

**OFDMA İÇİN ENİYİ VE ENİYİYE YAKIN  
KAYNAK TAHSİSİ**

**OPTIMAL AND SUBOPTIMAL RESOURCE  
ALLOCATION FOR OFDMA**

**HASAN AKBULUT**

Hacettepe Üniversitesi

Lisansüstü Eğitim – Öğretim ve Sınav Yönetmeliğinin

ELEKTRİK ve ELEKTRONİK Mühendisliği Anabilim Dalı İçin Öngördüğü

**YÜKSEK LİSANS TEZİ**

olarak hazırlanmıştır.

2012

Fen Bilimleri Enstitüsü Müdürlüğü'ne,

Bu çalışma jürimiz tarafından **ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI 'nda YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Başkan :.....  
(Prof. Dr. Mehmet ŞAFAK)

Üye (Danışman) :.....  
(Doç. Dr. Cenk TOKER)

Üye :.....  
(Doç. Dr. Emre AKTAS)

Üye :.....  
(Yrd. Doç. Dr. Tolga GİRİCİ)

Üye :.....  
(Yrd. Doç. Dr. Umut SEZEN)

ONAY

Bu tez Hacettepe Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliğinin ilgili maddeleri uyarınca yukarıdaki jüri üyeleri tarafından ...../...../..... tarihinde uygun görülmüş ve Enstitü Yönetim Kurulunca ...../...../..... tarihinde kabul edilmiştir.

Prof.Dr. Fatma SEVİN DÜZ

Fen Bilimleri Enstitüsü Müdürü

# OFDMA İÇİN ENİYİ VE ENİYİYE YAKIN KAYNAK TAHSİSİ

HASAN AKBULUT

## ÖZ

Çok kullanıcıli sistemlerde kullanıcılar ortak bir kanalı kullanmaktadır. Kullanıcıların bu ortak kanalın kaynaklarının (alt-taşıyıcılar ve güç) nasıl kullanacağıının belirlenmesi kaynak tahsis probleminin temelini oluşturmaktadır. OFDMA sistemlerde kaynak tahsis problemi güç uyarlamalı ve hız uyarlamalı olmak üzere iki başlık altında incelenmektedir. Tez kapsamında her iki kaynak tahsis problemi de incelenmiştir.

OFDMA sistemlerde kaynak tahsis problemi doğası gereği ayrık bir yapıya sahiptir. Bu nedenle tamsayı programlama problemi olarak değerlendirilmektedir. Tez kapsamında kaynak tahsis probleminin tamsayı kısıtları gevşetilerek problem ayrık doğasından çıkartılıp dışbükey hale dönüştürülmüş ve kaynak tahsis problemi matris ve vektörler ile ifade edilebilir bir yapıya dönüştürülerek doğrusallaştırılmıştır.

Doğrusallaştırılmış kaynak tahsis problemi, tez kapsamında önerilen yeni bir yöntem ile çözülmüştür. Bu yöntem işlem karmaşıklığı az olan doğrusal programlama ve aç gözlü algoritma kullanılarak oluşturulmuştur. Önerilen yöntem ile literatürde kullanılan benzer yöntemlere göre çok daha hızlı bir şekilde sonuca ulaşılmaktadır.

Önerilen yöntem dışında, tamsayı programlama yöntemlerinden yükselt ve izdüşür algoritması kaynak tahsis problemlerine uygulanmış ve gelecekte yapılabilecek çalışmalar için önemli bilgiler elde edilmiştir.

Ayrıca tez kapsamında kanalın bir önceki alt-çerçeve zamanına olan ilintisi kullanılarak bir çözüm önerilmiştir. Bu yöntem ile kaynak tahsis probleminin çözüm süresinin daha da azaltılabildiği görülmüştür.

**Anahtar Kelimeler:** OFDM, OFDMA, LTE, Kaynak Tahsisi, Doğrusal Programlama, Çokyüzlüler, Tamsayı Programlama, Yükselt ve İzdüşür Algoritması

**Danışman:** Doç. Dr. Cenk TOKER, Hacettepe Üniversitesi, Elektrik ve Elektronik Mühendisliği Bölümü

# OPTIMAL AND SUBOPTIMAL RESOURCE ALLOCATION FOR OFDMA

HASAN AKBULUT

## ABSTRACT

In multiuser systems, users share the same channel. The origin of the resource allocation problem is formed by determining the channel resources sharing (subcarrier and power) among the users. In OFDMA systems, resource allocation problem is divided into margin adaptive (MA) and rate adaptive (RA). Both of these problems are researched in the thesis.

In OFDMA systems, resource allocation problem has a discrete structure. So, the problem is solved by using integer programming. In the thesis, the constraint of the resource allocation problem is relaxed and structure of the problem is changed from discrete form to a convex form. With this approach the problem is linearized that can be expressed with matrices and vectors.

In thesis, a new proposed algorithm is used to solve the linearized resource allocation problem. The proposed algorithm is modeled with using the linear programming and the greedy algorithm. Faster results are achieved by using the proposed algorithm than using the algorithms in the literature.

Except proposed algorithm, Lift and Project algorithm is applied to the resource allocation problem and important datas are collected for a future work.

Moreover, another solution is advised by using dependency of previous subframe time of the channel. Evaluation time of the resource allocation problem can be decreased with using this method.

**Keywords:** OFDM, OFDMA, LTE, Resource Allocation, Linear Programming, Polyhedra, Integer Programming, Lift and Project Algorithm

**Advisor:** Assist. Prof. Dr.. Cenk TOKER, Hacettepe University, Department of Electrical and Electronics Engineering.

## **TEŐEKKÜR**

Her Őeyden önce, bu tezin oluŐturulmasında bilgi ve tecrübelerini sonuna kadar paylaşan, gösterdiđi ilgi ve anlayıŐı hiçbir zaman esirgemeyen deđerli tez danışmanım Sayın Doç. Dr. Cenk TOKER' e sonsuz teŐekkür ederim.

Tüm eđitim hayatım boyunca bana güvenen ve destek olan aileme çok teŐekkür ederim.

Benimle tanıştıđı günden itibaren gösterdiđi anlayıŐ, sabır ve destekten dolayı sevgili eŐime çok teŐekkür ederim.

## İÇİNDEKİLER DİZİNİ

	<u>Sayfa</u>
ÖZ .....	iii
ABSTRACT .....	iv
TEŞEKKÜR.....	v
İÇİNDEKİLER DİZİNİ .....	vi
ŞEKİLLER DİZİNİ.....	viii
SİMGELER VE KISALTMALAR DİZİNİ .....	x
SÖZLÜK DİZİNİ .....	xi
1. GİRİŞ .....	1
2. OFDMA SİSTEMLERDE KAYNAK TAHSİSİ.....	5
2.1 Sistem Yapısı ve Problem Tanımı .....	7
2.1.1 MA Problem Tanımı .....	9
2.1.2 RA Problem Tanımı.....	10
2.2 Problemin Doğrusallaştırılması.....	15
2.3 Kaynak Tahsis Problemlerinde Olurlu Kümenin Geometrisi .....	21
3. DOĞRUSAL TAMSAYI PROGRAMLAMA ve YÜKSELT-ve-İZDÜŞÜR ALGORİTMASI.....	23
3.1 Yükselt-ve-İzdüşür Algoritması .....	24
3.1.1 Yükseltme (Lifting) .....	25
3.1.2 İzdüşüm (Projection).....	26
3.2 Lift and Project İçin Önerilen Algoritma.....	33
4 ÖNERİLEN ENİYİYE YAKIN ALGORİTMA .....	38
4.1 LTE Sistem Yapısı .....	45
5. BENZETİM SONUÇLARI .....	50
5.1 MA Problem Benzetimleri .....	51
5.2 RA Problem Benzetimleri.....	59
5.3 Kanal Özilinti Değerine Göre Kaynak Tahsisi Benzetimleri .....	66
5.4 Lift and Project Algoritması Benzetimleri .....	72
6. SONUÇLAR .....	75
KAYNAKLAR.....	77
EKLER .....	81

