisel Yaklaşımın Ana Hatları	74
4.3. AYRINTILARI İLE ÇÖZÜM YÖNTEMİ	77
4.3.1. Başlangıç Çözümünün Bulunması	77
4.3.2. Geliştirme Yordamı	78
4.3.3. Olurluluk Denetimi	86
4.4. ALGORİTMA	88
4.5. ÖRNEK PROBLEM	92
5. AYRITLARA HİZMET GÖTÜRÜLEN PROBLEMLERDE TAŞIT SAYISI VE KAPASİTELERİ İÇİN BİR SEZGİSEL ALGORİTMA	97
5.1. GİRİŞ	97
5.2. PROBLEMİN TANIMI	98

	<u>SAYFA</u>
5.3. ANA HATLARI İLE ÇÖZÜM YÖNTEMİ	98
5.4. AYRINTILARI İLE ÇÖZÜM YÖNTEMİ	100
5.4.1. Kentli Postacı Turunun Elde Edilmesi	101
5.4.2. Yeni Serimin Elde Edilmesi	102
5.4.3. En Kısa Yol Algoritması	107
5.4.4. Algoritma	108
5.5. ÖRNEK PROBLEMLER	109
5.6. KISITLI TAŞIT SAYISI İÇİN BİR ÇÖZÜM YÖNTEMİ	115
5.6.1. Dallanma Kuralı	116
5.6.2. Sınırlama Kuralı	117
5.6.3. Dallanılacak Düğümün Seçimi Kuralı	117
5.6.4. Algoritma	118
5.7. ÖRNEK PROBLEMLER	119
6. SONUÇ	125
6.1. ASKERİ UÇAKLARIN İNTİKAL PLANLAMASI İÇİN KESİN ÇÖZÜM YÖNTEMİNİN DEĞERLENDİRİLMESİ	125
6.1.1. Kesin Çözüm Yönteminin Denenmesi	125
6.1.2. Bazı Yeni Öneriler	127
6.2. ASKERİ UÇAKLARIN İNTİKAL PLANLAMASI İÇİN BİR SEZGİSEL ÇÖZÜM YÖNTEMİNİN DEĞERLENDİRİLMESİ	128
6.3. ATGB PROBLEMİNDE TAŞIT SAYISI VE KAPASİTESİ İÇİN BİR SEZGİSEL ÇÖZÜM YÖNTEMİNİN DEĞERLENDİRİLMESİ	130
KAYNAKÇA	134

ŞEKİLLER

	SAYFA
Şekil 3.1. 4 Uçak ve 19 Yüklü Örnek	37
Şekil 3.2. Havaalanının Üç Düğüm ile Temsili	40
Şekil 3.3. Üç Havaalanı İçin Matematiksel Programlama Formülasyonu Serimi	45
Şekil 3.4. Dönüşüm Uygulanacak Serim	46
Şekil 3.5. Dönüşüm Neticesi Elde Edilen Serim	47
Şekil 3.6. Kesin Çözüm Yönteminin Akış Şeması	63
Şekil 4.1. Örnek Serim	75
Şekil 4.2. Ulaştırma Probleminin Çözüldüğü Serim	78
Şekil 4.3. Boş Uçuş Enküçüklemesi Örneği	79
Şekil 4.4. Boş Bir Uçuşa İki Uçuş Yerleştirilmesi Örneği	81
Şekil 4.5. Boş Uçuş Kullanmadan İki Uçuş Eklenmesi Örneği	82
Şekil 4.6. Boş Uçuş Kullanarak İki Uçuş Eklenmesi Örneği	83
Şekil 4.7. Bir Uçuşun Boş Bir Uçuşa Yerleştirilmesi Örneği	84
Şekil 4.8. Boş Uçuş Kullanmadan Bir Uçuşun Eklenmesi Örneği	85
Şekil 4.9. Boş Uçuş Kullanarak Bir Uçuşun Eklenmesi Örneği	85
Şekil 4.10. Ana Hatları ile Çözüm Yöntemi	89
Şekil 4.11. Örnek Problem Serimi	93
Şekil 5.1. Yeni Serim Örneği	99
Şekil 5.2. Esas Serim Örneği	104
Şekil 5.3. En Kısa Yolun Altturlara Bölüştürülmesi	108
Şekil 5.4. Örnek Problemlerin Serimi	109
Şekil 5.5. Birinci Örnek Problemin Çözümü	113
Şekil 5.6. Dallanma Kuralı	116
Şekil 5.7. Dal-Sınır Ağacının İlk İki Düzeyi	120
Şekil 5.8. İlk Kısıtlı Örnek Probleminin Çözümü	122

TABLOLAR

	<u>SAYFA</u>
Tablo 3.1. Yk Talepleri Matrisi	46
Tablo 3.2. A Matrisi	57
Tablo 3.3. B Matrisi	57
Tablo 3.4. Daraltılmıř X Kmesi	71
Tablo 4.1. Yk Talep Matrisi	75
Tablo 6.1. zel Kme Paralama Algoritması Deney Problemleri	126
Tablo 6.2. Deney Problemleri Neticeleri	129
Tablo 6.3. Deney Problemleri Neticeleri	132

I. GİRİŞ

Yöneylem Araştırmasının önemli uğraşlarından birisi mal ve hizmetlerin arz noktalarından talep noktalarına nasıl eniyi bir biçimde ulaştırılacağına araştırılmasıdır. Bu ulaştırmanın araştırılma, planlanma ve gerçekleştirilme faaliyetlerinin tümüne lojistik yönetimi diyoruz. Lojistik yönetimi bir hiyerarşi içinde ve belirli aşamalar halinde işlevini gerçekleştirir. Bu aşamalardan biri de dağıtımdır. Dağıtım burada bir faaliyet türünü belirtmek için kullanılmıştır. Gerçekte bu faaliyet toplama da olabilirdi. Örneğin posta toplanması gibi. Bu tezin genel konusu olan taşıt güzergahı belirleme (TGB) problemleri dağıtımın planlanmasında çözülmesi gereken bir sorundur.

TGB problemi, talepleri bilinen nokta ve/veya ayrıtlar (arcs) bir amaç işlevini, verilmiş kısıtlar içinde eniyileyecek ve bütün talepleri karşılayacak şekilde sıralanması ile bir güzergahın belirlenmesi problemidir.

Problem, Garvin v.d.(1957) ve Dantzig ve Ramser(1959) tarafından ortaya atıldığından beri çok sayıda adam-yüzyıllık çalışma bu probleme hasredilmiştir. Bunun nedeni hem pratik hem teorik açılardan problemin ilginçliğidir. Problem günlük hayatımızın parçası olan somut bir konuyu içermektedir. Getirilecek çözümlerin hemen uygulanabilme ve etkilerinin görülebilmeye olanağı vardır. Bütün bunlar pratik yönden sorunu çekici kılmaktadır. Problem teorik açıdan bakıldığında kombinatoryal bir problemdir. Problemin boyutları büyük bir hızla büyümektedir. Bu itibarla probleme kesin çözüm yolları aramak yöneylem araştırmacıları açısından zorluğundan dolayı çekici olmuştur. Problemin gerek teorik açıdan zorluğu gerekse çözüm den planlamaya geçme aşamasında esnek modellerden gelen çözümlerin daha yararlı olması probleme oldukça büyük sayıda sezgisel (heuristic) yaklaşımın yapılmasına neden olmuştur.

Uzun süre hemen hemen sadece fabrika-depo-toptancı-tüketici zincirinin içinde düşünülen TGB problemleri özellikle 1970 li yıllarda kent sorunlarına uygulanmaya başlanmış ve okul otobüsleri, kar küreme makinaları, çöp kamyonları gibi kente hizmet veren taşıtların güzergah belirleme problemleri de TGB problemleri kapsamına girmiştir.

TGB problemini en soyut hali ile ifade eden ve burada çekirdek problem olarak nitelendirilen serim (network) problemleri sezgisel ve kesin çözüm yöntemlerini derinden etkilemişlerdir. Bu problemler, gezgin satıcı problemi (traveling salesman problem), Çinli postacı problemi (Chinese postman problem) ve kentli postacı problemidir (rural postman problem)*. Bunlardan ilki noktalara, diğer ikisi ayrıtlara hizmet götüren TGB problemleri ile ilişkilidirler. Çekirdek problemlerin kesin çözüm yöntemlerinde ve genellikle kombinatoriyal programlamada yapılan ilerlemeler TGB problemlerine yansıtılamamıştır. Çabaların sezgisel yöntemlerde yoğunlaşmış olması bunun en önemli nedenlerinden biridir. Ancak son yıllarda kesin çözümlerde bir ilerleme gözlenmektedir ve bu konuda önemli gelişmeler beklenmektedir. Ayrı ayrı geliştirilmeleri yerine kesin ve sezgisel çözüm yöntemlerinin birlikte kullanılmaları daha üstün sezgisel ve kesin çözüm yöntemlerinin geliştirilmesine neden olacaktır kanısındayız.

Çalışmanın 2.Bölümünde TGB problemlerine bir genel bakış getirilmektedir. Henüz tatminkâr bir çözüm veya hiç bir çözüm getirilememiş bazı problemlere altbölüm 2.6 da değinilmiştir.

Bu çalışmada böyle iki problem ele alınmaktadır. Bunların birincisi olan askeri uçakların intikal planlamasına bir kesin çözüm yöntemi 3.Bölümde geliştirilmiştir. Yöntemin

* Bu çekirdek problem genellikle kent problemlerinde kullanıldığı için bu şekilde Türkçeleştirilmiştir.

eniyei çözümi bulduđu ispatlanmıř ve çözüml yöntemi bir örnek probleme uygulanmıřtır.

4.Bölümde aynı problem için geliştirilen bir sezgisel yöntem sunulmaktadır. Yöntem örneklerle anlatılmakta ve bir örnek probleme uygulanmaktadır.

5.Bölümde ayrıtlara hizmet götürülen durumlarda taşıt sayısı ve kapasiteleri problemi için geliştirilen bir sezgisel yöntem ayrıntıları ile tanıtılmakta ve değişik örnek problemlere uygulanmaktadır.

6. Bölümde her yöntemle ilgili değerlendirme yapılmıř ve deney problemleri neticeleri sunulmaktadır. Askeri uçakların intikal planlaması problemi için geliştirilen kesin ve sezgisel yöntemlerin tek bir yöntem halinde birleştirilmesi tartışılmaktadır.

Bu ve diğer Bölümlerde bazı teknik terimlerin Türkçe karşılıklarından emin olunmadığı durumlarda İngilizce karşılıkları parantez içinde verilmiştir. Amaç okumadaki akışı bölmektir.

2. TAŞIT GÜZERGAHI BELİRLEME PROBLEMLERİ: BİR GENEL BAKIŞ

2.1. TAŞIT GÜZERGAHI BELİRLEME (TGB) PROBLEMİNDE TANIMLAMA VE SINIFLANDIRMA

Bu altbölümde, TGB problemi önce tanımlanacak sonra problemin değişik niteliklerine göre sınıflandırılacaktır.

Taşıt güzergahı bir nokta ve ayrıt dizisinden oluşur. Her ayrıt başlangıç ve bitiş noktaları ile belirlenir. Taşıt başlangıç noktasından hareketle güzergahındaki bütün noktaları ve ayrıtları dolaşarak başlangıç noktasına geri döner. Bu başlangıç noktasına depo diyeceğiz.

TGB problemi, talepleri bilinen nokta ve/veya ayrıtların bir amaç işlevini, verilmiş kısıtlar içinde eniyileyecek ve bütün talepleri karşılayacak şekilde sıralanması ile bir güzergahın belirlenmesi problemidir.

TGB probleminde taşıt güzergahı üzerinde hizmet, (i) noktalarda, (ii) ayrıtlarda, (iii) nokta ve ayrıtlarda verilebilir. Taşıt güzergahının üzerinde tanımlandığı serimin tüm nokta veya ayrıtlarına hizmet götürülmeyebilir.

Götürülen hizmete olan talebin niteliği problemin formülasyonunda belirleyici bir unsurdur. Talep, (i) gerekirci (deterministic) veya (ii) rassal (stochastic) olur.

TGB problemlerinde değişik türde amaç işlevleri kullanılmaktadır. Bunlar, (i) değişken maliyetlerin enküçüklenmesi, (ii) değişken ve sabit maliyetlerin enküçüklenmesi, (iii) taşıt sayısının enküçüklenmesi. Taşıt sayısının enküçüklenmesi sabit maliyetinin enküçüklenmesidir. Değişken maliyetlerin ifadesi genellikle doğrusal olmuştur. Sabit mali-

yetler ise taşıt maliyeti olarak (0-1) şeklinde formüle edilmişlerdir.

Taşıt filosunun büyüklüğü de formülasyonda göz önüne alınan bir unsurdur. TGB problemlerinin formülasyonu (i) bir taşıt veya (ii) birden fazla taşıt durumlarını içerir. Bir taşıt durumu çoğu kez kapasite kısıtsız durumu ima eder. Taşıt sayısı üzerine getirilen bu kısıt genellikle kapasite kısıtları ile birlikte ele alınır.

Taşıtların kapasite kısıtı problemlerin gerçekçiliği açısından gerekli bir kısıttır. Tek taşıt olduğu durumlarda problemin doğru tanımlanmış olması için toplam talebin eldeki taşıtın kapasitesinden büyük olmaması gerekir. Bu nedenle tek taşıtlı problemlerde ayrıca kapasite sınırı kullanılmaz. Bunun dışında, TGB problemlerinde, (i) bütün taşıtlar için kısıt vardır ve hepsi için aynıdır (ii) bütün taşıtlar için kısıt vardır fakat değişik kapasite kısıtları olabilir.

Taşıt tipleri itibarı ile TGB problemleri, (i) bir taşıt tipi, (ii) birden fazla taşıt tipi olarak sınıflandırılırlar. Taşıt tipleri taşıtların değişik özellikleri ile belirlenir. Bunlardan biri kapasite başına sabit maliyet ve birim katedilen mesafe başına değişken maliyet toplam maliyeti etkileyen önemli parametreleri oluştururlar. Taşıtların TGB problemi için diğer önemli özellikleri kapasiteleri ve güzergah uzunluğu üstsınırlarıdır. Bunlar TGB problemine kısıt olarak girerler.

TGB problemi üzerinde diğer bir kısıt taşıt güzergah uzunluğu üstsınırlıdır. Bu kısıt iki türlü olabilir: (i) üstsınır var fakat bütün taşıtlar için aynı; (ii) üstsınır var fakat bütün taşıtlar için aynı değil. Böyle bir kısıtın gerçekçi olduğu durumlar çoğunluktadır. Eldeki filo değişik türde taşıtlardan oluşuyorsa o durumda taşıt türüne göre üstsı-

nır getirmek gerekli olacaktır. Ancak bazı durumlarda da taşıt güzergah uzunluğu üzerinde (iii) üstsınır yoktur. Güzergah uzunluğu üzerindeki üstsınır sadece taşıtların bir özelliği olarak belirmez. Bu aynı zamanda dağıtılan veya toplanan malın niteliklerinden de bağımlı olabilir.

Bazı problemlerde bir nokta ve ayrıta hizmet verme zamanı hizmetin önemli bir unsuru olabilmektedir. Böyle durumlarda, hizmet verme zamanı (i) önceden belirlenmiş ve sabit, (ii) bir zaman dilimi olarak belirlenmiş, (iii) belirlenmemiş olabilir.

Hizmet verilen nokta veya ayrıtlarda yapılan işlemler, (i) salt yükleme, (ii) salt boşaltma, veya (iii) yükleme ve boşaltma karışık olabilir.

İşlemlerin niteliği ötesinde özellikle maliyet açısından önemli olabilecek bir nokta güzergah üzerinde yapabilecek en fazla işlem sayısıdır. İşlemler için belirli bir zaman harcandığı göz önüne alınırsa işlem sayısının taşıt zamanı üzerinde doğrudan etkisi olduğu görülür. Problemin yapısına göre, (i) işlem sayısı üzerinde bir kısıt vardır, veya (ii) işlem sayısı üzerinde bir kısıt yoktur.

TGB probleminin üzerinde belirlendiği serimin yapısı problem formülasyonunda ve özellikle çözümünde önemli bir rol oynar. Serim, (i) yönlü, (ii) yönsüz, (iii) karışık olabilir. Tanım icabı yönlü ve karışık serimler simetrik değil, yönsüz serimler ise simetrikler. Bir serimin simetrik oluşu veya olmayışı çözüm yönteminin başarısını etkileyen önemli etmenlerdendir.

TGB problemini daha geniş bir perspektif içinde ele alan dağıtım problemi ile TGB problemi arasındaki bir bağ da depodur. TGB probleminde bir sınıflandırma da (i) bir depo veya (ii) birden fazla depo olması durumuna göre yapılır.

Görüldüğü gibi yukarıda yapılan sınıflandırma oldukça kapsamlıdır. Buna rağmen gerçek problemleri tüm ayrıntıları ile ele almaktan uzaktır. Özellikle bu sınıflandırmanın matematiksel modellemeye geçiş sırasında daha da sınırlanacağı göz önüne alınırsa modelin çözümünden uygulanması istenen çözüme geçiş sürecinin önemi daha belirgin olarak ortaya çıkar. Yöneylem araştırması uygulamalarının başarısı üzerinde en önemli etmenlerden olan (gerçek durum → model → gerçek durum) dönüşümlerindeki isabet TGB problemleri için de geçerlidir. Bunun da ötesinde, TGB problemlerinin bu konuda yöneylem araştırmacıları için çözümü hiç de kolay olmayan sorunları vardır.

2.2. TGB PROBLEMLERİNDE GERÇEKÇİ MODELLEMENİN GÜÇLÜKLERİ

Yukarıda yapılan sınıflandırma bazı varsayımlara dayandırılan formülasyonlar üzerindendir. Bu sınıflandırmada kullanılan bazı unsurların basitleştirici varsayımlar kaldırıldığında yaratabilecekleri zorluklar ve bu nedenle gerçekçi bir formülasyonun zorlukları burada tartışılacaktır.

Önce talebi ele alalım. Talep gerçekte son ana kadar bilinmeyebilir. Çoğu kez bu olasılıklar hakkındaki bilgi güvenilir bir olasılık dağılımının elde edileceği bir düzeyde olmayabilir.

Hergün için bir eniyi güzergah belirlemesi saptanabildiğini varsayalım. Talepteki oynamalar neticesi hergünkü eniyi güzergah belirlemesi değişik olabilecektir. Bu ise aynı sürücünün değişik uzunluktaki seferlerde değişik yerlere uğraması gibi istenmeyen bir durum yaratabilecektir. Bazı durumlarda günler arasında taşıt talebinde de farklılaşma olabilir; yani bazı günler bazı sürücüler boş kalacak veya elde mevcut sürücü sayısından fazlasına ihtiyaç olacaktır. Bütün bunlar personel yönetimi açısından sakıncalı hususlardır. İşin sürücüler arasında mümkün olduğunca eşit dağıtılması yönetimce bir

amaç olarak benimsenebilir. Ancak bu amaç gerçekleştirilmek istendiğinde eniyi çözümden oldukça uzaklaşmak gerekebilir. Gerek talebin rassal yapısına gerekse talebi karşılayamama sorunlarına çözüm getirmek amacı ile ortalama taleplere dayandırılan bir iskelet çözüm ve karşılanamayan talebi belirli bir olasılık düzeyinin altında tutan bir ek kapasite bulunabilir. Ancak bu yaklaşım taşıt sayısını enküçükleme amacı ile ters düşecektir.

Güzergah belirlemesinin üzerinde yapıldığı serim ve bu serime ilişkin veri tabanının güvenilirliği çözümün geçerliliğini doğrudan etkileyen faktörlerdir. Gerçek yol seriminin üzerinde çözümün yapıldığı serime dönüştürülmesindeki ayrıntı sorunu model çözümünün kesinliği ile çözüm süresi arasında bir dengeleme sorunudur. Gerçek serimde olan bazı noktalar model serimine geçirilmeyerek model seriminin daha ufak olması sağlanır.

Bir serim, noktalar, ayrıtlar ve bunlar arasındaki ilişkilerin tümü ile tanımlanır. Noktalar ve ayrıtların serim üzerindeki konumları ve ayrıtların uzunluk ve katedilme sürelerine ilişkin verilerdeki hassasiyet çözümü etkileyecektir. Özellikle kentiçi yol serimlerinde kavşaklar ve U-dönüşlerine ilişkin süreler ve ayrıtların katedilme süreleri günboyu değişiklik gösterirler. Burada yaklaşım, günü zaman dilimlerine bölmek ve her zaman dilimi için ayrı bir çözümleme yapmaktır (Ulusoy ve Or 1979a). Yollar, özelliklerine göre değişik kategorilere ayrılırlar ve her kategorideki yol için debi-katetme süresi ilişkisi gözlemlere dayanılarak bulunur (FHA, 1973). Böylece bir ayrıttın yol kategorisi saptandıktan sonra debisi tahmin edilerek ayrıtı katedilme süresi bulunabilir. Görüldüğü gibi yol kategorileri ayrıt uzunluklarını sınırlayan bir kısıt olarak ortaya çıkmaktadır. Günü zaman dilimlerine ayrılmasının yarattığı zorluk bir taşıtın seferini aynı zaman dilimi içinde bitirememesidir. Ancak zaman dilimlerinin az tutulması da hassasiyet kaybına yol açabilir.

Ayrıtların bir karakteristiği olan uzunluk maliyetler yolu ile amaç işlevine yansıtılır. Değişken maliyetler (yakıt, yağlar, lastik, amortisman ve bakım) uzunluktan bağımlıdır. Dolayısı ile kilometre başına maliyet olarak ifade edilirler. Yakıt ve lastik ve taşıt yıpranmaları bir dereceye kadar yol şartlarından ve taşınan yükün ağırlığından da bağımlıdır. Maliyetin diğer bir unsuru olarak sabit maliyetleri gösterebiliriz (sürücü ve diğer işçi ücretleri, sigorta ve diğer vergi ve harçlar, garaj giderleri ve taşıt maliyeti faiz giderleri). Sabit maliyetler zamandan bağımlıdır.

Amaç işlevinin salt uzunluğu yansıtması halinde değişken maliyetler; taşıt sayısını yansıtması istendiğinde sabit maliyetler; uzunluk ve taşıt sayısını yansıttığı durumda ise değişken ve sabit maliyetlerin toplamı kullanılır. Amaç işlevinin birim zaman başına maliyet cinsinden ifade edilmesi genelde en uygundur. Bu amaçla, kilometre başına maliyet, süre, süre ve uzunluk verileri kullanılır. Bütün uğraşın amacı bu amaç işlevinin eniyilenmesi olduğuna göre, bunun ifadesinde kullanılan katsayıların sağlam verilere dayandırılması gerekir. Normal muhasebe kayıtları dışındaki verilerin elde edilmesi çok güç olabilir. Veri hassasiyetinin gereken düzeyi duyarlılık analizi ile bulunarak veri toplama çabasındaki yoğunluğun buna göre ayarlanması yararlı olur.

Klasik TGB problemi formülasyonunda yapılan iki varsayım bazı gerçek TGB durumlarında geçerli olmamaktadır. Bu varsayımlar: (i) eldeki taşıtların enküçük kapasitesinin enbüyük talepten büyük veya eşit olduğu, (ii) güzergahların depodan başlayıp depoda bittiğidir.

Birinci varsayımın geçerli olmadığı durumlarda o noktadaki talebi bölmek gerekecektir. Talebin bir bölümünün eldeki enbüyük kapasiteli bir taşıtla geri kalan bölümünün ise di-

ger bir taşıtla taşınmasının her zaman eniyi çözümü vermeyeceği kolaylıkla gösterilebilir. Bu sorun çoğu kez sezgisel yöntemlerle çözülmeye çalışılır. İkinci varsayımın geçerli olmadığı durumlarda ise model seriminde bir değişiklik ile yeni durum gözönüne alınabilir.

Bir takım gerçek problemler ise sadece TGB problemi değildirler; başka bir problemin TGB problemi ile müştereken ele alınmasını gerektirirler. Örneğin değişik tür yükleri aynı anda beraberce taşıyabilen taşıtlarda taşıt yüklemesi en az güzergah belirlemesi kadar önemlidir. Diğer böyle bir problem taleplerin değişik gruplar halinde birleştirilerek bir zaman birimine atanmaları ve bu zaman dilimi içinde karşılanmalarıdır. Böyle bir gruplandırmanın nasıl yapılacağı TGB problemi ile müştereken çözülmek durumundadır.

Gerçekçi modellemenin bazı güçlükleri yukarıda belirtilmeye çalışılmıştır. Ancak gerek yöneylem araştırmasında gerekse diğer modelleme yapılan bilim dallarında gözlenen sürece göre gerçek problemlere önce basit modellerle yaklaşılmakta; giderek modeller daha gerçekçi olmaktadır. TGB problemlerinde de geçen zaman içinde bu gözlenmiştir. Bu nedenle önümüzdeki yıllarda gerek bilgisayar donanımlarında gerekse kombinatoryal programlamada yapılmakta olan gelişmelerin TGB problemlerine aktarılması ile daha gerçekçi modellerin kurulup çözülmesi mümkün olacaktır kanısındayız.

2.3. TGB PROBLEMLERİNE MODELLEME YAKLAŞIMLARI

TGB problemleri için oldukça geniş sayıda model önerilmiştir. Bu modeller (i) nokta, (ii) ayrıt, (iii) nokta ve ayrıtta hizmet verilen modellerdir. Önce noktaya hizmet verilen TGB problemleri (NTGB) incelenecektir. Üç matematiksel programlama modeli bu konuda temel modelleri oluşturmaktadır. Bunlar: (i) çok ürün akışına dayandırılmış model, (ii) küme parçalama (set partitioning) modeli, (iii) taşıt akışına dayandırılmış mo-

deldir.

Burada sunulacak olan bu modellere, çok depo, güzergah uzunluğu üstsınırı veya enbüyük güzergah süresi, taleplerin karşılanması istenen zaman aralıkları gibi uzantılar rahatlıkla eklenebilir.

2.3.1. Çok Ürünlü Akış Problemine Dayandırılmış Model

TGB problemine literatürde yapılan ilk atıf Garvin v.d. (1957) tarafından yapılmıştır. Verdikleri formülasyon (P1) olarak gösterilmiştir.

(P1) modelde,

$$\sum_{i=1}^N y_{ijs} = \sum_{i=1}^N y_{jis} \quad \forall s, j \quad s \neq j \quad (2.1)$$

$$\sum_{i=1}^N y_{iss} = D_s \quad \forall s \quad (2.2)$$

$$\sum_{s=1}^N \sum_{j=1}^N y_{ijs} = \sum_{s=1}^N D_s \quad (2.3)$$

$$\sum_{i=1}^N x_{ijp} = \sum_{i=1}^N x_{jip} \quad \forall j, p \quad (2.4)$$

$$\sum_{s=1}^N y_{ijs} - \sum_{p=1}^P C_p x_{ijp} \quad \forall i, j \quad i \neq j \quad (2.5)$$

$$x_{ijp} = (0,1) \quad \forall i, j, p \quad (2.6.a)$$

$$y_{ijs} > 0 \quad \forall i, j, s \quad (2.6.b)$$

kısıtları altında

$$Z = \sum_{i=1}^N \sum_{j=1}^N \sum_{p=1}^P c_{ijp} x_{ijp} \quad (2.7)$$

amaç işlevi enküçüklenecektir.

Burada:

c_{ijp} = i noktasından j noktasına p taşıtının birim taşıma maliyeti

y_{ijs} = i noktasından j noktasına s noktası teslim noktası olmak üzere yollanan yük miktarı

$x_{ijp} = \begin{cases} 1, & \text{eğer } p \text{ taşıtı } i \text{ noktasından } j \text{ noktasına gidiyorsa} \\ 0, & \text{aksi durumlarda} \end{cases}$

D_s = s talep noktasının talebi ($D_1 = 0$)

C_p = p taşıtının kapasitesi

N = nokta sayısı (1 numaralı nokta depodur)

P = taşıt sayısıdır.

(2.1) numaralı kısıt kümesi herhangi bir teslim noktası için yollanan yüklerin diğer noktalarda birikimini önler.

(2.2) numaralı kısıt kümesi taleplerin karşılanmasını sağlar.

(2.3) numaralı kısıt depodan çıkışların toplam talebe eşit olmasını sağlar.

(2.4) numaralı kısıt kümesi herhangi bir noktaya giren p taşıtının o noktadan çıkması gerektiğini belirtir.

(2.5) numaralı kısıt kümesi (i,j) ayrıtında yollanan yük miktarını mevcut taşıma kapasitesi ile ilişkilendiren bir eşitsizlikler kümesidir. (2.7) numaralı ilişki ile ifade edilen amaç işlevi toplam taşıma maliyetini belirtmektedir.

Bu formülasyona çok ürünlü akış problemine dayandırılmış denmesinin nedeni y_{ijs} değişkenlerinin çok ürünlü akış probleminde (i,j) ayrıtı üzerindeki s ürününün akış miktarına karşı gelmeleridir.

(P1) modelinde $(2N^2 + NP - N+1)$ kısıt ile (N^3+N^2P) karar değişkeni bulunmaktadır. Görüldüğü gibi az sayıda bir karışık-tam sayı programlama problemi ortaya çıkmaktadır. Garvin v.d. (1957) problemi doğrusal programlama kullanarak yaklaşık olarak çözmüşlerdir. Ancak tatmin edici neticeler almaları mümkün olmamıştır.

2.3.2. Küme Parçalama Modeli

NTGB problemine daha değişik bir yaklaşım Balinski ve Quandt (1964) tarafından önerilmiştir. Burada problem bir küme parçalama problemi olarak ele alınmıştır. Bu amaçla olurlu güzergahlar kümesi tanımlanmıştır. Amaç, bütün talepleri karşılayan ve enküçük toplam maliyete sahip güzergahlar altkümelerini bulmaktır. (P2) olarak gösterilen formülasyon aşağıda verilmiştir.

(P2) Modelde,

$$\sum_{j=1}^J a_{ij} x_j = 1 \quad \forall i \quad (2.8)$$

$$x_j = \begin{cases} 1, & \text{eğer güzergah } j \text{ kullanılıyorsa} \\ 0, & \text{aksi durumlarda} \end{cases} \quad (2.9)$$

kısıtları altında

$$Z = \sum_{j=1}^J C_j x_j \quad (2.10)$$

amaç işlevi enküçüklenecektir.

Burada:

C_j = j güzergahının maliyeti

$$a_{ij} = \begin{cases} 1, & \text{eğer } i \text{ düğümü } j \text{ güzergahında ise;} \\ 0, & \text{aksi durumlarda} \end{cases}$$

J = olurlu güzergah sayısıdır.

Yaklaşımında temelde önce düğümler gruplandırılmakta daha sonra her grup içinde bir güzergah saptanmaktadır. C_j maliyeti j grubundaki düğümler için çözülen bir gezgin satıcı problemi neticesinde bu turun amaç işlevi değeri olarak bulunur. Taşıt kapasitesini C ile gösterirsek taşıt kapasitesi yönünden olurlu bir j güzergahı aşağıdaki eşitsizliği sağlamak zorundadır:

$$\sum_{i=1}^N a_{ij} d_i \leq C \quad (2.11)$$

Model (P2) de aksayan taraf büyük problemler için olurlu güzergahlar kümesinin büyüklüğüdür. Ancak problem konusunda oldukça geniş bir bilgi varsa ve/veya problem oldukça kısıtlı ise bu durumda olurlu güzergahlar kümesi fazla büyük olmayabilir. Balinski ve Quandt (1964) "uygun" bir küme seçerek olurlu güzergahlar kümesini daha da daraltma yolunu seçmişlerdir. Küme parçalama problemini çözmek için kesen düzlemler (cutting planes) kullanan bir tamsayı programlama algoritması önermişlerdir. 15 talep noktası ve 270 olurlu güzergahı içeren bir örnek problem 23 iterasyon ile çözülmesine karşın benzer bir problemde 200 iterasyon ile bir netice elde edilememiştir.

Foster ve Ryan (1976) depodan kaynaklanan bir ışınlı düğümler düzlemini tarayarak radyal konumlu noktaları birbirine bağlayıp daha olası güzergahlar yaratma yoluna gitmişlerdir. Bu sezgisel yaklaşımla tamsayı programlamanın birleştirilmesi başarılı olmuş ve 100 düğüme kadar problemler çözülmüştür.

(P1) modeli değişik taşıt tiplerini içerdüğinden bu formülasyonun da bu özellik için nasıl değiştirilmesi gerektiğini göstermek bütünlük açısından yararlı olacaktır.

X_p , p taşıtı için olurlu güzergahlar kümesini göster-sin. Değişik taşıt tipi uzantısını (P2) ye dahil etmek için (2.12) kısıt kümesi (P2) ye eklenir.

$$\sum_{j \in X_p} x_j = 1 \quad (2.12)$$

Böylece her taşıtın bir güzergah grubuna atanması sağlanır. (2.11) ile gösterilen taşıt kapasitesi kısıtı taşıt bağımlı olduğundan şöyle ifade edilir.

$$\sum_{i=1}^N a_{ij} d_j \leq C_p \quad \forall j \in X_p \quad (2.13)$$

2.3.3. Taşıt Akışına Dayandırılmış Model

Taşıt akışına dayandırılmış formülasyon gezgin satıcı probleminin bir uzantısıdır. Golden v.d. (1977) tarafından ve-rilen model burada (P3) olarak nitelendirilmiştir.

(P3) Modelde,

$$\sum_{i=1}^N \sum_{p=1}^P x_{ijp} = 1 \quad \forall j, j \neq 1 \quad (2.14)$$

$$\sum_{i=1}^N x_{ijp} - \sum_{i=1}^N x_{ijp} = 0 \quad \forall j, p \quad (2.15)$$

$$\sum_{i=1}^N D_i \left(\sum_{j=1}^N x_{ijp} \right) \leq C_p \quad \forall p \quad (2.16)$$

$$\sum_{j=2}^N x_{1jp} \leq 1 \quad \forall p \quad (2.17)$$

$$x_{ijp} = (0,1) \quad \forall i,j,p \quad (2.18)$$

$$X \in S \quad (2.19)$$

kısıtları altında

$$Z = \sum_{i=1}^N \sum_{j=1}^N \sum_{p=1}^P c_{ijp} x_{ijp} \quad (2.20)$$

amaç işlevi enküçüklenecektir.

Burada

$X =$ elemanları $x_{ij} = \sum_{p=1}^P x_{ijp}$ olan bir matrisdir ki taşıt

tipinden bağımsız olarak bağlantıları verir.

$S =$ alttutur önleme kısıtları kümesidir.

(2.14) numaralı kısıtlar kümesi her talebin sadece ve sadece tek bir taşıtla karşılanmasını sağlamaktadır. (2.15) numaralı kısıt kümesi eğer bir taşıt bir düğüme gelirse bu düğümden çıkması gerektiğini ifade etmektedir. (2.16) numaralı kısıt kümesi taşıtlar üzerine kapasite sınırıdır. (2.17) numaralı kısıtlar eldeki mevcut taşıt sayısının aşılmasını sağlarlar. (2.18) karar değişkenleri üstüne olan kısıtları, (2.19) ise altturları önleyici kısıtları verir.

Altturları önleme kısıtları S için değişik seçenekler vardır. Bunlar:

$$S = \{(x_{ij}) \mid \sum_{i \in Q} \sum_{j \notin Q} x_{ij} \geq 1 \text{ bütün boş olmayan düğüm uygun alt-} \\ \text{kümeleri } Q \text{ için}\} \quad (2.21)$$

$$S = \{(x_{ij}) \mid \sum_{i \in Q} \sum_{j \in Q} x_{ij} \leq |Q|-1, \{2,3,\dots,N\} \\ \text{kümesinin bütün boş} \\ \text{olmayan altkümeleri} \\ \text{için}\} \quad (2.22)$$

$$S = \{(x_{ij}) \mid y_i - y_j \cdot N x_{ij} \leq N-1, 2 < i \neq j \leq N \text{ ve bazı gerçektek} \\ \text{sayılar } y_i \text{ için}\} \quad (2.23)$$

Görüldüğü gibi (2.21) ve (2.22) yaklaşık 2^N alttur önleme kısıtı içermesine rağmen, Miller v.d. (1960) tarafından önerilen (2.23) sadece $(N^2 - 3N + 2)$ kısıt içermektedir. Uygulanacak çözüm yöntemine göre bu üç seçenekten biri seçilir.

(P3) formülasyonunda bir düğüme tek bir kere gidişle talebin karşılanacağı varsayılmıştır. Buna karşın (P1) de bir düğüme birden fazla gidiş izni verilmektedir

Dikkat edilirse (P3) de taşıt sayısı $P=1$ olduğunda kısıt kümeleri (2.15) ve (2.16) gereksiz olmaktadır. Burada taleplerin toplamının tek taşıtın kapasitesi üstünde olmadığı kabul edilmektedir ki aksi takdirde problem yanlış tanımlanmış olur. Bu kısıtların elenmesi ile problem bir gezgin satıcı problemine dönüşür.

Bu altbölümde NTGB problemine yapılan üç modelleme yaklaşımı tanıtılmıştır. Genelde modelleme yaklaşımlarına temel teşkil eden bu modellere çözüm yaklaşımları daha ileride verilecektir.

2.3.4. Ayrıtlara Hizmet Verilen TGB Problemi Formülasyonu (ATGB)

Burada (P4) olarak sunulacak olan formülasyon Golden ve Wong (1980) tarafından verilmiştir.

(P4) Modelde,

$$\sum_{j=1}^N x_{ijp} - \sum_{j=1}^N x_{jip} = 0 \quad \forall i, p \quad (2.24)$$

$$\sum_{p=1}^P (h_{ijp} + h_{jip}) = \left[\frac{d_{ij}}{C_p} \right] \quad \forall (i, j) \in A \quad (2.25)$$

$$x_{ijp} > h_{ijp} \quad \forall (i, j) \in A, \forall p \quad (2.26)$$

$$\sum_{i=1}^N \sum_{j=1}^N h_{ijp} d_{ij} \leq C_p \quad \forall p \quad (2.27)$$

$$\sum_{j=1}^N f_{ijp} - \sum_{j=1}^N f_{jip} = \sum_{j=1}^N h_{ijp} \quad \forall i, p; i \neq 1 \quad (2.28)$$

$$f_{ijp} \leq N^2 x_{ijp} \quad \forall (i, j) \in A, \forall p \quad (2.29)$$

$$x_{ijp} = (0, 1), h_{ijp} = (0, 1), f_{ijp} \geq 0 \quad \forall i, j, p \quad (2.30)$$

kısıtları altında,

$$Z = \sum_{i=1}^N \sum_{j=1}^N \sum_{p=1}^P c_{ijp} x_{ijp} \quad (2.31)$$

amaç işlevi enküçüklenecektir.

Burada:

A = hizmet talep eden ayrıtların kümesi

$d_{ij} = (i,j)$ ayrıtlarının talebi

$h_{ijp} = \begin{cases} 1, & \text{eğer taşıtı } (i,j) \text{ ayrıtlarına hizmet veriyorsa} \\ 0, & \text{aksi durumlarda} \end{cases}$

$f_{ijp} = (i,j)$ ayrıtlarında p taşıtı için akış değişkeni (sıfırdan büyük değer alması ancak $x_{ijp}=1$ olduğu zaman mümkündür.)