EK-1: Çokyüzlüler ve Doğrusal Programlama .....	81
EK-2: Yükselt ve İzdüşür Algoritması Kılavuz.....	91
EK-3: Yükselt ve İzdüşür Algoritması İçin Önerilen Yöntem İspatı .....	95

## ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
<b>Şekil 2.1:</b> OFDMA Sistem Modeli [13] .....	7
<b>Şekil 2.2:</b> $\mathbf{x}^T$ bit tahsisi vektörünün yapısı.....	16
<b>Şekil 2.3:</b> Güç Tahsisi Vektörünün Genel Yapısı.....	17
<b>Şekil 3.1:</b> (3.13)'de verilen Eşitsizliklerin Tanımladığı Olurlu Bölge.....	29
<b>Şekil 3.2:</b> $x_1$ eksenine göre yükseltme sonrası yükseltilmiş uzay.....	30
<b>Şekil 3.3:</b> Yükseltilmiş Uzayın XY Düzlemine İzdüşümü (Yeni Olurlu Bölge) ..	31
<b>Şekil 3.4:</b> $x_2$ Eksenine Göre Yükseltme Sonrası Olurlu Bölge.....	32
<b>Şekil 3.5:</b> $x_2$ Eksenine Göre Yükseltme ve İzdüşüm Sonrası Olurlu Bölge.....	32
<b>Şekil 3.6:</b> 5 Adet Uç Noktaya Sahip Çokyüzlü.....	34
<b>Şekil 3.6:</b> a) $x_1$ Eksenine Göre Yükseltme ve İzdüşüm Algoritmasının Uygulanması Sonucu Oluşan Çokyüzlü .....	35
b) $x_2$ Eksenine Göre Yükseltme ve İzdüşüm Algoritmasının Uygulanması Sonucu Oluşan Çokyüzlü .....	35
<b>Şekil 3.7:</b> 6 Adet Uç Noktaya Sahip Çokyüzlü.....	36
<b>Şekil 3.8:</b> a) $x_1$ Eksenine Göre Yükseltme ve İzdüşüm Algoritmasının Uygulanması Sonucu Oluşan Çokyüzlü .....	36
b) $x_2$ Eksenine Göre Yükseltme ve İzdüşüm Algoritmasının Uygulanması Sonucu Oluşan Çokyüzlü.....	36
<b>Şekil 4.1:</b> $\tilde{\mathbf{x}}^T$ ve $\tilde{\mathbf{p}}^T$ vektörlerinin genel yapısı .....	38
<b>Şekil 4.2:</b> $\tilde{\mathbf{A}}\tilde{\mathbf{x}} \leq \tilde{\mathbf{b}}$ eşitsizliklerinin genel yapısı.....	41
<b>Şekil 4.3:</b> 1.4 MHz Bant Genişliğinde RB Genel Yapısı.....	45
<b>Şekil 4.4:</b> Farklı CQI değerlerinde SNR' a karşılık Veri Kapasitesi Grafiği .....	46
<b>Tablo 5.1:</b> K=8 Kullanıcı için Verilen Farklı Bit Yükleme Senaryoları.....	51
<b>Şekil 5.1:</b> Tablo 5.1' de Verilen Senaryolar için Toplam Verici Gücü.....	52
<b>Şekil 5.2:</b> Tablo 5.1'de Verilen Senaryolar için Ortalama İşlem Süresi.....	53
<b>Şekil 5.3:</b> Farklı Kullanıcı Sayısı (bkz. Tablo 5.4) için Harcanacak Toplam Güç .....	57
<b>Şekil 5.4:</b> Farklı Kullanıcı Sayısı için (bkz. Tablo 5.4) İşlem Süresi Grafiği.....	57



<b>Şekil 5.5:</b> K=8 ve Kullanıcıların Orantılı Oranlarının 1:1:1:1:1:1:1:1 Olduğu Durumda Farklı Güç Kısıtları Altında Toplam Veri Hızı.....	59
<b>Şekil 5.6:</b> K=8 ve Kullanıcıların Orantılı Oranlarının 1:1:1:1:1:1:1:1 Olduğu Durumda Farklı Güç Kısıtları Altında Kaynak Tahsisi Süresi .....	60
<b>Şekil 5.7:</b> K=8 ve Kullanıcıların Orantılı Oranlarının 4:4:2:2:1:1:1:1 Olduğu Durumda Farklı Güç Kısıtları Altında Toplam Veri Hızı.....	61
<b>Şekil 5.8:</b> K=8 ve Kullanıcıların Orantılı Oranlarının 4:4:2:2:1:1:1:1 Olduğu Durumda Farklı Güç Kısıtları Altında Kaynak Tahsisi Süresi .....	62
<b>Şekil 5.9:</b> Eşit Orantılı Oran Koşulunda ve Farklı Güç Limiti Kısıtlarında Kullanıcı Sayısına göre Bir Altçerçeve Süresinde Sistemin Toplam Veri Hızı.....	63
<b>Şekil 5.10:</b> Eşit Orantılı Oran Koşulunda ve Farklı Güç Limiti Kısıtlarında Kullanıcı Sayısına göre İşlem Süresi .....	64
<b>Şekil 5.12:</b> 10 km/saat hızla giden bir kullanıcının kanal özilinti fonksiyonu ....	68
<b>Şekil 5.13:</b> 50 km/saat hızla giden bir kullanıcının kanal özilinti fonksiyonu ....	69
<b>Şekil 5.14:</b> 100 km/saat hızla giden bir kullanıcının kanal özilinti fonksiyonu ..	70
<b>Şekil 5.15:</b> Kanalin Değişim Hızına Göre Kaynak Tahsisi Probleminin Çözüm Süresi.....	71
<b>Şekil 6.1:</b> $\mathbf{a}_i^T \mathbf{x} \leq b_i$ eşitsizliklerinin tanımladıkları bölgeler.....	82
<b>Şekil 6.2:</b> $\mathbf{a}_i^T \mathbf{x} \leq b_i, i=1,2,\dots,6$ için $\mathbb{R}^3$ ’ te tanımlı P Çokyüzlüsü. ....	83
<b>Şekil 6.3:</b> Dışbükey ve Dışbükey Olmayan Küme .....	84
<b>Şekil 6.4 :</b> Çokyüzlünün iç noktası,yüzlemi ve yüzlemsel noktası [6].....	85
<b>Şekil 6.5:</b> Uç noktalar (Vertices) ve Uç Işınlar (Extreme Rays) [6].....	88

## **SİMGELER VE KISALTMALAR DİZİNİ**

AWGN	: Additive White Gaussian Noise
BER	: Bit Error Rate
CSI	: Channel State Information
CQI	: Channel Quality Indicator
ECR	: Effective Code Rate
FDMA	: Frequency Division Multiple Access
(I)FFT	: (Inverse) Fast Fourier Transform
(I)LP	: (Integer) Linear Programming
LTE	: Long Term Evolution
M-QAM	: M-ary Quadrature Amplitude Modulation
MA	: Margin Adaptive
MCS	: Modulation-and-Coding Scheme
OFDM	: Orthogonal Frequency Division Multiplexing
OFDMA	: Orthogonal Frequency Division Multiple Access
PSD	: Power Spectral Density
QoS	: Quality of Service
RA	: Rate Adaptive
RB	: Resource Block
SINR	: Signal to Interference and Noise Ratio
SISO	: Single Input Single Output
SNR	: Signal to Noise Ratio
TUM	: Totally Unimodular Matrix
VehA	: ITU Vehicular-A Channel Model

## SÖZLÜK DİZİNİ

Aç gözlü	: Greedy
Alt çerçeve	: Subframe
Alt-kanal	: Sub-channel
Alt-taşıyıcı	: Sub-carrier
Ana	: Primal
Artık Eşitsizlik	: Redundant Inequality
Aşağı Yönlü	: Downlink
Birimsel	: Unimodular
Çiftleş	: Dual
Çoklu Ortam	: Multimedia
Çokyüzlü	: Polyhedron
Dallan ve Kes	: Branch and Cut
Dallan ve Sınırla	: Branch and Bound
Dışbükey	: Convex
Dışbükey Küme	: Convex Set
Dikgen	: Orthogonal
Doğrusal	: Linear
Doğrusal Programlama	: Linear Programming
Doğrusal Bağımsız	: Linearly Independent
Doğrusal Olmayan	: Nonlinear
Döngüsel Önek	: Cyclic Prefix
Düzlem Kesme	: Cutting Plane
Enbüyük	: Maximum
Enbüyütme	: Maximization
Enküçük	: Minimum
Enküçültme	: Minimization
Eniyi	: Optimum
Eniyileme	: Optimization
Eniyiye yakın	: Sub-Optimum
Evreyumlu	: Coherent
Gevşetme	: Relaxation

Gösterge	: Indicator
İç Nokta	: Interior Point
İlinti	: Correlation
İşlev	: Function
İzdüşüm	: Projection
Kaynak Blok	: Resource Block, RB
Kenar	: Edge
Kesme	: Cut
Kısıt	: Constraint
Kip Çözücü	: Demodulator
Kiplenim	: Modulation
Köşe (Uç) Nokta	: Vertex
Maliyet	: Cost (Objective)
Politop	: Polytope
Olurlu (Bölge)	: Feasible (Region)
Olurlu Çözüm	: Feasible Solution
Olursuz	: Infeasible
Örtük Eşitlik	: Implicit Equality
Simpleks	: Simplex
Tahsis	: Allocation
Tamsayı Programlama	: Integer Programming
(Tam) Kerte	: (Full) Rank
Tekil	: Singular
Temel Olurlu	: Basic Feasible
Uç Işın	: Extreme Ray
Uç Nokta	: Extreme Point
Uyarlamalı	: Adaptive
Uygunluk	: Fitness
Yinelemeli	: Recursive
Yükseltme	: Lifting
Yüz	: Face
Yüzlem	: Facet
Yüzlemsel Nokta	: Inner Point

# 1. GİRİŞ

Geçtiğimiz yıllar itibarı ile kablosuz haberleşme sistemlerinde önemli gelişmeler meydana gelmekte ve teknolojinin gelişmesi ile kablosuz haberleşme sistemlerine olan ilgi hızla artmaktadır. Bu gelişim ile birlikte özellikle veri iletişimi konusunda ihtiyaç her geçen gün artmaktadır. Kullanılan günlük uygulamalar ile birlikte yüksek veri iletim hızı ihtiyacı ortaya çıkmaktadır.

OFDM günümüzdeki haberleşme sistemleri içinde temel iletim yöntemlerinden biri haline gelmiştir (802.11a/g, ADSL, WiMAX). Gelecekte kullanılacak olan haberleşme sistemleri için de OFDM kullanımı desteklenmeye devam edilmektedir (örn. LTE-A aşağı yönlü iletimde).

OFDM tekniğinde mevcut geniş bant-aralığı birbirinden bağımsız dar bantlı alt-kanallara bölünerek çok taşıyıcılı kiplenim gerçekleştirilir. Dar bantlı bu alt-kanallar birbirlerine diktir. Bu nedenle alt-taşıyıcıların birbirleri arasında oluşabilecek çapraz karışım engellenmektedir. Alt-taşıyıcılar arası karışım engellendiği için tek taşıyıcılı bir sisteme göre daha basit yapıda kanal eşitleyicisi kullanılmaktadır. Alt-kanallarda düşük hızlarda veri taşınması rağmen toplamdaki veri hızı oldukça yüksek olabilmektedir.

Bir iletişim sisteminde kanalın ilinti bant genişliğinin mevcut bant genişliğinden dar olması durumunda kanal frekans seçici özellik gösterir. Bu durum mevcut kanal bant genişliğinin alt-kanalların sayısı kadar eşit parçaya bölünmesiyle giderilir ve artık bu alt-kanallar frekans seçici olmaktan çıkıp göreceli olarak frekans-düz sönümlenmeye maruz kalırlar.

Kablosuz haberleşme sistemlerinde ortak kanala ulaşmak isteyen kullanıcılar kanalı paylaşmak zorundadırlar. Çok kullanıcıli sistemlerde kanala erişim için farklı yöntemler kullanılmaktadır:

Frekans Bölüşümlü Çoklu Erişim (FDMA) [31], Zaman Bölüşümlü Çoklu Erişim (TDMA) [31], Kod Bölüşümlü Çoklu Erişim (CDMA) [31] [34] ve Uzay Bölüşümlü Çoklu Erişim (SDMA) [31] [34]. Bu tez kapsamında temel olarak FDMA ele alınacaktır. FDMA' de kanal, frekans alt-bantlarına bölünerek kullanıcılara paylaşılır.

maktadır. OFDM ve FDMA 'in birleştirilmesi Çoklu Erişimli Dikgen Frekans Bölüşümlü Çoğullama (OFDMA) olarak adlandırılmaktadır. OFDMA 'de kanal birbirine dikgen alt-kanallara bölünmekte ve alt kanallar belirli kurallara göre kullanıcılara paylaştırılmaktadır. Bu nedenle bir OFDM çerçevesinde birden fazla kullanıcının verisi taşınabilmektedir.

Kablosuz haberleşme sistemlerinde toplam iletim gücü ve toplam bantgenişliği sınırlı kaynaklardır. Bu nedenle çoklu erişimde kullanıcıların bu kaynaklardan verimli olarak faydalanabilmesi için bu kaynaklar kullanıcılara belirli kriterlere göre tahsis edilmektedir.

Statik kaynak tahsisi bilinen en basit kaynak tahsisi yöntemidir. Statik kaynak tahsisinde kullanıcıların sistemden faydalanabilmesi için FDMA veya TDMA kullanılmaktadır. Burada kullanıcıların sahip oldukları kanalın durumuna ve değişimine bakılmaksızın önceden belirlenmiş frekans bantlarında veya zaman aralıklarında iletişim yapılmaktadır. Statik kaynak tahsisi kullanıcıların kanal durumunu göz ardı ettiği için oldukça verimsiz bir kaynak tahsis yöntemi olarak bilinmektedir. Bununla birlikte, kullanıcıların sahip oldukları kanalın durumuna göre de kaynak tahsisi yapılabilir. Bu yöntem ise uyarlamalı kaynak tahsisi olarak adlandırılmaktadır.