[Z] = Z değerinden büyük veya eşit enküçük tamsayıdır.

(2.24) numaralı kısıtlar güzergahın devamlılığını sağlarlar. (2.25) numaralı kısıtlarda sıfırdan büyük talebi olan ayrıtlara ancak bir defa hizmet verilmesi gerektiği belirtilir. (2.26) numaralı kısıtlar ise bir ayrıta hizmet verilebilmesinin ancak o ayrıttan bir taşıtın geçmesi ile mümkün olduğunu ifade eder. (2.27) numaralı kısıtlar taşıt kapasite kısıtlarıdır. (2.28) ve (2.29) numaralı kısıtlar geçersiz altturların olmasını müştereken önlerler. Geçersiz altturlar depoyu yani 1.noktayı içermeyen altturlardır. Sıfırdan büyük olma ve tamsayı olma kısıtları (2.30) da verilmişlerdir.

Yapılan formülasyon yönsüz serimler içindir. Ancak yönlü serimlere de uzatılabilir.

(P4) formülasyonunun çözümü için Golden ve Wong (1980) tarafından bir sezgisel yöntem önerilmiştir. Bu yöntemde önce her talep düğümü için ayrı bir tur yaratılır, sonra bu turların en büyüğünden başlayarak küçük turlardaki ayrıtların büyük tur içinde hizmet verilip verilemeyeceği araştırılır. En büyük tasarrufu sağlayan turlar birleştirilir. Ancak bu arad taşıt kapasitesi gözönüne alınır. Aynı probleme diğer bir çözüm yöntemi Christofides (1973) tarafından önerilmiştir.

2.3.5. Nokta ve Ayrıtlara Hizmet Verilen TGB Problemi (GTGB)

Bir serim üzerinde mevcut nokta ve ayrıtların birer altkümesine hizmet götürülen problemler ilk defa Orloff (1974a) tarafından incelenmiştir. Orloff bu problemlere genelleştirilmiş taşıt güzergahı belirleme problemi (GTGB) demiştir. Problemin açık bir matematiksel programlama formülasyonu verilmemiştir. Yapılan yaklaşımda serim üzerinde uygulanan bir dönüşümle geçilmesi zorunlu düğümler seçilmesi zorunlu ayrıtlar haline getirilir ve yeni serim üzerinde bir Euler turu aranır. Dönüşüm sırasında bazen birden fazla geçilmesi zorunlu düğüm tek bir geçilmesi zorunlu ayrıta indirgenir. Dönüşüm neticesi, özellikle nokta taleplerinin baskın olduğu problemlerde problem büyüklüğünde önemli azalmalar meydana gelebilir. Yaklaşımın temelinde alttur önleme algoritması vardır. Bu algoritmada 1,2-eşleme (1,2-matching) problemi Edmonds'un b-eşleme algoritması kullanılarak çözülmüştür. Yönlü ve yönsüz serimlerde bu yaklaşımın başarılı olduğu gösterilmiş, ancak hem yönlü hem yönsüz ayrıtlar içeren ve karışık olarak nitelenen serimlerde Euler turu bulmanın zorlukları nedeni ile aynı başarı tekrarlanamamıştır.

Orloff (1974b) GTGB problemini m adet taşıt durumu içinde çözmüştür. Önerilen çözüm yöntemi dört aşamalıdır. Önce m adet taşıtlı GTGB problemi tek taşıtlı GTGB problemine dönüştürülür. Alttur önleyici kısıtları da içeren bir eşleme problemi çözülür. Bu çözüm kullanılarak Euler turu içeren bir serim oluşturulur. Son aşamada bu Euler turu bulunur. Bulunan Euler turu GTGB probleminin ve dolayısı ile m adet taşıtlı GTGB probleminin eniyi çözümünü verir. Karışık serimlerde tek tur problemlerinde yenilemeyen güçlükler burada da halledilememişlerdir.

2.3.6. TGB Formülasyonlarında Bazı Uzantılar

2.3.6.1. Çok Depolu TGB Problemleri

Depoları, $1, 2, \dots, M$ noktaları olarak gösterelim ve bunlar nokta dizisinin ilk M elemanını oluştursun. (P1) formülasyonunda i ve j indisleri $(1, 2, \dots, N)$, s indisi ise $(M+1, M+2, \dots, N)$ sınırları arasında tanımlansın. Bu durumda formülasyon çok depolu formülasyondur. (P2) formülasyonunda çok depolu durum değişik depolardan turların yaratılması ve küme parçalama algoritmasının her depodaki kaynak ve taşıt sayısı kısıtlarını gözönüne alacak şekilde değiştirilmesi ile mümkündür. (P3) formülasyonunda ise (2.14) kısıt kümesinde j indisi ($j = M+1, \dots, N$) olarak değiştirilir. (2.17) kısıtları ise şöyle yazılır:

$$\sum_{i=1}^M \sum_{j=M+1}^N x_{ijp} \leq 1 \quad \forall p \quad (2.32)$$

Ayrıca altturları önleme kısıtları kümesi S için verilen seçeneklerde şu değişiklikler yapılır: (2.21) de Q uygun altkümelerinin $1, 2, \dots, M$ düğümlerini içermesi istenir. (2.22) de Q altkümesinin de $\{M+1, M+2, \dots, N\}$ kümesinin boş olmayan bir altkümesi olması istenir. (2.23) de ise i ve j indisleri için aralık sınırları $(M+1 \leq i \neq j \leq N)$ şeklinde alınır.

ATGB ve GTGB problemlerinde çok depolu durum açıkça formüle edilmemiştir. Bu problemlerde düğümleridepolara atayarak gruplandırmak ve problemi böylece tek depo durumuna indirgeyip çözmek olurlu bir yaklaşım olarak gözükmektedir.

2.3.6.2. Güzergah Süresi Üzerine Kısıt

Güzergah süresi üzerine kısıt şu şekilde ifade edilebilir.

$$\sum_{i=1}^N t_{ip} + \sum_{j=1}^N x_{ijp} + \sum_{i=1}^N \sum_{j=1}^N t_{ijp} x_{ijp} \leq T_p \quad \forall p \quad (2.38)$$

Burada:

t_{ip} = p taşıtı için i noktasında işlem süresi

t_{ijp} = p taşıtı için i noktasından j noktasına seyahat süresi

T_p = p taşıtının olurlu çalışma süresidir.

(2.38) kısıtları (P1), (P3) ve (P4) formülasyonlarına eklenirler. Problem büyüklüğünü fazla etkilemezler. (P2) formülasyonunda ise olurlu güzergahların bulunmasında gözönüne alınır. GTGB probleminde ise alttur eliminasyonunda bu kısıtı ihlal eden turlar da alttur olarak tanımlanarak kısıt gözönüne alınmış olur.

2.3.6.3. Zaman Aralıkları Kısıtları

j talep noktasında yapılacak bir işlem b_j zamanında yapılmalı. Bu teslimin en erken $\underline{b}_j$ ve en geç $\bar{b}_j$ zamanında yapılması gereksin. Bu durumda bu işlem üzerindeki kısıt şöyle ifade edilir.

$$b_j = \sum_{p=1}^P \sum_{i=1}^N (b_i + t_{ip} + t_{ijp}) x_{ijp} \quad \forall j \neq 1 \quad (2.33)$$

$$b_1 = 0 \quad (2.34)$$

$$\underline{b}_j \leq b_j \leq \bar{b}_j \quad (2.35)$$

(2.26) numaralı kısıtlar doğrusal olmadığından problemi oldukça zorlaştıracaklardır. Bu nedenle bir dönüşüm uygulanarak doğrusal duruma getirilir:

$$b_j \geq (b_i + t_{ip} + t_{ijp}) - (1 - x_{ijp})T \quad \forall i, j, p; j \neq 1 \quad (2.36)$$

$$b_j \leq (b_i + t_{ip} + t_{ijp}) + (1 - x_{ijp})T \quad \forall i, j, p; j \neq 1 \quad (2.37)$$

Burada $T = \text{enb } T_p$, yani T taşıtlar için en uzun çalışma süresini belirtir. (2.34), (2.35), (2.36) ve (2.37) kısıtları birlikte zaman aralıkları kısıtlarını oluştururlar. Bu kısıtlar (P1), (P3) ve (P4) formülasyonlarında kısıt sayısında önemli artışa neden olurlar. (P2) formülasyonunda ise olurlu güzergah bulunması daha çok işleme gerek gösterecektir. GTGB problemi ise daha önce belirtilen yeni alttır tanımlaması ile bu kısıtlar gözönüne alınır.

2.3.7. TGB Problemlerine Temel Teşkil Eden Çekirdek Problemler

2.3.7.1. Noktalara Hizmet Götürülen Durum

(P1), (P2), (P3) formülasyonlarından görüldüğü gibi NTGB problemleri temelde gezgin satıcı probleminin birer uzantısıdır. Bu nedenle NTGB problemlerinde gezgin satıcı problemi çekirdek problem olarak nitelendirilecektir.

Gezgin satıcı problemi şöyle tanımlanabilir: Bir serimdeki N düğümü bir ama sadece bir defa geçen en küçük maliyetli turu bulunuz.

Gezgin satıcı problemi yöneylem araştırmasının en çok ilgi çeken ve üzerinde en çok çalışılan problemlerinden biridir. Probleme önerilen çözüm yöntemleri üzerinde burada durmayacaktır. Lawler ve Wood (1966), Bellmore ve Nemhauser (1969) ve Bellmore ve Malone (1971) yaptıkları taramalarla problemi 1970 başına kadar kapsamaktadırlar. Eilon v.d. (1971) gezgin satıcı problemine geniş yer ayırmaktadırlar. Golden ve Magnan (1977) ise yayın tarihine kadar olan gelişmeler için kapsam bir kaynak listesi sunmaktadırlar.

Gezgin satıcı problemi üzerinde yoğunlaşan çalışmalarda "hesap güçlüğü" (computational complexity) denilen bir kavram yol açmış (Karp, 1972) ve hesap güçlüğü 1970 lerin özellikle ikinci yarısından itibaren algoritma tasarımlarını etkileye

en önemli unsur olmuştur. Bu konuda geniş kapsamlı bir tarama Garey ve Johnson (1979) tarafından verilmektedir. Bir algoritmanın çözüme varmak için yapacağı işlemlerin problemin bir boyutu olan n cinsinden ifade edilen bir polinom $P(n)$ ile üstten sınırlanması mümkünse bu takdirde bu algoritma için, polinom zamanda çözülebilir denir. Buna mukabil bu üstsınır ancak $2^{P(n)}$ şeklinde ifade edilebiliyorsa bu takdirde bu algoritma üssel zamanda çözülebilir denir. Hesap güclüğü konusunda yapılan çalışmalar neticesinde problemler hesap güçlüklerine göre iki sınıfa ayrılmışlardır: (i) polinom zamanda çözülenler-P, ve (ii) polinom zamanda çözülemeyenler-NP. Polinom zamanda çözülemeyenler arasından belirli bir grup problem NP-bütün (NP-complete) sınıfını oluştururlar. Bu sınıfın özelliği, NP deki bütün problemlerin verimli bir dönüşümle bu problemlere indirgenebilir olmasıdır. Buna göre NP-bütün bir probleme polinom zamanda bir çözüm bulunsa bütün NP problemlere polinom zamanda çözüm bulunmuş olacaktır. Ancak genel kanı $P \neq NP$ dir (Garey ve Johnson, 1979).

Gezgin satıcı problemindeki tek satıcı m adet satıcıya çıkarıldığında problem özellikle TGB problemleri açısından daha ilginç bir duruma gelir. m adet satıcı durumuna bir çözüm yöntemi Eİloñ v.d. (1971) tarafından sunulmuştur. Burada serime $(m-1)$ adet düğüm eklenerek standart gezgin satıcı problemi bu serim üzerinde çözülmektedir. Genişletilmiş serim üzerinde diğer çözüm yöntemi Bellmore ve Hong (1974) tarafından verilmiştir. m adet taşıt içeren TGB problemleri için bu problem bir çekirdek problemdir ve amaç işlevi değeri bu tip TGB problemlerine bir altsınır teşkil eder.

Gezgin satıcı problemi sabit maliyetlerde eklenerek daha kapsamlı duruma getirilmiştir. Hong ve Padberg (1977) simetrik gezgin satıcılar problemine her i gezgin satıcısı için f_i sabit maliyetinin eklendiği problemlere bir çözüm getirmişlerdir. m adet gezgin satıcının artan sabit maliyet sıra-

sına göre numaralandırıldığı problem formülasyonunda N adet olan düğüm sayısı $(N+m+4)$ e çıkartılarak bu serim üzerinde standart gezgin satıcı problemi çözülmüştür. Discenza (1981) değişik bir dönüşüm uygulayarak bu ek 4 düğüme gerek olduğunu göstermiştir. Sabit maliyetlerin de göz önüne alındığı TGB problemlerinde bu problem çekirdek problemdir ve amaç işlevi değeri bu tip TGB problemlerine altsınır teşkil eder.

2.3.7.2. Ayrıtlara Hizmet Götürülen Durum

ATGB probleminin çekirdek problemleri Çinli postacı problemi ve kentli postacı problemidir.

Çinli postacı probleminde bir serim üzerindeki bütün ayrıtları en az bir defa geçen en küçük maliyetli tur aranır. İlk kez Kwan (1962) tarafından ortaya atılan probleme yine Kwan'ın önerdiği çözümde eniyi çözümü garanti etmek için bütün turları bulmak gerekmektedir. Bu ise büyük serimlerde pratik bir yol değildir. Edmonds (1965) yönsüz serimlerde bir Euler turu olabilmesi için bütün düğümlerle çift sayıda ayrıttan çakışması gerektiği gözleminden hareketle en kısa yol problemi ve 1-eşleme problemini çözümüne dayanan bir yöntem önermiştir. En kısa yol ve 1-eşleme yöntemleri polinom zamanda çözüldüklerinden bu yöntem de polinom zamanda çözülür. Yönlü serimlerde Çinli postacı turunun bulunması için en kısa yol ve ulaştırma problemlerinin çözüldüğü bir yöntem Edmonds ve Johnson (1973) tarafından verilmiştir. Bu problem de polinom zamanda çözülebilen bir problemdir. Karışık serimlerde Çinli postacı problemi ise polinom zamanda çözülemeyen bir problemdir (Frederickson v.d. 1978). Bir çözüm yöntemi Edmonds ve Johnson (1973) tarafından verilmektedir. Bu problemin ayrıntılı bir incelemesi Minieka (1979) tarafından yapılmıştır.

m adet Çinli postacı probleminde ise genellikle iki yoldan biri izlenir: (i) serim altserimlere bölünür ve her bir altserim tek başına bir Çinli postacı problemi şeklinde çözülür; (ii) tek bir Çinli postacı problemi çözülerek bir ana tur bulunur ve bu m adet alttura bölünür (Beltrami ve Bodin, 1974). Bu yaklaşımlardan hangisinin daha uygun olacağı problemin yapısından bağımlı olmaktadır. Gerek yönlü gerek yönsüz m adet Çinli postacı probleminin polinom zamanda çözülemez olduğu gösterilmiştir (Frederickson, 1978).

Kentli postacı problemi Orloff (1974a) tarafından tanımlanmıştır. Bu problemde serim üzerindeki ayrıtların bir altkümesini teşkil edenler en az bir defa geçilmek zorunludur. Yönlü ve yönsüz kentli postacı probleminde Çinli postacı probleminde uygulanan yöntemler kullanılır. Ancak geçilmesi zorunlu ayrıtların oluşturduğu serimin her zaman bağılı olmayabilişi çözümü zorlaştıran önemli bir unsur olmaktadır. Yönlü ve yönsüz kentli postacı problemlerinin polinom zamanda çözilemeyeceği Lenstra ve Rinnooy Kan (1976) tarafından gösterilmiştir.

Çekirdek problemlerin çözümünde görülen güçlükler işletme ve özel kısıtların da probleme eklenmesi nedeni ile TGB problemlerinde daha da büyüyecektir. Bunun tabii bir neticesi olarak TGB problemi genelde polinom zamanda çözülemez bir problemdir (Golden v.d. 1981).

2.4. TGB PROBLEMLERİNİN İKİ ÖNEMLİ UZANTISI

2.4.1. Taşıt Sayısı Problemi

TGB probleminde amaç işlevi değişken maliyetten oluşur. Sabit maliyetler ise gözönüne alınmamaktadır. Ancak sabit maliyetin de önemli bir unsur olduğu durumlarda bu mali-

yetin de deęişken maliyetle birlikte gözönüne alınması gerekir. Bu durumda taşıt sayısı bir karar deęişkeni olur. Taşıt kapasiteleri bu problemde bütün taşıtlar için aynı kabul edilir. Taşıt kapasitesi bir parametre kabul edilerek toplam maliyetin enküçüklendięi taşıt büyüklüğü aranabilir. Bir kaç mükün taşıt kapasitesi için problem defaeten çözülür ve çözümler arasından enküçüğü seçilir. Ancak taşıt kapasitesini enküçük maliyet dışında da belirleyen unsurlar vardır. Örneğin taşınan malın nitelięi, fiziki kısıtlar, v.s. gibi böyle durumlarda kapasite seçimi ayrı bir problem olarak ele alınabilir (Eilon v.d., 1971).

Taşıtlar kiralama veya satın alma yolu ile filoya dahil edilirler. Kiralanan ve satın alınan kapasiteler arasındaki denge özellikle müşteri talebindeki kararlılıktan bağımlıdır. Bu dengenin ne noktada kurulacaęı taşıt sayısı probleminin bir parçası deęildir. Ancak taşıt kapasitesinde olduđu gibi bir parametrik çalışma bu konuda karar vericiye yeterli bilgi sağlayabilir.

p taşıtı için sabit maliyeti F_p ile gösterirsek, (P1) ve (P3) formülasyonlarında amaç işlevine aşığıdaki sabit maliyet bileşeni eklenir.

$$F = \sum_{p=1}^P F_p \sum_{j=1}^N x_{ijp} \quad (2.38)$$

Bu formülasyonda P alabileceęi en yüksek deęer verilerek çözülebilir. Ancak bu yol problemin boyutlarını oldukça büyütebilir. Bu nedenle P parametrik olarak alınıp deęişik deęerleri için problem çözülür ve bu çözümler içinde enküçük toplam maliyetli olanı seçilebilir.

(P2) formülasyonunda ise sabit maliyet bileşeni güzergah maliyeti C_j içine alınarak problem aynı yöntemle çözülür.

Probleme bir sezgisel çözüm yöntemi Eilon v.d. (1971) tarafından önerilmiştir. Yöntem NTGB problemleri için geliştirilmiştir. Önce bir yükleme problemi (loading problem) çözümlere talepleri karşılayacak en küçük taşıt sayısı bulunmaktadır. Sonra salt değişken maliyetlere dayanılarak en kısa toplam tur uzunluğunu veren tur ve bu tura karşı gelen taşıt sayısı bulunmaktadır. Son aşamada bu iki taşıt sayısı arasında en küçük toplam maliyeti veren taşıt sayısı tespit edilmektedir.

2.4.2. Taşıt Sayısı ve Kapasiteleri Problemi

Bu problemin taşıt sayısı probleminden farkı taşıt kapasitelerinin aynı olması koşulunun burada kaldırılmış olmasıdır. NTGB problemleri için Eilon v.d. (1971) bir tamsayı programlama modeli kurmuşlar ancak çözüm getirmemişlerdir. Aynı probleme iki değişik sezgisel yaklaşım Levy v.d. (1980) tarafından önerilmiştir. Birincisinde tur birleştirmeye ilişkin bir tasarruf kavramı tanımlanıp kullanılmaktadır. İkincisinde ise bir ana tur oluşturulmakta ve sonra taşıt kapasitelerine ve diğer kısıtlara uygun bir biçimde altturlara bölünmektedir.

2.5. ÇÖZÜM YÖNTEMLERİ

2.5.1. Giriş

Hem teori hem pratik yönlerden çekiciliği nedeni ile TGB problemine büyük ilgi gösterilmiş ve çok sayıda yayın yapılmıştır. Bu nedenle burada sadece yaklaşımların bir gruplandırılması yapılacak ve sadece temel nitelikteki katkılara yer verilecektir.

Çözüm yöntemleri için bir gruplandırma şöyle yapılabilir:

- i. Tek depolu TGB problemleri için sezgisel yöntemler
- ii. Çok depolu TGB problemleri için sezgisel yöntemler
- iii. Kesin çözüm yöntemleri
- iv. Etkileşimli yöntemler

Algoritmaların çözüm süresi karşılaştırmalarına bu verilerin çok değişik şartlarda toplanmış olmaları nedeni ile burada girilmeyecektir.

2.5.2. Tek Depolu TGB Problemleri İçin Sezgisel Yöntemler

NTGB problemlerinde dört temel algoritma tanıtılacaktır. Birinci algoritma Clark ve Wright (1964) algoritmasıdır ve tanımladıkları bir tasarruf kavramı üzerine kurulmuştur. Bütün düğümler doğrudan doğruya depoya bağlanarak başlangıç çözümü elde edilir. İki düğümün ayrı ayrı iki taşıt kullanılarak taleplerinin karşılanması yerine bunun tek taşıtla yapılmasının olurluluğu araştırılır. (i,j) ayrıtının uzunluğunu U_{ij} ile gösterirsek,

$$S_{ij} = U_{1i} + U_{1j} - U_{ij} \quad (\text{simetrik}) \quad (2.39a)$$

$$S_{ij} = U_{i1} + U_{1j} - U_{ij} \quad (\text{simetrik değil}) \quad (2.39b)$$

iki taşıttan bir taşıta geçişin tasarrufudur. S_{ij} değerleri bütün i,j kombinasyonları için hesaplanır. Sıfırdan büyük tasarruflar büyükten küçüğe doğru sıralanarak tasarruf listesi elde edilir. Tasarruf listesinin başından başlayarak olurlu değişiklikleri yapılarak aşağıya doğru inilir. Bir kere yapılan değişiklik bir daha geri alınmaz. Tasarruf listesinin bütün elemanları incelendikten sonra durulur.

Clark ve Wright algoritmasının zayıf tarafı "unutkan" olmasıdır. Bu özellik bir kere çözüme alınan bir ayrıtın

bir daha çözümden çıkartılamamasından kaynaklanır. Tillman ve Cochran (1968) ancak birlikte enbüyük tasarrufu veren ayırıt çiftleri seçmeyi önermişlerdir. Bu üç veya daha fazlaya çıkartılabilir. Gaskell (1967) ve Yellow (1970) tasarruf tanımına sıfırdan büyük bir ölçek parametresi α getirmişlerdir. Buna göre, (2.39.a) ve (2.39b) de U_{ij} ölçek parametresi α ile çarpılır. Golden v.d. (1977) ölçek parametresi ve bazı bilgisayar programlama teknikleri kullanarak Clark ve Wright algoritmasında büyük gelişme sağlamışlardır. Tasarruf listesinde sıradaki tasarrufun bir çözüme dahil edilip öbürüne edilmediğine göre dallanan bir dal-sınır ağacı yaklaşımı Holmes ve Parker (1976) tarafından verilmiştir.

Tyagi (1968) tarafından önerilen algoritmada depo dışında bir noktadan başlanarak o nokta kendisine en yakın noktaya birleştirilir. Bu işlem sırasında kapasite ve güzergah uzunluğu kısıtları kontrol edilir. Bu şekilde talep noktaları gruplandırılmış olur. Böyle her grup için bir gezgin satıcı problemi çözümlenerek turlar elde edilir.

Gillet ve Miller (1974) talep noktalarını gruplandırmak için bir tarama yöntemi uygulamışlardır. Buna göre depo ve rassal olarak seçilen bir talep noktası arasına bir çizgi çekilir. Depo merkez alınarak bu çizgi saat yönünde veya ters yönünde çevrilerek çizginin üzerinden geçtiği talep noktaları kapasite kısıtına varılana kadar aynı taşıta atanırlar. İlk taşıt dolduktan sonra çizginin geçtiği noktalar ikinci taşıta atanırlar. Bu işlem tüm noktalar bir taşıta atanana kadar sürer. Daha sonra çizginin ilk yönünün tersine döndürülmesi, çizginin rassal seçilen uç noktasının yeniden seçilmesi gibi değişiklikler uygulanarak değişik çözümler aranır. Bir taşıta atanan noktaları bağlayan eniyi turu bulmak için gezgin satıcı problemi uygulanır.

Christofides v.d. (1969) tarafından önerilen yöntem Lin'in

(1965) gezginci satıcı problemi çözüm yönteminden yararlanmaktadır. Burada "r-eniyi" kavramı getirilmektedir. Örneğin, 2-eniyi çözüm, o çözümde ayırtların ikili halinde değiştirilmesinin daha iyi bir olurlu çözüm yaratılamayacağını ifade eder. Önerilen yöntem bir olurlu başlangıç çözümü ile başlayarak r-eniyi çözümü arar. Burada r bir parametredir. Başlangıç çözümünün daha önce anlatılan sezgisel yöntemlerden biri ile elde edilmesi ilginç bir birleşim yaratabilir.

ATGB problemlerinde daha önce belirtilen Christofides (1973) ve Golden ve Wong (1980) dışında bu konuda yapılan çalışmalar problem özelliklerine bağlı uygulamalar şeklinde olmuştur. Örneklemek gerekirse, Bodin ve Kursh (1978,1979), Stern ve Dror (1979) ve Beltrami ve Bodin (1974) gösterilebilir.

2.5.3. Çok Depolu TGB Problemleri İçin Sezgisel Yöntemler

Bu problem tek depolu problemin olduğu kadar incelenmiş değildir. İki temel yaklaşım uygulanmıştır.

Birinci yaklaşımda olurlu bir başlangıç çözümü ile başlanmakta ve çözümün geliştirilmesi için turlar arasında nokta alışverişi olanakları aranmaktadır. Bu değişikliklerin olumlu olup olmadığı denetlenmektedir. Bu tür yaklaşım örnekleri arasında Wren ve Holladay (1972) ve Cassidy ve Bennett'in (1972) çalışmalarını gösterebiliriz.

İkinci yaklaşımda ise önce talep noktaları gruplandırılmakta sonra her grup için bir tek depo problemi uygulanmaktadır. Gillet ve Johnson (1976) bu yaklaşımı kullanmışlar ve ikinci aşamada Gillet ve Miller (1974) algoritmasından yararlanmışlardır. Birinci aşamada her noktanın en yakın olduğu depoların oranı kullanılır. Bu oranın bire yakın olduğu noktaların ataması daha dikkatle yapılır. Tillman ve Cain (1972)

i ve j noktalarını k deposuna bağlı olacak şekilde birleştirmenin getireceği tasarrufu ve birleştirmemenin yaratacağı zararı tanımlarlar. Her noktanın kendisine en yakın depoya bağlandığı başlangıç çözümüne Clark ve Wright algoritmasını bu tasarruf ve cezanın bir ağırlıklı ortalamasını kullanarak uygularlar. Golden v.d. (1977) önerdikleri ilk çok depolu TGB algoritmasında Tillman ve Cain'in sadece tasarruf kavramını kullanmaktadırlar. En büyük katkıları getirdikleri bilgisayar programlama verimliliğidir. İkinci algoritmalarında ise noktalar en yakın oldukları iki depoya olan uzaklığın oranının belirli bir parametreden büyük olması halinde sınır noktaları olarak tanımlanmaktadırlar. Birinci aşamada sadece sınır noktaları gözönüne alınarak birinci algoritma uygulanmaktadır. İkinci aşamada diğer noktalar en yakın depoya atanmakta ve her depo için TGB problemi çözülerek bu noktalar da güzergahlara dahil edilmektedir. Elde edilen sonuç üçüncü aşamada bir irdeleme sürecinden geçirilmektedir.

2.5.4. Kesin Çözüm Yöntemleri

TGB problemlerinde kesin çözüm yöntemleri henüz oldukça geridedir. NTGB problemlerinin çekirdek problemi olan gezgin satıcı problemine 1318 noktaya kadar kesin çözüm bulunduğu halde (Crowder v.d., 1980) NTGB problemine ancak 10-12 nokta için kesin çözüm bulunabildiği görülmektedir. (Eilon v.d., 1971). Ancak bazı özel problemler için 31 talep noktasına kadar çözüm bulunmuştur (Christofides, 1976). Kesin çözümlerde bir yaklaşım Christofides v.d. (1981) tarafından verilmektedir. Burada değişik dal-sınır algoritmaları önerilmektedir. Bu algoritmalarda kullanılan sınırlar (i) önceden belirlenmiş bir noktada sabit bir kertes (degree) olan bir en kısa kapsar ağaç (spanning tree), (ii) en küçük q-güzergahı, $q = 1, \dots, Q$. Bu güzergahlar üstündeki yük kesin q miktarındadır. Bu sınırların hesaplanmasında Lagrange ceza yordamı (penalty procedure) kullanılmıştır. Algoritma 25 talep nokta

sına kadar kesin çözümü makul bilgisayar sürelerinde vermek-
tedir.

2.5.5. Etkileşimli Yöntemler

Etkileşimli yöntemlerin ilk başarılı örneği Krolak v.d. (1972) tarafından verilmiştir. Yöntem, (i) iyi bir olur-lu başlangıç çözümü ve (ii) etkileşim aşaması olmak üzere iki aşamalıdır. Başlangıç çözümünü elde etmek için gezgin satıcı problemi ve ulaştırma problemi kullanılmaktadır. Bu sezgisel çözüm oldukça başarılı bir başlangıç çözümü oluşturmaktadır. Etkileşim safhasında kişi kendi deneyimlerini çözüme aktarabildiğinden bu yaklaşım türü pratik uygulamalarda büyük yarar sağlayacaktır. Ancak gelişmesi bilgisayar donanımlarındaki gelişmeden bağımlıdır. Bu konuda çalışmalar giderek artmaktadır (Örneğin Baker, 1981).

2.6. BAZI ESKİ VE YENİ PROBLEM ALANLARI

Bu altbölümde bazı eski ve yeni problem alanlarına dikkat çekilecektir. Bunlar burada bir sıra gözetmeksizin sıralanacaklardır. Bu liste tüm problem alanlarını kapsamaktan ziyade örneklemek amacını gütmektedir.

Bu bölümde ele alınan problemlerde değişken maliyetler doğrusal kabul edilmişlerdi. Ancak bu geçerli olmayabilir. Doğrusal olmayan değişken maliyetler özellikle kesin çözüm yöntemlerini olumsuz etkileyeceklerdir.

Amaç işlevinin salt bir enküçükleme değil enk enb(min max) veya enb enk (max min) operatörleri uygulanabilir. Örneğin en uzun güzergahın enküçüklenmesi, enbüyük ayırıt kapasite ihtiyacının enküçüklenmesi gibi. Bu tür problemlere henüz geçerli çözümler getirilmemiştir.

Bu bölümde verilen TGB problemlerinde yapılan varsayım taşıt kapasitesinin enbüyük hizmet talebinden daha büyük olduğudur. Ancak bu varsayımın geçerli olmadığı durumlar gerçekte vardır. Bu konuda özel çözümlerden genele geçiş üzerinde daha çalışılması gereken bir alandır.

Üzerinde fazla çalışılmamış diğer bir araştırma alanı birden fazla dönemi içeren problemlerdir.

TGB problemlerinde belirsizlik durumları (örneğin yol süreleri, taşıt sayıları gibi) henüz yeterince işlenmemiş oldukça geniş bir problem alanıdır. Henüz analitik yöntemlerin yeterince gelişmemiş olmasının yanında sağlıklı veri eksikliği de bu problemlere girilmesinde bir sorun yaratmaktadır. Belirsizlik sorunları genellikle benzetim (simulation) çalışmaları ile incelenmektedir.

TGB problemlerinde taşıt sayısı ve kapasiteleri problemi henüz üzerinde çalışılması gereken alanlardır. Özellikle ATGB problemlerinde yazarın bilgisi dahilinde böyle bir çalışma yoktur.

TGB problemlerinin diğer bazı problemlerle birlikte ele alınması problemlerin ayrı ayrı ele alınmasından daha yararlı olmaktadır; bazen de gerekli olmaktadır.

Örneğin taşıtların yükleme problemi böyle bir problemidir. Ne yüklerin ne miktarda ve bazı durumlarda hangi sırada yükleneceği bir çok dağıtım probleminde karşılaşılan sorunlardır. Hacim ve ağırlık boyutlarının yanında taşıtlara ilişkin bir de zaman boyutu vardır. Taşıt personelinin çalışma süreleri ve toplu iş sözleşmesinin buna ilişkin maddeleri, güzer-gahları doğrudan etkileyebilir. Taşıt yükleme ve TGB problemlerinin müştereken ele alınmaları ilginç bir araştırma alanı olarak gözükmektedir.

Yine böyle bir problem depo ve dağıtım merkezlerinin yerleştirilmesi sorunudur. Böyle bir problemde TGB problemi bir altproblem olarak ele alınabilir.

Bir dönem içinde oluşan talebin biriktirilerek dönemin belli zamanlarında karşılanması pratik bir sorundur. Bir talep düğümleri kümesinin parçalanarak her altkümenin dönemin belirli zamanlarına atanması ve TGB ile birlikte çözülmesi henüz üzerinde gelişmeler sağlanabilecek bir araştırma alanı olarak gözükmektedir.

Çok geniş ve o oranda da ilgi çeken bir problem taşıt güzergahı belirleme ve çizelgeleme problemlerinin müştereken ele alınmasıdır. Bazen personel çizelgelemesi ile de birleştirilen bu problem büyük bir araştırma potansiyeli içermektedir.

Bu çalışmanın bundan sonraki bölümlerinde burada belirtilen problem alanlarından ikisi için çözüm yöntemleri önerilecektir. Bunlardan birisi amaç işlevine (enk enb) operatörü uygulanan bir problemdir. Diğeri ise ATGB problemlerinde taşıt sayısı ve kapasiteleri problemdir.

3. ASKERİ UÇAKLARIN İNTİKAL PLANLAMASI İÇİN BİR KESİN ÇÖZÜM YÖNTEMİ

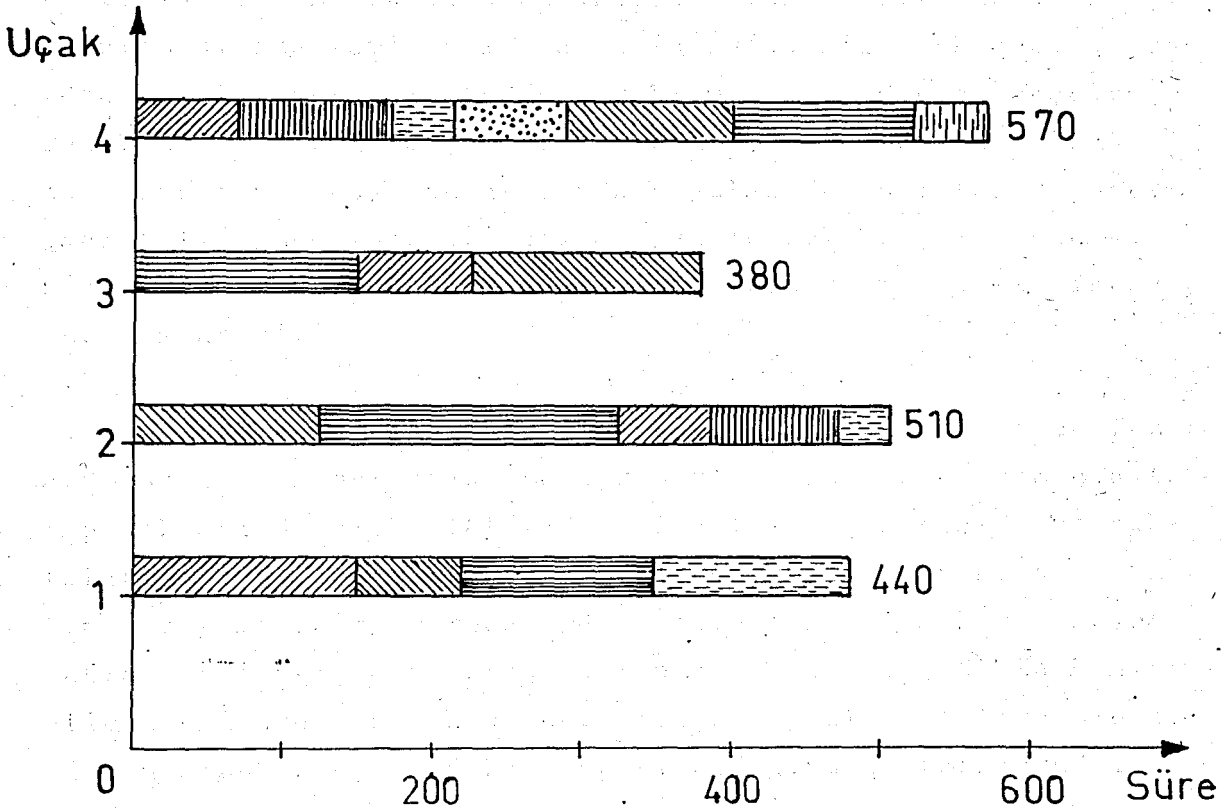
3.1. GİRİŞ

Bu bölümde askeri uçakların intikal planlaması için geliştirilmiş bir algoritma sunulacaktır. Kesin bir çözüm yöntemi olarak sunulan algoritma ayrıntıları ile verilecek, örneklenecek ve değerlendirilecektir.

Önce intikal planlamasının bir tanımını yapalım. İntikal planları, "ihtimal dahilinde olan askeri durumlar gerçekleştiğinde yapılması düşünülen ulaştırmalardır" (Över ve Yaramancı, 1980). Burada bir noktadan diğerine gidecek yük miktarı önceden bellidir. Bu yükler birim yük olarak ifade edilmektedirler ve her bir yük bir uçağı tam olarak dolduracak şekilde ayarlanmıştır. Alanlar arasındaki uçuş süreleri sabit olarak alınmaktadır. Her alandaki uçak boşaltma ve doldurma süreleri de sabittir. İntikal planlamasının neticesinde her uçağın güzergahı belirlenir, yani her uçağın hangi noktalar arasında hangi yükleri taşıyacağı ve görevinin hangi noktada biteceği belirlenir. Bundan da anlaşılabileceği gibi intikal planlaması problemi taşıt güzergahı belirleme problemleri sınıfına girer. Ancak aşağıda da gösterileceği gibi bu problemler arasında özel bir konumu vardır.

İntikal planlamasındaki amaca girmeden önce görev süresini tanımlayalım. Görev süresi, bir uçağın harekât başlangıç anından itibaren en son yükünü boşaltmasına kadar geçen süre olarak tanımlanır. İntikal planlamasında amaç enbüyük görev süresinin enküçüklenmesidir. Örneğin elimizde 4 uçak ve 19 yük olsun. Bu 19 yükün 4 uçak arasında paylaştırılması neticesi uçakların görev süreleri sırası ile 440, 510, 380, 570 dakika olsun. Bu durumda amaç işlevi değeri 570'dir. Eğer bu çözüm eniyi çözüm değilse yeni bir paylaştırma ile enbüyük

görev süresi 570'in altına düşürülebilir. Dikkat edilirse amaç 4 uçağın görev sürelerinin toplamının enküçüklenmesi değildir. Yukarıdaki örnekteki her bir görev süresini bir çubukla gösterelim (Şekil 3.1.). Amaç, bütün olurlu çözümler ara-



Şekil 3.1. 4 Uçak ve 19 Yüklü Örnek

sında en uzun çubuğu en kısa olan çözümü bulmaktır. O halde problem bir enk enb (min max) problemidir. Buna göre proble-

min amaç işlevi doğrusal olmamaktadır.