Uyarlamalı kaynak tahsisinde kullanıcıların sahip oldukları kanalın durumuna göre kaynak tahsisi yapılmaktadır. Baz istasyonu ile kullanıcılar arasındaki ortam düşünüldüğünde çeşitli çevresel etkenlerden dolayı her bir kullanıcının sahip olduğu kanal birbirlerinden farklılık göstermektedir. Örneğin, bir kullanıcı bazı frekans bantlarında iyi kanal koşullarına sahipken başka bir kullanıcı aynı frekans aralığında kötü kanal koşullarına sahip olabilmektedir. Bu durumda kullanıcılara tahsis edilecek olan alt-kanalların ve iletim gücünün iyi bir planlama ile seçilmesi gerekmektedir. Bu şekilde sistemin performansı yükseltilebilmektedir.

Bu kaynaklar üzerinden kullanıcılara iletilecek bitlerin eniyi şekilde yüklenmesi uyarlamalı kaynak tahsisi problemini oluşturmaktadır. Bu problemde kullanıcılara tahsis edilecek alt-taşıyıcıların ve bitlerin tamsayı olması gerekliliğinden dolayı problem, tamsayı programlama problemi olarak değerlendirilmektedir. Tamsayı programlama problemlerinin çözülebilmesi için literatürde farklı algoritmalar ta-

nımlanmaktadır. Bu yöntemlerin işlem karmaşıklıklarının fazla oldukları bilinmektedir. Uyarlamalı kaynak tahsisi yapıldığında kaynak tahsisi probleminin en kısa sürede çözülmesi gerekmektedir. Bu nedenle mevcut tamsayı programlama yöntemlerinden farklı olarak alternatif çözümler üretilmektedir. Sunulan alternatif çözümler ile işlem karmaşıklığı azaltılmakta buna rağmen problemin eniyiye yakın bir çözüm üretmesi sağlanabilmektedir. Bu nedenle tez kapsamında önerilen yeni bir yöntem ile OFDMA kaynak tahsisi probleminin çözülmesi gerçekleştirilecektir.

Tezin ikinci bölümünde OFDMA sistemlerde kaynak tahsisi problemi matematiksel olarak ifade edilmiştir. Burada kaynak tahsisi problemi güç uyarlamalı (margin adaptive, MA) ve hız uyarlamalı (rate adaptive, RA) kaynak tahsis problemi olarak iki başlık altında incelenmiştir. Literatürde iki kaynak tahsisi problemi için önerilen yöntemler araştırılarak bilgi verilmiştir. Sonrasında Kaynak tahsis probleminin tamsayı kısıtları gevşetilerek problem ayrık doğasından çıkartılıp dışbükey hale dönüştürülmüş ve kaynak tahsis problemi matris ve vektörler ile ifade edilebilir bir yapıya dönüştürülerek doğrusallaştırılmıştır.

Üçüncü bölümünde, tamsayı programlama yöntemlerinden yükselt-ve-izdüşür algoritması hakkında bilgi verilmiştir. Tamsayı programlama algoritmaları ile kaynak tahsisi probleminin eniyi çözümü elde edilebilmektedir. Bu kapsamda doğrusal tamsayı algoritmalarından yükselt ve izdüşür algoritması kullanılarak kaynak tahsisi probleminin eniyi çözümü elde edilmeye çalışılacaktır. Ayrıca, yükselt-ve-izdüşür algoritmasını işlem karmaşıklığı azaltılarak çözmek için bir yöntem önerilmiştir.

Dördüncü bölümünde, kaynak tahsis probleminin doğrusal programlama algoritmaları ile tamsayı değerler verebilecek şekilde çözülebilmesi için yeni bir yöntem önerilmiştir. Bu yöntemde kaynak tahsisi probleminin çözümü için iki aşamalı bir çözüm önerilmiştir. Ayrıca benzetimlerde kullanılacak olan LTE sistemi ve parametreleri hakkında bilgi verilmiştir. Önerilen yöntem ile kaynak tahsisi probleminin eniyiye yakın çözümünün elde edilmesi amaçlanmıştır.

Tezin beşinci bölümünde LTE benzeri bir sisteme göre çeşitli benzetim çalışmaları yapılarak, sonuçları yorumlanarak tez tamamlanmıştır.



## 2. OFDMA SİSTEMLERDE KAYNAK TAHSİSİ

Yeni nesil haberleşme sistemlerinde OFDM, popüler bir teknik olarak öne çıkmaktadır. OFDM yönteminde mevcut olan geniş bant aralığı, çok sayıda birbirinden bağımsız dar bantlı alt-kanallara (subchannel veya subcarrier) bölünerek çok taşıyıcılı kiplenim gerçekleştirilmektedir. OFDM yönteminde alt taşıyıcılar birbirlerine diktir. Bu nedenle, alt taşıyıcıların birbirlerine oluşturabilecekleri girişimler ortadan kalkmaktadır. Alt taşıyıcıların birbirlerine girişim oluşturmaları ortadan kalktığı için, OFDM kullanılan sistemlerde dar bant aralıklarında daha yüksek veri hızı sağlanabilmektedir.

OFDMA sistemlerde, kullanıcıların baz istasyonlarına olan uzaklıklarına, hızlarına ve diğer çevresel faktörlere bağlı olarak, kullanıcılar aynı alt taşıyıcıda farklı kanal koşullarına sahip olmaktadır. Bu nedenle bazı alt-taşıyıcılarda, kullanıcılar daha iyi kanal olanaklarına sahipken, kalan diğer alt-taşıyıcılarda daha kötü kanal olanaklarına sahip olabilmektedirler. Sistemden faydalanan kullanıcılar için tanımlanmış olan servis kalitesinin (Quality of Service, QoS) ve veri hızının sağlanabilmesi için kullanıcılara bu alt taşıyıcılardan hangilerinin tahsis edileceği, tahsis edilen alt taşıyıcılara kaç bit yükleneceği ve ne kadar güç verilebileceğinin belirlenmesi kaynak tahsisi probleminin temelini oluşturmaktadır.

Bu bölümde anlatılacak olan OFDMA sistemi için servis kalitesi aşağıda tanımlanmış olan faktörlere göre belirlenmektedir:

- a) İletim bant genişliği (Transmission Bandwidth) : Kullanıcıya atanan bant genişliğine göre kullanıcının hangi hızda ve hangi kalitede iletişim hizmetinden faydalanacağı belirlenmektedir.
- b) Bit Hata Oranı (Bit Error Rate): Kullanıcının veri alışverişindeki ortalama hata olasılığını vermektedir. Kullanıcının faydalanacağı servisin türüne göre, kullanıcı için belirlenecek bit hata oranları farklılık göstermektedir. Örneğin veri iletişimi yapılacak ise düşük bit hata oranı, ses iletişimi yapılacak ise daha yüksek bit hata oranı belirlenebilmektedir.
- c) Servis Yararlanabilirliği (Availability of Service): İletişim sisteminden yararlanmak isteyen kullanıcı, sistemden belirli bir bant genişliği ve veri hızı talep etmektedir. Sistemin o anki şartlarına göre,

kullanıcının talebine cevap verip veremeyeceği, cevap veremediği durumda, kullanıcıyı beklemeye mi alacağı yoksa sistemden mi atacağı gibi durumlar kullanıcının sistemden yararlanabilirliğini belirlemektedir.

Literatürde temel olarak kaynak tahsisi problemleri eniyileştirilecek kaynağın türüne göre: Güç Uyarlamalı (Margin Adaptive, MA) [7], [8], [9] ,[10] ve Hız Uyarlamalı (Rate Adaptive, RA) [11],[15] olarak iki başlık altında tanımlanmaktadır.

Güç Uyarlamalı Kaynak Tahsisi problemi; kullanıcı için tanımlanan bit hızı ve bit hata oranı servis kalitesinin sağlandığı ve bir alt taşıyıcının birden fazla kullanıcı tarafından kullanılmasının önlendiği durum altında, toplam harcanan gücü enküçültmek olarak tanımlanmaktadır. Bir alt taşıyıcının birden fazla kullanıcıya tahsis edilmemesi kullanıcılar arasındaki karışımı engellemektedir [7].

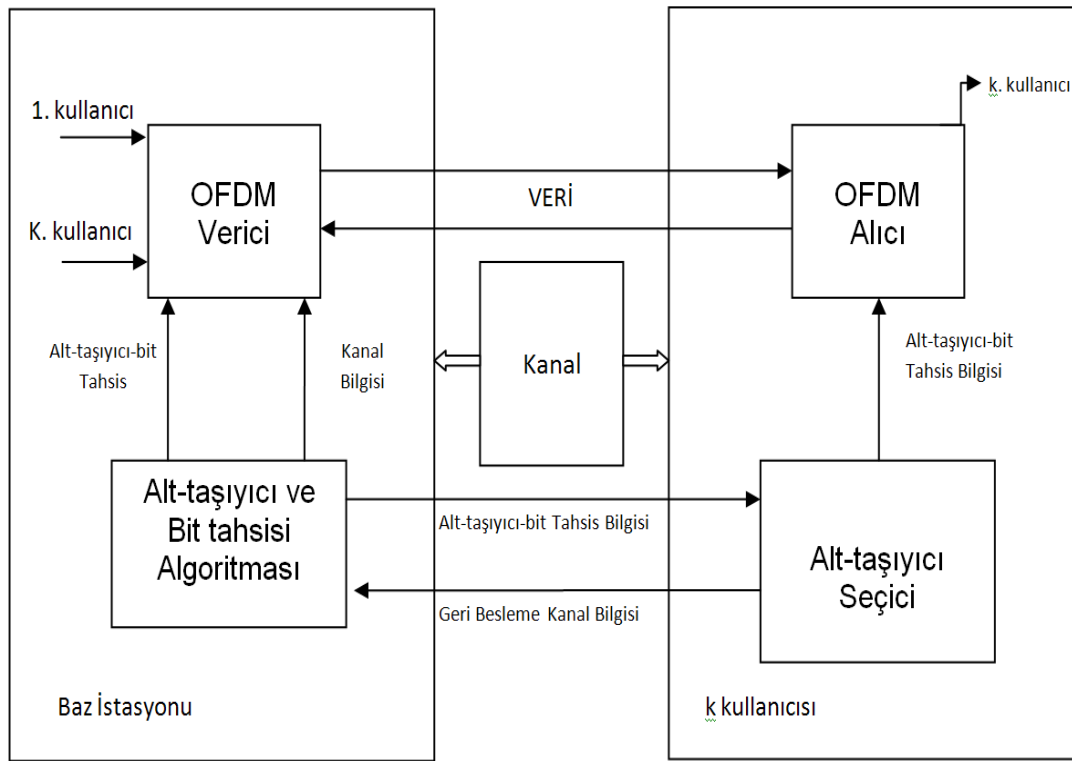
Hız Uyarlamalı Kaynak Tahsis problemi; kullanıcılar için belirlenen sınırlı güç altında, bit hata oranı ve kullanıcılar arası orantılı adalet (Proportional Fairness) [11] sağlanacak şekilde, kullanıcıların bit hızlarının enbüyütülmesi olarak tanımlanmaktadır. Burada da aynı şekilde bir alt taşıyıcının aynı anda birden fazla kullanıcıya tahsis edilmesi yasaklanmaktadır.

Kullanıcılar arası adaletin dikkate alındığı veya göz ardı edildiği sistem modelleri mevcuttur. MA kaynak tahsisini kullanan sistemler, MA problemin doğası gereği kullanıcılar açısından adil bir sistem yapısı oluşturmaktadır. Fakat RA kaynak tahsisinin yapıldığı sistem modellerinde kaynak tahsisinin adil olmayan biçimde yapıldığı durumlar bulunmaktadır [15].

Tanımlanan kaynak tahsisi problemlerinin en iyi çözümü, problemin tamsayı programlama algoritmaları; Dallan-Sınırla Algoritması (Branch-and-Bound) [3] ve Düzlem Kesme algoritmaları (Cutting-Planes) [3] gibi algoritmalar ile çözülmesiyle elde edilebilmektedir. Fakat bu algoritmalarının işlem karmaşıklığı çok yüksek olduğundan, pratik sistemlerde kullanımları pek mümkün olmamaktadır. Bu nedenle kaynak tahsisi problemlerinin çözümü için literatürde çok sayıda eniyeye yakın yöntem önerilmektedir. Önerilen yöntemlerde, genellikle problemin en iyi çözümünden ödün verilerek, problemin daha az karmaşık yapıda çözülmesine yönelik çözümler sunulmaktadır.

## 2.1 Sistem Yapısı ve Problem Tanımı

Şekil 2.1 'de verilen OFDMA sistem modelinde K tane kullanıcı ve bir adet verici (baz istasyonu) bulunmaktadır. Kullanıcılarda ve baz istasyonunda bir adet anten bulunmaktadır. Sistem için tüm alt- taşıyıcıların evre uyumlu bant genişliğinin (coherence bandwidth) kanalın evre uyumlu bant genişliğinden daha küçük olduğu varsayılmaktadır. Ayrıca baz istasyonunda kullanıcılara ait kanal bilgisinin eksiksiz olarak bilindiği varsayılmaktadır.



Şekil 2.1: OFDMA Sistem Modeli [13]

Tez kapsamında kullanılacak OFDMA sistemini daha iyi anlayabilmek için aşağıdaki tanımlamalar yapılmaktadır.

K : Sistemde bulunan toplam kullanıcı sayısı

N : Sistemde tanımlı toplam alt-taşıyıcı sayısı

$c_{k,n}$  : k-ıncı kullanıcıya tahsis edilen n-inci alt-taşıyıcıya yüklenecek bit sayısı

$c_{k,n} \in D = \{0,1,2,\dots,M\}$  olarak ifade edilmektedir. Klasik bir OFDMA sistemde uyarlamalı kipleyci (adaptive modulator),  $D = \{0,1,2,\dots,M\}$  kümesi içerisinde değerlere

sahip olmaktadır. Bu nedenle bir alt-taşıyıcıya yüklenebilecek en fazla bit sayısı “M” ile sınırlandırılmaktadır. Sistem genelinde bir alt-taşıyıcının birden fazla kullanıcı tarafından kullanılması yasaklandığı için k kullanıcısı dışındaki kullanıcılar için  $c_{k,n} = 0$  (tüm  $k' \in (1,2,\dots,K)$  ve  $k' \neq k$  değerleri için) olmaktadır.

k-ıncı kullanıcının bir OFDM sembol süresinde sahip olduğu veri hızı  $R_k$ , kullanıcıya tahsis edilen alt-taşıyıcılara yüklenen bitlerin toplanması ile elde edilir.

$$R_k(c_{k,n}) = \sum_{n=1}^N c_{k,n}, \quad k \in (1,2,\dots,K) \quad (2.1)$$

Yukarıda verilen tanımlamalar sonrasında, OFDMA sistem yapısında baz istasyonu ve kullanıcılarda gerçekleşen işlemler genel olarak aşağıdaki gibi tanımlanmaktadır.