Problemin kısıtları ise talep noktalarındaki taleplerin karşılanmasıdır. Literatürde tartışılan taşıt güzergahı belirleme problemlerinde amaç olarak daha önce belirtildiği gibi (Alt bölüm 2.1), değişken maliyetlerin enküçüklenmesi; sabit maliyetlerin enküçüklenmesi ve sabit ve değişken maliyetlerin toplamının enküçüklenmesi alınmaktadır. İntikal planlaması probleminde ise amaç işlevine bir (enk enb) operatörü uygulanmaktadır. Bu ise problemin çözümünü oldukça güçleştirmektedir. Yazarın bilgisi dahilinde literatürde amaç işlevine (enk enb) operatörünün uygulandığı bir taşıt güzergahı belirleme problemi açık olarak formüle edilip çözülmemiştir. Ancak Pierce (1969) bu tür problemlerin varlığından söz etmektedir.

Problemin diğer bir ilginç yönü problemde birden fazla arz noktasının bulunması ve arz noktalarının aynı zamanda talep noktası da olabilmeleridir. Araçlar başlangıçta arz noktalarında bulunmayabilirler ve sadece aynı arz noktasına bağımlı değildirler. Ayrıca, görev süresi sonunda bir uçağın tekrar başlangıç noktasında bulunması gerekmez. Bütün bunlar elimizdeki problemin klasik taşıt güzergahı belirleme probleminden oldukça değişik bir yapısı olduğunu göstermektedir.

İntikal planlaması problemine açık literatürde görülen sezgisel bir çözüm önerisi Ulusoy ve Altinkemer (1980) tarafından getirilmektedir. Kullanılan yöntem yaklaşık bir çözüm bulmayı amaçlamaktadır. Halen bulundukları havaalanı gözönüne alınarak uçakların mevcut yükleri taşıma süreleri hesaplanmakta ve bu bilgiler (uçak x yük) matrisine girilmektedir. Bu matris üzerinde Macar atama yöntemi kullanılarak uçaklar yüklerle tahsis edilmektedir. Bu tahsis neticesi atanan uçaklardan birisi yükünü yerine ulaştırıp boşaltdıktan sonra (uçak x yük)-matrisi güncelleştirilmektedir. Seferini bitirmiş olan

uak bulunduėu havaalanında, diėer uaklar ise seferlerini bitirdiklefi zaman bulunacakları havaalanlarında kabul edilerek atanmamış yklere ilişkin yeni bir (uak x yk) matrisi hesaplanmaktadır. Bu matris zerinde Macar atama yntemi uygulanarak seferini tamamlamış olan uak bir yke atanmaktadır. Her bir sefer tamamlandığında bu iřlemler tekrarlanır. Bu en son yk atanana kadar devam eder. (Uak x yk) matrisinin hesaplanmasında boř uuřlar da gznne alınmaktadır.

ver ve Yaramancı (1980) bu zm yntemine dayanarak bir bilgisayar uygulaması getirmişlerdir.

Bu altblmde szl olarak tanımlanan intikal planlaması probleminin bundan sonraki altblmde matematiksel programlama formlasyonu verilecektir.

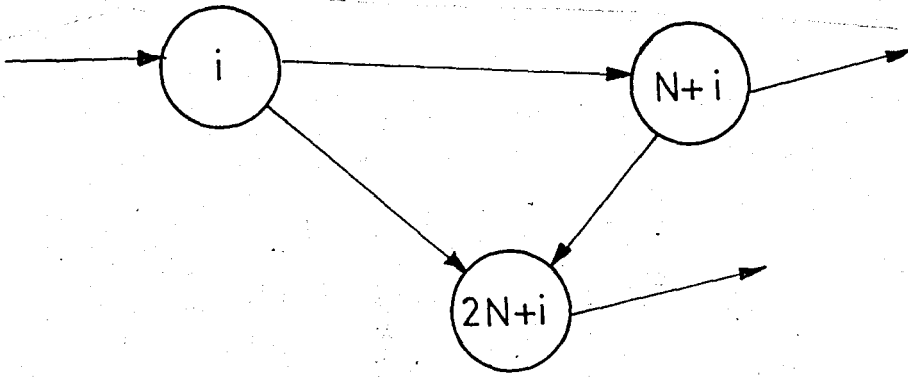
3.2. MATEMATİKSEL PROGRAMLAMA FORMLASYONU

Intikal planlamasının matematiksel programlama formlasyonu bir 0-1 tamsayı programlama formlasyonudur.

Problemin dinamik yapısı nedeni ile formlasyonda zamanın bir boyut olarak eklenmesi kaçınılmaz olmuřtur. Formlasyon srekli olmayacağına gre zamanın problemde yer alan operasyonların en byk ortak blenine gre aralıklar řeklinde ifade edilmesi uygun olacaktır. Toplam sreyi bu birim aralıėa blmüş olalım. Her aralık t indisi ile numaralandırılacaktır. $t = 1$ bařlangı noktasına karřı gelecektir. $t = 2$ ikinci aralıėın bařlangı zamanını verecektir. $t = T$ ise toplam sreyi ifade edecektir.

Formlasyonu gerekleřtirebilmek amacı ile havaalanlarını temsil etmekte olan her bir dėm  dėm ile ifade edilmiştir. Havaalanının  dėm ile temsili řekil 3.2.'de

gösterilmiştir. Havaalanı sayısına N dersek, i havaalanını temsil eden düğümler i, (N+i), (2N+i) olarak numaralanmışlardır. Bunlardan i düğümü tüm o havaalanına yapılan uçuşların olduğu giriş düğümüdür. Uçağın boşaltma işlemi (N+i) düğümünde, yükleme işlemi ise (2N+i) düğümünde yapılır. Bu havaalanından dışarıya yapılan çıkışlar ise Şekil 3.2.'de de görüldüğü gibi (N+i) ve (2N+i) düğümlerinden yapılır ve sırası ile boş ve dolu uçuşlara karşı gelir.



Şekil 3.2. Havaalanının Üç Düğüm İle Temsili

Yapılacak olan formülasyon düğümlerde uçak birikimine izin vermeyen bir formülasyondur. Buna göre görev süresi biten uçaklar sıfır uçuş süresinde (3N+2) olarak gösterilen bir düğümde toplanırlar ve yine sıfır uçuş süresinde (3N+1) olarak gösterilen bir düğümden bağlı oldukları havaalanına giderler.

Modelde,

$$\sum_{m=1}^M \sum_{t=1}^T x_{m,2N+i,j,t} = D_{ij} \quad i, j = 1, \dots, N, i \neq j \quad (3.1)$$

$$\sum_{m=1}^M \sum_{t=1}^T x_{m,i,N+i,t} = \sum_{j=1}^N D_{ij} \quad i = 1, \dots, N \quad (3.2)$$

$$\sum_{m=1}^M \sum_{t=1}^T x_{m,i,2N+i,t} = \sum_{j=1}^N D_{ij} \quad i = 1, \dots, N \quad (3.3)$$

$$\sum_{m=1}^M \sum_{t=1}^T x_{m,3N+1,j,t} = M_j \quad j = 1, \dots, N \quad (3.4)$$

$$\sum_{m=1}^M (x_{m,i,2N+i,t} + x_{m,N+i,2N+i,t}) \quad (3.5)$$

$$= \sum_{m=1}^M \sum_{j=1}^N x_{m,2N+i,j,t} + s_{ij} \quad i = 1, \dots, N \\ t = 1, \dots, T$$

$$\sum_{m=1}^M \sum_{j=1}^N x_{m,N+j,i,t} + \sum_{m=1}^M x_{m,3N+1,i,t} \quad (3.6)$$

$$= \sum_{m=1}^M x_{m,i,N+i,t+b} + \sum_{m=1}^M x_{m,i,2N+i,t+d} \quad i = 1, \dots, N \\ t = 1, \dots, T$$

$$\sum_{m=1}^M x_{m,i,N+i,t} = \sum_{m=1}^M \sum_{j=1}^N x_{m,N+i,j,t} + s_{ij} \quad (3.7)$$

$$+ \sum_{m=1}^M x_{m,N+i,2N+i,t+b} + \sum_{m=1}^M x_{m,N+i,3N+2,t} \quad i = 1, \dots, N \\ t = 1, \dots, T$$

$$\sum_{t=1}^T \sum_{m=1}^M \sum_{i=1}^N x_{m,N+i,3N+2,t} = \sum_{i=1}^N M_i \quad (3.8)$$

$$x_{m,i,N+i,t} = (0,1) \quad m = 1, \dots, M \quad (3.9.a) \\ i = 1, \dots, N \\ t = 1, \dots, T$$

$$\begin{aligned} x_{m,i,2N+i,t} &= (0,1) & m &= 1, \dots, M & (3.9.b) \\ & & i &= 1, \dots, N \\ & & t &= 1, \dots, T \end{aligned}$$

$$\begin{aligned} x_{m,N+i,j,t} &= (0,1) & m &= 1, \dots, M & (3.9.c) \\ & & i &= 1, \dots, N \\ & & t &= 1, \dots, T \\ & & j &= 1, \dots, N, \quad j \neq 1 \end{aligned}$$

$$\begin{aligned} x_{m,N+i,2N+i,t} &= (0,1) & m &= 1, \dots, M & (3.9.d) \\ & & i &= 1, \dots, N \\ & & t &= 1, \dots, T \end{aligned}$$

$$\begin{aligned} x_{m,2N+i,j,t} &= (0,1) & m &= 1, \dots, M & (3.9.e) \\ & & i &= 1, \dots, N \\ & & t &= 1, \dots, T \\ & & j &= 1, \dots, N, \quad j \neq 1 \end{aligned}$$

$$\begin{aligned} x_{m,3N+1,i,t} &= (0,1) & m &= 1, \dots, M & (3.9.f) \\ & & i &= 1, \dots, N \\ & & t &= 1, \dots, T \end{aligned}$$

$$\begin{aligned} x_{m,N+i,3N+2,t} &= (0,1) & m &= 1, \dots, M & (3.9.g) \\ & & i &= 1, \dots, N \\ & & t &= 1, \dots, T \end{aligned}$$

kısıtları altında

$$\begin{aligned} Z = \quad & \text{enb} \quad \left\{ \sum_{i=1}^N \sum_{j=1}^N \sum_{t=1}^T s_{ij} (x_{m,N+i,j,t} + x_{m,2N+i,j,t}) \right. \\ & + \sum_{i=1}^N \sum_{t=1}^T \{ b x_{m,i,N+i,t} + d (x_{m,i,2N+i,t} + x_{m,N+i,2N+i,t}) \} \} \\ & (3.10) \end{aligned}$$

amaç işlevi enküçüklenecektir.

Burada:

m = uçak indisi

i, j = düğüm indisleri

t = zaman indisi

s_{ij} = i düğümünden j düğümüne uçuş süresi

D_{ij} = j düğümünün i düğümünden karşılanacak talebi

M_i = i düğümünde başlangıçta mevcut uçak sayısı

M = toplam uçak sayısı

b = uçak boşaltma süresi

d = uçak yükleme süresi

$$x_{m,i,j,t} = \begin{cases} 1, & \text{eğer } m \text{ uçağı } i \text{ düğümünden } j \\ & \text{düğümüne } t \text{ zamanında ulaşırsa,} \\ 0, & \text{aksi durumlardadır.} \end{cases} \quad (3.11)$$

Birinci, ikinci ve üçüncü kısıt kümeleri talebin karşılanmasına ilişkin kısıtları içermektedir.

Dördüncü kısıt kümesi yapay başlangıç noktası $(3N+1)$ den uçakların üslendikleri havaalanlarına sıfır sürede gönderilmelerini sağlar. Bu kısıta göre bir havaalanındaki her uçak $t = 1$ zamanında işleme girmek zorunda değildir.

Beşinci kısıt kümesi yükleme düğümlerinde, altıncı kısıt kümesi giriş düğümlerinde, yedinci kısıt kümesi ise boşaltma düğümlerinde herhangi bir birikim olmamasını sağlayan kısıtlardır.

Sekizinci kısıt bütün uçakların görev süreleri sonunda yapay toplanma noktası olan $(3N+2)$ noktasında toplanmalarını sağlar.

Dokuzuncu kısıt kümesi ise (0,1) değeri alacak karar değişkenlerini belirler. Karar değişkenlerinin bu şekilde sınıflandırılması ile (0-1) tamsayı karar değişkenlerinin sayısı $(9MN^2T)$ den $(2MN^2T + 3MNT)$ ye düşürülmüştür.

3.3. MATEMATİKSEL PROGRAMLAMA FORMÜLASYONUNUN DEĞERLENDİRİLMESİ

Bir önceki altbölümde daha önce tanımlanmış olan intikal planlaması probleminin (0-1) tamsayı formülasyonu verilmiştir. Karar değişkeni sayısı $(2MN^2T + 3MNT)$, kısıt sayısı ise dokuzuncu kısıt kümesi haricinde $(3NT+N^2+2N+1)$ dir.

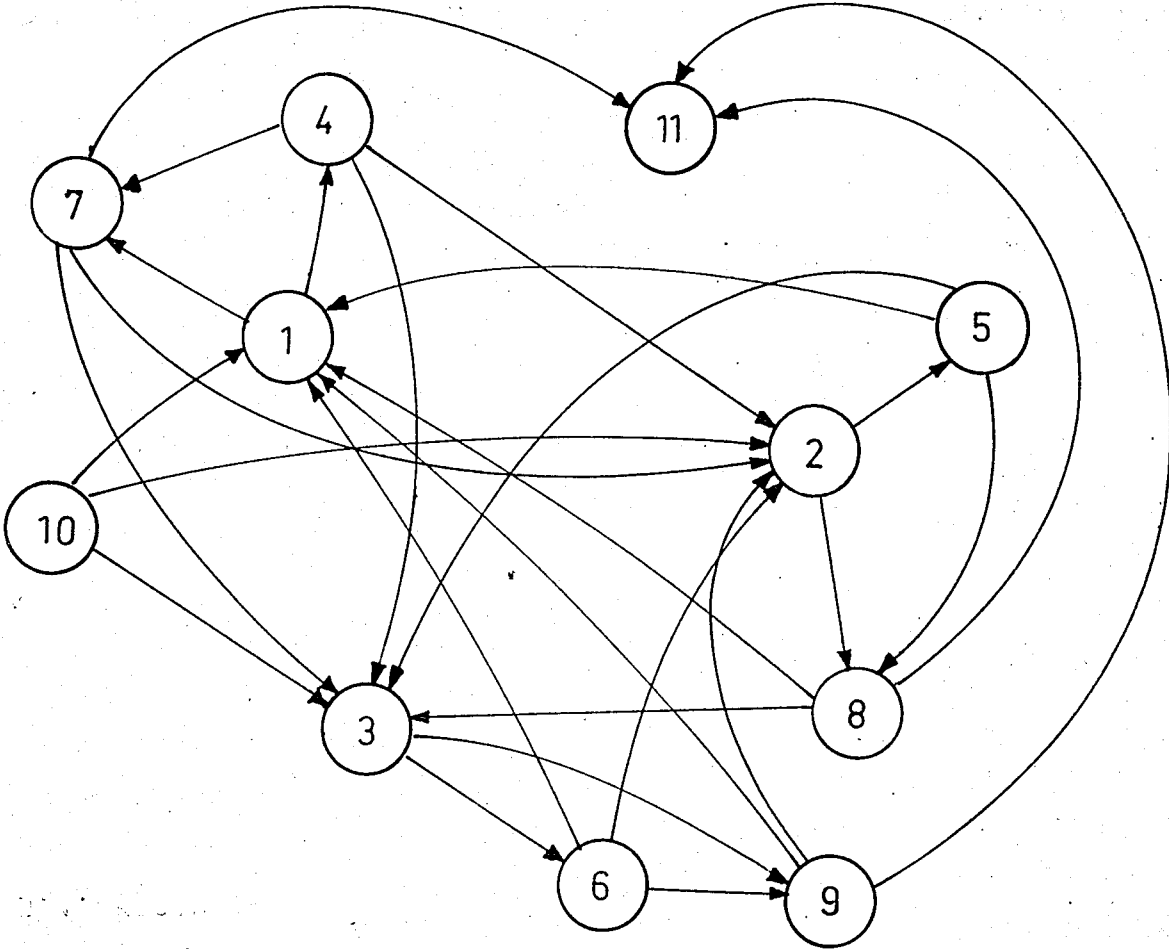
Bu formülasyon kombinatoryal niteliktedir ve dolayısı ile kombinatoryal programlamanın tüm zorluklarını içermektedir. En temel zorluk problemin büyüklüğünü belirleyen M,N,T parametrelerindeki artışların karar değişkenlerinin sayısında önemli artışlara neden olmasıdır. Örneğin, $M = 20$, $N = 20$, $T = 100$ için karar değişkeni sayısı 1.720.000, kısıt sayısı ise 6441'dir. Parametre değerlerinin $M = 40$, $N = 25$, $T = 100$ olduğu durumda karar değişkeni sayısı 5.300.000, kısıt sayısı ise 8176'dır. Görüldüğü gibi problem oldukça büyük boyutludur. Mevcut (0-1) tamsayı programlama problemleri bilgisayar paketlerinin kabul edilebilir hesaplama süreleri içinde ancak 90-100 adet (0-1) tamsayı karar değişkeni olan problemleri çözebildiği anımsanırsa eldeki probleme doğrudan bilinen (0-1) tamsayı programlama yöntemlerinin uygulanmasının bir netice vermeyeceği görülür (Ulusoy, v.d. 1979b).

Bu durumda işlenebilecek bir yol bazı dönüşümlerle problemdeki karar değişkenleri sayısını makul bir düzeye indirmektir. Elimizdeki problemde bu dönüşüm problemin serimi üzerinde gerçekleştirilecektir.

3.4. KESİN ÇÖZÜM YAKLAŞIMI

3.4.1. Serim Dönüşümü Uygulaması

Altbölüm 3.2.'de verilen formülasyonda oluşturulan serimde son iki yapay düğüm dışındaki düğümler havaalanlarını, işlevlerine göre parçalanmış olarak; ayrıtlar ise düğümler arasındaki olurlu bağlantıları göstermektedir. Örneğin üç havaalanından oluşan ve her havaalanında başlangıçta bir uçağın bulunduğu bir serimde bütün havaalanları arasındaki bağlantıların olurlu olduğu bir durumu gözönüne alalım (Şekil 3.3.). Problem bu serim üzerinde formüle edilmektedir. Yine aynı ör-



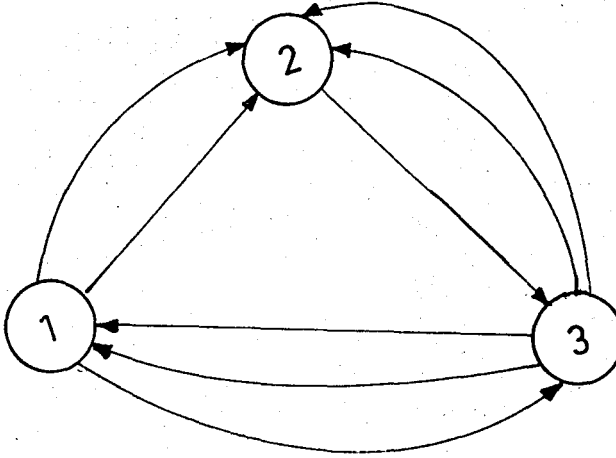
Şekil 3.3. Üç Havaalanı İçin Matematiksel Programlama Formülasyonu Serimi

nek için aşağıdaki (Tablo 3.1) talep ilişkisini kabul edelim.

Tablo 3.1. Yük Talepleri Matrisi

	1	2	3
1	-	2	1
2	-	-	1
3	2	2	-

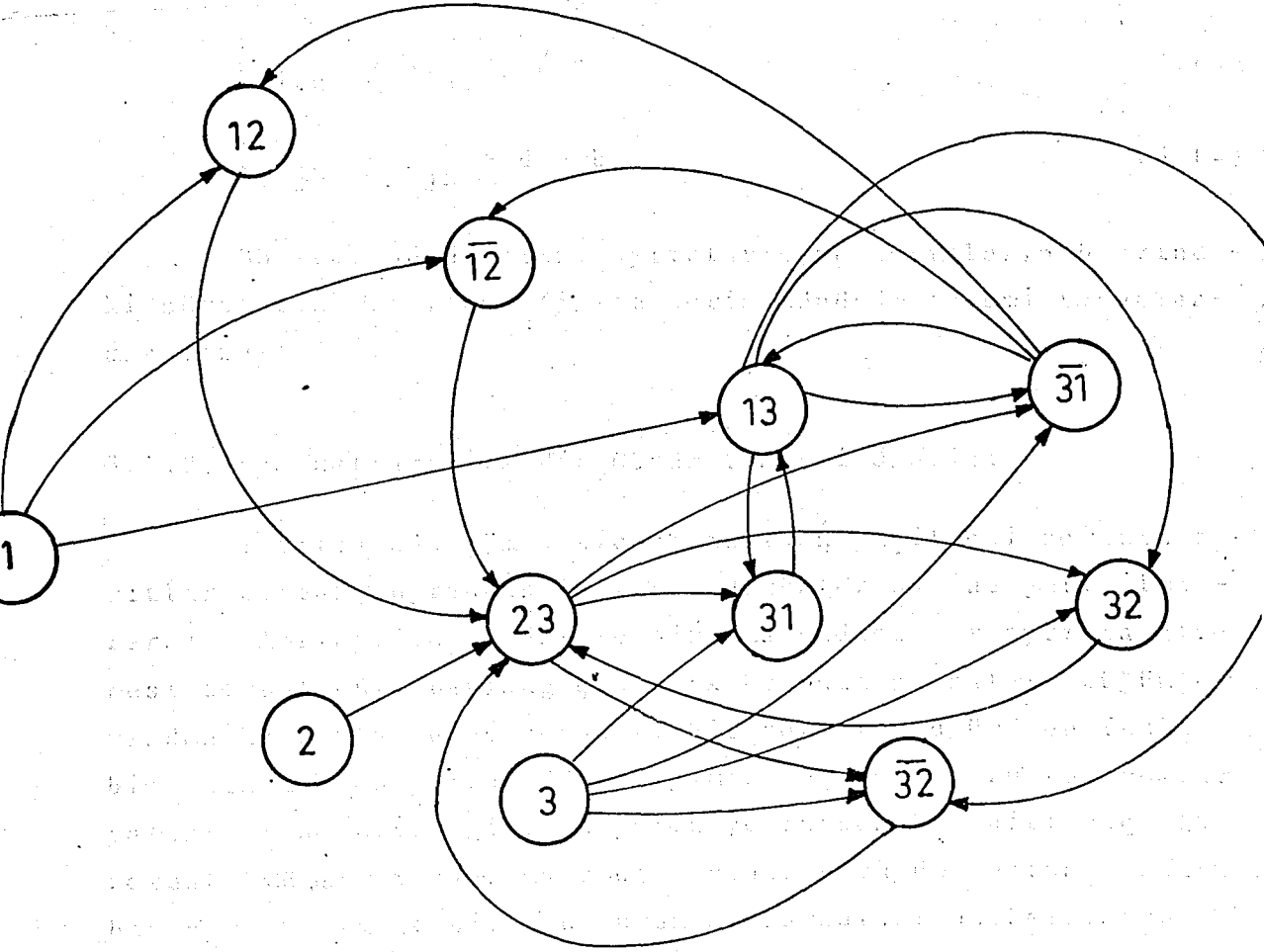
Üç havaalanını işlevlerine göre düğümlere parçalamadan ve ayrıtları talep edilen uçuşlar olarak tanımlayarak bir serim çerzersek Şekil 3.4'deki serimi elde ederiz. Bu serim üzerindeki ayrıtlar mutlaka üzerinden geçilmesi gerekli ayrıtlardır. Ye-



Şekil 3.4. Dönüşüm Uygulanacak Serim

ni oluşturulacak serimde her bu tür ayrıtı bir düğümlerle temsil edelim. Düğüm kümesinin geri kalan elemanlarını başlangıçta uçak bulunan havaalanları oluştursun. Bu durumda düğüm kümesi V şöyle oluşur: $V = \{1, 2, 3, (12), (\overline{12}), (13), (23), (31), (\overline{31}), (32), (\overline{32})\}$. Burada, i düğümünden j düğüme yapılan uçuşlar $(ij), (\overline{ij}), \dots$ olarak gösterilmektedir. Ayrıtlar kümesi ise şöyle oluşturulur: (ij) düğümü i den j ye olan uçuşu temsil ettiği-

ne göre, (ij) düğümünden bu tür düğümler arasında ancak (jk) düğümüne olurlu bağlantı vardır. Ayrıca, i , havaalanı düğümünden (ij) düğümüne olurlu bağlantı vardır. Buraya kadar tanımlanan ayritlar talep edilen uçuşlara ilişkin ayritlardır. Ancak genelde zorunlu dolu uçuşların yanında boş uçuşlar da olabilir. Bunlara ilişkin ayritlar ise (ij) düğümünden k düğümüne şeklinde oluşturulur. Burada kesinlikle $j \neq k$ dir. Şekil 3.4'de verilen serime bu dönüşüm uygulandığında Şekil 3.5'deki serim elde edilir. Ancak bu serimi daha karmaşık yapmamak için boş uçuşlara karşı gelen ayritlar eklenmemiştir. Bunlar,



Şekil 3.5. Dönüşüm Neticesi Elde Edilen Serim

(12,1), (12,3), (12,1)(12,3), (13,1), (13,2), (23,1), (23,2), (31,2), (31,3), (31,2), (31,3), (32,1), (32,3), (32,1), (32,3) ayrıştlarıdır.

Dönüşüm neticesi elde edilen serimin ayrıştlarının uzunlukları olan t_{ij} değeri, havaalanları arasındaki uçuş sürelerini veren süre matrisinin elemanları olan s_{ij} değeri ve yükleme ve boşaltma süreleri olan d ve b kullanılarak bulunur. Buna göre,

$$t_{ij,jh} = s_{jh} + d + b \quad (3.12)$$

$$t_{ij,h} = s_{jh}, \quad j \neq h \quad (3.13)$$

$$t_{j,jh} = s_{jh} + d + b \quad (3.14)$$

Böylece düğümlerin, ayrıştların ve ayrıştların üzerindeki sürelerin belirlenmesi ile serim dönüşüm işlemi tamamlanmış olmaktadır.

3.4.2. Ana Hatları İle Bir Çözüm Yöntemi Önerisi

Yapılan dönüşüm neticesi mutlaka geçilmesi zorunlu ayrıştlar mutlaka uğranması zorunlu düğümler haline getirilmişlerdir. Ayrıca, bu düğümlere bir ama sadece bir ayrıştın girmesi zorunludur. Başlangıçta uçak bulunan havaalanı düğümlerinden kaynaklanan ve herhangi bir zorunlu düğümü en fazla bir defa içeren yollar türettiğimiz düşünelim. Bu tür yollara yalın yollar diyeceğiz. Bu yalın yollardan her biri geçtiği zorunlu düğümleri kaplamaktadır, yani bu düğüme yolun elemanı bir ayrışt girmektedir. Bu durumda, havaalanı düğümlerinden kaynaklanan ve orada başlangıçta bulunan uçak sayısı kadar yalın yolu içeren bir yalın yollar kümesi ile tüm kaplanması gerekli düğümler bir ama sadece bir kere kaplanırlarsa probleme olurlu bir çözüm elde edilmiş olur. Bu yalın yollardan

yalın yolu içeren bir yalın yollar kümesi ile tüm kaplanması gerekli düğümler bir ama sadece bir kere kaplanırlarsa problem olurlu bir çözüm elde edilmiş olur. Bu yalın yollardan her biri bir uçağın güzergahını temsil eder ve uzunluğu (3.12), (3.14) ve (3.14) ile tanımlanan ayırıt uzunlukları ile bulunur."

Burada izah edildiği şekilde bulunan olurlu çözümler kümesi kolaylıkla görülebileceği gibi literatürde "küme parçalama problemi" olarak anılan problemin olurlu çözümler kümesi ile benzer şekilde ifade edilir. Küme parçalama problemini anımsayalım. Küme parçalama probleminde amaç belirli bir maliyetle düğüm kümesinin bir altkümesini kapsayan birimler (ki burada yalın yollara karşı gelmektedirler) arasından tüm düğümleri bir ama sadece bir defa kaplayan enküçük maliyetli birimler demetini bulmaktır. Bu tanımla ile küme parçalama probleminin elimizdeki problemden iki temel farkı olduğunu görürüz. Bunlardan birincisi, geçilmesi zorunlu düğümleri kaplayan yalın yolların sayısının ve hangi noktalardan kaçartanesinin kaynaklanması gerektiğinin kısıt olarak verilmiş olmasıdır. Anımsanacağı gibi yalın yollar uçak güzergahlarına karşı gelmektedirler ve başlangıçta uçak bulunduran havaalanı düğümlerinden kaynaklanmaktadır. Her böyle düğümden kaynaklanan yalın yol sayısı p düğümden başlangıçta mevcut uçak sayısına eşit olmak zorunluluğundadır. İkinci temel farklılık ise amaç işlevindedir. Küme parçalama probleminin amacı tüm kullanılan yalın yolların uzunluklarının toplamının enküçüklenmesi iken, intikal planlaması probleminde amaç kullanılan yalın yolların uzunluklarının enbüyüğünün enküçüklenmesidir.

Birinci farklılığı halletmek için burada izlenen yol küme parçalama algoritmasında bazı değişiklikleri gerçekleşt-

tirecek bu kısıtları gözönüne almaktır. Bu amaçla bir özel küme parçalama algoritması geliştirilmiştir.

İkinci farklılığı ortadan kaldırmak için burada önerilen yöntem darboğaz kavramının küme parçalama problemi çözümlerine uygulanması olmuştur. Buna göre, küme parçalama probleminin eniyi çözümü bulunmakta ve bu süreç sırasında bulunan enküçük enbüyük yalın yol uzunluğu saptanarak bu ve bundan daha uzun yalın yollar, yalın yollar kümesinden çıkarılarak küme parçalama problemi yeniden çözülmektedir. Böylece, eğer bir olurlu çözüm varsa, şimdi elde edilecek enküçük enbüyük yalın yol uzunluğunun bir önceki aşamada bulunan enküçük enbüyük yalın yol uzunluğundan daha küçük olması garanti edilmiş olmaktadır. Buna darboğaz uygulaması diyoruz.

Buraya kadar yapılan tartışmalardan hareketle burada önerilen çözüm yönteminin ana katlarını aşağıdaki gibi verebiliriz:

1. Aşama: Başlangıçta uçak mevcut olan havaalanı düğümleri ve buralardaki uçak sayıları, bir noktadan diğerine gönderilmesi gerekli yük sayılarını belirleyen talep matrisi, noktalar arasındaki uçuş sürelerini veren süre matrisi, yükleme ve boşaltma süreleri değerleri kullanılarak serim dönüşümü gerçekleştirilir ve ayrıt süreleri hesaplanır.

2. Aşama: Başlangıçta uçak mevcut olan havaalanı düğümleri kaynak alınarak yalın yollar türetilir. Bu yalın yol-

lar dolu uçuşlara karşı gelen düğümlerde sona erer.

3. Aşama: Yalın yollar kümesi kullanılarak geçilmesi zorunlu düğümler üzerinde özel küme parçalama algoritması uygulanır. Eğer bir olurlu eniyi çözüm elde edilirse 4. Aşamaya geçilir. Aksi taktirde en son elde edilmiş olan eniyi çözüm intikal planlaması probleminin eniyi çözümü olarak kabul edilir.

4. Aşama: Bir önceki aşamadan gelen enküçük enbüyük yolu yol uzunluğu kullanılarak yalın yollar kümesinde bu ve bundan büyük yalın yollar elenir ve 3.aşamaya gidilir.

Ana hatları ile yukarıda verilen çözüm yönteminin ayrıntıları bir sonraki altbölümde verilecektir.

3.5. AYRINTILARI İLE ÇÖZÜM YÖNTEMİ

3.5.1. Giriş

Altbölüm 3.4.2.'de dört aşama halinde verilen çözüm yönteminin bu aşamaları burada ayrıntılı olarak incelenecektir. Ancak birinci aşamayı oluşturan serim dönüşümü işlemi altbölüm 3.4.1.'de ayrıntılı olarak ele alındığından burada tekrarlanmayacaktır.

3.5.2. Yalın Yolların Türetilmesi

Yeni serim üzerinde başlangıçta uçak mevcut olan havaalanlarından başlayan ve geçilmesi mutlaka zorunlu düğümleri kaplayan yalın yolların türetilmesinde kullanılan yöntem bu altbölümde anlatılacaktır.

Yalın yolların türetilme işlem sürelerini azaltmak ve

daha önemlisi, bunların girdi olduğu özel küme parçalama algoritmasının büyüklüğünü makul düzeylerde tutmak için türetilen yalın yol sayısını problemin eniyi çözümünü kaybetmeden mümkün olduğunca az tutmak yararlı olacaktır. Yalın yollar özel küme parçalama algoritmasının üzerinde uygulandığı matrisin kolonlarını oluşturduğundan sayıları bu problemin çözülebilirliğini doğrudan etkilemektedir. Bu nedenle burada sayılarını azaltma önlemleri üzerinde özellikle durulacaktır.

Verilmiş bir çizge (graph) üzerinde mümkün yalın yolların türetimi çizge kuramında (graph theory) ilgi çeken, ancak o derecede de büyük bir problemdir. Pratikte uygulanabilirliği nedeni ile de oldukça üzerinde durulmuş olan bu probleme iki temel yaklaşım yapılmaktadır. Bunlardan birincisi dallanma esası üzerine kurulmuş ve özellikle iki nokta arasındaki tüm yalın yolları türetmek üzere geliştirilmiştir (örneğin, Kroft 1967). İkincisi ise matris yöntemlerine dayandırılan bir yaklaşımdır (Danielson, 1968). Bu yaklaşımda çizgenin bitişiklik (adjacency) matrisi A nın k . kuvveti A^k hesaplanır ki, A_{ij}^k , i ve j düğümleri arasında m uzunluğundaki tüm yolları içerir. Bunların arasından yalın olmayanlar elenir. Fratta ve Montanari (1975) bu problem için bir "yol cebiri" önermekte ve yalın yolların türetilmesi problemini bu cebirin uygulandığı bir doğrusal denklemler sistemi çözümü şeklinde ifade etmektedirler. Bu doğrusal denklemler sisteminin Gauss eliminasyon yönteminden yararlanmaktadırlar.

Burada uygulanan yöntem ise birinci temel yaklaşıma yakın bir yöntemdir. Yaklaşım iki nokta arasındaki bütün yalın yolları bulmak içindir. Ancak bazı basit dönüşümlerle bunu genelleştirmek mümkündür.

Yöntemin birinci aşamasında nihai noktadan bütün diğer noktalara en kısa yol algoritması uygulanır.

İkinci aşamada başlangıç noktası ile nihai nokta arasındaki en kısa yol alınarak bu yol üzerinde başlangıç noktası ile başlanarak en kısa yol üzerinde olmayan bitişik noktalara dallanılır. Her dallanılan nokta en kısa yol ile nihai noktaya bağlanır ve bu noktadan dallanmaya devam edilir. Her yolun yalın olup olmadığı kontrol edilir. Burada en kısa yolların kullanılmasının nedeni, şayet bütün yalın yolların türetilmesi istenmezse mümkün olduğunca iyi yalın yolları türetmek içindir.

Serime iki düğüm eklenir. Bütün düğümler bu düğümlerden birine bağlanır, diğer düğüm ise bütün diğer düğümlere bağlanır. Elde edilen serim üzerinde yukarıdaki yöntemin uygulanması bütün düğümler arasındaki yalın yolları verecektir.

Yukarıda kısaca genelde nasıl uygulandığı anlatılan yöntemin intikal planlaması probleminde güzergah türetime uygulanmasına bazı kolaylıklar getirilebilmektedir. Geçilmesi zorunlu m adet düğümden bir ama sadece bir defa geçilmesi ve yalın yolların ancak başlangıçta uçak mevcut havaalanlarından başlaması koşulları türetilecek yalın yolların sayısında önemli azalmalar meydana getirmektedir. Diğer bir husus yalın yolların geçilmesi gerekli düğümlerden birinde bitirilmesi zorunluluğudur. Aksi halde güzergahın son ayağı boş uçuş olacaktır ve bu gereksizdir.

Türetilen yalın yolların sayısında önemli bir azalma da bu yalın yolların uzunlukları üzerine iyi birer üstsınır ve bir altsınır getirildiğinde sağlanacaktır. Burada önerilen ve kullanılan üstsınır şöyle bulunmaktadır:

$$\bar{U}S = \left[\frac{\text{Dolu uçuş sayısı}}{\text{Uçak Sayısı}} \right] \left(\text{Ortalama dolu uçuş süresi} + \frac{1}{2} \times \text{Ortalama boş uçuş süresi} \right) \quad (3.15)$$

Burada **CC** işaretinin anlamı, bu işaretin içinde hesaplanan sayının tam sayı olmadığı taktirde bir üst tamsayıya yuvarlatılmasıdır. Bu üstsınırın bulunması tamamen sezgisel bir neticedir ve çözülen problemlerde makul bir üstsınır oluşturduğu gözlenmiştir.

Altsınır ise şöyle bulunur:

$$AS = \left[\frac{\text{Dolu uçuş sayısı}}{\text{Uçak sayısı}} \right] (\text{Enküçük boş uçuş süresi}) \quad (3.16)$$

Sayısal çalışmalarda gözlenen, yalın yol uzunluklarının hızla belirli bir düzeye eriştiği ve AS'ın çoğu kez yalın yol sayısında fazla bir azalma sağlamadığıdır. Ancak AS'ı daha yükseltmek eniyi çözümü kaybettirebilir. Bu nedenle daha sıkı ve güvenilir AS değerlerinin aranması gerekmektedir.

Elde edilen yalın yollar özel küme parçalama algoritmasına girmeden önce şu işlemlerden geçirilir. Yalın yol sadece geçilmesi zorunlu m düğüm cinsinden ifade edilir ve diğer düğümler yalın yoldan elenir. Başlangıç düğümünü teşkil eden havaalanı yalın yol düzeyinin (array) arkasına bir etiket olarak takılır. Aynı şekilde yalın yolun süresi ikinci bir etiket olarak takılır. Üçüncü bir etiket ise yalın yola atanan yapay taleptir. Yapay talep değeri, aynı düğümleri değişik bir sırada izleyen ve aynı havaalanından kaynaklanan yalın yolları tespit etmek için kullanılır. Böyle birden fazla yalın yol bulunduğunda bunlardan sadece birisi özel küme parçalama algoritmasına girdi olur; diğerleri elenir. Kolayca görüleceği gibi böyle yalın yollar salt yol uzunluğundan yakalanamazdı. Üçüncü etiket sadece eleme için kullanıldığından yalın yollar özel küme parçalama algoritmasına girerlerken çıkarılır.

Saptanmış bir alt ve üstsınır arasında türetilip geçilmesi zorunlu m düğüm cinsinden ifade edilen ve başlangıç

havaalanı ve süre etiketleri takılı yalın yollar özel küme parçalama algoritmasına girdi olurlar.