Baz istasyonunda, kitleyici ile modüle edilmiş olan kompleks değerlere sahip semboller, kitleyici sonrasında Ters Hızlı Fourier Dönüşümü (Inverse Fast Fourier Transform, IFFT) alınarak zaman bölgesine dönüştürülmekte ve her bir OFDM çerçevesi için döngüsel önek (cyclic prefix) eklenmektedir. Döngüsel önekin eklenmesiyle alt-taşıyıcılar arasındaki dikgenlik (orthogonality) sağlanarak, alt-taşıyıcıların birbirlerine olabilecek girişimleri ortadan kaldırılmaktadır.

Alıcı tarafında, tek bir anten ile alınan OFDM sembolünün döngüsel öneki atılmakta ve sonrasında zaman bölgesinde olan sinyallerin Hızlı Fourier Dönüşümleri (Fast Fourier Transform, FFT) alınarak, alt-taşıyıcılardaki sinyaller elde edilmektedir. Elde edilen bu sinyaller kullanıcı için tanımlı olan kaynak tahsis bilgisine göre kip çözücülerden geçirilerek, kip çözülmüş bitler elde edilmektedir.

Anlatılan OFDMA sistemde frekans seçici bir kanal bulunduğu varsayılmaktadır. Frekans seçici kanallarda alt-taşıyıcılar, her bir kullanıcı için farklı sönümlere maruz kalmaktadır. Bu nedenle her bir kullanıcı için hesaplanan kanal katsayıları birbirlerinden bağımsız ve farklı olmaktadır. Bundan sonra anlatılacak kısımlar için k-ıncı kullanıcının n-inci alt-taşıyıcı için sahip olduğu kanal katsayısı,  $\alpha_{k,n}$  ile gösterilecektir. Kullanılan OFDMA sistemde, kanalın tek-yanlı izgesel güç yoğunluğu (Power Spectral Density, PSD),  $N_0 = 1$  olarak kabul edilmektedir.

Bir alt-taşıyıcı için, kanal katsayısının  $\alpha_{k,n} = 1$  olduğu durumda, k-ıncı kullanıcının n-inci alt-taşıyıcıda taşınan  $c_{k,n} = m$  bitlik veriyi belirli bir bit hata oranını sağlaya-

cak şekilde alabilmesi için harcanması gereken güç  $f_k(m)$ ,  $k \in (1,2,\dots,K)$  ile ifade edilmektedir. Burada  $f_k(m)$ 'nin alabileceği değerler, kullanıcılara bağlı olarak değişebildiği için farklı kullanıcılar için farklı QoS gereksinimleri tanımlanabilmektedir.

Kullanıcılar için belirlenmiş QoS 'ın sağlanabilmesi için, baz istasyonunda k-ncı kullanıcıya atanan n-inci alt-taşıyıcı için harcanması gereken güç  $p_{k,n}$  ile ifade edilmektedir. k-ncı kullanıcı için harcanan toplam güç ise  $P_k$  ile gösterilmektedir.

$$p_{k,n} = \frac{f_k(c_{k,n})}{\alpha_{k,n}^2}, \quad k \in (1,2,\dots,K) \text{ ve } n \in (1,2,\dots,N) \quad (2.2)$$

$$P_k = \sum_{n=1}^N p_{k,n} = \sum_{n=1}^N \frac{f_k(c_{k,n})}{\alpha_{k,n}^2}, \quad k \in (1,2,\dots,K) \text{ ve } n \in (1,2,\dots,N) \quad (2.3)$$

$c_{k,n} \in D = \{0,1,2,\dots,M\}$  olarak verilmektedir.

Yukarıda anlatılan OFDMA mimarisi altında literatürde genel olarak kaynak tahsisi problemleri, farklı maliyet işlevlerine göre ikiye ayrılmaktadır.

### 2.1.1 MA Problem Tanımı

MA OFDMA kaynak tahsisi eniyileme problemi aşağıdaki şekilde tanımlanmaktadır [7].

$$\begin{aligned} \text{Maliyet işlevi} \quad & \min_{c_{k,n} \in D} P_T = \sum_{k=1}^K P_k(c_{k,n}) \\ \text{Kısıtlar} \quad & R_k(c_{k,n}) \geq r_k, \quad \text{tüm } k \in (1,2,\dots,K) \text{ değerleri için} \\ & \text{eğer } c_{k,n} \neq 0 \text{ ise } c_{k',n} = 0, \quad \text{tüm } n \text{ ve tüm } k' \neq k \text{ değerleri için} \\ & c_{k,n} \in D = \{0,1,2,\dots,M\} \end{aligned} \quad (2.4)$$

$R_k(c_{k,n})$  (2.1)' de,  $P_k(c_{k,n})$  ise (2.3)' te verilmektedir.

(2.4) 'de tanımlanan MA problemi, tüm kullanıcılara veya alt-taşıyıcılara harcanan toplam iletim gücünü azaltmayı amaçlamaktadır. Problemin ilk kısıtı ile bir OFDM sembolü süresince tüm kullanıcılara en az  $r_k$  bit iletim yapılması garanti edilmektedir. İkinci kısıt ise bir alt-taşıyıcının birden fazla kullanıcı tarafından kullanılmasını engellemek amacı ile kullanılmaktadır. Son kısıtta ise alt-taşıyıcılara yüklenecek

bit sayısının tam sayı değerler almak zorunda olduğu belirtilmektedir. Bu nedenle (2.4)' de verilen problem, Tamsayı Programlama Problemi olarak görülmektedir.

Tanımlanan problem için amaç fonksiyonu doğrusal olmadığından, problemin mevcut tamsayı programlama algoritmaları (Dallan-Sınırla veya Dallan-Kes algoritmaları) [3],[4] ile çözümü oldukça zordur.

### 2.1.2 RA Problem Tanımı

RA OFDMA kaynak tahsisi probleminin tanımı aşağıda verilmektedir.

$$\text{Maliyet işlevi } \max R_T = \sum_{k=1}^K R_k(c_{k,n})$$

$$\begin{aligned} \text{Kısıtlar} \quad & \sum_{k=1}^K P_k(c_{k,n}) \leq P_T \\ & R_k \geq r_k = \gamma_k R_T, \text{ tüm } k \in (1,2,\dots,K) \text{ değerleri için} \\ & \text{eğer } c_{k,n} = 1 \text{ ise } c_{k',n} = 0, \text{ tüm } n \text{ ve tüm } k' \neq k \text{ değerleri için} \\ & c_{k,n} \in D = \{0,1,2,\dots,M\} \end{aligned} \quad (2.5)$$

Burada kısıt kümesinde toplam harcanan iletim gücü, üst eşik değer olan  $P_T$  ile sınırlandırılmaktadır. Bu durum altında amaç tüm kullanıcıların toplam veri hızını enbüyütmektir. RA problemlerde kullanıcıların veri hızları için bir alt sınır belirlenmezse kullanıcılar arası adalet (fairness) ihlal edilmiş olur. Öyle ki, baz istasyonuna yakın olan kullanıcının veri hızı, uzak olan kullanıcının veri hızından daha fazla olabilmektedir.(Baz istasyonuna yakın kullanıcının kanal koşulları uzak olan kullanıcıya göre daha iyi olduğundan bu durum meydana gelebilmektedir.) RA problemleri için adalet sorununu ortadan kaldırmaya yönelik literatürde orantılı adalet (proportional fairness) kavramı önerilmekte ve aşağıdaki şekilde tanımlanmaktadır

$$r_1 : r_2 : \dots : r_k = \gamma_1 : \gamma_2 : \dots : \gamma_k \text{ tüm kullanıcılar için} \quad (2.6)$$

(2.6) ' da verilen  $\gamma_k$  orantı sabiti olarak tanımlanmakta ve tüm kullanıcılar için orantı sabitin toplamı 1' e eşit olmaktadır.

$$\sum_{k=1}^K \gamma_k = 1 \text{ ve } R_T = \sum_{k=1}^K R_k \text{ için } r_k = \gamma_k R_T \text{ olarak verilmektedir.} \quad (2.7)$$

(2.7)' de  $R_T$  tüm kullanıcıların veri hızı toplamını yani RA problem için enbüyütülecek amaç fonksiyonunu tanımlamaktadır.  $r_k$  ise k-ıncı kullanıcıya sağlanması gereken en küçük veri hızını tanımlamaktadır [10].

(2.5) 'de tanımlanan RA probleminde, ilk kısıt toplam iletim gücünün sınırlandırılmasını garanti altına almaktadır. İkinci kısıt ise kullanıcıların veri orantı sabitlerine göre veri hızının belirli bir eşik değerin üzerinde olmasını sağlamaktadır. RA probleminin son iki kısıtı ise MA problemindeki kısıtlar ile aynıdır.

Tanımlanan MA problemi için Wong, problemin tamsayı kısıtlarını gevşeterek problemi dış bükey hale getirmiştir. Problemin gevşetilmesi için  $c_{k,n}$ 'in alabileceği değerler tamsayı olmaktan çıkmakta ve  $[0, M]$  arasındaki gerçek sayı değerlerine dönüşmektedir. Bu nedenle, problemin gevşetilmesi sonrasında elde edilecek çözümler için bir alt-taşıyıcı birden fazla kullanıcıya tahsis edilebilir hale gelmektedir. Gevşetilmiş problem çözümünde alt-taşıyıcıların hangi kullanıcı tarafından ne oranda kullanılacağı  $\rho_{k,n} \in [0, 1]$  ile gösterilmektedir. Burada tüm  $k \in \{1, 2, \dots, K\}$  de-

ğerleri için  $\sum_{k=1}^K \rho_{k,n} = 1$  durumu sağlanmalıdır.  $\rho_{k,n}$  ile ilgili olarak aşağıdaki tanımlamalar yapılmaktadır:

$\rho_{k,n} = 0$  ise k 'ıncı kullanıcı n 'inci alt-taşıyıcı kullanmamaktadır.

$\rho_{k,n} = 1$  ise k 'ıncı kullanıcı n 'inci alt taşıyıcıyı tek başına kullanmaktadır.

$\sum_{k \in K'} \rho_{k,n} = 1$  ise n 'inci alt-taşıyıcı,  $k \in K' \subseteq K$  olan  $K'$  kümesi içerisinde yer

alan kullanıcılar tarafından paylaşılmaktadır.

Yukarıda verilen tanımlamalara göre Wong [7], gevşetilmiş problemi aşağıdaki gibi tanımlamıştır.

$$\text{maliyet işlevi} \quad P_T = \min_{\substack{c_{k,n} \in [0, M] \\ \rho_{k,n} \in [0, 1]}} \sum_{k=1}^K \sum_{n=1}^N \frac{\rho_{k,n}}{\alpha_{k,n}^2} f_k(c_{k,n})$$

$$\begin{aligned} \text{Kısıtlar} \quad & \sum_{n=1}^N \rho_{k,n} c_{k,n} = R_k, \quad \text{tüm } k \in \{1, 2, \dots, K\} \text{ değerleri için} \\ & \sum_{k=1}^K \rho_{k,n} = 1, \quad \text{tüm } n \in \{1, 2, \dots, N\} \text{ değerleri için} \end{aligned} \quad (2.8)$$

Gevşetilmiş problemde (2.9) 'de verilen durum sağlandığında gevşetilmiş problem ile (2.4)' de verilen orijinal problem birbirlerine eşit olmaktadır.

$$\rho_{k,n} = \begin{cases} 1 & , \text{ eğer } c_{k,n} \neq 0 \text{ ve } k \in \{1,2,\dots,K\}, n \in \{1,2,\dots,N\} \\ 0 & , \text{ eğer } c_{k,n} = 0 \text{ ve } k \in \{1,2,\dots,K\}, n \in \{1,2,\dots,N\} \end{cases} \quad (2.9)$$

Gevşetilmiş problemde  $\rho_{k,n} \in [0,1]$  arasında değerler alabildiği için, gevşetilmiş problemin tanımlı olduğu olurlu uzay, orijinal problemin tanımlı olduğu olurlu uzaydan daha büyüktür. Daha geniş olurlu uzay için problemin çözümü en iyi sonucu vermektedir. Bu nedenle, gevşetilmiş problemin çözümü, bundan sonraki bölümler için en iyi çözüm olarak kabul edilecektir. Aynı zamanda diğer en iyiye yakın problem çözümleri için bir üst sınır teşkil edecektir.

(2.8) 'de verilen gevşetilmiş kaynak tahsisi problemini, dış bükey hale getirmek için  $r_{k,n} = \rho_{k,n} c_{k,n}$  ,  $k \in \{1,2,\dots,K\}$ ,  $n \in \{1,2,\dots,N\}$  şeklinde yeni bir değişken tanımlanmaktadır [7]. Buna göre dış bükey kaynak tahsis problemi aşağıdaki gibi verilmektedir.

$$\begin{aligned} \text{maliyet işlevi} \quad P_T &= \min_{\substack{r_{k,n} \in [0, \rho_{k,n} M] \\ \rho_{k,n} \in [0,1]}} \sum_{k=1}^K \sum_{n=1}^N \frac{\rho_{k,n}}{\alpha_{k,n}^2} f_k \left( \frac{r_{k,n}}{\rho_{k,n}} \right) \\ \text{Kısıtlar} \quad \sum_{n=1}^N r_{k,n} &= r_k, \quad \text{tüm } k \in \{1,2,\dots,K\} \text{ değerleri için} \\ \sum_{k=1}^K \rho_{k,n} &= 1, \quad \text{tüm } n \in \{1,2,\dots,N\} \text{ değerleri için} \end{aligned} \quad (2.10)$$

Wong [7], (2.10) 'da verilen doğrusal olmayan dış bükey kaynak tahsisi problemini doğrusal olmayan programlama yöntemleri ile çözmektedir. Problemin çözümü iki aşamalı olarak gerçekleştirilmiştir. İlk olarak, alt-taşıyıcıların kullanıcılara tahsisi yapılmaktadır. Burada eğer n-inci alt-taşıyıcı birden fazla kullanıcıya tahsis edilmiş ise, n-inci alt-taşıyıcı için en büyük  $\rho_{k,n}$  değerine sahip k-ıncı kullanıcıya tahsis edilmektedir. Buna göre  $\rho_{k,n} = 1$  olarak değiştirilmektedir. İlk etapta n-inci alt-taşıyıcıya atanan diğer kullanıcılar için  $\rho_{k',n} = 0$ ,  $k' \neq k$  ve  $k' \in \{1,2,\dots,K\}$  olarak değiştirilmektedir. Böylece bu kullanıcılar n-inci alt-taşıyıcıyı kullanamamaktadır. Bu sayede bir alt-taşıyıcı yeniden en fazla bir kullanıcıya atanabilir hale gelmektedir.



Bu işlem tüm alt-taşıyıcılar için gerçekleştirilmektedir. Bu işlem sonunda kullanıcılar için alt-taşıyıcı tahsisi tamamlanmaktadır. Kullanıcılara artık alt-taşıyıcılar tahsis edildiği için ilk aşama böylece tamamlanmaktadır.