3.5.3. Özel Küme Parçalama Algoritması

Burada önce özel küme parçalama probleminin tanımı verilecektir: Elimizde geçilmesi (kaplanması) zorunlu (yani dolu uçuşlara karşı gelen) m adet düğümü içeren bir $\bar{K}$ kümesi ve n adet yalın yolu (güzergahı) içeren bir X kümesi olsun. Bu yalın yolların her biri başlangıçta belli bir sayıda uçak bulunduran bir havaalanından kaynaklanır. Yalın yolların herbiri $\bar{K}$ kümesinin bir altkümesi olan düğümleri kaplamaktadır. Amaç, $\bar{K}$ kümesinin m adet düğümünden bir ama sadece bir defa geçen ve her bir havaalanından kaynaklanan yalın yolların bu havaalanında başlangıçta mevcut uçak sayısına eşit olduğu öyle bir yalın yollar altkümesi seçmektir ki bu yalın yolların toplam süresi enküçük olsun.

Özel küme parçalaması algoritması literatürde bilinen küme parçalama algoritmalarından birine dayandırılacaktır. Daha önce sözlü tanımı ve matematiksel formülasyonu verilen küme parçalama problemine burada esas olarak alınacak olan çözüm yöntemi Pierce (1968) ve Garfinkel ve Nemhauser (1969) tarafından önerilen yöntemdir. Bu algoritma esasta bir dal-sınır algoritmasıdır. Ancak klasik dal-sınır algoritması yapısı içinde değil de bilgisayar programlamaya daha yatkın bir şekilde sunulacaktır.

Algoritma kısıt matrisi A' 'dan bir dönüşümle elde edilen ve daha sonra genişletilen bir matris üzerinde uygulanır. Kısıt matrisinin elemanları a_{ij} şöyle tanımlanır:

$$a_{ij} = \begin{cases} 1, & \text{eğer } j \text{ güzergahı } i \text{ düğümünden geçiyorsa} \\ 0, & \text{aksi durumlarda} \end{cases} \quad (3.17)$$

Dönüşümle elde edilen yeni matris $\bar{B}$ matrisi diyelim. $\bar{B}$ matrisi şöyle elde edilir. A matrisinin kolonları kolon ve satırlarında yapılan permutasyonlarla en fazla m adet grupta toplanır. K kümesi elemanları $\bar{B}$ matrisinin satırlarını simgelerler ve bunların birinci satırdan başlayarak $k_1, k_2, \dots, k_m$ olarak sıralandığını kabul edelim. $\bar{B}$ matrisinin kolonları m grupta toplanır. Herhangi bir j grubunda bu grubu oluşturan kolonlarda $k_1, \dots, k_{j-1}$ elemanları sıfırdır; bu kolonların k_j elemanı ise mutlaka birdir; $k_{j+1}, \dots, k_m$ elemanları ise sıfır veya bir değeri alırlar. $\bar{B}$ matrisini elde etmek için yapılan permutasyonlar neticesinde mutlaka m adet grup oluşmayabilir; yani bazı gruplar yukarıda verilen gerekli koşulları yerine getiren kolonlar bulunamadığından boş kalırlar. $\bar{B}$ matrisi elde edildikten sonra en son satırının altına iki satır eklenir. h ile tanımlanacak ilk satırın her elemanı bağlı olduğu kolonun karşı geldiği yalın yolun kaynaklandığı havaalanını gösterecektir; c ile tanımlanacak ikinci satırın elemanları ise bu yalın yolların görev sürelerini belirtecektir. Bu genişletilmiş $\bar{B}$ matrisine B matrisi diyelim. Algoritma bu matris üzerinde uygulanacaktır.

Örnek olarak Tablo 3.2.'de verilen A matrisinden kolon ve satır permutasyonları ile elde edilen B matrisi Tablo 3.3. de verilmiştir. Tablo 3.3. den görüleceği gibi K kümesinde altı eleman olmasına karşın grup sayısı beş olmuştur. Genelde belirtilmesi gereken bir husus, yapılan kolon ve satır permutasyonlarına bağımlı olarak bir A matrisinden değişik B matrislerinin elde edileceğidir. Bu çözümü etkilemez ancak çözüm süresi üzerinde ne gibi bir etkisi olacağı henüz incelenmemiştir.

Örnekten de kolayca görüleceği gibi, A matrisinden dönüşümle B matrisinin elde edilmesinin nedeni kısıt kümesine dahil her noktanın bir ama sadece bir defa kaplanması kısıtını gerçekleştirecek çözümlerin türetilmesini kolaylaştırmak içindir. "Bu kısıtı sağlayan bir çözümde aynı gruptan iki

Tablo 3.2. A Matrisi

Nokta Numarası	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}
3	0	0	0	1	1	0	0	1	1	0	1	0	0	1	0	1	0
4	1	0	1	0	1	0	1	0	1	1	1	1	0	0	1	0	0
5	1	1	0	0	0	0	1	1	0	1	0	0	0	1	0	0	1
6	0	0	1	0	1	0	0	1	0	0	1	0	1	0	1	0	0
7	0	1	0	1	0	1	0	0	1	1	0	1	0	0	1	0	1
8	0	1	1	0	1	1	1	1	0	0	0	0	1	0	1	1	0

Tablo 3.3. B Matrisi

i	k_i	x_{13}	x_3	x_{11}	x_8	x_5	x_{15}	x_{14}	x_4	x_{16}	x_9	x_1	x_{12}	x_7	x_{10}	x_{17}	x_2	x_6
1	6	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
2	3	0	0	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0
3	4	0	1	0	0	1	1	0	0	0	1	1	1	1	1	0	0	0
4	5	0	0	0	1	0	0	1	0	0	0	1	0	1	1	1	1	0
5	7	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1
6	8	1	0	1	1	1	1	0	0	1	0	0	0	1	0	0	1	1
	h	1	1	2	1	2	2	2	1	2	2	1	1	2	1	2	2	2
	c	20	26	31	54	62	66	22	26	28	31	19	24	27	39	29	59	32

yalın yol bulunamaz" gözlemi önemli sayıda kombinasyonu göz-önüne dahi almadan elemektedir.

Her grup içindeki yalın yollar enküçük maliyetliden başlayarak artan maliyet sırasına göre dizilir ve numaralanırlar. Bu çözüm süresini kısaltmaya yönelik bir sezgisel kuraldır.

Algoritmanın uygulanması sırasında herhangi bir anda elde mevcut eniyi çözümü $\bar{Y}$ ve toplam görev süresini $\bar{Z}$; en son üzerinde çalışılan kısmi çözümü Y ve toplam görev süresini Z ; g grubunun p havaalanından kaynaklanan j elemanını $y_{gj}(p)$ ve görev süresini c_{gj} , g grubundaki eleman sayısını M_g ; havaalanı sayısını H ; Y 'nin kapladığı K kümesi elemanlarının kümesini E ; i havaalanında başlangıçta mevcut uçak sayısını U_i ; kısmi çözümün i havaalanından kullandığı uçak sayısını ise u_i ile gösterelim.

Bu açıklama ve tanımlardan sonra özel küme parçalama algoritmasını yazabiliriz.

Adım 1: B matrisi elde edilir. $Y = \{\emptyset\}$, $z = 0$, $\bar{Y} = \{\emptyset\}$, $\bar{z} = \infty$, $E = \{\emptyset\}$, $u_i = 0 \forall i$, yapılır.

Adım 2: $g = \text{enk} \{i | k_i \notin E\}$ bulunur. g grubu boş değil ise ilk elemanı işaretlenir, $j = 1$ yapılır ve Adım 3'e gidilir. Eğer g grubu boş ise Adım 4'e gidilir.

Adım 3: g grubunda işaretlenmiş olan $y_{gj}(p)$ yalın yolu ile başlanarak bu yalın zincirler artan j sırası ile incelenir. $y_{gk}(p) \cap E = \emptyset$, $z + c_{gk} < \bar{z}$ ve $u_p + 1 \leq U_p$ koşulları ilk defa sağlandığında $y_{gj}(p)$ üzerindeki işaret kaldırılır ve bu koşulu sağlayan $y_{gk}(p)$ işaretlenerek Adım 5'e gidilir. Eğer g grubunun incelenen elemanları bu koşulları sağlayamazsa veya ilk defa bir $y_{gk}(p)$ için $z + c_{gk} \geq \bar{z}$ olduğunda $y_{gj}(p)$ üzerin-

Adım 4: Y kısmi çözümü daha iyi bir sonuç türetemiyecektir. Eğer $Y = \{\emptyset\}$ ise durulur; eğer $\bar{Y} \neq \{0\}$ ise eldeki $\{\bar{Y}, \bar{z}\}$ çözümü eniyi çözümdür; eğer $Y = \{\emptyset\}$ ise, Y kümesine en son eklenmiş olan yalın yol ki buna $y_{ab}(p)$ diyelim, Y kümesinden çıkartılır, $z \leftarrow (z - C_{ab})$, $E \leftarrow E - y_{ab}(p)$, $u_p \leftarrow (u_p - 1)$ yapılır, $y_{ab}(p)$ nın işareti kaldırılır. Eğer $b = M_a$ ise, bu Adım tekrarlanır. Eğer $a = 1$ ve $b = M_1$ ise durulur. Aksi hallerde, $y_{a,(b+1)}(p)$ işaretlenir, $a \leftarrow a + g$ ve $(b+1) \leftarrow j$ yapılır ve Adım 3'e gidilir.

Adım 5: $Y \leftarrow Y \cup \{y_{gk}(p)\}$, $z \leftarrow (z + C_{gk})$, $E \leftarrow E \cup y_{gk}(p)$, $u_p \leftarrow (u_p + 1)$ güncelleştirmeleri yapılır. Eğer $E = K$ ve $u_i = U_i \forall i$ ise daha iyi bir olurlu çözüm bulunmuştur; $\bar{Y} = Y$, $\bar{z} = z$ yapılır ve Adım 4'e gidilir. Eğer $E \neq K$ ise Adım 2'ye gidilir. Eğer $E = K$ ve en az bir i indisi için $u_i \neq U_i$ ise, bu durumda Adım 4'e gidilir.

Özel küme parçalama algoritması üzerinde uygulandığı yalın yollar kümesi X 'in bir olurlu çözümü varsa mutlaka eniyi çözümü bulacaktır. Bu netice bir teorem şeklinde ifade edilmiş ve ispatlanmıştır.

Teorem 3.1. Yukarıda verilen algoritma özel küme parçalama probleminin, eğer bir olurlu çözümü varsa, eniyi çözümü bulacaktır.

İspatı: Daha önce de belirtildiği gibi algoritma bir dal-sınır algoritmasıdır. Problemin eniyi amaç işlevi üzerine koyulan üstsınır başlangıçta sonsuzdur. Ancak ilk olurlu çözüm elde edildiğinde bunun amaç işlevi değeri üstsınır olarak alınır. Bundan sonra her yeni olurlu çözüm elde edildiğinde onun amaç işlevi değeri yeni üstsınır olarak alınır. Algoritmanın türettiği her kısmi çözümün bulundurduğu yalın yolların toplam görev süresi bu üstsınır ile karşılaştırılır. Bu değer üstsınıra eşit veya büyükse kısmi çözüm elenir yani bu kısmi çözümü tamamlayacak bütün kombinasyonlar teker teker değer-

lendirilmeden elenirler çünkü hiçbirisi üstsınırdan daha düşük değerde bir amaç işlevi değerine sahip olamayacaklardır. Görüldüğü gibi yukarıdaki karar kuralı kullanılarak elenen çözümler arasında eniyi çözümün bulunması olanaksızdır. Türetilen her kısmi çözümün, geçilmesi gerekli düğümleri bir ama sadece bir defa kaplaması ve kısmi çözümdeki yalın yolların kaynaklandıkları havaalanlarında başlangıçta mevcut uçak sayısını aşmamaları kısıtlarına göre olurluluk denetimi yapılmakta ve olurlu olmayan kısmi çözümler elenmektedir. Kullanılan işaretleme mekanizması gruplandırma ile yapılan örtük elemelerin dışındaki tüm yalın yol kombinasyonlarını türetebilmektedir. Buna göre, algoritma sadece olurlu olmayan ve/veya elde edilmiş bir amaç işlevi değerinden daha iyi bir değere götürmeyecek olan kısmi çözümleri ve dolayısı ile bunlardan türetilecek çözümleri elemekte ve elemelerin dışında kalan tüm olurlu çözüm alanını taramaktadır. O halde algoritmanın bulduğu çözüm problemin eniyi çözümüdür.

Özel küme algoritmasının çıktısı önerilen kesin çözüm yönteminin son aşaması olan yalın yollar kümesinin daraltılması işlemine girdi olur.

3.5.4. Yalın Yollar Kümesinin Daraltılması

$\bar{Y}$ kümesi özel küme parçalama algoritmasının başında boş kümeye, T_{enb} ise sonsuza eşitlenirler ve her bir daha iyi olurlu çözüm bulunduğunda güncelleştirilirler. Böyle bir olurlu çözümü meydana getiren yalın yolların süreleri uçakların görev sürelerine karşı gelir. $\bar{Y}$ kümesi her güncelleştirildiğinde bu yeni olurlu çözümün görev süreleri arasından en büyüğü seçilir ve mevcut T_{enb} değeri ile karşılaştırılır. Eğer daha küçükse yeni T_{enb} değeri olarak alınır. Özel küme parçalama algoritması eniyi çözümü bulduktan sonra elimizde olan T_{enb} değerine eşit veya daha büyük görev süreleri olan yalın yollar X kümesinden elenirler. Bu eleme işlemine X kümesinin daraltılması diyoruz. Bu işlem neticesinde, daraltıl-

miş X kümesi üzerinde özel küme parçalama algoritması uygulandığında, eğer bir olurlu çözüm varsa, bu uygulamanın T_{enb} değerinin bir evvelki uygulamanın T_{enb} değerinden daha küçük olması garanti edilmektedir. Olurlu çözüm bulunabildiği süreçte 1., 2., ..., k. uygulamadan elde edilen $\{T_{enb}^1, T_{enb}^2, \dots, T_{enb}^k\}$ dizisinin elemanları sürekli azalan bir diziyi oluştururlar.

Sonlu sayıda bir daraltmadan sonra X kümesi üzerinde uygulanan özel küme parçalama algoritması olurlu bir çözüm vermeyecektir. Bu durumda en son elde edilmiş olan T_{enb}^k değeri ve buna karşı gelen uçuş güzergahları intikal planlaması probleminin eniyi çözümü olarak alınır.

Intikal planlaması probleminin olurlu bir çözümü olduğu takdirde burada önerilen kesin çözüm yönteminin eniyi çözümü bulacağı aşağıdaki teorem ve ispatı ile gösterilmiştir.

Teorem 3.2. Özel küme parçalama algoritmasına i. uygulanmasında, eğer bir olurlu çözüm varsa, elde edilen T_{enb}^i değeri kullanılarak X kümesinin daraltılması ve daraltılmış X kümesi üzerinde özel küme parçalama algoritmasının yeniden uygulanması işlemlerinin tekrarlanması ile elde edilen $\{T_{enb}^1, T_{enb}^2, \dots, T_{enb}^k\}$ dizisi sürekli azalan sonlu sayıda elemandan oluşur. Eğer $(k+1)$. uygulamada olurlu çözüm yoksa, T_{enb}^k intikal planlaması probleminin eniyi çözümüdür.

İspatı: $\{T_{enb}^1, T_{enb}^2, \dots, T_{enb}^k\}$ dizisinin elemanları $T_{enb}^1 > T_{enb}^2 > \dots > T_{enb}^k$ ilişkilerini sağlarlar. T_{enb}^{v+1} değeri T_{enb}^v değerine eşit ve daha büyük elemanların elendiği bir daraltılmış X kümesinden elde edildiğine göre $T_{enb}^{v+1} < T_{enb}^v$ dir. X kümesi elemanlarının değişik kombinasyonları arasından özel küme parçalama probleminin kısıtlarını sağlayanlar problemin olurlu çözümlerini oluştururlar. X kümesi sonlu sayıda eleman içerdiğine göre olurlu çözüm kümesinin de sonlu sayıda elemanı olacaktır. Özel küme parçalama algoritmasının her uygulanışında X kümesinin en az bir elemanı elendiğine göre özel küme

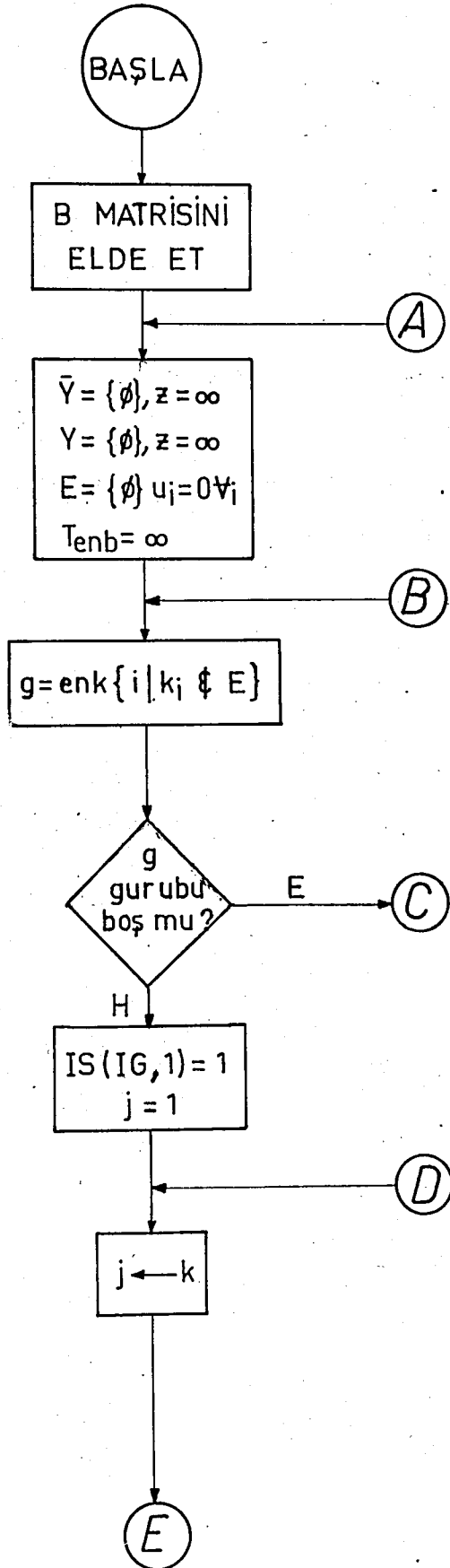
parçalama algoritmasının sonlu sayıda uygulanması neticesi daraltılmış X kümesinden bir olurlu çözüm türetilmeyecektir. Buna göre, $T_{enb}^1, T_{enb}^2, \dots, T_{enb}^k$ dizisi sonlu bir dizidir. $(k+1)$. uygulama ilk defa olurlu sonuç vermeyen uygulama ise T_{enb}^k , X kümesinin olurlu çözümlerine ait enküçük enbüyük görev süresidir; yani intikal planlaması probleminin eniyi çözümdür.

Böylece intikal planlaması problemi için önerilen ve dört aşamadan oluşan kesin çözüm yöntemi ayrıntıları ile sunulmuş ve eğer problemin olurlu çözüm kümesi boş değil ise eniyi çözümü bulacağı ispatlanmıştır. Yöntemin akış şeması Şekil 3.6.'da verilmiştir. Akış şemasında serim dönüşümü neticesi girdi olarak alınmış ve yalın yolların türetiminin ayrıntılarına girilmemiştir. Burada IS algoritma esnasında konulduğu belirtilen işaretleri simgelemektedir. Diğer simgeler aynen bu bölümde kullanıldığı gibidir.

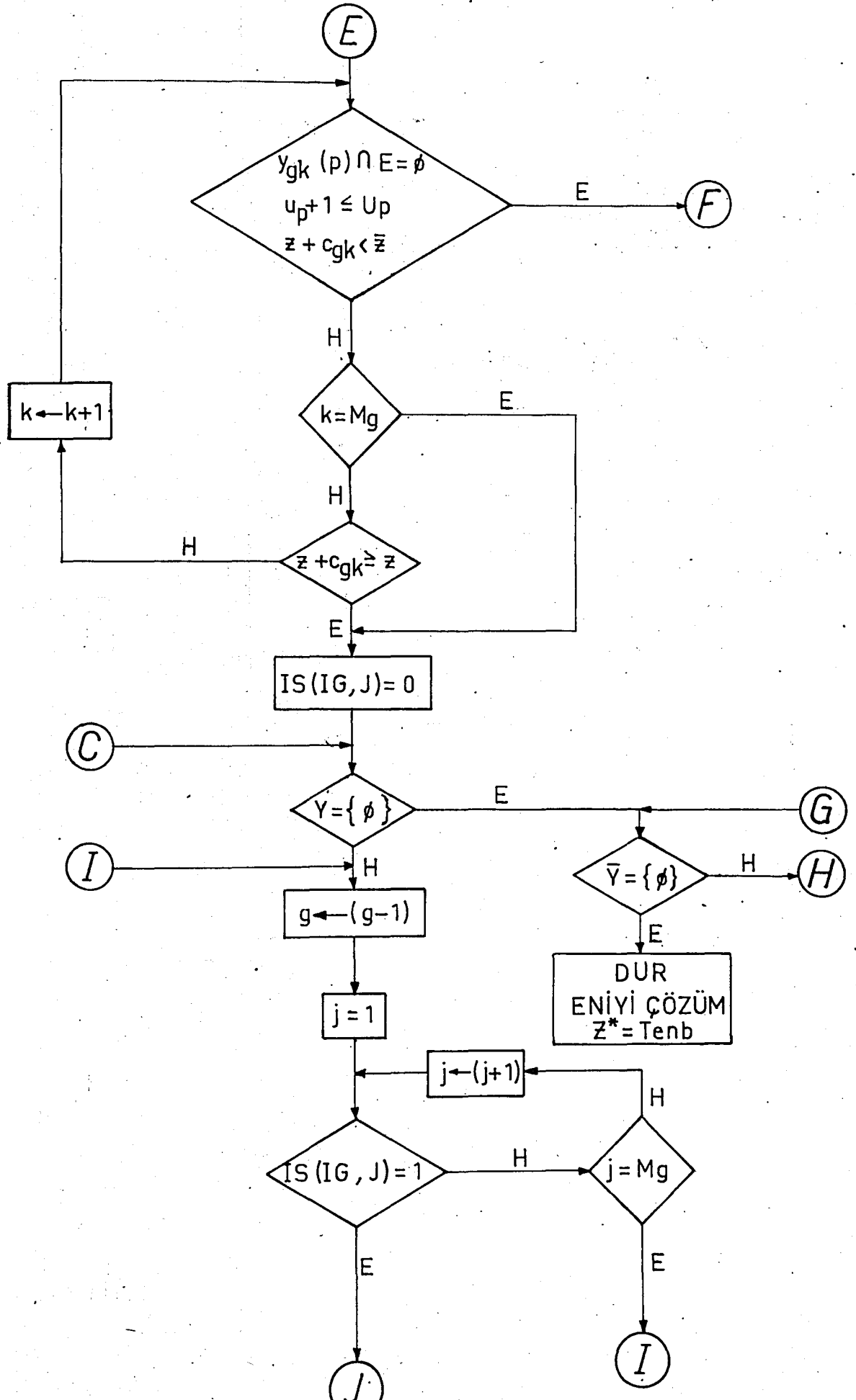
3.6. ÖRNEK PROBLEM

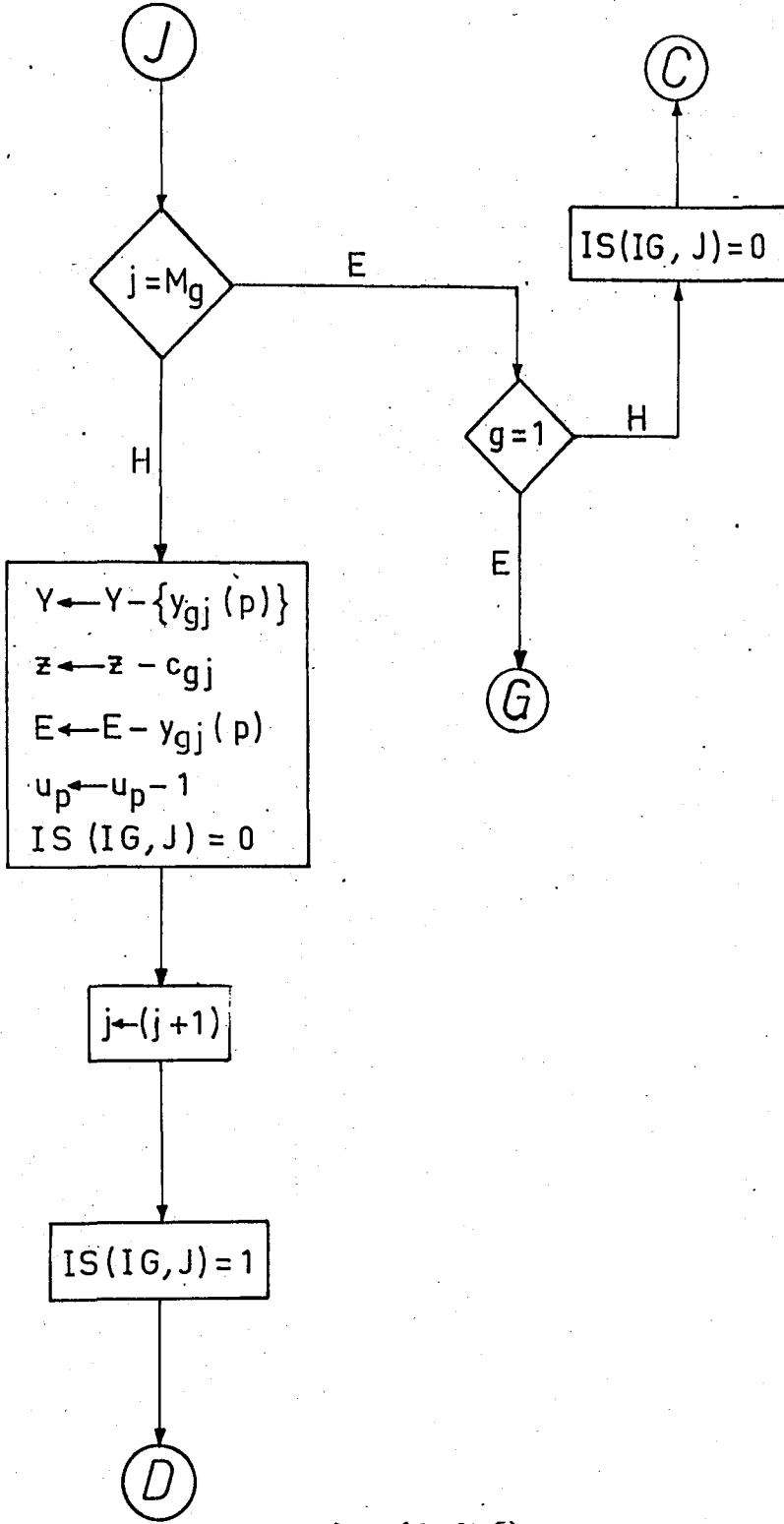
Kesin çözüm yöntemi aşağıdaki probleme uygulanacaktır. Problem çözümünden amaç yöntemin uygulanışını örneklemek olduğundan X kümesindeki yalın yolların sayısı düşük tutulmuştur.

Elimizde altı adet yük olsun; yani altı adet dolu uçuş gerekli olsun. İki havaalanı ve bunların birincisinde bir, ikincisinde ise iki uçak bulunsun; yani $U_1 = 1, U_2 = 2$. Serim dönüşümü ve yalın yolların türetilmesi aşamalarının neticesi olarak Tablo 3.2.'de verilen A matrisi elde edilmiş olsun. Yalın yolların türetilmesi altbölümünde belirtildiği gibi havaalanlarına karşı gelen 1. ve 2. düğümler A matrisinde yer almamıştır.

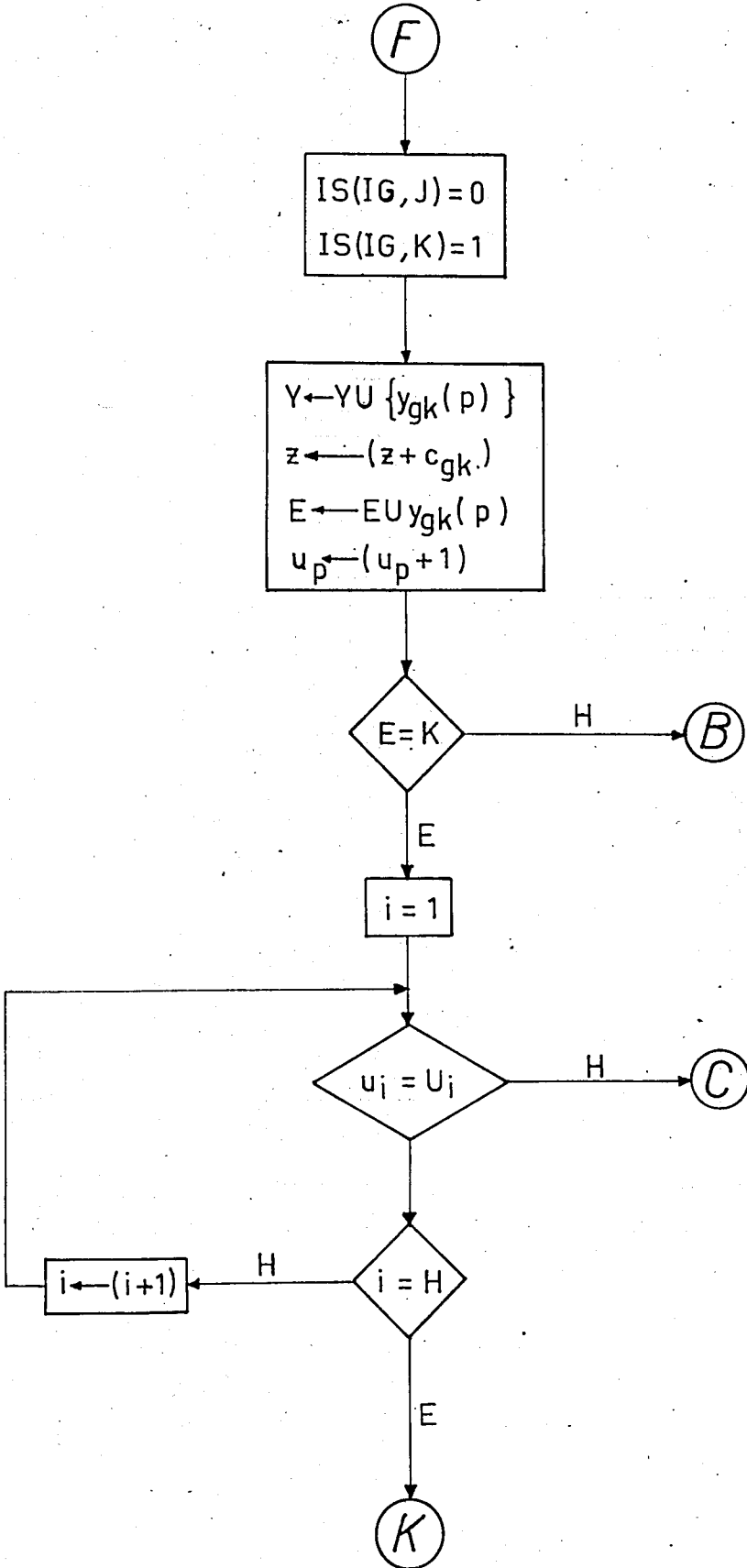


Şekil 3.6. Kesin Çözüm Yönteminin Akış Şeması

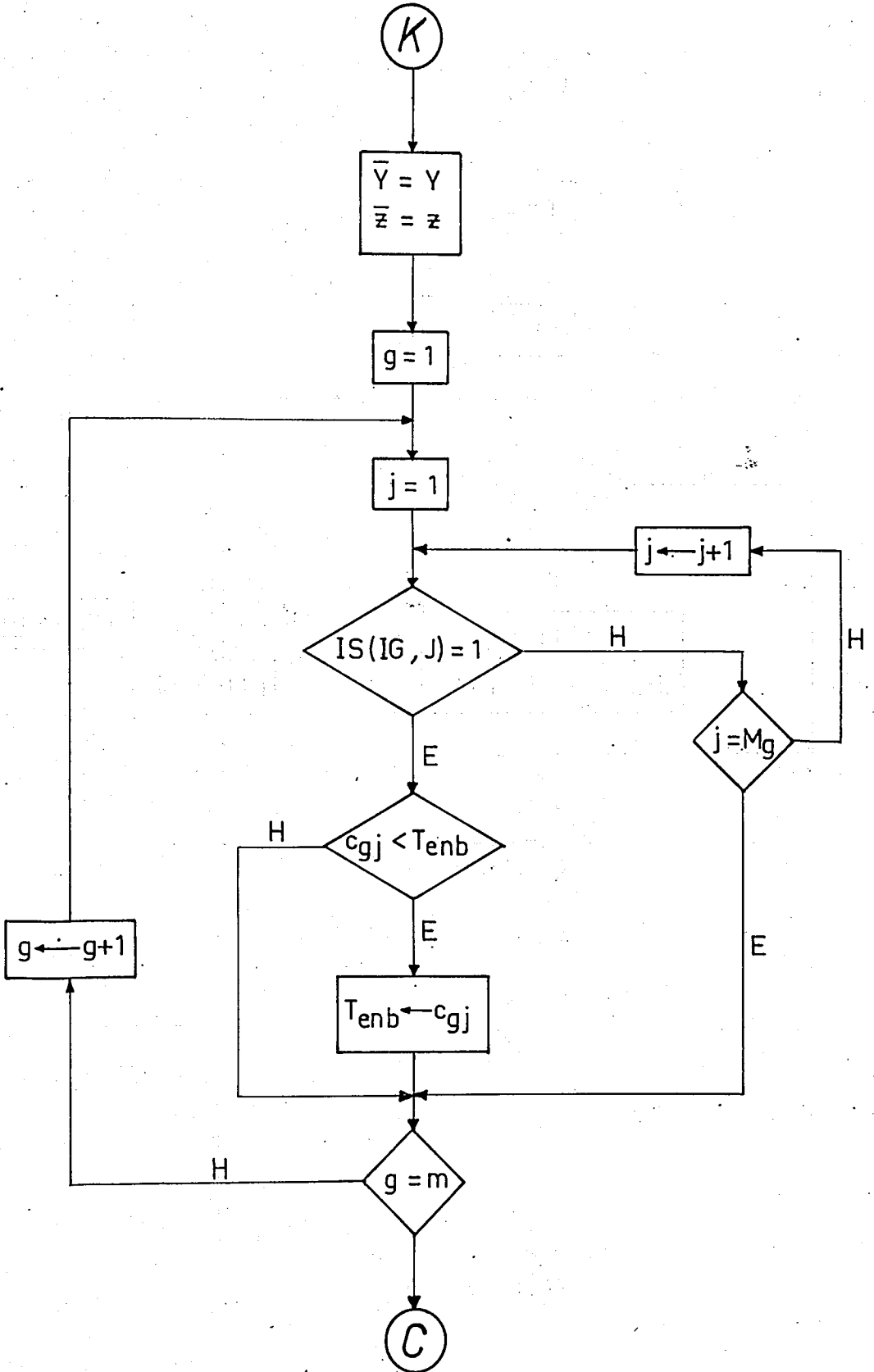




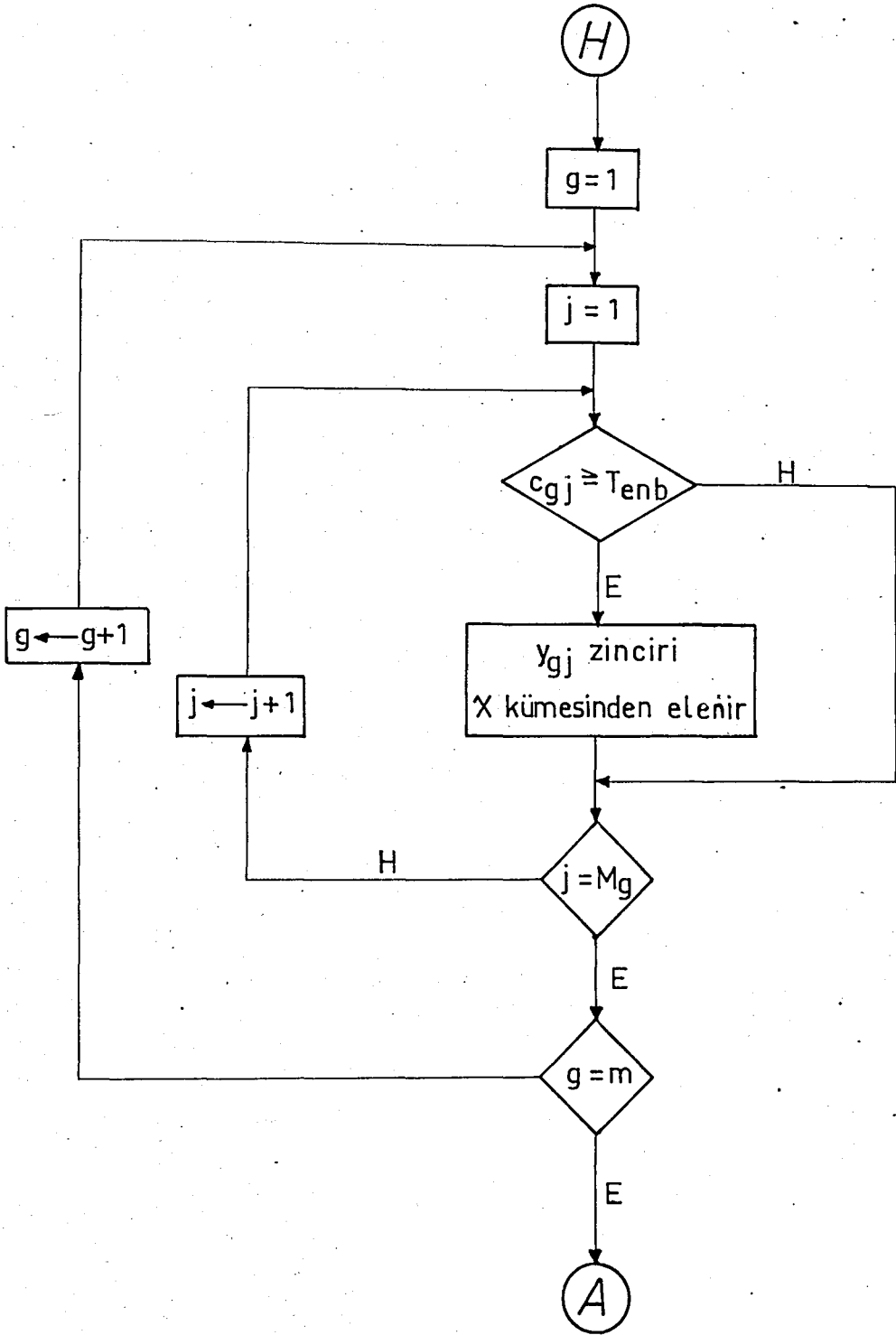
(Şekil 3.6)



(Şekil 3.6)



(Şekil 3.6)



(Şekil 3.6)

Özel küme parçalama algoritması ve X kümesini daraltma aşamaları aşağıda verilmiştir.

Adım 1: B matrisi elde edilmiş ve Tablo 3.3. de verilmiştir. $Y = \{\emptyset\}$, $z = 0$, $\bar{Y} = \{\emptyset\}$, $z = \infty$, $E = \{\emptyset\}$, $u_1 = u_2 = 0$, $U_1 = 1$, $U_2 = 2$, $T_{enb} = \infty$.

Adım 2: $g = \text{enk}\{i|k_i \notin E\} = 1$. $y_{11}(1)$ işaretlendi.
 $j = 1$.

Adım 3: $y_{11}(1) \cap E = \{\emptyset\}$, $z + c_{11} = 0 + 20 < \infty$,
 $u_1 + 1 = 0 + 1 = 1$.

Bu koşullar sağlandığından $y_{11}(1)$ işareti muhafaza edilir ve Adım 5'e gidilir.

Adım 5: $Y = \{y_{11}(1)\}$, $z = 20$, $E = \{6,8\}$, $u_1 = 1$.
 $E \neq K$, Adım 2 ye gidilir.

Adım 2: $g = \text{enk}\{i|k_i \notin E\} = 2$. $y_{21}(2)$ işaretlendi.
 $j = 1$.

Adım 3: $y_{21}(2) \cap E = \emptyset$, $20 + 22 < \infty$, $0 + 1 < 2$. Bu koşullar sağlandığından $y_{21}(2)$ nin işareti muhafaza edilir ve Adım 5'e gidilir.

Adım 5: $Y = \{y_{11}(1), y_{21}(2)\}$, $z = 42$, $E = \{3,5,6,8\}$,
 $u_2 = 1$. $E \neq K$, Adım 2'ye gidilir.

Adım 2: $g = \text{enk}\{i|k_i \notin E\} = 3$. $y_{31}(1)$ işaretlendi.
 $j = 1$.

Adım 3: $y_{31}(1) \cap E \neq \emptyset$. $j + j+1$. $y_{32}(1) \cap E = \emptyset$,
 $42 + 24 < \infty$, $1 + 1 \neq 1$; $j + j + 1$. $y_{33}(2) \cap E \neq \emptyset$. $j + j + 1$.
 $y_{34}(1) \cap E \neq \emptyset$. $y_{31}(1)$ üzerindeki işaret kaldırılır ve Adım

4 e gidilir.

Adım 4: $Y \neq \{\emptyset\}$. $y_{21}(2)$ Y kümesinden çıkartılır. $z = 42 - 22 = 20$, $E = \{3, 5, 6, 8\} - \{3, 5\} = \{6, 8\}$, $u_2 = 1 - 1 = 0$. $y_{21}(2)$ işareti kaldırılır. $y_{22}(1)$ işaretlenir, $g = 2$, $j = 2$. Adım 3'e gidilir.

Adım 3: $y_{22}(1) \cap E = \emptyset$, $20 + 26 < \infty$, $E = \{6, 8, 3, 7\}$, $1 + 1 \leq 1$. $j \leftarrow j + 1$. $y_{23}(2)$ $E \neq \emptyset$, $j \leftarrow j + 1$. $y_{24}(2) \cap E = \emptyset$, $20 + 31 < \infty$, $E = \{6, 8, 3, 4, 7\}$, $0 + 1 < 2$. Bu koşullar sağlandığından $y_{24}(2)$ işaretlenir, $y_{22}(1)$ in işareti kaldırılır, Adım 5 e gidilir.

İlk olurlu çözüm, $\bar{Y} = \{y_{12}(1), y_{21}(2), y_{51}(2)\}$ ve $z = 80$ dir. $C_{12} = 26$, $C_{21} = 22$, $C_{51} = 32$. Adım 5 de, $32 < T_{enb} = \infty$ olduğundan $T_{enb} = 32$ yapılır. Adım 4 e gidilir.

Algoritmanın uygulanmasına devam edildiğinde elde edilen ikinci daha iyi olurlu çözüm, $\bar{Y} = \{y_{13}(2), y_{21}(2), y_{32}(1)\}$, ve $z = 77$ dir. $C_{13} = 31$, $C_{21} = 22$, $C_{32} = 24$. Adım 5 de, $31 < 32$ olduğundan $T_{enb} = 31$ yapılır. Adım 4 e gidilir.

Bu çözüm aynı zamanda özel küme parçalama algoritmasının ilk uygulamasının eniyi çözümüdür. $T_{enb}^1 = 31$ olduğundan X kümesinden görev süresi 31 ve daha büyük olan zincirler ele-
nir. Daraltılmış X kümesi, $X = \{x_{13}, x_3, x_{14}, x_4, x_{16}, x_1, x_{12}, x_7, x_{17}\}$ olarak belirlenir (Tablo 3.4).

Özel küme parçalama algoritması bu daraltılmış X kümesi üzerinde yeniden uygulanır. Bu uygulamada tek bir olurlu çözüm bulunur ve bu eniyi çözümü oluşturmaktadır:

$\bar{Y} = \{y_{12}(1), y_{23}(2), y_{41}(2)\}$, $z = 83$. $C_{12} = 26$, $C_{23} = 28$, $C_{41} = 29$. Bu durumda $T_{enb}^2 = 29$ dur.

Tablo 3.4. Daraltılmış X Kümesi

i	k_i	x_{13}	x_3	x_{14}	x_4	x_{16}	x_1	x_{12}	x_7	x_{17}
1	6	1	1	0	0	0	0	0	0	0
2	3	0	0	1	1	1	0	0	0	0
3	4	0	1	0	0	0	1	1	1	0
4	5	0	0	1	0	0	1	0	1	1
5	7	0	0	0	1	0	0	1	0	1
6	8	1	0	0	0	1	0	0	1	0
	h	1	1	2	1	2	1	1	2	2
	c	20	26	22	26	28	19	24	27	29

$T_{enb}^2 = 29$ değeri kullanılarak daraltılmış X kümesinde görev süresi 29 ve daha büyük olan yalın yollar elenir. Daraltılmış X kümesi, $X = \{x_{13}, x_3, x_{14}, x_4, x_{16}, x_1, x_{12}, x_7\}$ dir; yani Tablo 3.4. de verilen B matrisinin sadece son kolonu elenmiştir.

Bu matris üzerinde yeniden uygulanan özel küme parçalama algoritması olurlu bir çözüm bulamaz. Buna göre intikal planlaması probleminin eniyi çözümü:

$Z^* = 29$, ve uçuş güzergahları x_3 , x_{16} ve x_{17} dir. Yani hareketin üç uçakla gerçekleştirilebileceği en kısa süre 29 zaman birimidir.

Bu örnekte ilginç bir nokta özel küme parçalama algoritmasının ikinci uygulamasında bulunan çözümün birinci uygulama-
lanışta olurlu olmasına rağmen gözükmemesidir. Bu olgu çözüm yönteminde özel küme parçalama algoritmasının defaeten uygulanması ardındaki mantığı sergilemektedir.

4. ASKERİ UÇAKLARIN İNTİKAL PLANLAMASI İÇİN BİR SEZGİSEL ALGORİTMA

4.1. GİRİŞ

Bir önceki bölümde matematiksel programlama formülasyonu ve bir kesin bir çözüm yöntemi verilen intikal planlaması problemi için bu bölümde bir sezgisel algoritma sunulmaktadır. Probleme burada kapasite sınırı baştan itibaren eklenmektedir. Böylece problem daha gerçekçi bir hale getirilmiş olmaktadır. Önerilen sezgisel algoritma ayrıntıları ile anlatılacak ve çalışma mekanizmasını açıklamak amacı ile bir örnek problem çözülecektir. Sezgisel yöntem beş adet örnek probleme uygulanacak ve elde edilen sonuçlar bu problemlerin eniyi çözümleri ile karşılaştırılacak ve değerlendirilecektir. Sezgisel ve kesin çözüm yöntemlerinin müşterek bir değerlendirmesi altbölüm 6.2 de verilecektir.

4.2. SEZGİSEL YAKLAŞIM

4.2.1. Sezgisel Yaklaşımın Nedenleri

Altbölüm 3.3'de de gösterildiği gibi intikal planlaması problemi kombinatoryal niteliğinden ötürü oldukça büyük bir problemdir. Uçak, havaalanı ve zaman dilimi sayılarında ufak artışlarla problemin karar değişkeni sayısı büyük artışlar göstermektedir. Bölüm 3'de önerilen kesin çözüm yöntemi başarılı olmakla birlikte yalın yolların türetilmesi yordamı bir darboğaz oluşturmaktadır.

Gerek probleme önerilen kesin çözüm yönteminin büyük problemlerde nispeten büyük çözüm süreleri gerektirmesi, gerekse problemin iyi tanımlanmış bir problem olması sezgisel

yöntemleri denemek için gereklilik ve olurluluk koşullarını sağlamışlardır. Burada problemin iyi tanımlanmış olmasından maksat, karar değişkenleri değerleri bilindiğinde sonucun olurlu olup olmadığının saptanabilmesi ve amaç işlevinin değerinin hesaplanabilmesidir. Sezgisel yöntemin burada bir avantajı formülasyonda esnekliğe yer verebilmesidir. Problemin giderrek daha gerçekçi olabilmesi için yapılacak değişikliklerin çözüm yapısı içine sindirilebilmesi sezgisel yöntemlerde kesin yöntemlere göre daha olasıdır.

Yöneylem araştırması literatüründe genellikle eniyileme yöntemleri sezgisel yöntemlerin bir altyordamı olarak kullanılırlar. Burada sezgisel bir yöntem araştırmanın bir nedeni de bunun tersi bir yaklaşımın, yani eniyileme yöntemi içinde sezgisel yöntemin kullanılabilirliğinin bu probleme uygulanabilirliğini denemektir.

4.2.2. Temel Sezgisel Yöntem Yaklaşımları

Burada önerilen sezgisel yöntemin genelde sezgisel yöntemler arasında nereye düştüğünü bilmek yöntemi değerlendirmek açısından yararlı olacaktır, kanısındayız. Sezgisel yöntemler altı grupta toplanabilirler (Silver v.d., 1980):

i. Parçalama Yöntemleri. Burada problemler yapıya uygun olarak parçalara ayrılırlar ve her bir parçaya kendi içinde çözüm aranır. Ancak parçalar arasında bir geçiş kurulmaya çalışılır. Dikkat edilmesi gereken husus parçalama ile bütünün özelliklerinden kaybetmemektir.

ii. Genelleme Yöntemleri. Burada eldeki problem daha küçük boyutlarda çözülerek esas problemin çözüm ve parametre özellikleri üzerinde bir fikir oluşturulmaya çalışılır.

iii. Özellik İndirgeme Yöntemleri. Probleme ilişkin

bazı sayısal örneklere eniyi çözüm bulunur. Bu çözümlerden hareketle eniyi çözümlerin bazı müşterek özellikleri bulunmaya çalışılır; örneğin bir kısıtın hiç bir zaman bağlayıcı olmaması gibi. Bu özellikler kullanılarak problemin basitleştirilmesine çalışılır.

iv. Model Yapısında Basitleştirme. Model yapısında yapılacak değişikliklerle probleme çözüm getirilmeye çalışılır; örneğin doğrusal olmayan bir amaç işlevini doğrusal kabul etmek, bazı kısıtları ortadan kaldırmak gibi. Değişikliğe uğramış modelden elde edilen sonuçların yapılan değişiklikler doğrultusunda değerlendirilmesi gerekir.

v. Bina Eden Yöntemler. Bu tür yöntemlerde olurlu çözüm bir çekirdek etrafında adım adım bina edilir. Her adımda eniyi karar alınmaya çalışılır. Ancak bütün bir kararlar dizisi birlikte gözönüne alınmadığı için elde edilen sonuç genelde bir alteniyi çözüm olacaktır.

vi. Yerel Geliştirme Yöntemleri. Bina eden yöntemlerin aksine yerel geliştirme yöntemleri olurlu bir çözümle işe başlarlar. Bu başlangıç çözümünde bazı esaslara uygun olarak değişiklikler yapılarak daha iyi sonuçlara gidilmeye çalışılır. Hangi değişikliklerin yapılması ve değişikliklerin ne zaman sona erdirilmesi gerektiğinin saptanması sezgisel yöntemin temel kararlarını teşkil eder. Sezgisel yöntemin başarısını belirleyen diğer bir husus başlangıç çözümünün kalitesidir.

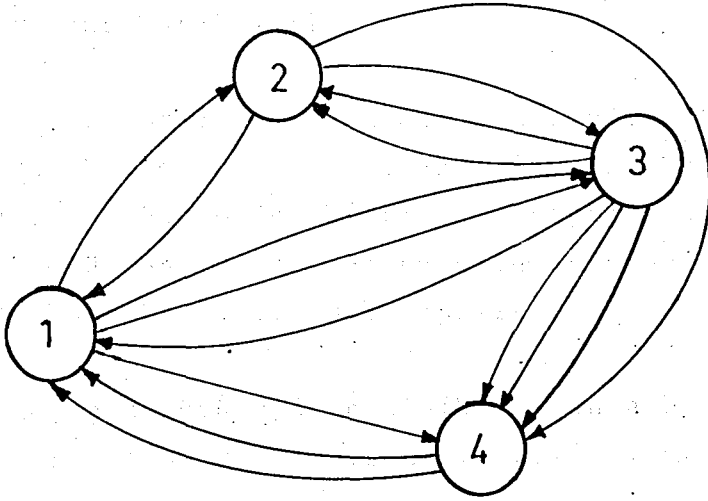
4.2.3.

Önerilen Sezgisel Yaklaşımın Ana Hatları

Burada önerilen sezgisel yaklaşım yukarıdaki temel yaklaşımlar arasındaki son gruba girer, yani bir yerel geliştirme yöntemidir. Buna göre, önce bir başlangıç çözümü yaratılacak; daha sonra bunun üzerinde yapılacak değişikliklerle

geliştirilmesine çalışılacaktır.

Başlangıç çözümünün yaratılmasında kullanılan yaklaşım taşıt güzergahı belirleme probleminin önce güzergahın belirlenip sonra talep noktalarının gruplandırıldığı türdeki yaklaşımı olmuştur. Dört düğümlü bir örnekte gerekli uçuşların birer ayrıtla gösterildiği serim Şekil 4.1'deki gibi olsun. Burada iki düğüm arasındaki ayrıt sayısı ayrıtın bitiş nokta-



Şekil 4.1. Örnek Serim

sının başlangıç noktasından yük talebini gösterir. Buna göre Şekil 4.1. deki problemin yük talep matrisi Tablo 4.1. deki gibidir.

Tablo 4.1. Yük Talep Matrisi

	1	2	3	4
1	0	1	2	1
2	1	0	1	1
3	1	2	0	3
4	2	0	0	0

Şekil 4.1. deki serimin ayrıtları yapılması zorunlu uçuşları göstermektedir. Ancak intikal planlaması problemin serimindeki her iki nokta doğrudan iki yönlü bağlıdır. Yük taleplerine karşı gelen bütün ayrıtları kaplayan ancak en küçük tur uzunluğuna sahip bir tur tek uçakla en kısa sürede bütün talepleri karşılayan turdur. Bu tur Çinli postacı probleminin çözümü olarak bulunur. Ancak geçilmesi zorunlu ayrıtların dışındaki ayrıtların da gerekli boş uçuşlar kullanılmasına izin verildiğinde genelde daha kısa bir tur elde edilecektir. Bu tur kentli postacı turuna karşı gelir.

Elimizdeki problemde M adet uçak vardır. Bu nedenle bu tur M alttura bölündüğünde her uçak için bir güzergah elde edilmiş olur. Bu bölünme uçakların başlangıçta bulundukları havaalanı gözönüne alınarak mümkün olduğunca eşit uzunlukta güzergahlar yaratılacak biçimde yapılır.

Başlangıç çözümü elde edildikten sonra bu çözüme geliştirme yordamı (routine) uygulanır. Geliştirme yordamı şu altıyordamlardan oluşur:

- i. Boş uçuş enküçüklemesi
- ii. İki uçuş kullanılarak dengeleme
- iii. Tek uçuş kullanılarak dengeleme
- iv. Boş uçuş enküçüklemesi

Bu aşamada uçuşların bir güzergah içinde ve güzergahtan güzergaha yerlerinin değiştirilmesi ile başlangıç çözümü geliştirilmeye çalışılır. Özellikle boş uçuşlar burada önemli bir rol oynarlar.

Son aşama olurluluk denetimidir. İlk iki aşama neticesinde elde edilen çözümün alan kapasiteleri açısından olurlulukları araştırılır. Olurlu ise kabul edilir; değilse enküçük uçuşlu güzergahtan başlayarak yeni olursuzluğa neden olmadan

gerekirse güzergah süresini uzatarak mevcut olursuzluk ortadan kaldırılır.

4.3. AYRINTILARI İLE ÇÖZÜM YÖNTEMİ

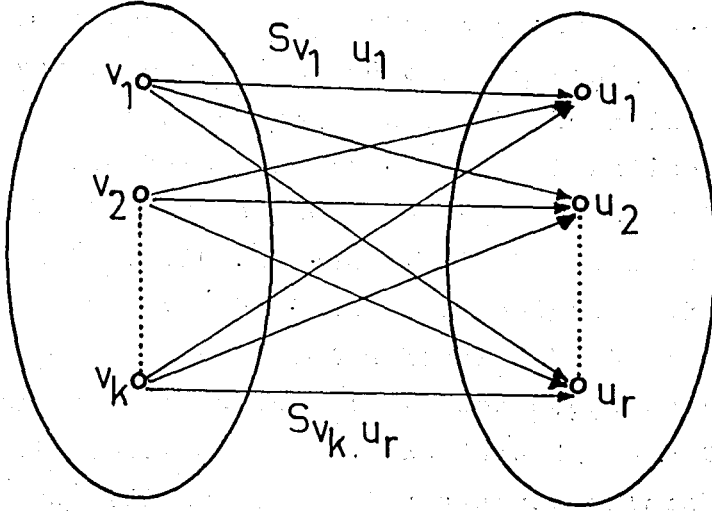
4.3.1. Başlangıç Çözümünün Bulunması

Başlangıç çözümü daha önce belirtildiği gibi iki aşamada bulunmaktadır: (1) kentli postacı turu bulunur; (2) tur M parçaya bölünür.

Ayrıtların yönlü olması nedeni ile sezgisel yöntemde sadece yönlü kentli postacı problemi çözülecektir. Problemin iki özelliğinden yararlanılarak yönlü kentli postacı problemi çözüm yöntemine (Orloff, 1974a) benzeyen ancak çözümü daha kolay bir yöntem kullanılmıştır. Bu iki özellik şunlardır: (i) problemde her iki düğüm iki yönlü olarak birbirine bağlıdır; (ii) ayrıtlar uçuşlara karşı geldiğinden iki düğüm arasında düz çizgi ile ifade edilirler ve üçgen eşitsizliğine (triangle inequality) uyarlar.

i düğümüne giren ayrıtların sayısını d_i^+ , çıkan ayrıtların sayısını d_i^- ile gösterelim. $d_i = (d_i^+ - d_i^-)$ değeri i düğümünün arzını gösterir. Eğer bu değer (+) ise, i düğümü arz düğümü; (-) ise, i düğümü talep düğümü; (0) ise bir aktarma (transshipment) noktasıdır. Dolu uçuşlara karşı gelen ayrıtlar kullanılarak bütün düğümlerin d_i değeri bulunur. Sıfırdan değişik d_i değeri olan düğümler yeni bir serimin düğüm kümesini oluştururlar. Bu düğümler iki altkümeye ayrılırlar. Bir altkümede d_i değeri (>0) olan düğümler; diğer altkümede d_i değeri (<0) olan düğümler yer alır. Yeni serimde ayrıtlar sadece $d_i > 0$ altkümesi elemanlarından $d_i < 0$ altkümesi elemanlarına giderler. Bu problemin (ii). özelliğinin bir neticesidir. (i,j) ayrıtının ağırlığı doğru-

dan i ve j düğümleri arasındaki boş uçuş süresine eşittir. Böylece belirlenen yeni serim (Şekil 4.2) tipik ulaştırma problemi serimidir. Bu serim üzerinde çözülen ulaştırma problemi esas serimde kentli postacı turu elde etmek için gerekli ayrıtlar demetinin enküçük maliyetli olanını bulacaktır.



Şekil 4.2. Ulaştırma Probleminin Çözüldüğü Serim

Ulaştırma problemi neticesi bulunan boş uçuş ayrıtlarının dolu uçuş ayrıtları ile birlikte oluşturduğu kentli postacı turu başlangıç çözümünün ilk aşaması olan ana turdur.

İkinci aşamada uçak güzergahları yaratılır. Burada yaklaşım sezgiseldir. Mümkün olduğunca eşit parçalar halinde olması amaçlanarak, ana tur uçakların başlangıçta bulundukları havaalanları ilk düğüm olmak şartı ile M adet alttura bölünür. Bu işlem sırasında güzergahların son uçuşunun boş uçuş olamayacağı gözönüne alınır.

4.3.2. Geliştirme Yordamı

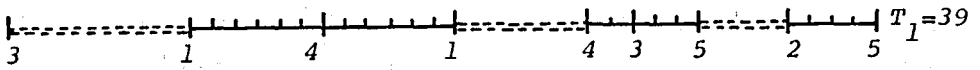
4.3.2.1. Boş Uçuş Enküçükleme

Burada her uçak için boş uçuşların toplam süresi enkü-

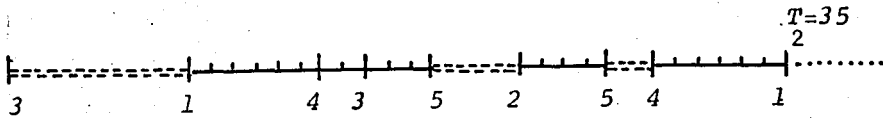
çüklenmeye çalışılır. Değişiklikler aynı güzergah üzerinde yapılır. Algoritma her boş uçuş için ilk boş uçuştan başlayarak bunun güzergah içinde nasıl yok edilebileceğini araştırır. Bu iki şekilde gerçekleştirilir.

i. Geriden Tarama

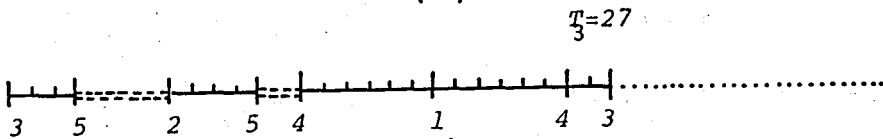
Eğer boş uçuşun bitiş noktası güzergah üzerinde daha önce geçiliyorsa, o takdirde güzergahın son noktası ile bu nokta arasındaki boş uçuşun süresi yok etmek istediğimiz boş uçuşun süresi ile karşılaştırılır. Eğer kısa ise incelenen boş uçuşu yok etmek yararlıdır. Örneğin Şekil 4.3.a.(*) daki güzergaha bakalım. İlk boş uçuş (3,1) boş uçuşudur. İlk uçuş olduğundan 1 noktası daha önce geçilmemiştir. İkinci boş uçuş olan (1,4) uçuşuna geçilir. 4 noktası daha önce geçilmiştir. (1,4) uçuşunu yok etmek için (5,4) boş uçuşunun süresi ile karşılaştırılır. $S_{54} = 2$ olsun. Bu durumda, $6 > 2$ olduğundan (1,4) uçuşu yok edilir ve (5,4) uçuşu yaratılarak daha önce geçilen 4 noktası ile (1,4) boş uçuşunun başlangıç noktası olan 1 noktası arasındaki blok (5,4) uçuşunun arkasına takılır (Şekil 4.3.b.). Toplam tasarruf 4 zaman birimidir.



(a)



(b)



(c)

Şekil 4.3. Boş Uçuş Enküçüklemesi Örneği

(*) Bu altbölümdeki şekillerde boş uçuşlar çift kesikli çizgi, dolu uçuşlar ise düz çizgi ile gösterilmişlerdir.

ii. Önden Tarama

Bu taramada da yok edilmek istenen boş uçuşun başlangıç noktasının bu boş uçuştan sonra tekrar geçilip geçirilmediği araştırılır. Geçiliyorsa, güzergahın son noktası ile boş uçuşun bitiş noktası arasındaki boş uçuş süresi elimizdeki boş uçuş süresi ile karşılaştırılır. Eğer küçükse incelenen boş uçuşun yok edilmesi yararlıdır. Örneğin, Şekil 4.3.b. de boş uçuş (3,1) i inceleyelim. Başlangıç noktası tekrar geçilmektedir. Güzergahın bitiş noktası 1 noktasıdır (1,1) süresi sıfır olduğuna göre (3,1) uçuşu yok edilebilir. Boş uçuşun bitiş noktası 1 ile tekrar geçilen 3 noktası arasındaki blok tümü ile güzergahın sonuna yeni boş uçuş eklendikten sonra takılır. Ancak bu örnekte yeni boş uçuş (1,1) olduğundan blok doğrudan güzergah sonuna eklenir (Şekil 4.3.c.). Tasarruf 8 zaman birimidir.

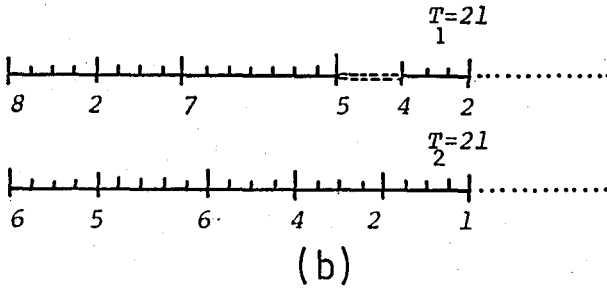
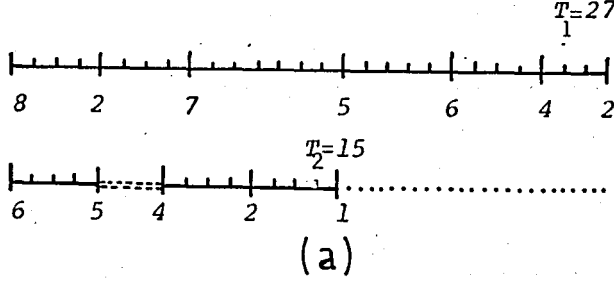
4.3.2.2. İki Uçuş Kullanarak Dengeleme

Burada arka arkaya gelen iki dolu uçuşu bir güzergah-tan diğer bir güzergaha taşıyarak bir tasarruf yapma olanağı araştırılacaktır. Böyle bir değişiklik ancak iki güzergah arasında değişiklik yapılmadan önceki net farkın değişiklik yapıldıktan sonra oluşacak net farktan büyük olması durumunda olur-ludur. Arka arkaya iki dolu uçuş aramaya güzergahın sonundan başlanır.

i. Boş Bir Uçuşa İki Uçuş Yerleştirilmesi

Bir güzergahın arka arkaya dolu iki uçuşu diğer güzergahın beş uçuşu yerine yerleştirilmeye çalışılır. Örnek olarak Şekil 4.4.a. daki iki güzergahı alalım. Uzun güzergahta arka arkaya iki dolu uçuşa bakılır. (6,4) ve (4,2) böyle uçuş-lardır, ancak (5,4) boş uçuşuna yerleştirilemezler. Bundan sonraki kombinasyon (5,6) ve (6,4) kombinasyonudur. Bu kombi-

nasyon ikinci güzergahtaki (5,4) boş uçuşa yerleştirilebilir.

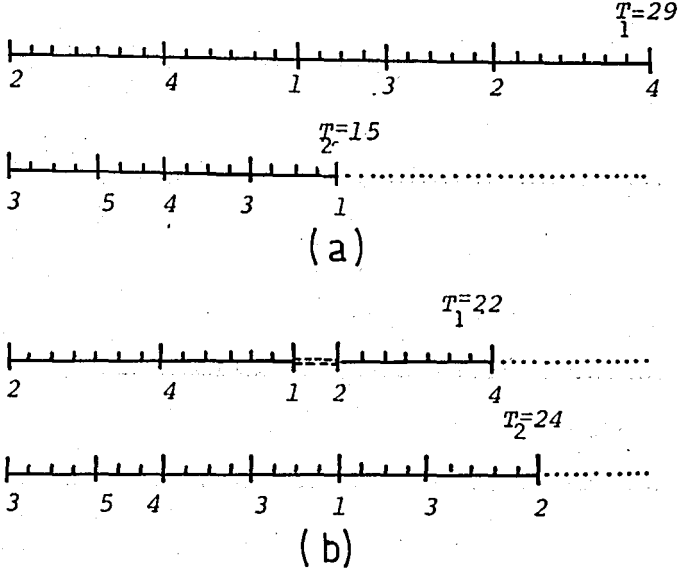


Şekil 4.4. Boş Bir Uçuşa İki Uçuş Yerleştirilmesi Örneği

Ancak bu ilk güzergahta (5,4) boş uçuşunu yaratacaktır. $(T_2 - S_{54} + (S_{56} + b + d) + (S_{64} + b + d))$ değerinin $T_{enb} = T_1$ değerinden daha küçük olması halinde değişiklik gerçekleştirilir. $(15-2 + 5 + 4) < 27$ dir. Bu nedenle değişiklik yapılır (Şekil 4.4.b.) ve 6 zaman birimlik tasarruf sağlanır.

ii. Boş Uçuş Kullanmadan İki Uçuş Eklenmesi

Bir güzergahın arka arkaya dolu iki uçuşın diğer güzergahın sonuna boş uçuş kullanmadan eklemenin olurluluğu araştırılır. Buna göre iki uçuş bloğunun başlangıç noktası ile diğer güzergahın son noktasının aynı olması durumunda bu değişiklik gerçekleştirilir. Örneğin Şekil 4.5.a. daki ilk güzergahta (3,2) ve (2,4) arka arkaya iki dolu uçuştur. Bu blok boş uçuş kullanmadan ikinci güzergaha eklenemez. Bundan sonraki iki dolu uçuş (1,3) ve (3,2) dir. Görüldüğü gibi bu iki uçuş ikinci güzergaha eklenebilir. Ancak bu ilk güzergahta

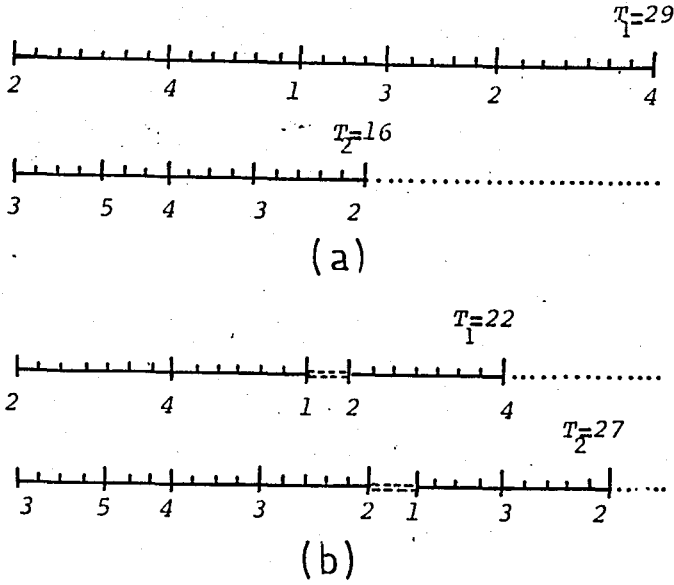


Şekil 4.5. Boş Uçuş Kullanmadan İki Uçuş Eklenmesi Örneği

bir boş uçuşa neden olacaktır. Bunun olurluluğu incelenir. $(T_2 + (S_{13} + b + d) + (S_{32} + b + d))$ değerinin $T_{enb} = T_1$ değerinden küçük olması halinde değişiklik gerçekleştirilir. $(15 + 4 + 5) < 29$ olduğu için değişiklik yapılır (Şekil 4.5.b.) ve 5 birimlik bir tasarruf sağlanır.

iii. Boş Uçuş Kullanarak İki Uçuş Eklenmesi

(ii). olasılık gerçekleşmediğinde algoritma bu sefer aynı işlemi boş bir uçuş kullanarak gerçekleştirmenin olurluluğunu araştırır. Şekil 4.8. deki örneğe bakalım. Şekil 4.6.a. daki ilk güzergahın ilk arka arkaya dolu iki uçuşu (3,2) ve (2,4) dür. Bunu ikinci güzergaha eklemek için (2,3) boş uçuşuna gereksinme olacaktır. $(T_2 + S_{23} + (S_{32} + b + d) + (S_{24} + b + d) + (S_{24} + b + d))$ değerinin $T_{enb} = T_1$ değerinden küçük olması halinde bu değişiklik yapılır. $(16+3+5+7)>29$ olduğundan değişiklik yapılmayacaktır. İkinci arka arkaya dolu iki uçuş aranır. Bunlar (1,3) ve (3,2) dir. Bu durum için, $(16+2+4+5)<29$ dur ve değişiklik yapılarak Şekil 4.6.b. deki



Şekil 4.6. Boş Uçuş Kullanarak İki Uçuş Eklenmesi Örneği

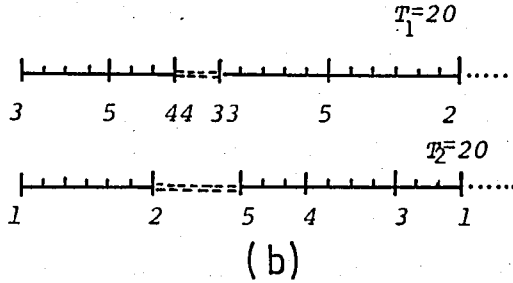
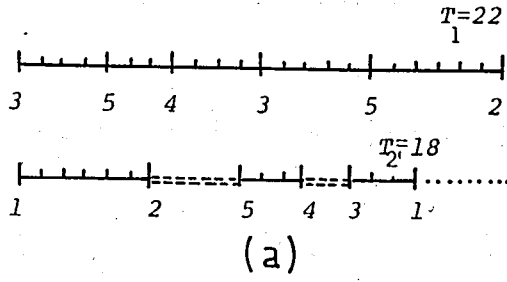
iki güzergah elde edilir. Tasarruf miktarı 2 zaman birimidir.

4.3.2.3. Bir Uçuş Kullanarak Dengeleme

Algoritma bu aşamada T_{enb} 'e sahip güzergahın son uçuşundan başlayarak dolu uçuşları diğer bir güzergaha (mümkünse enküçük görev süreli güzergaha) aktarmanın olurluluğunu araştırır. Eğer bu değişiklik neticesi oluşan yeni bir güzergahın görev süreleri T_{enb} 'den küçükse bu değişiklik olurludur. Bu tür dengeleme üç ayrı şekilde yapılabilir.

i. Bir Uçuşun Bir Boş Uçuşa Yerleştirilmesi

Algoritma T_{enb} 'e sahip güzergahın her dolu uçuşunu diğer bir güzergahın boş uçuşuna yerleştirme olanağını arar. Netice olarak yapılan, $(b+d)$ süresini görev süresi küçük olan güzergaha transfer etmektir. Eğer bunun neticesinde bu güzergahın görev süresi T_{enb} 'den küçük olursa bu işlem olurludur.



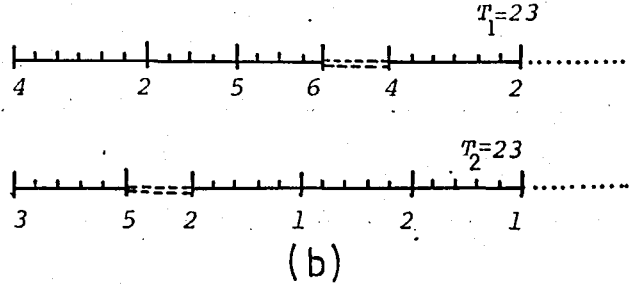
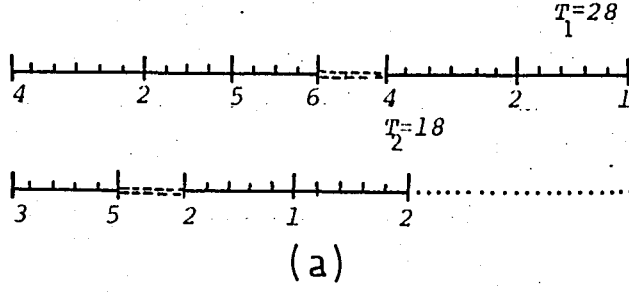
Şekil 4.7. Bir Uçuşun Bir Boş Uçuşa Yerleştirilmesi Örneği

Şekil 4.7.a. daki ilk güzergahın dolu uçuşları sondan başlayarak tarandığında (4.3) uçuşunun birinci güzergahta dolu, ikinci güzergahta boş olduğu görülür. $(T_2 + b + d)$ değeri T_{enb} ile karşılaştırılır. $(18 + 1 + 1) < 22$ olduğundan (4,3) uçuşu birinci güzergahta boş ikinci güzergahta dolu uçuş yapılarak Şekil 4.7.b. deki güzergahlar elde edilir. Tasarruf miktarı 2 zaman birimidir.

ii. Boş Uçuş Kullanmadan Bir Uçuşun Eklenmesi

Burada algoritma en uzun görev süreli güzergahın bir dolu uçuşunu diğer bir güzergahın sonuna boş uçuş kullanmadan eklemenin olurluluğunu araştırır. Bunun neticesinde ekleme yapılan güzergahın görev süresi T_{enb} değerinden küçük olamazsa bu ekleme olurludur. Şekil 4.8.a. da birinci güzergahta ilk incelenen dolu uçuş (2,1) uçuşudur. İkinci güzergahın son noktası 2 olduğundan bir ekleme yapılabilir. Bunun olurluluğunu araştırmak için $(T_2 + (S_{21} + b + d))$ değeri T_{enb} ile karşılaştırılır. $(18 + 5) < 28$ olduğundan yapılacak ekleme olurludur ve Şekil 4.8.b. de gösterilmiştir. Bu değişiklik neticesi

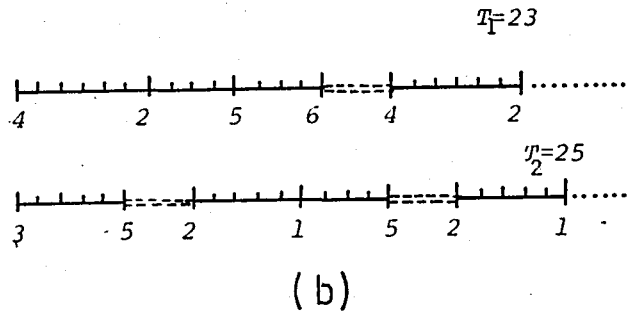
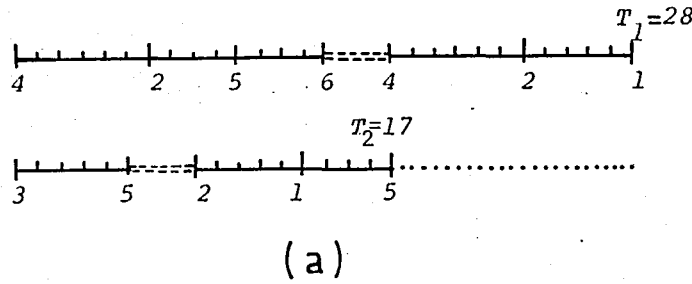
5 birimlik bir tasarruf sağlanmıştır.



Şekil 4.8. Boş Uçuş Kullanmadan Bir Uçuşun Eklenmesi Örneği

iii. Boş Uçuş Kullanarak Bir Uçuşun Eklenmesi

Eğer (ii.) olasılık gerçekleşmezse algoritma aynı değişikliği bir boş uçuş kullanarak yapmanın olurluluğunu araştırır. Örneğin Şekil 4.9.a. daki iki güzergaha bakalım.



Şekil 4.9. Boş Uçuş Kullanarak Bir Uçuşun Eklenmesi Örneği

Birinci güzergahın ilk incelenecek dolu uçuşu (2,1) dir. İkinci güzergahın son noktası 5 olduğundan bir ekleme doğrudan yapılamamakla birlikte (5,2) boş uçuşu kullanılarak yapılabilir. Eğer $(T_2 + S_{52} + (S_{21} + b + d))$ değeri T_{enb} 'den küçükse ekleme olurludur. $(18 + 2 + 5) < 28$ olduğundan ekleme yapılır ve Şekil 4.9.b. deki iki güzergah elde edilir. Yapılan tasarruf 3 zaman birimidir.

Algoritmanın uygulanmasında geliştirme yordamı içinde bir uçuş kullanarak dengeleme işlemlerinden sonra 4.3.2.1. de izah edilen boş uçuş enküçüklenmesi işlemleri tekrarlanır. Geliştirme yordamı neticesi elde edilen çözüme olurluluk denetimi uygulanır.

4.3.3. Olurluluk Denetimi

Olurluluk denetiminde problemin matematiksel programlama formülasyonunda kullanılan zaman birimi olan bütün sürelerin enbüyük ortak böleni birimi kullanılır. Gerçek zaman t bu birim cinsinden ifade edilir. Her uçak için bir dizey açılır. Dizeyin j . elemanı $t = j$ 'e karşı gelir ve eğer o uçak herhangi bir havaalanında $t = j$ 'de işlem görüyorsa o havaalanının numarasını değer olarak alır; aksi taktirde 0 değeri alır. Bütün uçakların görev süreleri için bir dizey oluşturulduktan sonra bu dizeylerin her girdisinde kapasite denetimi yapılır.

Örnek olarak aşağıdaki dizeylere bakalım. Burada dört uçağın güzergahları yukarıda tanımlanan biçimde dizeyler ha-

1.	1	1	0	0	0	2	2	2	0	0	0	0	3							
2.	3	3	0	0	0	0	2	0	0	0	4	4	0	0	0	2	2			
3.	3	3	0	0	1	1	1	0	0	0	0	4	4	4	0	0	0	2	2	
4.	2	2	0	0	0	0	3	3	3	0	0	4	4	4	0	0	0	0	1	1
$t =$	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0

linde ifade edilmişlerdir. $t = 7$ zaman diliminde 2 numaralı havaalanında 2; 1 numaralı havaalanında 1; 3 numaralı havaalanında ise 1 uçak bulunmaktadır. Dizeylerden her zaman dilimi için kolaylıkla elde edilen bu bilgi havaalanlarının kapasiteleri ile karşılaştırılır. Eğer 4 numaralı havaalanının kapasitesi 2 uçak ise bu kapasite kısıtı $t = 12$ 'de ihlal edilmektedir.

Havaalanı kapasiteleri ihlal edildiğinde bunun düzeltilmesi için aşağıdaki adımlar sırası ile uygulanır.

Adım 1: Kapasitenin kaç uçakla ihlal edildiği bulunur. Eğer kapasite ihlali kalmamışsa durulur; olurlu çözüm elde edilmiştir. Aksi takdirde kapasiteyi ihlal eden uçaklar aralarında enküçük olurlu tehir sırasına göre sıralanırlar.

Adım 2: Enküçük olurlu tehirli uçaktan başlanarak uçağın güzergahı üstünde daha önce geçilen havaalanlarında kapasite sınırını ihlal etmeden bekletilme olanağı araştırılır. Eğer böyle bir olanak varsa bu kullanılır. Yeniden Adım 1'e gidilir. Eğer yoksa Adım 3 e gidilir.

Adım 3: Enbüyük görev süresi bir zaman birimi arttırılarak Adım 1 e gidilir.

Örneğin, yukarıda $t = 12$ de 4 numaralı havaalanında kapasitenin bir uçak ile ihlal edildiğini varsayalım. İhlal eden uçaklar 2,3 ve 4 dür. 2. ve 4. uçaklarda yapılacak tehirler (yani daha önce geçtikleri havaalanlarında bekletme) olursuzdur çünkü $t = 20$ yi aşmaktadır. 3. uçağın bir zaman birimi tehiri ise olurludur. İkinci adımda daha önce geçilen havaalanları açısından bu tehirin olurluluğu araştırılır. Aşağıdaki dizeylerden görüleceği gibi bu olurludur ve olurlu çözüm elde edilir.

1.	1	1	0	0	0	2	2	2	0	0	0	0	3								
2.	3	3	0	0	0	0	2	0	0	0	4	4	0	0	0	2	2				
3.	3	3	0	0	1	1	1	1	0	0	0	0	4	4	0	0	0	2	2		
4.	2	2	0	0	0	0	3	3	3	0	0	4	4	4	0	0	0	0	1	1	
t =	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
										10										20	

Bu altbölümde ayrıntıları ile sunulan sezgisel yöntem sonraki altbölümde algoritma şeklinde verilecektir. Algoritma ana hatları ile Şekil 4.10. da sunulmuştur.

4.4. ALGORİTMA

Başlangıç Çözümünün Bulunması

Adım 1: Kentli postacı problemi çözülür ve ana tur elde edilir.

Adım 2: Ana tur elimizdeki M adet uçak için her uçağın başlangıçta bulunduğu havaalanından başlayan M alttura bölünür.

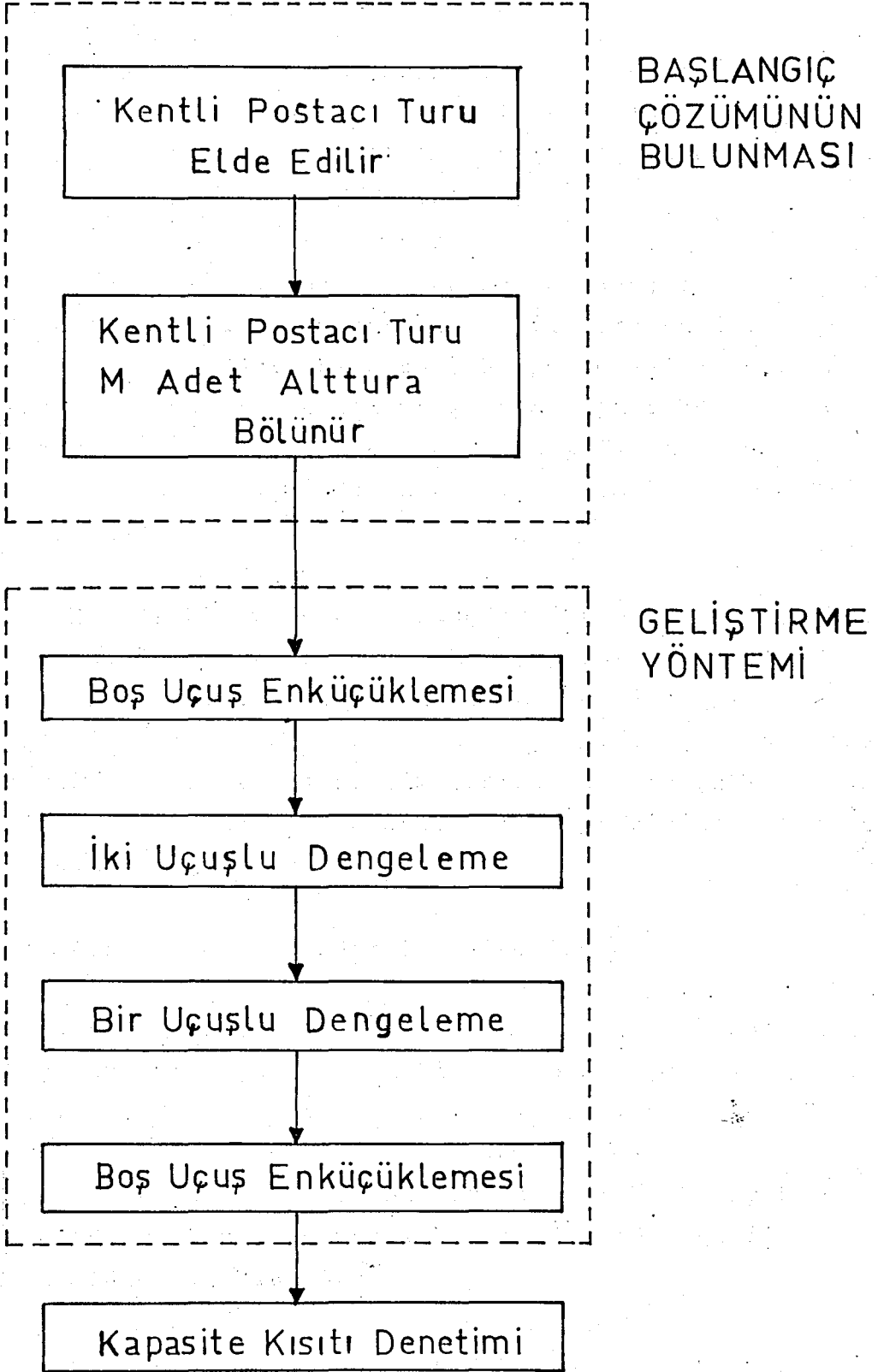
Boş Uçuş Enküçüklemesi

Adım 3: Her güzergaha önce geriden tarama, sonra önden tarama uygulanarak boş uçuş enküçüklemesi yapılır.

İki Uçuş Kullanarak Dengeleme

Adım 4: Enbüyük görev süresi olan k güzergahı seçilir. $T_{enb} = T_k$ ve $IM = 1$ yapılır.

Adım 5: k güzergahı son uçuşundan başlayarak ilk uçuşuna doğru taranarak arka arkaya iki dolu uçuşluk blok aranır. Böyle bir (u,v) bloku bulunduğu ve güzergahın ilk uçuşunun



Şekil 4.10. Ana Hatları ile Çözüm Yöntemi

blokun ilk uçuşu olmadığı durumlarda blok numarası $j = 1$ yapılır; blok uzunluğu T_b ile gösterilir ve $IM = 1$ ise Adım 7 ye; $IM = 2$ ise Adım 9 a gidilir. Böyle bir (u,v) bloku bulunmadığı takdirde Adım 11 e gidilir.

Adım 6: j bloğunun ilk uçuşu son uçuş olarak alınır ve güzergah ilk uçuşa doğru taranarak arka arkaya dolu iki uçaklık blok aranır. Böyle bir (u,v) bloku bulunduğu ve güzergahın ilk uçuşunun bloğun ilk uçuşu olmadığı takdirde $j \leftarrow j + 1$ yapılır ve blok uzunluğu T_b ile gösterilir ve Adım 7 ye gidilir. Bulunmadığı takdirde $IM = 1$ ise Adım 8 e; $IM = 2$ ise Adım 11 e gidilir.

Adım 7: Görev süresi enküçükten enbüyüğe doğru bütün $i \neq k$ güzergahları taranarak ilk (u,v) boş uçuşu olan h güzergahı aranır. Eğer yoksa Adım 6 ya gidilir. Eğer varsa, ancak $T_h + T_b + S_{uv} \geq T_{enb}$ ise Adım 6 ya gidilir. Aksi takdirde değişiklik yapılır. Değişiklik sonrası k güzergahı yine enbüyük güzergah uzunluğuna sahipse Adım 6 ya gidilir. Aksi takdirde Adım 4 e gidilir.

Adım 8: $IM = 2$. Adım 5 e gidilir.

Adım 9: Görev süresi enküçükten enbüyüğe doğru bütün $i \neq k$ güzergahlarından son noktası u olan ilk h güzergahı aranır. Bulunmadığı takdirde Adım 10 a gidilir. Bulunduğu takdirde, eğer $T_h + T_b \geq T_{enb}$ ise Adım 10 a gidilir. Aksi takdirde değişiklik yapılır. Değişiklik sonrası k güzergahı yine enbüyük güzergah uzunluğuna sahipse Adım 10 a gidilir. Aksi takdirde Adım 4 e gidilir.

Adım 10: Görev süresi enküçükten enbüyüğe doğru bütün $i \neq k$ güzergahları taranarak (y_i, u) boş uçuşu ile birlikte j bloğunun güzergah sonuna eklenmesi halinde $T_i + S_{y_i u} + T_b < T_{enb}$ eşitsizliği sağlanamazsa Adım 6 ya gidilir. Aksi takdir-

de ilk sağlayan güzergahta değişiklik yapılır. Değişiklik sonrası k güzergahı yine enbüyük güzergah uzunluğuna sahipse Adım 6 ya gidilir. Aksi takdirde Adım 4 e gidilir.

Bir Uçuş Kullanarak Dengeleme

Adım 11: Enbüyük görev süresi olan k güzergahı seçilir. $T_{enb} = T_k$ ve $ID = 1$ yapılır.

Adım 12: k güzergahının son uçuşundan başlayıp ilk uçuşa doğru taranarak bir dolu uçuş aranır. İlk (u,v) dolu uçuşu j = 1 olarak numaralanır. Eğer $ID = 1$ ise Adım 14 e, $ID = 2$ ise Adım 16 ya gidilir.

Adım 13: j uçuşundan başlanarak k güzergahı ilk uçuşa doğru taranır ve bulunan ilk (u,v) dolu uçuşu eğer güzergahın ilk uçuşu değilse $j = j + 1$ yapılır ve $ID = 1$ ise Adım 14 e, $ID = 2$ ise Adım 16 ya gidilir. (u,v) dolu uçuşu güzergahın ilk uçuşu ise $ID = 1$ durumunda Adım 15 e, $ID = 2$ durumunda Adım 18 e gidilir.

Adım 14: Görev süresi enküçükten enbüyüğe doğru bütün $i = k$ güzergahları taranarak ilk (u,v) boş uçuşu olan h güzergahı aranır. Eğer yoksa Adım 13 e gidilir. Eğer varsa, ancak $T_h + b + d \geq T_{enb}$ ise Adım 13 e gidilir. Aksi takdirde değişiklik yapılır. Değişiklik sonrası k güzergahı yine enbüyük güzergah uzunluğuna sahipse Adım 13 e gidilir. Aksi takdirde Adım 11 e gidilir.

Adım 15: $ID = 2$. Adım 12 ye gidilir.

Adım 16: Görev sırası enküçükten enbüyüğe doğru bütün $i \neq k$ güzergahlarından son noktası u olan ilk h güzergahı aranır. Bulunamadığı takdirde Adım 17 ye gidilir. Bulunduğu takdirde, $T_h + S_{uv} + b + d \geq T_{enb}$ ise Adım 17 ye gidilir. Aksi

takdirde değişiklik yapılır. Değişiklik sonrası k güzergahı yine enbüyük güzergah uzunluğuna sahipse Adım 13 e gidilir. Aksi takdirde Adım 11 e gidilir.

Adım 17: Görev süresi enküçükten enbüyüğe doğru bütün $i \neq k$ güzergahları taranarak (y_i, u) boş uçuşu ile birlikte (u, v) uçuşunun güzergah sonuna eklenmesi halinde $T_i + S_{y_i u} + (S_{uv} + b + d) < T_{enb}$ eşitsizliği sağlanamazsa Adım 13 e gidilir. Aksi takdirde eşitsizlik ilk sağlandığında değişiklik yapılır. Değişiklik sonrası k güzergahı yine enbüyük güzergah uzunluğuna sahipse Adım 13 e gidilir. Aksi takdirde Adım 11 e gidilir.

Boş Uçuş Enküçüklemesi

Adım 18: Her güzergaha önce geriden tarama, sonra önden tarama uygulanarak boş uçuş enküçüklemesi yapılır.

Kapasite Kısıtı Denetimi

Adım 19: Kapasite kısıtı denetimi yapılır. Kapasite kısıtlarına uygulamıyorsa, $T_{enb} \leftarrow T_{enb} + t$ yapıp yeniden kapasite denetimi yapılır.

Algoritma bir örnek probleme uygulanacaktır.

4.5. ÖRNEK PROBLEM

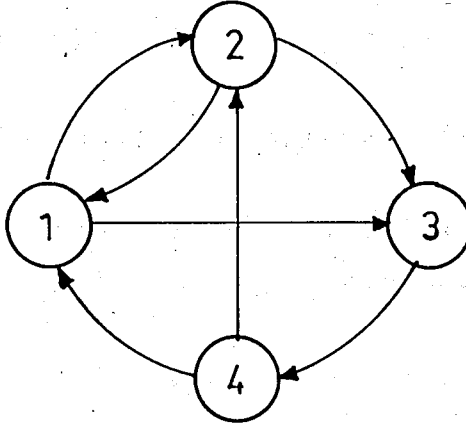
Alt bölüm 4.4. de verilen algoritma bir örnek probleme uygulanacaktır.

Problemde 4 havaalanı ve 3 uçak vardır. Bu uçaklar başlangıçta 3. ve 4. havaalanlarında bulunurlar; yani $U_3 = 1$, $U_4 = 2$. Her havaalanının kapasitesi ikidir. Yükleme süresi

$d = 15$, boşaltma süresi $b = 10$ alınmışlardır. Talep matrisi D ve süre matrisi S aşağıdaki gibi verilmişlerdir.

$$D = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} - & 1 & 1 & 0 \\ 1 & - & 1 & 0 \\ 0 & 0 & - & 1 \\ 1 & 1 & 0 & - \end{bmatrix} \end{matrix} \quad T = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} & \begin{bmatrix} - & 25 & 45 & 53 \\ 25 & - & 28 & 50 \\ 45 & 28 & - & 30 \\ 53 & 50 & 30 & - \end{bmatrix} \end{matrix}$$

Problemin serimi Şekil 4.11. de verilmiştir.



Şekil 4.11. Örnek Problem Serimi

Adım 1: Kentli postacı problemi çözümü için d_i^- ve d_i değerleri bulunup, d_i değerleri hesaplandığında şu neticeler elde edilir; $d_1 = d_2 = 0$, $d_3 = 1$, $d_4 = -1$. Buna göre 3. düğümün 1 birimlik arzı, 4. düğümün 1 birimlik talebi vardır. Boş uçuş (3,4) ayrıtı olarak bulunur. Böylece bütün düğümlere giren ve çıkan ayrıt sayısı aynı olduğundan boş ayrıt eklenmiş serimde Euler turu vardır. Bir Euler turu şöyle bulunur:

{3,4,2,1,3,4,1,2,3}

Adım 2: Bu ana tur 3. ve 4. düğümlerden başlayacak üç alttura bölünür. Bu altturlar şöyle bulunurlar:

	<u>Alttur</u>	<u>Uzunluk</u>
1.	4,2,1,3	195
2.	4,1,2,3	181
3.	3,4	55

Bu bölünme sırasında (3,4) boş ayrıtı elenmiştir.

Adım 3: Boş uçuş olmadığından boş uçuş enküçüklemesi uygulanmaz.

Adım 4: $k = 1$. $T_{enb} = T_1 = 195$. $IM = 1$.

Adım 5: $(u,v) = (2,3)$, $T_b = 50 + 70 = 120$. $j = 1$.

Adım 7: Boş uçuş olmadığından Adım 6 ya gidilir.

Adım 6: Bulunan blokun (4,1) ilk uçuşu güzergahın ilk uçuşu olduğundan ve $IM = 1$ olduğundan Adım 8 e gidilir.

Adım 8: $IM = 2$. Adım 5 e gidilir.

Adım 5: $(u,v) = (2,3)$, $T_b = 50 + 70 = 120$. $j = 1$.

Adım 9: 3. ve 2. güzergahlara ekleme yapılamaz. Adım 10 a gidilir.

Adım 10: 3. ve 2. güzergahlara sırası ile (3,2) ve (4,2) boş uçakları ile (2,3) bloku eklenemez. Adım 6 ya gidilir.

Adım 6: Bulunan blokun (4,1) ilk uçuşu güzergahın ilk uçuşu olduğundan ve $IM = 2$ olduğundan Adım 11 e gidilir.

Adım 11: $k = 1$. $T_{enb} = T_1 = 195$. $ID = 1$.

Adım 12: $(u,v) = (1,3)$, $j = 1$. Adım 14 e gidilir.

Adım 14: Boş uçuş olmadığından Adım 13 e gidilir.

Adım 13: $(u,v) = (2,1)$, $j = 2$. Adım 14 e gidilir.

Adım 14: Boş uçuş olmadığından Adım 13 e gidilir.

Adım 13: $(u,v) = (4,2)$ güzergahın ilk uçuşu ve $ID = 1$ olduğundan Adım 15 e gidilir.

Adım 15: $ID = 2$. Adım 12 ye gidilir.

Adım 12: $(u,v) = (1,3)$, $j = 1$. Adım 16 ya gidilir.

Adım 16: Son noktası 1 olan güzergah bulunamadığından Adım 17 ye gidilir.

Adım 17: $(u,v) = (1,3)$ bir boş uçuşla 3. güzergaha eklenir. $55 + 53 + 70 = 178 < 195$. Bu nedenle değişiklik yapılır. Enbüyük güzergah uzunluğu 2. güzergahındır. Adım 11 e gidilir.

Adım 11: $k = 2$. $T_{enb} = 181$. $ID = 1$.

Buradan sonra bir uçuş kullanarak dengeleme yordamı ile bir değişiklik daha gerçekleştirilir ve geliştirme yordamından şu çözümle çıkılır. (Burada ve bundan sonra $(i,*j)$ gösterimi (i,j) uçuşunun boş uçuş olduğunu gösterecektir.)

	<u>Güzergah</u>	<u>Uzunluk</u>
1.	4,2,1,2	175
2.	4,1,*2,3	156
3.	3,4,*1,3	178

Bu çözüme boş uçuş enküçüklemesi uygulanır. Ancak bir değişiklik olmaz. Bundan sonra çözüm kapasite denetiminden geçirilir.

Adım 19: Gerçek t değeri bir zaman birimi olmasına rağmen burada gösterimi kolaylaştırmak açısından beş zaman birimi alınmış ve gerekli yuvarlamalar yapılmıştır. Buna göre T_{enb} sırasında havaalanı doluluk pozisyonları şöyle gösterilirler:

1.	4	4	4	0	0	0	0	0	0	0	0	0	0	2	2	2	2	2	0	0
2.	4	4	4	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
3.	3	3	3	0	0	0	0	0	0	4	4	0	0	0	0	0	0	0	0	0
t =	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0
										10										20

1.	0	0	0	1	1	1	1	1	0	0	0	0	0	2	2					
2.	0	2	2	2	0	0	0	0	0	0	3	3								
3.	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	3	3			
t =	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6				
																	30			

Görüldüğü gibi bu çözüm olurlu bir çözümdür.

Aynı problem bir önceki bölümde verilmiş olan kesin çözüm yöntemi ile çözülmüş ve eniyi çözüm elde edilmiştir.

	<u>Güzergah</u>	<u>Uzunluk</u>
1.	4,2,1,2	175
2.	3,4,*2,3	158
3.	4,1,3	148

Sezgisel yöntemin türetmiş olduğu çözüm eniyi çözümden pek farklı değildir. B.Ü.Bilgisayar Merkezindeki UNIVAC 1106 da sezgisel yöntem 18.1 CPU saniye, kesin yöntem ise 11.7 CPU saniyede çözülmüşlerdir.

5. AYRITLARA HİZMET GÖTÜRÜLEN PROBLEMLERDE TAŞIT SAYISI VE KAPASİTELERİ İÇİN BİR SEZGİSEL ALGORİTMA

5.1. GİRİŞ

Ayrıtlara hizmet götürülen tür problemler noktalara hizmet götürülen problemlerle birlikte taşıt güzergahı belirleme problemlerinin iki temel problem türünü oluştururlar. Taşıt sayısı ve kapasiteleri problemleri klasik TGB problemlerinden iki noktada farklıdırlar:

i. Problemin amaç işlevi salt toplam tur uzunlukları olarak ifade edilen değişken maliyetlerden oluşmamakta; değişken maliyete taşıt sabit maliyetleri de eklenerek elde edilen bir toplam maliyet şeklinde ifade edilmektedir. Böylece taşıt sayısı toplam tur uzunluklarını enküçükleyen çözümün bir neticesi olarak oluşmamakta; bir karar değişkeni olmaktadır.

ii. Kullanılan taşıtların kapasiteleri birden fazla değer alabilmektedir. Böylece değişik taşıt kapasitelerinin özelliklerinden yararlanabilme olanağı yaratılabilmektedir.

Bu iki noktada yapılan değişikliklerle TGB problemi daha gerçekçi bir duruma gelmektedir. Problem bu iki noktanın dışında yeni bir kısıtla daha gerçekçi bir duruma getirilmeye çalışılmıştır. Buna göre, değişik kapasitedeki taşıtların sayıları üzerine bir üstsınır getirilmiştir. Bu problemi çözmek üzere bir dal-sınır yöntemi geliştirilmiştir.

Ayrıtlara hizmet götürülen problemlerde sabit maliyetleri de gözönüne alan bir çalışmaya açık literatürde rastlanmamıştır. Burada önerilen sezgisel yaklaşım böyle bir boşluğu doldurmak ihtiyacından doğmuştur.

5.2. PROBLEMİN TANIMI

Ayrıtlara hizmet götürülen sistemlerde taşıt sayısı ve kapasiteleri problemi sözlü olarak şöyle tanımlanabilir. Elimizde değişik kapasitelere sahip taşıtlar olsun. Verilmiş bir serim üzerinde hizmet talep eden ve taşıt kapasitelerini kullanmak isteyen bir ayrıt altkümesi olsun. Amaç, bu hizmet talebini sabit ve değişken maliyetlerin toplamını enküçük düzeyde tutarak karşılayacak taşıt sayısı, kapasiteleri ve güzergahlarını saptamaktır. Değişik kapasitedeki taşıt sayıları üstünde bir kısıt yoktur.

5.3. ANA HATLARI İLE ÇÖZÜM YÖNTEMİ

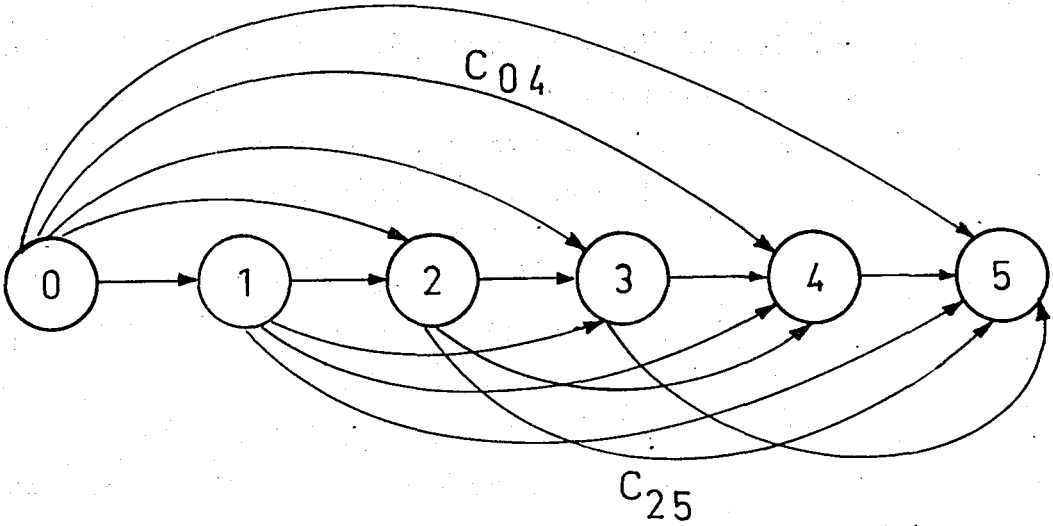
Taşıt sayısı ve kapasiteleri probleminin burada çözümü için önerilen yöntemde önce bütün geçilmesi gerekli ayrıtları kapsayan bir ana tur bulunur. Daha sonra bu ana tur altturlara bölünür. Bu altturların sayısı taşıt sayısını; her altturun toplam hizmet talebi ise o alttura atanan taşıtın kapasitesini belirler. Buna göre önerilen yaklaşım iki aşamadan oluşmaktadır:

- i. Ana turun bulunması.
- ii. Ana turun altturlara bölünmesi.

Ana tur tanım gereği bütün geçilmesi gerekli ayrıtları kapsayan bir turdur. Buna göre ana tur problemi, sonsuz kapasiteli tek bir taşıtın geçilmesi gerekli ayrıtları kapsayan enküçük maliyetli turudur. Taşıt tek, kapasitesi ise sonsuz olduğundan, ana tur probleminin amaç işlevi değişken maliyetlerdir. Burada değişken maliyetler doğrudan katledilen mesafeye bağlı maliyetlerdir. O halde, ana tur problemi, geçilmesi gerekli ayrıtları kapsayan ve enküçük uzunluğa sahip güzergahın bulunması problemidir. Bu problem daha önce tanımlı

verilmiş olan kentli postacı porbelmine karşı gelmektedir. Bu durumda, önerilen çözüm yönteminin birinci aşaması bir kentli pastacı problemine indirgenmektedir. Burada serimin yapısına göre, (i) yönlü, (ii) yönsüz kentli postacı problemleri gözönüne alınacaklardır.

Yaklaşımın ikinci aşaması ana turun altturlara bölünmesidir. Bu işlem taşıt sayısını ve taşıt kapasitelerini belirleyeceğinden ikinci aşamada sabit maliyetler de gözönüne alınırlar. İkinci aşamayı daha iyi açıklamak için makine yenileme problemini anımsayalım. Bu problemde, bilinen bir süre içinde, yıllara göre satın alma maliyeti, yaşına göre hurda geliri, yaşına göre işletme ve bakım masrafları bilinen bir makine için en küçük maliyetli yenileme politikası aranır. İşletme ve sabit maliyetleri içeren bu probleme en kısa yol algoritması uygulanır (Whitehouse, 1973). Küçük bir örnek Şekil 5.1. de verilmiştir. Düğüm numaraları yıl sonlarını, ayrıt maliyetleri ise başlangıç ve bitiş düğümleri arasında kalan



Şekil 5.1. Yeni Serim Örneği

süre içindeki toplam net maliyeti vermektedir. Örneğin, (2,4) ayrıtının maliyeti 2. yıl sonunda yeni makine alınması ve bu-

nun iki yıl kullanılarak 4. yıl sonunda satılmasının maliyetidir. Bu serim üzerinde en kısa yol algoritmasının uygulanması eniyi yenileme politikasını verir. Örneğin, en kısa yol $\{0,2,5\}$ ise, başlangıçta alınan makinayı iki yıl kullanmak, 2. yıl sonunda satıp yenisini almak ve bunun üç yıl kullandıktan sonra 5. yıl sonunda satmaktır.

Sabit ve değişken maliyetleri içeren ve belirli bir "uzunluğu" eniyi biçimde bölmeye yönelik bu en kısa yol uygulamasının elimizdeki probleme uygulanabilmesinin olurluluğu araştırılmış ve en kısa yol algoritmasının uygulanacağı serimin ve ayırt maliyetlerinin uygun bir şekilde tanımlanması halinde bunun mümkün olduğu görülmüştür.

İkinci aşamada önce en kısa yol algoritmasının uygulanacağı serim ayırt maliyetleri ile birlikte elde edilecektir. Daha sonra en kısa yol algoritması uygulanacaktır. En kısa yol algoritmasının uygulandığı serime yeni serim; problemin üzerinde tanımlandığı serime ise esas serim diyeceğiz. En kısa yol çözümünde yer alan ayırtlar kullanılarak esas serim üzerinde karşı geldikleri altturlar bulunacaktır. En kısa yol çözümündeki ayırt sayısı gerekli taşıt sayısını, her altturla geçilmesi gerekli ayırtlarının talepleri ise bu taşıtların kapasitelerini belirleyecektir.

5.4. AYRINTILARI İLE ÇÖZÜM YÖNTEMİ

Çözüm yöntemi bu altbölümde üç kısımda verilecektir. Birinci kısımda anaturu oluşturan kentli postacı turunun bulunması için yönlü ve yönsüz ayırtlarda çözüm yöntemleri izah edilecektir. İkinci kısımda bu ana tur kullanılarak yeni serimin ne gibi işlemlerle elde edileceği anlatılacaktır. Üçüncü kısımda en kısa yol algoritmasının uygulanması ve sonuçlarının değerlendirilmesi izah edilecektir.

5.4.1. Kentli Postacı Turunun Elde Edilmesi

Yönlü serimlerde kentli postacı turunun nasıl elde edileceği daha önce altbölüm 4.3.1. de ayrıntıları ile verilmişti. Bu nedenle tekrarlanmayacaktır. Burada yönsüz serimlerde kentli postacı turunun elde edilmesi için bir yöntem anlatılacaktır (Orloff, 1974a).

S serimi üzerinde geçilmesi zorunlu ayrıtlar A kümesini, diğer ayrıtlar ise $\bar{A}$ kümesini oluştursunlar. Sadece $\bar{A}$ ayrıtları-ın çakıştığı düğümler $\bar{V}$ kümesini, diğer düğümler ise V kümesini oluştursunlar.

V kümesinin elemanları iki altkümeye ayrılırlar. Bir düğüm kendisi ile çakışan ayrıtların sayısının tek veya çift olmasından bağımlı olarak tek kerteli (odd degree) ve çift kerteli (even degree) olarak nitelendirilirler. Tek kerteli düğümler bir altküme, V_T , çift kerteli düğümler ise diğer bir altkümeyi, $V_{\bar{T}}$, oluştururlar.

Bütün düğümler çift kerteli ise serimde bir Euler turu vardır ve problemin eniyi çözümü budur.

Ancak iki veya daha fazla tek kerteli düğüm varsa o takdirde mevcut ayrıtları sadece bir kere kullanarak bir Euler turu yaratılamaz. Bu durumda kentli postacı turu için bazı ayrıtların birden fazla geçilmesine ve gerekirse $\bar{A}$ daki bazı ayrıtların kullanılmasına gerek vardır.

Bunun enküçük maliyetle gerçekleştirilmesi için şu yöntem uygulanır. Tek kerteli düğümler altkümesi V_T ve $\bar{V}$ kümesinin düğüm kümesini, problemde başlangıç ve bitiş düğümlerinin ikisinin de bu düğüm kümesinde olduğu ayrıtların ayrıt kümesini oluşturduğu bir S_1 serimi üzerinde V_T nin bütün düğümleri arasında en kısa yol algoritması uygulanır. Bu amaçla

burada Floyd algoritması (Floyd, 1962) kullanılmıştır.

V_T düğüm kümesi olan ve bütün düğümleri birbirine bağlı ve ayrıt maliyetleri Floyd algoritmasının çıktısı olan yeni bir S_2 serimi oluşturalım. Bu serim üzerinde 1-eşleme (1-matching) problemi çözülür (Edmonds v.d., 1973). Bu problemin neticesinde enküçük maliyetle hangi düğümlerin hangileri ile birleştirileceği bulunur. Bu birleştirme tek bir ayrıtla olabileceği gibi iki düğüm arasında bir yola da karşı gelebilir. Esas serim üzerinde gerekirse $\bar{A}$ ve $\bar{V}$ elemanları da kullanılarak V_T düğümleri birleştirilir ve S_3 serimi elde edilir.

S_3 seriminde bütün düğümler çift kertelidir ve bu serim üzerinde bir kentli postacı turu vardır. Böylece yönsüz kentli postacı turu çözülmüş olur.

5.4.2. Yeni Serimin Elde Edilmesi

Kentli postacı turu girdi olarak yeni serimin bulunmasında kullanılır. Serimi tanımlamak için düğüm kümesi, ayrıt kümesi, çakışım ilişkileri ve ayrıt maliyetlerinin belirlenmesi gerekmektedir.

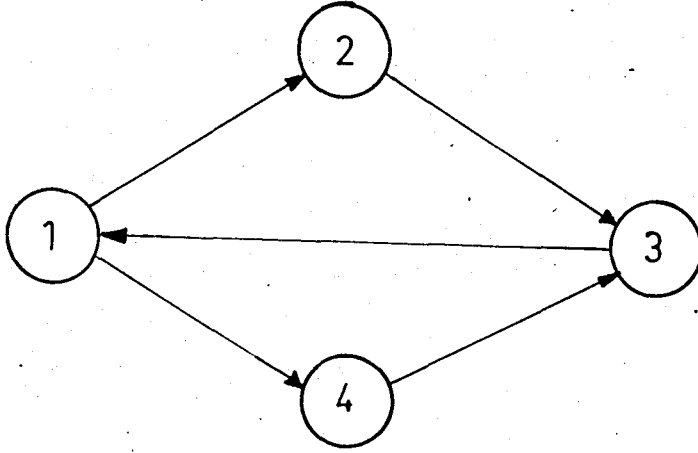
Düğüm kümesinde sıfır numaralı yapay düğüm ve kentli postacı turunun ayrıtlarına karşı gelen düğümler yer alır. Kentli postacı turunun her ayrıtı için bir düğüm yaratılır. Buna göre düğüm kümesinde kentli postacı turundaki ayrıt sayısından bir fazla düğüm vardır. Tur başlangıç ve bitiş düğümüne merkez düğümü diyelim. Merkez düğümden başlayarak ayrıtları geçildikleri sıraya göre numaralandıralım; $\{1, 2, \dots, m\}$. Her ayrıtın karşı geldiği düğüme de aynı numarayı verelim.

Yeni serimin ayrıt kümesini belirlemeden önce ayrıtla-

rı tanımlayalım. Ayritlar ancak küçük numaralı düğümlerden kaynaklanıp büyük numaralı düğümlere girebilirler ve her bir ayrit bir altturu temsil eder. Örneğin, $(i, i+3)$ ayritını alalım ve $(i+1)$ ve $(i+3)$ düğümlüne karşı gelen ayritların boş ayrit olmadıklarını varsayalım. $(i, i+3)$ ayritı şu altturu temsil eder: $\{0, i+1, i+2, i+3, 0\}$. Bu tanımlamaya göre ayritın kaynaklandığı nokta bu ayritın temsil ettiği turda yer almamaktadır. Ayrit tanımından hareketle ayrit kümesinin kentli postacı turunun tüm altturlarını temsil eden ayritlardan oluştuğunu söyleyebiliriz. Ancak esas problemin amacı gözönüne alındığında bazı altturlar ayrit kümesine alınmadan elenebilirler. Bunlar boş ayritlarla ilişkili altturlardır ve çakışım ilişkilerini belirlerler.

Genelde ayritlar sadece küçük numaralı düğümlerden büyük numaralı düğümlere giderler; (i, j) , $j > i$. Ancak j düğümlünün boş ayrita karşı geldiğini düşünelim. Bu durumda (i, j) ayritının temsil ettiği altturun sondan bir evvelki ayritı boştur ve bu ayritın girdiği düğümden yapay düğüme giden ve altturu tamamlayan ayrit da doğal olarak boştur. Ana turun bulunması sırasında kentli postacı turunun tamamlanmasında bir işlevi olan j boş ayritının bu altturda hiçbir işlevi yoktur ve boş yere altturu uzatmaktadır. Bu gözlemden hareketle boş ayritlara karşı gelen düğümlere yeni serimde ayrit girmemesi kuralı oluşturulur. (i, j) , $j > i+1$, ayritında $(i+1)$ düğümlü boş ayrit düğümlü ise o takdirde (i, j) ayritının temsil ettiği altturun ikinci ayritı boş bir ayrit olacaktır ve bu alttur içinde bir gerekliliği olmayacaktır. Bu nedenle elenmesi gerekir. Buna göre, eğer $(i+1)$ düğümlü boş ayrit düğümlü ise, i düğümlünden ayrit çıkmaz kuralı oluşturulur. Ancak bu kurallar böylece uygulandığında yeni serim bağlı olmayabilir; yani merkez düğümlünden her düğüme bir yol olmayabilir. Bunu önlemek için, eğer $(i+1)$ boş ayrit düğümlü ise $(i, i+1)$ ayritı serime eklenir. En kısa yol algoritması çözümünü etkilememesi için bu ayritlara sıfır maliyet atanır.

Sadece geçilmesi zorunlu ayrıtları Şekil 5.2. de gösterilmiş olan esas serimden elde edilmiş olan yeni serim Şekil 5.1. de sunulmuştur. Bütün düğümleri birbirine çift yönlü bağlı kabul edilen esas serim üzerinde bir kentli pastacı turu olarak $\{(1,2), (2,3), (3,1), (1,4), (4,3), (3,1)\}$ bulunmuş olsun. Bu tur üstünde (3,1) ayrıtı boş ayrıttır. Yeni serim oluşturulurken (3,1) ayrıtına karşı gelen 6. düğüm kullanılmamıştır. Bunun nedeni 6. düğümün bir boş düğüm olmasıdır ve daha önce belirlenen bir kurala göre bu düğüme bir



Şekil 5.2. Esas Serim Örneği

önceki düğümün dışında hiçbir düğümden bir ayrıt gelemez. Buna göre esasta yeni serim 5. düğüm ile bitmektedir ve bu nedenle 6. düğüm yeni serime eklenmemiştir.

Ayrıt kümesi ve çakışım ilişkileri böylece oluşturulduktan sonra ayrıt maliyetleri bulunur.

Yeni serimin (i,j) ayrıtının maliyeti C_{ij} sabit ve değişken maliyet bileşenlerinden oluşmaktadır. Sabit maliyeti F işlevi ile gösterelim. (i,j) ayrıtının sabit maliyeti bu ayrıtın karşı geldiği alttur üstündeki toplam talebi karşılanacak enküçük kapasitedeki taşıtın sabit maliyetidir. k düğümüne karşı gelen ayrıtın talebini d_k ile gösterelim. Boş ayrıt

düğümünün d değeri sıfır alınmıştır. Buna göre, (i,j) ayrıtının sabit maliyeti

$$F_{ij} = F\left(\sum_{k=i+1}^j d_k\right) \quad (5.1)$$

olarak bulunur.

Örneğin, Şekil 5.2. de (2,5) ayrıtının sabit maliyetini bulalım. (2,5) ayrıtının altturuna dahil olan 3, 4 ve 5 numaralı düğümlere karşı gelen ayrıtların talepleri 20, 30 ve 15 olsun ve elimizdeki taşıt tiplerinin kapasiteleri 30, 50 ve 80 olsun. Bu durumda (2,5) ayrıtının sabit maliyeti 80 kapasiteli taşıtın sabit maliyeti olacaktır.

Tur üstündeki n ayrıtının değişken maliyetini c_r ile gösterelim. i düğüme karşı gelen ayrıt (k,h) olsun. Yeni serim üzerinde c_{oi} ve c_{io} değişken maliyetleri sırası ile esas serimin (1,k) ve (h,1) ayrıtlarının maliyetlerine karşı gelirler. Buna göre, (i,j) ayrıtının toplam maliyeti şöyle hesaplanır:

$$C_{ij} = c_{o,i+1} + \sum_{k=i+1}^j c_k + c_{jo} + F\left(\sum_{k=i+1}^j d_k\right) \quad (5.2)$$

Şayet sabit maliyet bileşeninde toplam talep enbüyük taşıt kapasitesini aşarsa o takdirde C_{ij} sonsuza eşitlenir. Yine aynı şekilde, taşıtlar üzerinde bir tur uzunluğu üstsınıırı varsa ve bir altturun uzunluğu bu sınırı aşarsa bu altturun maliyeti sonsuza eşitlenir. Maliyetleri sonsuz olan ayrıtlar serim üzerinde gösterilmezler.

Altturnun yapısında gerçekleştirilecek değişikliklerle C_{ij} maliyetinin gereksiz yere yüksek çıkması önlenabilir. Buna yönelik değişiklik durumları aşağıdaki gibi sıralanabilir:

5.4.2.1. (i,j) altturunun $(i+1)$ düğümüne karşı gelen ayrıt merkez düğümünden kaynaklanıyorsa $c_{o,i+1}$ sıfır olarak alınır.

5.4.2.2. j düğümüne karşı gelen ayrıt merkez düğümde bitiyorsa c_{oj} sıfır olarak alınır.

5.4.2.3. $(i+1)$ düğümüne karşı gelen ayrıt merkez düğümünde bitiyorsa ve j düğümüne karşı gelen ayrıt bu ayrıtın başlangıç düğümünde bitiyorsa, bu alttur içinde $(i+1)$ düğümü j düğümünün arkasına eklenir ve C_{ij} hesaplanırken $c_{o,i+1}$ ve c_{jo} sıfır alınır.

5.4.2.4. j düğümüne karşı gelen ayrıt merkez düğümünden kaynaklanıyorsa ve $(i+1)$ düğümüne karşı gelen ayrıt bu ayrıtın bitiş düğümünden başlıyorsa, bu alttur içinde j düğümü $(i+1)$ düğümünün önüne eklenir ve C_{ij} hesaplanırken $c_{o,i+1}$ ve c_{jo} sıfır alınır.

Bu altbölümü bitirmeden önce yeni serimi oluşturmak için gerekli işlemleri aşağıdaki gibi özetleyelim.

Adım 1: Kentli postacı turunda merkez noktasından başlanarak ayrıtlar $\{1,2,\dots,m\}$ olarak numaralandırılırlar. Bunlar yeni serimin düğüm numaralarıdır. Bu düğümlere yapay düğüm olan sıfır düğümü de eklenir. Eğer m düğümü boş ayrıt düğümü ise yeni serime eklenmez.

Adım 2: Eğer $(i+1)$ boş ayrıt düğümü değilse veya $i=m$ değilse, i düğümü boş ayrıt düğümleri dışındaki tüm düğümlere bağlanır. Bu işlem bütün $i \neq m$ düğümleri için tekrarlanır.

Adım 3: Kapasite kısıtını ihlal eden altturları temsil eden ayrıtlar yeni serimden elenir.

Adım 4: Tur uzunluğu kısıtı varsa, 5.4.2.1., 5.4.2.2., 5.4.2.3. ve 5.4.2.4. de gözönüne alınarak alttur toplam uzunlukları bulunur ve bu kısıtı ihlal eden altturları temsil eden ayrıtlar yeni serimden elenir.

Adım 5: Kalan ayrıtların maliyetleri (5.2) kullanılarak ve 5.4.2.1., 5.4.2.2., 5.4.2.3. ve 5.4.2.4. gözönüne alınarak hesaplanır.

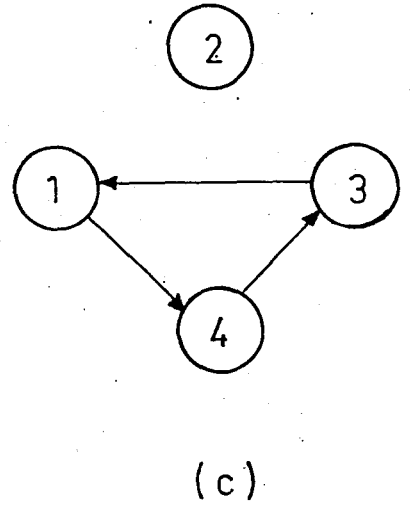
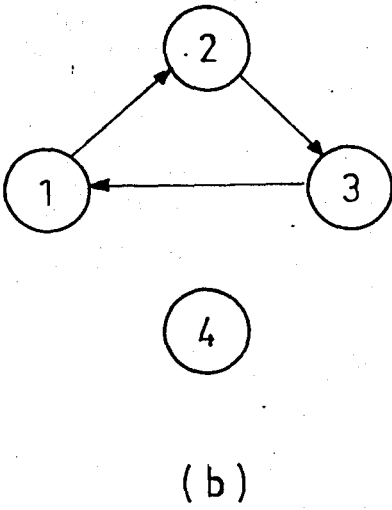
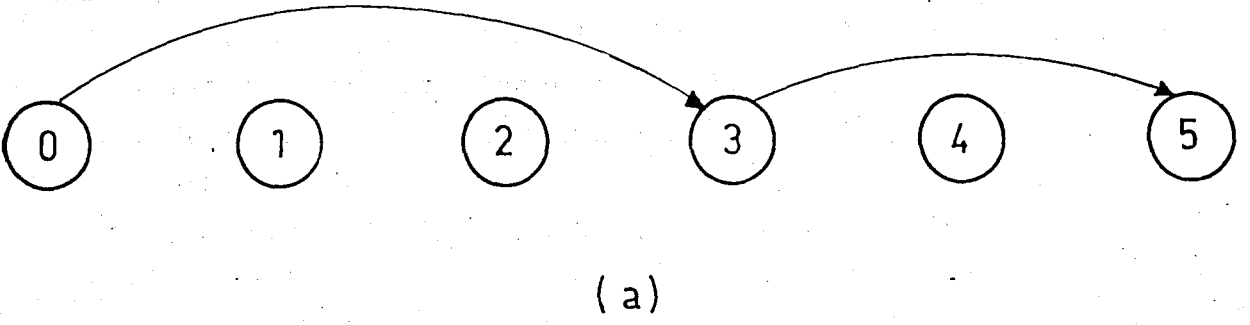
Adım 6: Eğer $(i+1)$ boş ayrıt düğümü ise, $(i, i+1)$ ayrıtı yeni serime eklenir ve sıfır maliyet atanır.

Bu işlemler bittikten sonra elde edilen yeni serimde en kısa yol algoritması uygulanacaktır.

5.4.3. En Kısa Yol Algoritması

En kısa yol algoritması yeni serim üzerinde sıfır noktası ile kentli postacı turunun son ayrıtına karşı gelen m düğümü arasındaki en kısa yolu bulmak üzere uygulanır. Uygulanacak en kısa yol algoritmasını belirlemekte şu iki gözlem önemli olacaktır: (i) yapılan dönüşüm neticesi elde edilen yeni serim çevrimsizdir (acyclic), (ii) yeni serimin tüm ayrıt maliyetleri sıfırdan büyüktür. Bu iki gözlemin belirlediği durumda en verimli algoritmalardan biri Dijkstra algoritmasıdır (Dreyfus, 1969) ve burada bu algoritma uygulanacaktır. Dijkstra algoritması oldukça yaygın bilinen bir algoritma olduğundan burada tekrarlanmayacaktır (Dijkstra, 1959).

Şekil 5.2. deki esas serimden elde edilen yeniserim (Şekil 5.1) en kısa yol algoritması uygulanmış ve Şekil 5.3a da gösterilen çözüm elde edilmiş olsun; yani $(0,3)$ ve $(3,5)$ ayrıtları en kısa yolu oluştursunlar. Bu en kısa yolun esas serim üzerinde karşı geldiği altturlar Şekil 5.3b ve c de gösterilmişlerdir. Buna göre iki taşıt kullanılacaktır. Çözüm



Şekil 5.3. En Kısa Yolun Altturlara Dönüştürülmesi

neticesi ikinci altturdaki (3,1) ayrıtı boş ayrıt olsun. Bu durumda birinci altturun taşıtının kapasitesini (1,2), (2,3) ve (3,1) ayrıtlarının toplam talebi; ikinci taşıtın kapasitesini ise (1,4) ve (4,3) ayrıtlarının toplam talebi belirler.

Bu altbölümde ayrıtları verilen çözüm yöntemini toplu halde sunmak amacı ile yöntem bir algoritma halinde ifade edilmiştir.

5.4.4. Algoritma

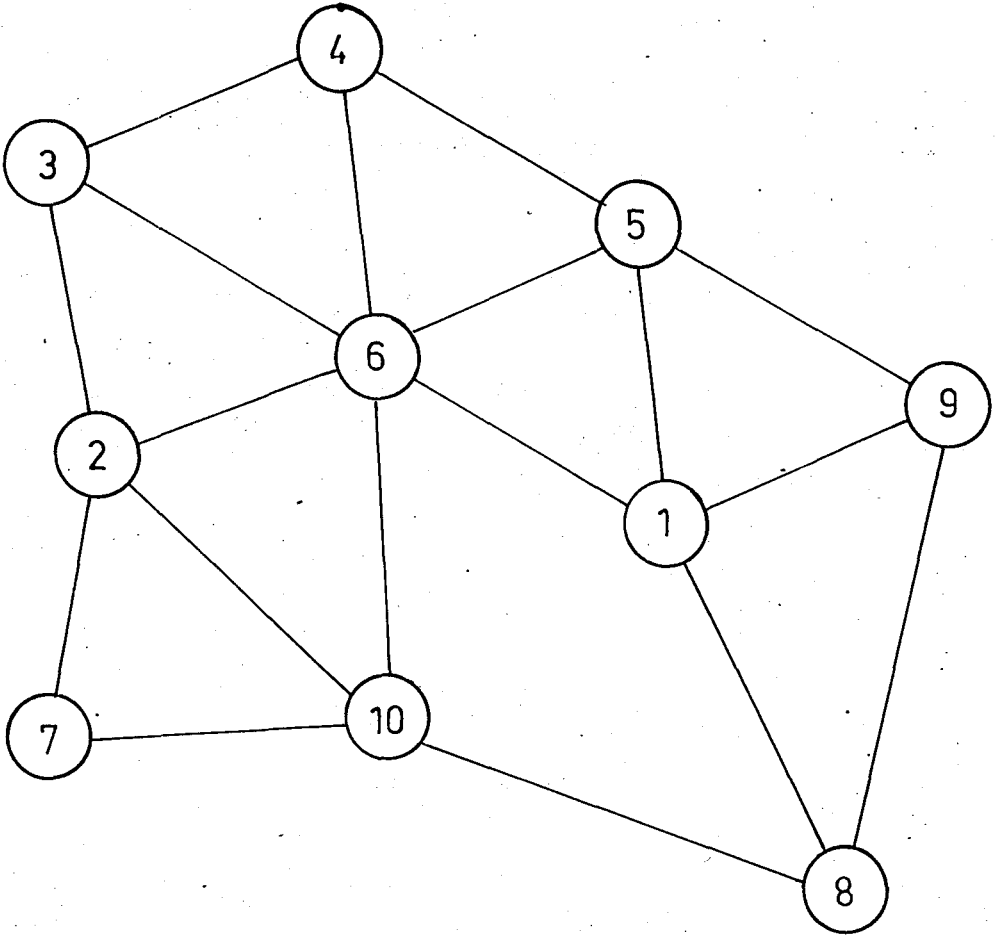
Adım 1: Verilmiş olan serim üzerinde kentli postacı problemi çözülerek ana tur elde edilir.

Adım 2: Ana tur kullanılarak yapılan bir dönüşümle yeni serim hazırlanır. Bunun için altbölüm 5.4.2. de adımlar halinde verilen işlemler uygulanır.

Adım 3: Yeni serim üzerinde en kısa yol algoritması uygulanır. Elde edilen çözüm esas serim üzerinde değerlendirilerek altturlar elde edilir. Her alttura atanacak olan taşıtın kapasitesi o altturun toplam talebi ile belirlenir.

5.5. ÖRNEK PROBLEMLER

Problemin serimi Şekil 5.4. de verilmiştir. Ayrıtlar yönsüzdür, yani istenilen yönde katedilerek talep edilen hizmet verilebilir.



Şekil 5.4. Örnek Problemlerin Serimi

Problemin talep ve uzunluk matrisleri problemin simetrik olmasından yararlanılarak beraberce verilmiştir. D/U matrisinin diyagonal üstü uzunluk matrisini temsil eder.

	1	2	3	4	5	6	7	8	9	10
1	-	∞	∞	∞	26	23	∞	25	32	∞
2	-	-	31	∞	∞	38	33	∞	∞	40
3	-	12	-	42	∞	53	∞	∞	∞	∞
4	-	-	42	-	30	40	∞	∞	∞	∞
D/U = 5	46	-	-	13	-	27	∞	∞	28	∞
6	38	17	19	3	14	-	∞	∞	∞	27
7	-	23	-	-	-	-	-	∞	∞	45
8	15	-	-	-	-	-	-	-	45	41
9	65	-	-	-	36	-	-	41	-	∞
10	-	6	-	-	-	16	19	26	-	-

Problemde 3 tür taşıt vardır ve bunların kapasiteleri 50, 80 ve 100 dür. Bir alttturun toplam maliyeti şöyle hesaplanır:

$$\text{Toplam Maliyet} = \text{Toplam Uzunluk} + F(\text{Toplam Talep})(5.3)$$

İlk adım olarak kentli postacı problemi çözülür. Tek kerteli düğümler 3, 4, 8, 9 dur. Bu düğümler arasında Floyd algoritması uygulanır ve şu neticeler elde edilir.

	3	4	8	9
3	-	42	101	100
4	42	-	81	58
8	101	81	-	45
9	100	58	45	-

Bu matris kullanılarak yapılan eşleme neticesinde (3,4) ve (8,9) boş ayrıtları serime eklenerek serim Euler turu elde

edilmeye uygun duruma getirilir. Bulunan bir Euler turu şöyledir:

{1,5,6,4,3,4,5,9,8,9,1,8,10,7,2,3,6,2,10,6,1}

Yeni serimin düğümleri ve Euler turunun ayrıtları aşağıdaki gibi eşlendirilmişlerdir:

<u>Düğüm</u>	<u>Ayrıt</u>	<u>Düğüm</u>	<u>Ayrıt</u>	<u>Düğüm</u>	<u>Ayrıt</u>
0	-	7	(5,9)	14	(7,2)
1	(1,5)	9	(9,8)	15	(2,3)
2	(5,6)	9	(8,9)	16	(3,6)
3	(6,4)	10	(9,1)	17	(6,2)
4	(4,3)	11	(1,8)	18	(2,10)
5	(3,4)	12	(8,10)	19	(10,6)
6	(4,5)	13	(10,7)	20	(6,1)

Yeni serim üzerindeki maliyetlere örnek olarak birisini hesaplayalım:

$$C_{36} = c_{04} + c_4 + c_5 + c_6 + c_{60} + F(d_4 + d_5 + d_6)$$

$$C_{36} = 56 + 42 + 42 + 30 + 26 + F(42 + 0 + 13)$$

$$C_{36} = 196 + 80 = 276$$

Bu şekilde tüm olurlu maliyetleri hesaplanan yeni serime uygulanan en kısa yol algoritması şu eniyi yolu verir:

{0,2,7,8,9,10,15,20}

Bu en kısa yolun karşı geldiği altturlar ve taşıt kapasite ve kullanımları şöyledir:

1.Alttur. {1,5,6,1}

Taşıt kapasitesi= 80
Taşıt kullanımı = 60

2.Alttur. {1,6,4,3,4,5,9,1}

Taşıt kapasitesi=100
Taşıt kullanımı = 94

3.Alttur. {1,9,8,1}

Taşıt kapasitesi= 50
Taşıt kullanımı = 41

4.Alttur. {1,9,1}

Taşıt kapasitesi= 80
Taşıt kullanımı = 65

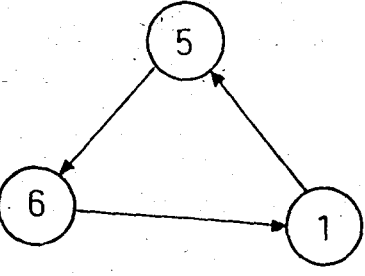
5.Alttur. {1,8,10,7,2,3,1}

Taşıt kapasitesi=100
Taşıt kullanımı = 95

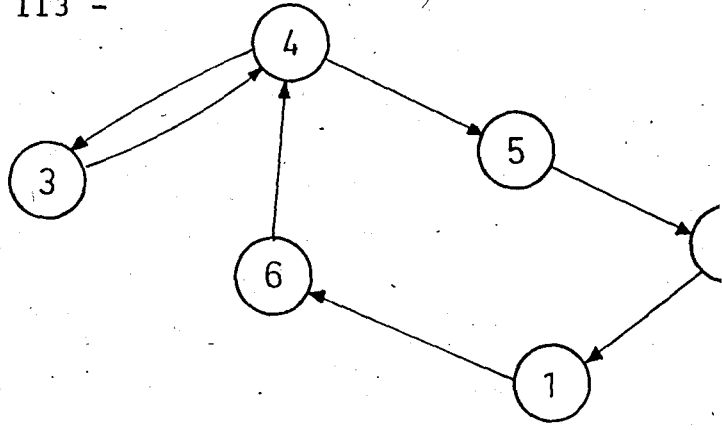
6.Alttur. {1,6,3,6,2,10,6,1}

Taşıt kapasitesi=100
Taşıt kullanımı = 96

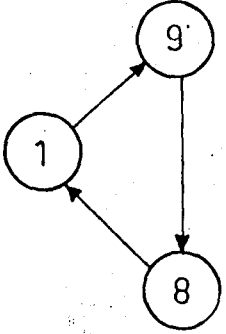
{1,8,9,1} altturu sadece boş ayırıt (8,9) u kapsadığı için altturlar arasına alınmamıştır. Birinci örnek problemin çözümü Şekil 5.5. de verilmiştir.



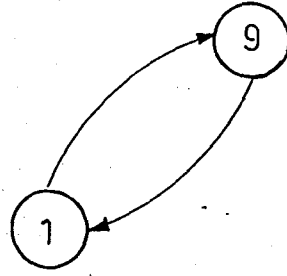
1. Altıtur (K=80)



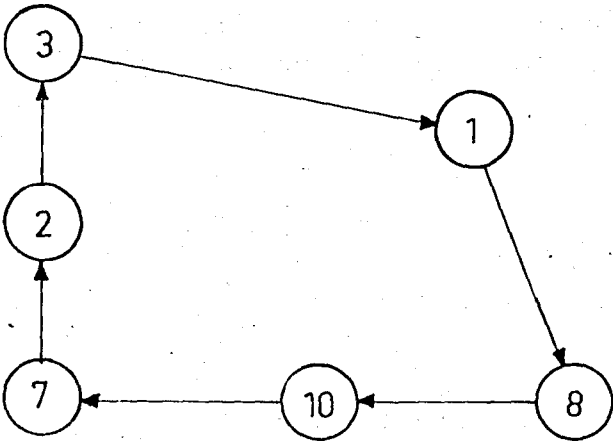
2. Altıtur (K=100)



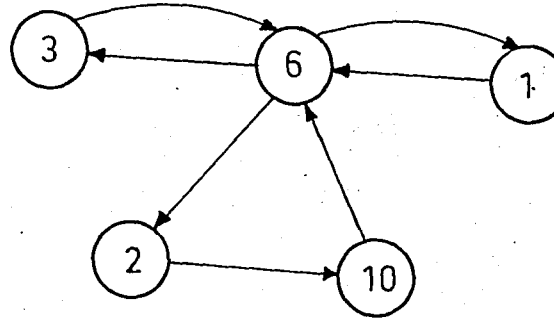
3. Altıtur (K=50)



4. Altıtur (K=80)



5. Altıtur (K=100)



6. Altıtur (K=100)

Çözümün toplam maliyeti 1126 dir. Hizmet talep eden 18 ayrıtı kapsamak için 6 taşıt ile 28 ayrıt kullanılmıştır. Ana turda ise tek taşıt ile 20 ayrıt kullanılmıştı.

Aynı problem değişik bir Euler turu ile de çözülmüştür. Yeni serimin düğümleri ve Euler turunun ayrıtları aşağıdaki gibi eşlendirilmiştir:

<u>Düğüm</u>	<u>Ayrıt</u>	<u>Düğüm</u>	<u>Ayrıt</u>	<u>Düğüm</u>	<u>Ayrıt</u>
0	-	7	(6,2)	14	(4,3)
1	(1,8)	8	(2,10)	15	(3,4)
2	(8,10)	9	(10,6)	16	(4,5)
3	(10,7)	10	(6,1)	17	(5,9)
4	(7,2)	11	(1,5)	18	(9,8)
5	(2,3)	12	(5,6)	19	(8,9)
6	(3,6)	13	(6,4)	20	(9,1)

Bu yeni serimde çözülen en kısa yol algoritmasından elde edilen eniyi yol:

{0,5,9,11,16,18,19,20} dir.

Bu en kısa yola karşı gelen altturlar ve taşıt kapasite ve kullanımları şöyledir:

1.Alttur. {1,8,10,7,2,3,1}	Taşıt kapasitesi=100 Taşıt kullanımı = 95
2.Alttur. {1,6,3,6,2,10,6,1}	Taşıt kapasitesi= 80 Taşıt kullanımı = 58
3.Alttur. {1,6,1,5,1}	Taşıt kapasitesi=100 Taşıt kullanımı = 84
4.Alttur. {1,5,6,4,3,4,5,1}	Taşıt kapasitesi= 80 Taşıt kullanımı = 72

5. Alttur. {1,5,9,8,1}

Taşıt kapasitesi==80
Taşıt kullanımı ==77

6. Alttur. {1,9,1}

Taşıt kapasitesi= 80
Taşıt kullanımı = 65

{1,8,9,1} altturu sadece boş ayırıt (8,9) u kapsadığı için altturlar arasına alınmamıştır.

Çözümün toplam maliyeti 1137 dir. Bir önceki çözümden amaç işlevi çok az farklıdır, ancak bazı altturlar değişik olarak elde edilmiştir. 6 taşıt ile 30 ayırıt kullanılmıştır. İlginç bir gözlem {1,9,1} altturunun yerleşmiş bir alttur olarak belirmesidir. Bunun nedeni (9,1) ayırıtının talebinin 65 olması ve bitişik olduğu (8,9) veya (5,9) ayırıtlarından biri ile birleştirildiğinde kapasite talebinin 100 ü aşmasıdır. Bu tür gözlemler model çözümünden uygulanacak plana geçişte yararlı olurlar.

Bu çözümlerin bilgisayarda CPU kullanımları ayrıca ölçülmemiş, bir sonraki altbölümde verilecek dal-sınır yönteminin başlangıç çözümü olarak elde edilmişlerdir.

5.6. KISITLI TAŞIT SAYISI İÇİN BİR ÇÖZÜM YÖNTEMİ

Problemin tanımında değişik tipteki taşıt sayıları için bir üstsınır getirilmemişti. Bu altbölümde böyle bir üstsınır getirildiği durum için bir çözüm yöntemi önerilecektir.

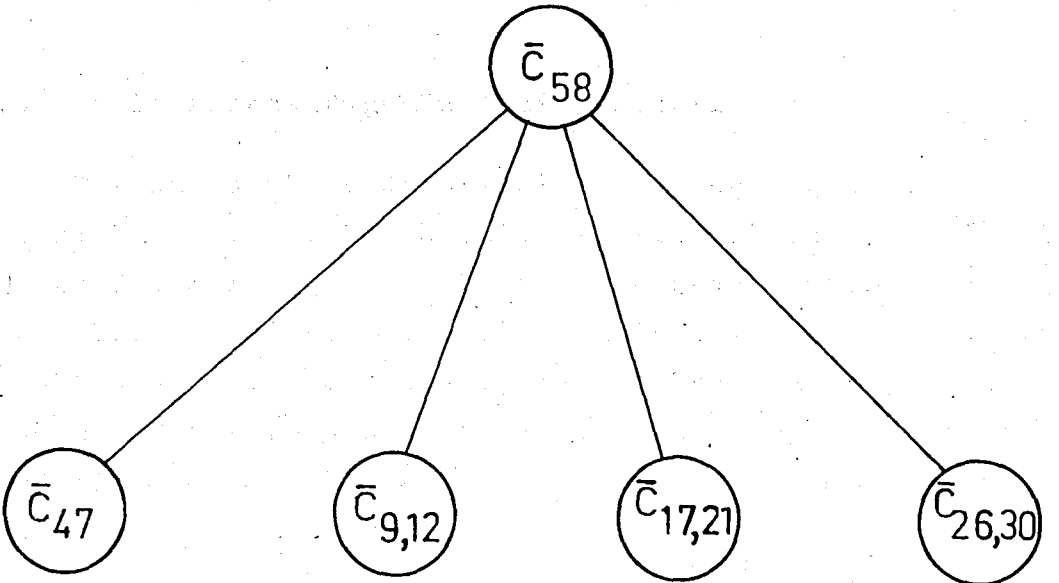
Önerilen çözüm yöntemi bir dal sınır yöntemidir. Dal-sınır yöntemi örtük birerlemeyi (implicit enumeration) yeni serim üzerinde uygulayacaktır. Yani kentli postacı turundan türetilebilecek olurlu altturlarla sınırlıdır. Bu olurlu altturlar dal-sınır yöntemi için tüm olurlu çözümler uzayını

teşkil ederler. Dal-sınır yöntemi bu olurlu çözümler uzayının eniyi çözümünü bulmayı garanti eder. Ancak bu eniyi çözümün tüm problemin eniyi çözümü olduğu garanti edilemez.

Bir dal-sınır yöntemini belirleyen yöntemin dallanma ve sınırlanma kurallarıdır. Buradaki yönteme ilişkin kurallar aşağıda verilmiştir.

5.6.1. Dallanma Kuralı

Dal-sınır ağacının h düğümünde olalım ve bu düğümde problemin çözümü olursuz bir çözüm olsun. Burada taşıt üstsınırını aşan tüm taşıt türleri arasında enküçük taşıt sayısı olan tur seçilir. Bu sayıya k dersek, h düğümünden k dal çıkacaktır, yani k yeni düğüm oluşturulacaktır demektir. Olursuz çözümde k taşıtın altturuna k ayırıt karşı gelmektedir. Her i dalının düğümünde çözülecek problemin yeni serimini elde etmek için i dalının kaynaklandığı düğümün yeni seriminde i altturuna karşı gelen ayırıtın maliyeti sonsuza eşitlenir. Bu şekilde elde edilmiş olan yeni serim üzerinde en kısa yol problemi çözümlere düğümün ASı bulunur



Şekil 5.6. Dallanma Kuralı

Örneğin Şekil 5.6. da dallanacağımız düğümde C_{58} sonsuza eşitlenmiş olsun ve bunu $\bar{C}_{58}$ şeklinde gösterelim. Elde edilen çözümde enküçük taşıt sayısına sahip olup taşıt üst-sınırını aşan taşıt türünün dört taşıtı olsun. Bu dört taşıtın altturlarına çözümde karşı gelen ayrıtlar (4,7), (9,12), (17,21), (26,30) olsun. Bu ayrıtlardan herbirinin maliyeti ayrı ayrı sonsuza eşitlenerek dört yeni düğüm oluşturulur. Bu durum Şekil 5.6. da gösterilmiştir.

5.6.2. Sınırlama Kuralı

Yöntemin başlangıcında problem kısıtsız olarak çözülür. Bu çözümün olursuz olduğunu varsayarsak, çözüm değeri probleme bir altsınırdır (AS). Eğer elimizde başlangıçta bir olurlu çözüm yoksa üstsınır (ÜS) olarak sonsuz alınır. Eğer varsa bu çözümün değeri ÜS olur. Her olurlu çözüm bulunduğu da değeri mevcut ÜS ile karşılaştırılır ve küçük ise yeni ÜS değeri olarak alınır. Aksi takdirde elenir. Her düğümün çözüm değeri o düğümün AS nı oluşturur.

Dal-sınır yönteminin uygulama stratejisini belirleyen ise dallanılacak düğümün seçimi kuralıdır.

5.6.3. Dallanılacak Düğümün Seçimi Kuralı

Dallanılacak düğümün seçimi için üç seçenek vardır: (i) enküçük AS lı düğümden dallanılır, (ii) en son bulunulan düğümden dallanılır, (iii) önce bir düzeydeki tüm düğümler oluşturulacak şekilde dallanılır (dal-sınır ağacının başlangıç noktasından bir düğüme geçilen enküçük dal sayısı o düğümün düzeyini verir; buna göre başlangıç düzeyi sıfırdır) (Barthés, 1975).

Burada uygulanan kural (ii). kural olmuştur. Bu seçimin nedeni bu kuralın (i). ve (iii). kuralların gerektirdiğinden çok daha az bilgisayar hafızası gerektirmesidir. Bu şekilde daha büyük boyutlu problemleri çözmek mümkün olabilmektedir.

Önerilen dal-sınır yönteminin esaslarını böylece belirledikten sonra bunu bir algoritma şeklinde aşağıdaki gibi verebiliriz.

5.6.4. Algoritma

Adım 1: Verilmiş olan serim üzerinde kentli postacı problemi çözülerek ana tur elde edilir.

Adım 2: Ana tur kullanılarak yapılan bir dönüşümle yeni serim hazırlanır. Bunun için altbölüm 5.4.2. de adımlar halinde verilen işlemler uygulanır.

Adım 3: Yeni serim üzerinde en kısa yol algoritması uygulanır. Elde edilen çözümde değişik tür taşıtlara yapılan atamaların taşıt sayısı kısıtlarını aşıp aşmadığı kontrol edilir. Eğer etmiyorsa, durulur - eniyi çözüm elde edilmiştir. Aksi takdirde altsınır $\underline{z}_0$ mevcut çözüm değerine, üstsınır $\bar{z}$ sonsuza ve çözüm vektörü $\bar{Y}$ ise $\emptyset$ a eşitlenir. $j=0$, $n=1$, $k=1$ yapılır.

Adım 4: j düzeyinde k düğümünde en kısa yol algoritması çözülür. Eğer çözüm olurlu ise Adım 5 e gidilir. Aksi takdirde çözümün değeri altsınır $\underline{z}_k$ olarak alınır ve eğer $\underline{z}_k \geq \bar{z}$ ise k düğümü budanır ve Adım 6 ya gidilir. Aksi takdirde çözümün taşıt kısıtını ihlal eden taşıt türleri arasında en küçük taşıt sayısına sahip olan bulunur. Bu taşıt türünün yaptığı altturlara karşı gelen en kısa yol ayrıtları saptanır. Bunların birincisinin maliyeti sonsuza eşitlenerek yeni serim bu-

lunur. $j \leftarrow j+1$, $n \leftarrow n+1$ ve $k=n$ yapılarak bu adım tekrarlanır.

Adım 5: Düğüm k budanır. Olurlu çözüm değeri $z_k \geq \bar{z}$ ise Adım 6 ya gidilir. Aksi takdirde $\bar{z} = z_k$, $\bar{Y} = Y_k$ olarak alınır ve Adım 6 ya gidilir.

Adım 6: k düğümünü elde etmek için maliyeti sonsuza eşitlenmiş ayrıtın maliyeti sonlu değerine eşitlenir. k düğümünün dallanmış olduğu h düğümü bulunur. h düğümünde maliyeti sonsuza eşitlenecek ayrıt kalmamışsa Adım 7 ye gidilir. Aksi takdirde ilk böyle ayrıtın maliyeti sonsuza eşitlenerek yeni serim elde edilir. Bu yeni serimin düğümü $n \leftarrow n+1$ ve $k=n$ yapılır. Adım 4 e gidilir.

Adım 7: $k=h$ ve $j \leftarrow j-1$ yapılır. Eğer $j \geq 1$ ise Adım 6 ya gidilir. Aksi takdirde bütün ağaç taranmıştır, durulur. Eğer $\bar{z} = \infty$ ise problemin olurlu çözümü yoktur. Aksi takdirde $(\bar{z}, \bar{Y})$ eniyi çözüm olarak alınır.

5.7. ÖRNEK PROBLEMLER

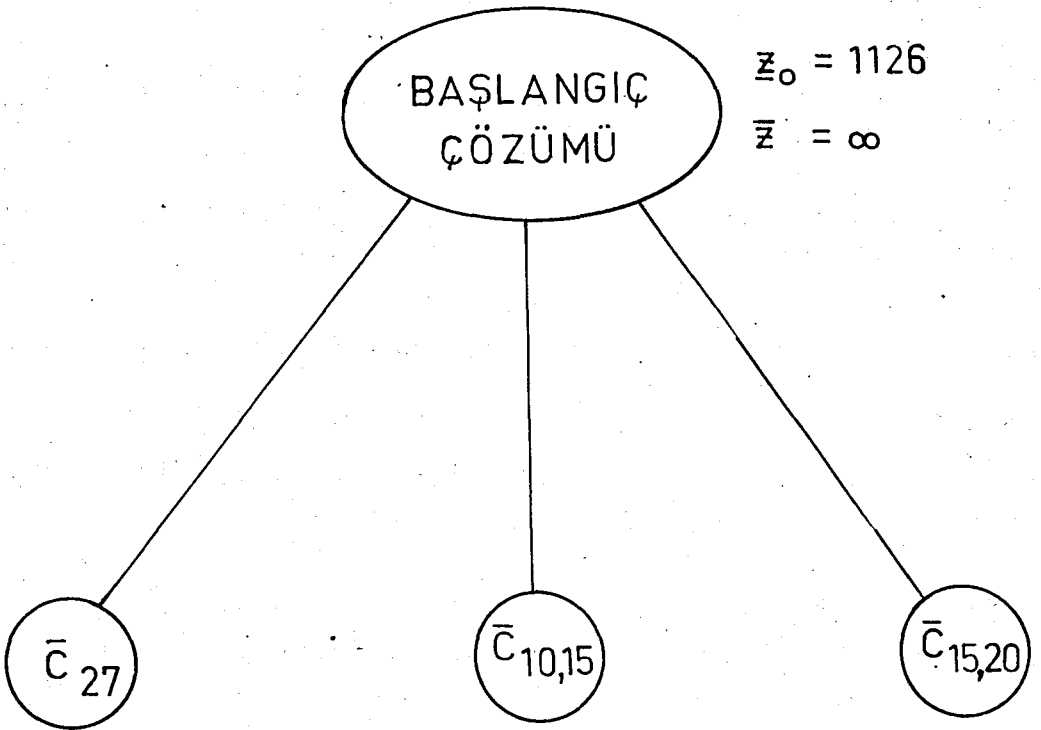
Altbölüm 5.5. de verilen örnek problemde ayrıca taşıt sayılarında şu kısıtları kabul edelim:

Taşıt Türü	1	2	3
Taşıt Sayısı	3	3	2

Dal-sınır yöntemine altbölüm 5.5. de verilen çözüm başlangıç çözümüdür ve olurlu olmaması nedeni ile dallanma gerekmektedir. Bu çözümde yapılan taşıt atamaları şöyledir:

Taşıt Türü	1	2	3
Atanan Taşıt Sayısı	1	2	3

Görüldüğü gibi 3. tip taşıta yapılan atama taşıt sayısı kısıtını ihlal etmektedir. Kısıtları ihlal eden tek taşıt tipi bu olduğundan bu taşıta ait altturlara dallanma yapılır. Bunlar, 2., 5. ve 6. altturlardır. Bu altturların maliyetleri sırası ile C_{27} , $C_{10,15}$ ve $C_{15,20}$ 'dir. Başlangıç düğümünden üç dallanma yapılacak ve herbirinin düğümünde sırası ile C_{27} , $C_{10,15}$ ve $C_{15,20}$ sonsuza eşitlenecek ve bu şekilde elde edilen yeni serim üzerinde en kısa yol algoritması uygulanacaktır (Şekil 5.7).



Şekil 5.7. Dal - Sınır Ağacının İlk İki Düzeyi

Probleme dal-sınır uygulaması sırasında 43 düğüm incelenmiştir. 10 düğümde maliyetleri sonsuza eşitlenen ayrıtların iptali ile elde edilen yeni serimler parçalı serimler olmuşlardır. Bu nedenle bir en kısa yol bulunamadığından bu düğümler budanmıştır. 19 düğüm altsınırları mevcut üstsınırdan küçük olmadıklarından elenmişlerdir. 2 düğümde ise olurlu çözüm elde edilmiştir.

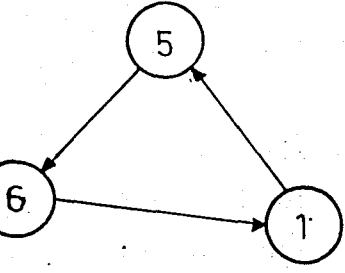
Algoritmanın bulduğu en kısa yol şöyledir:
{0,2,7,8,9,11,15,20}. Bu yolun maliyeti 1140 dır.

Çözümün altturları, taşıt kapasitesi ve taşıt kullanımı aşağıdaki gibidir:

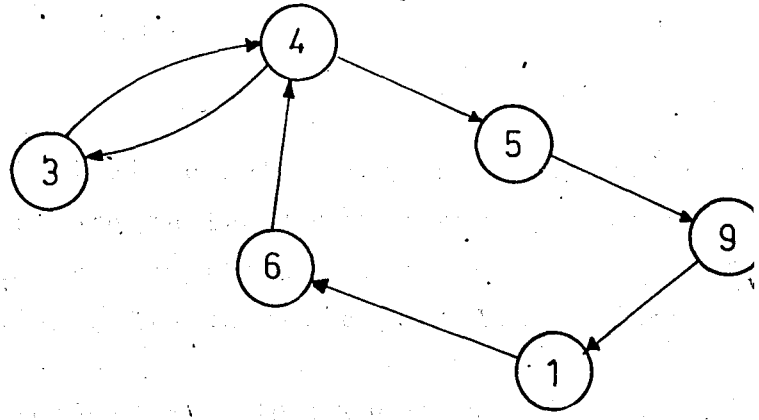
Alttur 1. {1,5,6,1}	Taşıt kapasitesi= 80 Taşıt kullanımı = 60
Alttur 2. {1,6,4,3,4,5,9,1}	Taşıt kapasitesi=100 Taşıt kullanımı = 94
Alttur 3. {1,9,8,1}	Taşıt kapasitesi= 50 Taşıt kullanımı = 41
Alttur 4. {1,9,1,8,1}	Taşıt kapasitesi= 80 Taşıt kullanımı = 80
Alttur 5. {1,8,10,7,2,3,1}	Taşıt kapasitesi= 80 Taşıt kullanımı = 80
Alttur 6. {1,6,3,6,2,10,6,1}	Taşıt kapasitesi=100 Taşıt kullanımı = 96

{1,8,9,1} altturu (8,9) boş ayrıt olduğundan yukarıya alınmamıştır. Elde edilen altturlar Şekil 5.8 de gösterilmişlerdir.

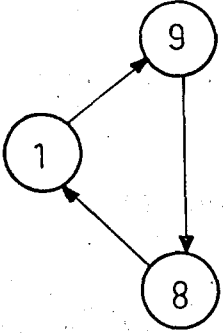
İlginç bir nokta kısıtsız problemde kararlı bir alttur olarak gözüken {1,9,1} altturu burada alttur uzunluğunu uzatarak taşıt kullanımını arttırmış ve böylece diğer altturların bir miktar yükünü taşımıştır. Başlangıç çözümüne göre değişiklik 4. ve 5. altturlarda yapılmış ve kısıtlar böylece sağlanmıştır. 4. alttura düğüm 8 ve dolayısı ile (1,8) dolu ayrıtı eklenmiş ve 5. altturda (1,8) ayrıtı boş ayrıt yapılarak taşıt kapasitesi 100 den 80 e düşürülmüştür. Diğer turlar aynen muhafaza edilmişlerdir.



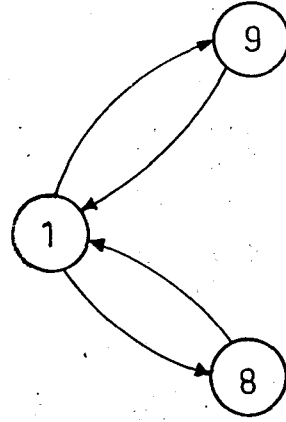
1. Alttur (K=80)



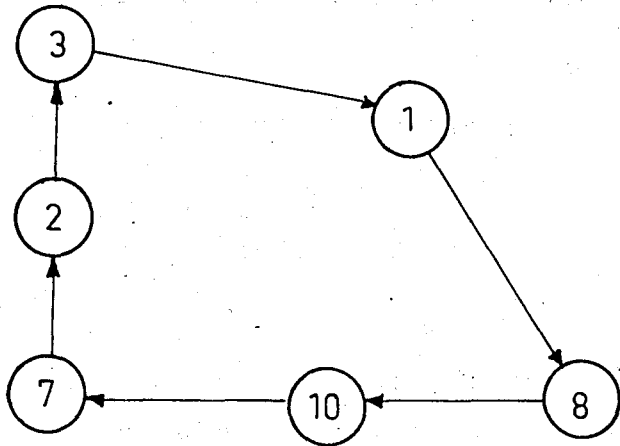
2. Alttur (K=100)



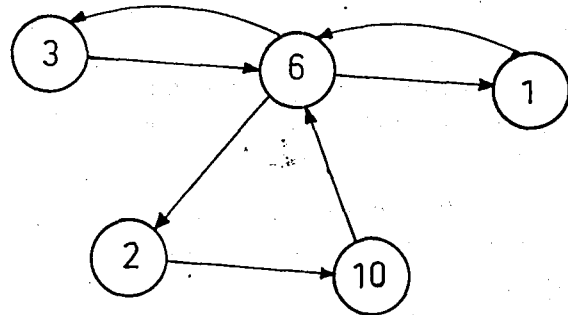
3. Alttur (K=50)



4. Alttur (K=80)



5. Alttur (K=80)



6. Alttur (K=100)

Şekil 5.8. İlk Kısıtlı Örnek Probleminin Çözümü

Altbölüm 5.5.de ikinci bir örnek olarak aynı problemde değişik bir kentli postacı turu kullanılarak altturlar bulunmuştu. Aynı kentli postacı turuna yukarıda verilen taşıt sayısı kısıtları uygulandığında aşağıdaki neticeler elde edilir:

En kısa yol şöyle bulunmuştur: {0,1,6,9,11,16,18,19,20} .
Bu yolun maliyeti 1171 dir.

Çözümün altturları, taşıt kapasiteleri ve taşıt kullanımı aşağıdaki gibidir:

1. Alttur. {1,8,1}	Taşıt kapasitesi= 50 Taşıt kullanımı = 15
2. Alttur. {1,8,10,7,2,3,6,1}	Taşıt kapasitesi=100 Taşıt kullanımı = 99
3. Alttur. {1,6,2,10,6,1}	Taşıt kapasitesi= 50 Taşıt kullanımı = 39
4. Alttur. {1,6,1,5,1}	Taşıt kapasitesi=100 Taşıt kullanımı = 84
5. Alttur. {1,5,6,4,3,4,5,1}	Taşıt kapasitesi= 80 Taşıt kullanımı = 72
6. Alttur. {1,5,9,8,1}	Taşıt kapasitesi= 80 Taşıt kullanımı = 77
7. Alttur. {1,9,1}	Taşıt kapasitesi= 80 Taşıt kullanımı = 65

1,8,9,1 altturu (8,9) boş ayrıt olduğundan bu listeye dahil edilmemiştir. Kısıtsız problemde 4 adet 80 kapasiteli taşıt kullanılmıştı ve burada koyulan taşıt kısıtını ihlal eden sadece bu taşıt tipidir. Başlangıç çözümü olan kısıtsız problemden dört dallanma yapılmıştır. Budallanmalar sırası ile C_{59} , $C_{11,16}$ ve $C_{16,18}$ ve $C_{19,20}$ 'nin sonsuza eşitlenmesi

ile yapılır. Problemin çözümü sırasında incelenen düğüm sayısı 64 olmuştur. Bu düğümlerden 13 ü serimleri parçalandığından budanmışlardır. 31 düğüm ise altsınırları mevcut üstsınırdan küçük olmadıklarından budanmışlardır. Olurlu çözüm 1 düğümde elde edilmiştir.

Başlangıç çözümüne göre taşıt sayısı 1 adet artmıştır. Bu çözümünden değişiklik başlangıç çözümünün ilk iki altturlunda kaplanan ayırtların burada değişik bir kombinasyonla üç altturla kaplanması olmuştur. Diğer altturlar aynı kalmıştır.

Aynı problem çözümünden türetilen iki değişik kentli postacı turu kullanılarak elde edilen kısıtlı problem sonuçlarında birbirinden oldukça farklı altturlar elde edilmiştir.

B.Ü.Bilgisayar Merkezindeki UNIVAC 1106'da birinci kısıtlı problem 2.42 CPU saniye, ikinci kısıtlı problem ise 2.90 CPU saniyede çözülmüşlerdir.

Bundan sonraki Bölümde bu çalışmada geliştirilen üç yöntemin bir değerlendirmesi yapılacak ve bazı öneriler getirilecektir.

6. SONUÇ

6.1. ASKERİ UÇAKLARIN İNTİKAL PLANLAMASI İÇİN KESİM ÇÖZÜM YÖNTEMİNİN DEĞERLENDİRİLMESİ

6.1.1. Kesin Çözüm Yönteminin Denenmesi

Problemin daha önce açık literatürde çok az işlenmiş olması nedeni ile algoritmayı başka algoritmalarla karşılaştırmak olanağı bulunamamıştır. Ancak algoritmanın verimliliği ve özellikleri hakkında bilgi edinebilmek amacı ile değişik problemler çözülmüştür. Bu problemlerden edinilen deneyimler yalın yolların türetilmesi ve özel küme parçalama algoritmaları açısından ayrı ayrı değerlendirilmiştir. Problemleri çözen bilgisayar programları üç ayrı birimden oluşmaktadır: yalın yolları türeten birim, geçiş birimi ve darboğaz uygulamasını da içeren özel küme parçalama algoritması birimi. Bu üç birim uygulamada ayrı ayrı birimler olarak çalıştırılmışlardır. Bunun nedeni aksi halde kütükten (file) çalışılacağı için programın yavaşlaması ve B.Ü.Bilgisayar Merkezindeki bazı pratik zorluklardır. Verilecek olan bütün CPU saniye süreler B.Ü.Bilgisayar Merkezindeki UNIVAC 1106'da alınmış sürelerdir.

Yalın yolların türetilmesi için önerilen yöntem çözülen problemlerde başarılı olmuştur. Kesin olarak sınanmıştır ki yöntem bütün olurlu yalın yolları türetmektedir. Türetilen yalın yolların sayısının nokta ve ayırıt sayısı ile doğrudan bağıntılı olduğu gözlenmiştir. Ancak iyi bir üstsınır getirildiği takdirde olurlu yalın yolların sayısı makul düzeylerde tutulabilmektedir. Şu anda eldeki bilgisayar programı 1500 olurlu yalın yolu hafızasında tutabilecek şekilde hazırlanmıştır. Ancak daha yavaş bir çalışma şekli olan kütükten çalışma yapılırsa bunu 10 000 e çıkarmak bir sorun olmayacaktır. Yalın yol türetme programı 1500 olurlu yalın yolu 60 CPU saniye civarında türetmektedir.

Özel küme parçalama algoritmasının çözölen 10 adet problemdeki performansı Tablo 6.1'de gösterilmektedir. Dördüncü kolonda gösterilen B matrisinin kolon sayısı olurlu ve

Tablo 6.1- Özel Küme Parçalama Algoritması Deney Problemleri.

No	Düğüm Sayısı	Uçuş Sayısı	B Matrisi Kolon Sayısı	Çözüm Süresi (CPU Saniye)
1	4	7	119	6.75
2	4	10	70	3.60
3	4	12	186	12.22
4	4	13	359	28.56
5	4	15	255	19.05
6	5	17	242	21.63
7	6	18	286	29.19
8	6	20	342	59.28
9	10	23	305	124.93
10	10	25	421	196.18

gerekli yalın yolların sayısını vermektedir. Çözüm süreleri sadece bir fikir vermek amacı ile rapor edilmektedirler. Tablo 6.1 de verilen ve diğer çözölen problemler göstermiştir ki çözüm süresini etkileyen bir çok parametre vardır. Bu nedenle bütün bu parametreleri kontrol edecek bir deneysel çalışma için gerekli sayı ancak rassal olarak problem türetilmesi ile elde edilebilecektir. Fakat rassal olarak problem türetimi esas problemden bizi çok uzaklaştırabilecektir. Bu nedenle az sayıda ancak anlamlı problemlerin çözümü ile problemin ve programın daha iyi tanınması yeğlenmiştir. Bu problemlerden elde edilen süreler oldukça başarılı bir neticeyi göstermektedir. Problem büyüklüğünün çözüm süresini üssel olarak etkilemediği elde edilen neticelerden görölmektedir. Buna göre,

daha büyük problemlerin çözümünde bu birim bir darboğaz oluşturmayacaktır.

Tablo 6.1 de verilen en büyük problem 10 düğümlü ve 25 yüklüdür. Denenen daha büyük boyutlu problemlerde yalın yolları türeten birimde 1500 lük hafızanın yetersiz kaldığı görülmüştür. Bu nedenle daha büyük problemler için kütükten çalışma ihtiyacı doğmaktadır. Çözüm süresinde bir artış söz konusu olmakla birlikte çok büyük boyutlu bir (0-1) tamsayı programlama problemine yapılan serim dönüşümü ve uygulanan algoritma ile görelî olarak çok düşük sürelerde çözüm getirilmiştir.

6.1.2. Bazı Yeni Öneriler

Problemın yapısına eklenmesi gerekli bir boyut havalanlarının kapasite kısıtıdır. Kapasite kısıtı bu çözüm yöntemine şöyle eklenebilir. Problem burada önerilen yöntemle çözülür. Bulunan netice kapasite kısıtları açısından kontrol edilir. Olurlu ise eniyi çözüm elde edilmiştir. Aksi takdirde kapasite kısıtını ihlal eden yalın yolların güzergahlarında süreyi uzatmadan bir değişiklik yapılması olanağı araştırılır. Bu işlem başarılı olamazsa aynı mecburi noktalardan ancak değişik sırada ve daha uzun sürede geçtikleri için B matrisine alınmayan yalın yollar gözönüne alınır. Harekât süresini uzatmayacaksa değişiklik yapılabilme olanağı araştırılır. Bu işlem de başarılı olmazsa kapasite kısıtı ihlal edilen havalanları arasında kısıtı ihlal eden yalın yol demetlerinden enb enk görev süreli olanı seçilir. Problem bu demetteki yalın yolların hep birlikte çözümde bulunamayacağı şekilde çözülür. Bu işlemler algoritma ve programda ufak değişikliklerle gerektirmektedirler.

Probleme getirilebilecek diğer bir boyut yükler arasındaki önceliktir. Bu öncelik mutlak bir önceliktir, yani ikinci öncelikli yüklere ait çözüm birinci öncelikli yüklere ait çözümü etkilemez. Birinci öncelikli yükler için çözüm bu-

lunsun ve alternatif çözüm olmasın. Bu durumda ikinci öncelikli yüklerin en erken işleme girmesi bu çözümdeki en kısa görev süresinin sonunda olabilir. Bu andan itibaren uçaklara birinci öncelikli yüklerle ilgili görev süreleri sonuna kadar yapay uçuş atanır. Problem yeniden ikinci öncelikli yükler için çözülür. Aynı süreç diğer öncelikler için tekrarlanır. Alternatif çözüm olduğunda bir sonraki öncelik için değişik başlangıç çözümleri var demektir. Bunların gözönüne alınması problemin çözüm süresini oldukça etkileyecektir. Bu durumda kesin çözümden vazgeçilerek bazı sezgisel kurallar geliştirilip yönteme eklenebilir.

İntikal planlaması olayının yapısında bir belirsizlik hali vardır. Örneğin harekât sırasında uçak sayısında bir azalma veya kapasite kısıtlarında değişimler olabilir. Bunların rassal bir formülasyonla gözönüne alınması son derece karmaşık hatta olanaksız olabilir. Bu nedenle belirsizlik halinin önceden yapılacak bir muhtemel olaylar analizi ve o anda uygulanacak etkileşimli yaklaşımla çözülmeye çalışılması en gerçekçi yol olacaktır kanısındayız.

6.2. ASKERİ UÇAKLARIN İNTİKAL PLANLAMASI İÇİN BİR SEZGİSEL ÇÖZÜM YÖNTEMİNİN DEĞERLENDİRİLMESİ

TGB problemlerinin temel sezgisel yaklaşımlarından biri olan önce güzergahın bulunup sonra talep noktalarının gruplanması yaklaşımını uygulayarak bir sezgisel çözüm yöntemi geliştirilmiştir. Kentli postacı turu uçak sayısı kadar alt-tura bölünerek elde edilen güzergahlara geliştirme yordamı uygulanır. Geliştirme yordamı tamamen sezgiseldir. Görüldüğü gibi bu sezgisel kurallar kolaylıkla uygulanabilirler ve basit bir mantığa dayandırılmışlardır. Sezgisel kurallar eniyi çözümde güzergahların özelliklerinin ne olabileceği tespit edilerek geliştirilmişlerdir. Bu amaçla kesin çözüm yöntemi-

nin sağladığı eniyi çözümler kullanılmıştır.

Sezgisel yöntem bir dizi probleme uygulanmıştır. Sonuçlar Tablo 6.2 de verilmiştir. Bu sonuçlardan görüleceği gibi sezgisel çözüm eniyi çözüme oldukça yakın neticeler türetebilmektedir.

Tablo 6.2- Deney Problemleri Neticeleri*

No	Eniyi Çözüm		Sezgisel Çözüm	
	Güzergah	T	Güzergah	T
1	1, 4, 1, 2, 4, 1, 3	48	1, 4, 1, 4, 2, 4, 1	48
	4, 2, 1, 4, 1, 3*, 4, 1	<u>50</u>	4, 1, 3*, 4, 1, 2, 1, 3	<u>50</u>
2	1, 2, 4*, 3, 4, 1, 4	42	1, 3, 2, 3, 4*, 3, 4	41
	3, 2, 1, 3, 4*, 3, 1	<u>43</u>	3, 2, 4, 1, 3, 4	41
	4, 1, 3, 2, 3, 4	<u>40</u>	4*, 3, 1, 2, 1, 4, 1	<u>43</u>
3	1, 3, 9, 6, 2, 3, 5, 2, 1	63	1, 7, 6, 1, 3, 5, 2, 6, 2	65
	2, 4, 6, 1, 7, 8, 6, 2, 8	63	2, 10, 7, 8, 2, 8, 2, 6, 8, 6	64
	2, 6, 8, 2, 10, 7, 6, 8, 2, 6	<u>65</u>	2*, 6, 2, 4, 6, 8*, 2, 3, 9, 6*, 2, 1	<u>67</u>
4	1, 3, 4, 1, 4, 3	29	1, 2, 4, 3, 4	30
	4, 1, 2, 3	<u>28</u>	4, 1, 4, 1*, 4, 2	<u>33</u>
	2, 4, 3, 4, 1	28	2, 3, 4, 3, 4, 3	31
	3, 4, 2, 1	28	3*, 4, 1, 4, 3, 4, 1	30
	3, 4, 3, 4, 1, 4	27	3*, 2, 1, 3, 4	31
5	4, 7, 8, 6, 2	<u>31</u>	4, 7, 8, 6, 2	<u>31</u>
	1, 4, 5, 1, 3	29	1, 4, 5, 1, 3	29
	3, 8, 2, 9, 2, 3	29	3, 8, 2, 9, 2, 3	29
	6, 10, 2	28	6, 10, 2	28
	6, 8, 3, 2, 9	27	6, 8, 3, 2, 9	27
	2, 6, 3, 9, 6	27	2, 6, 3, 9, 6	27

* T_{enb} değerinin altı çizilmiştir.

Yapılan deneysel çalışmaların çerçevesinde varılan sonuç çözüm süresinin yük ve uçak sayısına doğrudan bağımlı olduğudur. Sezgisel yöntem problem boyutları büyüdükçe kesin çözüm yöntemine göre oldukça kısa sürede çözülebilmektedir. Örneğin Tablo 3.5 deki 9.problem kesin çözüm yöntemi ile

124.93 CPU saniyede çözülürken sezgisel yöntemle 30.42 CPU saniyede çözülmüştür.

Sezgisel yöntemin bu özelliği bu yöntemi kesin çözüm yöntemi ile birleştirmek önerisine olurluluk kazandırmaktadır. Buna göre, sezgisel yöntemin bulduğu olurlu çözüm kesin çözüm yönteminde türetilecek yalın yolların uzunluğuna bir üstsınır olarak alınır. Böylece yalın yolları türetme algoritması için oldukça sıkı bir üstsınır bulunmuş olmaktadır. Bu üstsınır yalın yolları türetme algoritmasının daha kısa çalışmasını sağlayacağı gibi hafıza yükünü de azaltacaktır. Aynı zamanda bu yalın yolları girdi olarak kullanan özel küme parçalama algoritmasında B matrisinin kolon sayısında azalma olacaktır.

Sezgisel ve kesin çözüm yönteminin müşterek çalışmalarının ilginç yönü kesin çözüm yönteminin sezgisel yöntemin bir parçası değil de sezgisel yöntemin kesin çözüm yönteminin bir parçası olarak çalışmasıdır. Ve bu birleştirme neticesi kesin çözüm yönteminin eniyi çözümü kaybetme olasılığı sıfırdır.

Sezgisel yöntem kapasite kısıtını hiçbir soruna neden olmadan gözönüne almıştır. Aynı şekilde öncelikler sorunununda halledilmemesi için yöntemin uygulamasından gelen bir zorluk yoktur. Bunlar geliştirilmiş olan sezgisel yöntemin esnek yapısını göstermektedir.

6.3. ATGB PROBLEMİNDE TAŞIT SAYISI VE KAPASİTESİ İÇİN BİR SEZGİSEL ÇÖZÜM YÖNTEMİNİN DEĞERLENDİRİLMESİ

Önerilen çözüm yöntemi serim kuramının temel algoritmalarından olan kentli postacı turu ve en kısa yol algoritmalarına dayandırılmıştır. İki algoritma da buradaki uygulama çerçevesinde kesin çözüm garanti etmekle birlikte çözüm yön-

temi eniyi çözümün bulunacağını garanti etmemektedir; yani sezgisel bir yöntemdir. Bunun nedeni kentli postacı problemi ile elde edilen ana turdan yeni serime geçilirken oluşturulan altturların bütün altturların ancak bir altkümesini oluşturmasıdır. Kentli postacı probleminin amaç işlevinin salt değişken maliyeti içerdiğini anımsarsak, kentli postacı turundan oluşturulan altturların diğer altturları örtük (implicit) bir biçimde elemiş olmalarının esas problemin amaç işlevi göz-önüne alındığında olanaksız olduğu görülür. Toplam maliyet sabit maliyeti de içerdiğinden, tur uzunluğunun öneminin amaç işlevindeki ağırlığından bağımlı olarak kentli postacı turunun altturlarından daha başka altturlar eniyi çözümü oluşturabilirler.

Algoritmanın bu yönünün geliştirilmesi için bir öneri kentli postacı probleminin sonucunu eklenmesi en küçük maliyetli ayrit altkümesi olarak aldıktan sonra bir Euler turu değil fakat değişik Euler turları almak ve her birinden elde edilen yeni serim üzerinde en kısa yol problemini çözerek değişik çözümler elde etmektir. Böylece değişik altturlar bulmak mümkün olacak ve olurlu altturlar kümesinin daha geniş bir bölüm taranmış olacaktır.

Diğer bir öneri yeni serim üzerinde boş ayrit düğümlerinin dağılımı üzerinde değişiklikler yaparak en kısa yol algoritmasını defaeten uygulamaktır. Burada her uygulama için sadece bazı ayrit maliyetlerinin yeniden hesaplanması gerekmektedir. Bu önerinin uygulanması bir evvelki önerinin uygulanmasına göre çok daha az işlem gerektirmektedir.

Bu iki öneri de çözülen problemlerde denenmiştir. Amaç işlevi değeri büyük farklılıklar göstermemekle birlikte değişik altturlar elde edilmiştir. Böylece karar verici için model sonucundan uygulamaya geçerken oldukça yararlanacağı bir dizi seçenek türetilmiş olmaktadır.

Dal-sınır uygulaması için önerilmiş olan dallanma, sınırlama ve dallanılacak düğümün seçimi kurallarının oldukça başarılı olduğu görülmektedir. Çözülen problemlerde incelenen düğümlerin önemli bir bölümünün altsınırlarının mevcut üst-sınırdan büyük olması nedeni ile budandıkları saptanmıştır.

Çözüm getirilen problem daha önce literatürde yer almamış bir problem olduğundan henüz algoritmaların karşılaştırılabileceği bir dizi deney problemi yoktur. Bu nedenle algoritma burada oluşturulan problemler üzerinde denenmiştir. Algoritmanın programının geliştirilmesinde NETPACK'daki (Ulusoy ve Kasaroğlu, 1981) programlardan yararlanılmıştır.

Yönsüz serimlerde çözülmüş olan 10 adet deney problemi Tablo 6.3 de sunulmuştur. Verilen süreler kısıtlı problemlerin çözüm süreleridir.

Tablo 6.3- Deney Problemleri Neticeleri

No	Düğüm Sayısı	Ayrıt Sayısı	Çözüm Süresi (CPU Saniye)
1	10	18	2.90
2	12	24	4.16
3	13	28	5.86
4	15	36	8.81
5	18	54	13.63
6	20	64	22.12
7	24	86	46.23
8	26	96	63.42
9	28	114	81.37
10	30	128	112.17

Algoritmanın çözüm süresinin öncelikle serimin ayırıt sayısına ve sonra düğüm sayısına bağlı olduğu görülmüştür. Bunun nedeni ise çözüm sürelerinin önemli bir bölümünün başlangıç çözümü olan kentli postacı turunun bulunmasına ait olmasıdır ve bu turun bulunma süresi de serimin ayırıt ve nokta sayısından bağımlıdır. Programın dal-sınır bölümü enkısa yol algoritması ile çalıştığından harcanan süreler nispeten küçüktür. Elde edilen sonuçlar problem büyüklüğü ve çözüm süresi arasında üssel bir ilişki olmadığını göstermiştir ki bu büyük problemlerin çözümünün olurluğu açısından önemli bir neticedir. Yapılan deneysel çalışmalar daha ziyade yönsüz serimlerde olmuştur. Yönlü ve yönsüz serimler arasındaki fark kentli postacı turunun elde edilmesi yönteminde olmaktadır. Ancak 15 nokta ve 68 yönlü ayrıta kadar olan problemlerde yönlü ve yönsüz ayrıtların çözüm sürelerinde önemli farklılıklara rastlanmamıştır.

Bundan sonra yapılacak çalışmalar şu iki soruna yöneltebilirler. Sezgisel yöntemin daha iyi değerlendirilmesi açısından çözüm üzerinde bir altsınır bulunması gerekmektedir. Ancak çözümü mevcut çekirdek problemler böyle bir altsınırı doğrulamamaktadırlar. Bu nedenle bu konuda yapılacak bir çalışma gerek bu algoritmanın gerekse bu probleme önerilecek başka algoritmaların değerlendirilmesinde yararlı olacaktır.

Problemde kapasite ihtiyacı bir altturda hizmet verilen ayrıtların toplam talebi olarak kabul edilmiştir. Ancak bir altturda hizmet verilen ayrıtların enbüyük talebi de kapasite ihtiyacını belirleyebilir. Bu kuramsal açıdan son derece ilginç bir problemdir ve bir kentli postacı darboğaz probleminin çözümünü gerektirir ki bu probleme literatürde henüz çözüm getirilmemiştir.

KAYNAKÇA

- BAKER, T.E. (1981), "Interactive Vessel Scheduling at Exxon", CORS/TIMS/ORSA 11. Ulusal Toplantısında verilen bildirisi, Toronto, Kanada.
- BALINSKY, M.L. ve R.E. Quandt (1964), "On an Integer Program for a Delivery Problem", *Operations Research*, 12, s. 300-304.
- BARTHEs, J.P. (1975), "Branching Methods in Combinatorial Optimization", *Topics in Combinatorial Optimization*, S. Rinaldi (düz.), s. 117-143, Springer Verlag, New York.
- BELLMORE, M. ve G.L. Nemhauser (1968), "The Travelling Salesman Problem: A Survey", *Operations Research*, 6, s. 538-558.
- BELLMORE, M. ve S. Hong (1974), "Transformation of Multisalesman Problem to the Standard TSP", 21, *J.A.C.M.*, s. 500-504.
- BELLMORE, M. ve J. Malone (1971), "Pathology of Travelling Salesman Subtour Elimination Procedures", *Operations Research*, 19, 2.
- BELTRAMI, E.J. ve L.D. Bodin (1974), "Networks and Vehicle Routing for Municipal Waste Collection", *Networks*, 4, s. 65-94.
- BODIN, L.D. ve S.J. Kursch (1978), "A Computer Assisted System for the Routing and Scheduling of Street Sweepers", *Operations Research*, 26, 4, p. 525-537.

BODIN, L.D. ve S.J.Kursch(1979), "A Detailed Description of a Computer System for the Routing and Scheduling of Street Sweeping", *Computers and Operations Research*, 6, s.181-198.

CASSIDY, P.J. ve H.S.Bennett (1972), "TRAMP-A Multi-Depot Vehicle Scheduling System", *Operational Research Quarterly*, 23, s.151-163.

CLARK, G. ve J.W.Wright(1964), "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operational Research Quarterly*, 12, s.568-581.

CHRISTOFIDES, N. ve S.Eilon(1969), "An Algorithm for the Vehicle Dispatch Problem", *Operational Research Quarterly*, 20, s.309-318.

CHRISTOFIDES, N.(1973), "The Optimum Traversal of a Graph", *OMEGA*, 1, 6, s.719-732.

CHRISTOFIDES, N.(1975), *Graph Theory-An Algorithmic Approach*, Academic Press, New York.

CHRISTOFIDES, N. (1976), "The Vehicle Routing Problem", *RFAIRO*, 10, 2, s.55-70.

CHRISTOFIDES, N.A. Mingozzi ve P.Toth (1981), "Exact Algorithms for the Vehicle Routing Problem Based on Spanning Tree and Shortest Path Relaxations", *Mathematical Programming*, 20, s.255-282.

CROWDER, H. ve M.Padbery (1980), "Solving Large Scale TSP to Optimality", *Management Science*, 26, s.495-509.

DANIELSON, G.(1968), "On Finding the Simple Paths and Circuits in a Graph", *IEEE Trans. on Circuit Theory*, s.294-295.

- DANTZIG, G.B. ve J.H. Ramser (1959), "The Truck Dispatching Problem, *Management Science*, 6, 1, s.80-91.
- DIJKSTRA, E.W. (1959), "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, 1, s.269-271.
- DISCENZA, J.H. (1981), "A More Compact Formulation of the Symmetric Multiple Traveling Salesman Problem with Fixed Charges", *Networks*, 11, s.73-75.
- DREYFUS, S.E. (1969), "An Appraisal of Some Shortest Path Algorithms", *Operations Research*, 17, 3.
- EDMONDS, J. (1965), "Maximum Matching and a Polyhedron With Oil Vertices", *J. of Research of the National Bureau of Standards*, 69B, s.35-40.
- EDMONDS, J. ve E.L. Johnson (1973), "Matching, Euler Tours and the Chinese Postman Problem", *Mathematical Programming*, 5, 1, s.88-124.
- EILON, S., C.D.T. Watson-Gandy, N. Christofides (1971), *Distribution Management*, Hafner Pub. Co., New York.
- FLOYD, R.W. (1962), "Algorithm 97, Shortest Path", *Comm. ACM*, 5, s.345.
- FOSTER, B.A. ve D.M. Ryan (1976), "An Integer Programming Approach to the Vehicle Scheduling Problem", *Operational Research Quarterly*, 27, s.367-384.
- FRATTA, L. ve U. Montanari (1975), "A Vertex Elimination Algorithm for Enumerating all Simple Paths in a Graph", *Networks*, 5, s.151-157.

- FREDERICKSON, G.N., M.S. Hecht ve C.E. Kim (1978), "Approximation Algorithms for Some Routing Problems", *SIAM J. Comput.*, 7, s.178-198.
- GAREY, M. ve D.S. Johnson (1979), *Computers and Intractability - A Guide to the Theory of NP - Completeness*, W.H. Freeman and Co., San Francisco.
- GARFINKEL, R.S. ve G.L. Nemhauser (1969), "The Set Partitioning Problem: Set Covering with Equality Constraints", *Operations Research*, 17, s.848-861.
- GARVIN, W., H. Crandall, J. John ve R. Spellman (1957), "Application of LP in Oil Industry", *Management Science*, 3, s.407-430.
- GASKELL, T.J. (1967), "Bases for Vehicle Fleet Scheduling", *Operational Research Quarterly*, 18, s.281.
- GILLETT, B.E. ve L.R. Miller (1974), "A Heuristic Algorithm of the Vehicle Dispatch Problem", *Operations Research*, 22, s.340-349.
- GILLETT, B.E. ve T. Johnson (1976), "Multi-terminal Vehicle Dispatch Algorithm", *OMEGA*, 4, s.711-718.
- GOLDEN, B. ve T. Magnanti (1977), "Deterministic Network Optimization - A Bibliography", *Networks*, 7, s.149-183.
- GOLDEN, B., T. Magnanti ve H. Nguyen (1977), "Implementing Vehicle Routing Algorithms", *Networks*, 7, s.113-148.
- GOLDEN, B. ve R.T. Wong (1980), "Capacitated Arc Routing Problems", College of Business and Management, University of Maryland, Maryland.

- GOLDEN, B., M. Ball ve L. Bodin (1981), "Current and Future Research Directions in Network Optimization", *Computers and Operations Research*, 8, s. 71-81.
- HOLMES, R. A. ve R. G. Parker (1976), "A Vehicle Scheduling Procedure Based Upon Savings and a Solution Perturbation Scheme", *Operational Research Quarterly*, 27, s. 83-92.
- HONG, S. ve M. W. Padberg (1977), "A Note on the Symmetric Multiple Travelling Salesman Problem With Fixed Charges", *Operations Research*, 25, s. 871-874.
- KARP, R. M. (1972), "Reducibility Among Combinatorial Problems", - *Complexity of Computer Computations*, R. E. Miller ve J. W. Thatcher (düz.), Plenum Press, New York.
- KROFT, D. (1967), "All Paths Through a Maze", *Proc. of IEEE*, s. 68.
- KROLAK, P., W. Felts ve J. Nelson (1972), "A Man-Machine Approach Toward Solving the Generalized Truck Dispatching Algorithm", *Transportation Science*, 6, s. 149-169.
- KWAN, Mei-Ko (1962), "Graphic Programming Using Odd or Even Points", *Chinese Math.*, 1, s. 273-277.
- LAWLER, E. L. ve D. E. Wood (1966), "Branch and Bound Methods: A Survey", *Operations Research*, 14, s. 699-719.
- LENSTRA, J. K. ve A. H. G. Rinnooy Kan (1976), "On General Routing Problems", *Networks*, 6, s. 273-280.
- LEVY, L., R. Golden ve A. Assad (1980), "The Fleet Size and Mix Vehicle Routing Problem", College of Business and Management, University of Maryland, Maryland.

- LIN, S. (1965), "Computer Solution of the Traveling Salesman Problem", *Bell Syst. Tech. Journal*, 44, s.2245-2269.
- MATTHAUS, F.W. (1976), "Savings Approach to the Multiple Terminal Delivery Problem", *Transportation Science*, 10, 2, s.216-221.
- MINIEKA, E. (1979), "The Chinese Postman Problem for Mixed Networks", *Management Science*, 25, s.643-648.
- ORLOFF, C.S. (1974a), "A Fundamental Problem in Vehicle Routing", *Networks*, 4, s.35-64.
- ORLOFF, C.S. (1974b), "Routing a Fleet of M Vehicles to/from a Central Facility", *Networks*, 4, s.147-182.
- ÖVER, E. ve U.Yaramancı (1980), "Hava Ulaştırmasının Planlanmasında Bilgisayar Kullanımı", *Hava Kuvvetleri Dergisi*, 58, 274, s.24-30.
- PIERCE, J.F. (1968), "Application of Combinatorial Programming to A Class of All-Zero-One Integer Programming Problems", *Management Science*, 15, s.191-220.
- PIERCE, J.F. (1969), "Direct Search Algorithms for Truck Dispatching Problems", *Transportation Research*, 3, s.1-42.
- STERN, H. ve M.Dror (1979), "Routing Electric Meter Readers", *Computers and Operations Research*, 6, s.209-233.
- SILVER, E.A., R.V.V.Vidal ve D.de Werra (1980), "A Tutorial on Heuristic Methods", *EJOR*, 5, s.153-162.

- TILLMAN, F. ve H. Chochran (1968), "A Heuristic Approach for Solving the Delivery Problem", *J. of Industrial Engineering*, 19, s. 354-358.
- TILLMAN, F. ve T. Cain (1972), "An Upper Bounding Algorithm for the Single and Multiple Terminal Delivery Problem", *Management Science*, 19, s. 664-681.
- Traffic Assignment*, (1973), FHA, U.S.D.O.T., Washington, D.C.
- TYAGİ, M. (1968), "A Practical Method for Truck Dispatching Problem", *J. of the ORS of Japan*, 10, s. 76-92.
- ULUSOY, G. ve İ. Or (1979a), "İETT Ulaşım Projesi-2. Ara Rapor", BÜTSAE, İstanbul.
- ULUSOY, G., N. Kesen ve B. Balaban (1979b), "A Mathematical Programming Package: LP-PLP-QP-BB-MIF", Endüstri Müh. Bölümü, Boğaziçi Üniversitesi, İstanbul.
- ULUSOY, G. ve K. Altınkemer (1980), "İntikal Planlamasına Bir Yöneylem Araştırması Yaklaşımı", Yöneylem Araştırması 6. Ulusal Kongresinde sunulan bildiri, Hacettepe Üniversitesi, Ankara.
- ULUSOY, G. ve M. Kasaroğlu (1981), "Introduction to a Network Optimization Package-NETPACK", Endüstri Müh. Bölümü, Boğaziçi Üniversitesi, İstanbul.
- WHITEHOUSE, G. E. (1973), *Systems Analysis and Design Using Network Techniques*, Prentice Hall, New Jersey.
- WREN, A. ve A. Holiday (1972), "Computer Scheduling of Vehicles from One or More Depots to a Number of Delivery Points", *Operational Research Quarterly*, 23, s. 333-344.

YELLOW, P.C. (1970), "A Computational Modification of the Savings Method of Vehicle Scheduling", *Operational Research Quarterly*, 21, s.281.