İkinci aşamada, her bir kullanıcı için tahsis edilen alt-taşıyıcılara bit ve güç yüklemesi yapılmaktadır. Bu yükleme işlemi aç gözlü algoritma (greedy algorithm) kullanılarak gerçekleştirilmektedir [7].

Aç gözlü algoritma ile her iterasyonda kullanıcıya tahsisli tüm alt-taşıyıcılara bir bit eklendiği durum için; ihtiyaç duyulan iletim gücünü hesaplamaktadır. Burada en düşük güce ihtiyaç duyan alt-taşıyıcının bir bit daha taşımaya izin verilmektedir. Bu işlem k-ıncı kullanıcı için  $R_k$  defa tekrar etmektedir. k-ıncı kullanıcıya tahsis edilen alt-taşıyıcıların kümesi  $N_k$  olduğu durumda, Aç gözlü algoritma aşağıda verilmektedir.

$n \in N_k$  için  $c_{k,n} = 0$  ve  $\Delta P_n = \frac{[f_{k,n}(1) - f(0)]}{\alpha_{k,n}^2}$  olarak tanımlanır.

Sonrasında aşağıdaki işlemler  $R_k$  defa tekrar edilir:

$$\begin{aligned} \hat{n} &= \underset{n}{\operatorname{argmin}} \Delta P_n & (2.11) \\ c_{k,\hat{n}} &= c_{k,\hat{n}} + 1 \\ \Delta P_n &= \frac{[f_{k,n}(c_{k,\hat{n}} + 1) - f(c_{k,\hat{n}})]}{\alpha_{k,n}^2} \end{aligned}$$

(2.11)' da verilen işlemler tüm kullanıcılar için yapıldığında ikinci aşama, yani bit tahsisi işlemi tamamlanmaktadır. Böylece Wong' un önerdiği yöntem ile en iyiye yakın bir çözüm elde edilmektedir. Fakat burada en iyiye yakın çözüm elde edilmesine rağmen, (2.8) 'de tanımlanan kaynak tahsisi problemi doğrusal olmadığından doğrusal olmayan eniyileme yöntemleri ile çözülebilmektedir. Bu yöntemlerin işlem karmaşıklığı yüksek olduğundan algoritma oldukça yavaş çalışmaktadır.

Literatürde önerilen çeşitli diğer yöntemler aşağıda verilmektedir.

Mao, OFDMA kaynak tahsisi problemini matris ve vektörler şeklinde ifade edip hem MA Problemi hem de RA Problemi tanımlamıştır [10]. Matris yapıda tanımladığı problemi Dallan ve Sınırla (Branch-and-Bound) [4], [5] tamsayı programlama algoritması ile çözmüştür.

Turgu, MA problemi dış bükey hale dönüştürdükten sonra, problemi matris ve vektörler ile tanımlamıştır [13]. Dış bükey tanımlanan problemi doğrusal programlama algoritması olan Simpleks Algoritması [2],[4] (Simplex Algorithm) ile çözmüş ve kullanıcıların alt-taşıyıcı tahsisini gerçekleştirmiştir. Sonrasında her kullanıcı için Aç gözlü algoritmayı kullanarak bit ve güç tahsisini gerçekleştirmektedir. Bu şekilde makul sürelerde çalışan en iyiye yakın bir çözüm elde etmektedir.

Zhang [18], tarafından önerilen MA algoritma ilk olarak kullanıcıları, kanalın durumuna bağlı olarak kendi uzamsal ilinti değerlerine göre gruplara ayırmaktadır. Sonrasında aynı grupta bulunan kullanıcıların aynı alt-taşıyıcıyı kullanması yasaklanmaktadır. Buna göre yapılan alt-taşıyıcı ataması sonrasında, her kullanıcı için Aç gözlü algoritma ile bit ve güç tahsisi yapılmaktadır.

Wong [14], tarafında önerilen RA algoritmasının da ilk olarak her kullanıcıya atanacak alt-taşıyıcıların sayısı belirli bir orana göre belirlenmektedir. Belirlenen alt-taşıyıcı sayısına göre Aç gözlü algoritma çalıştırılarak kullanıcılara alt-taşıyıcılar tahsis edilmektedir. Daha sonra her bir kullanıcıya ayrılacak toplam iletim gücü, başta verilen kullanıcı oranlarına göre belirlenmekte ve kullanıcıların alt-taşıyıcıları arasında toplam veri hızını arttıracak şekilde dağıtılmaktadır.

Ermolova [19] , tarafından kullanılan algoritma RA kategorisindedir. Burada Dardari[20], tarafından OFDMA sistemler için önerilen Sıralı Alt-taşıyıcı Seçme algoritması (Ordered Subcarrier Selection Algorithm, OSSA) kullanılmıştır. Bu algoritmada tüm alt-taşıyıcılar içerisinde en iyi performansa sahip alt-taşıyıcılar seçilerek, her iterasyonda rastgele kullanıcılara atanmaktadır. OSSA algoritması kullanıcı sayısı ve alt-taşıyıcı sayısı fazla olan sistemlerde bit hata oranı (BER) arttığı için algoritmanın başarımı düşmektedir.

Wonjong Rhee [21], ilk olarak problemi dış bükey probleme dönüştürmektedir. Sonrasında alt-taşıyıcılara eşit güç tahsisi yapılmasını sağlayarak en iyiye yakın çözümü elde etmektedir.

Kıvanç [22] ,tarafından önerilen MA algoritmasında ilk olarak kullanıcıların sinyal gürültü oranlarına (Signal to Noise Ratio, SNR) göre her kullanıcıya tahsis

edilecek alt-taşıyıcı sayısını belirlemektedir. Belirlenen alt-taşıyıcı sayısına göre kullanıcılara alt-taşıyıcılar tahsis edilmekte ve oran ya da genliğe göre farklı iki Aç gözlü algoritma için bit ve güç ataması yapmaktadır.

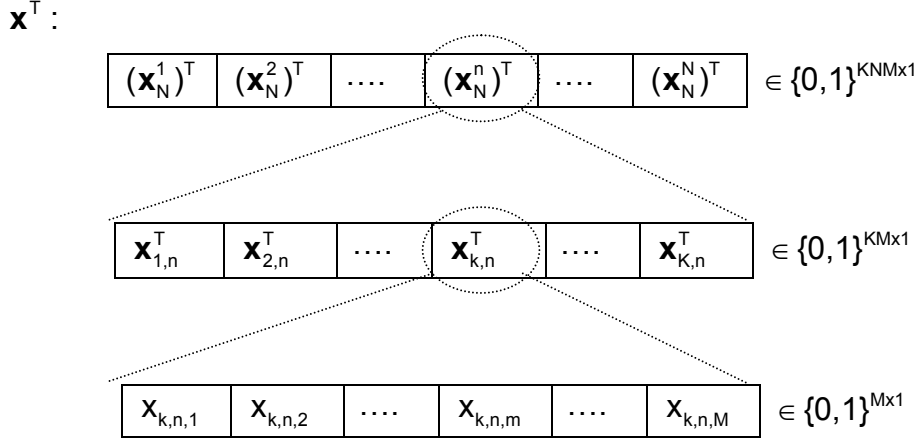
Bunlar dışında Munz [24], çoklu su doldurma algoritması geliştirerek kaynak tahsisini gerçekleştirmiş, Wang[25], Tang ve Reddy ise çeşitli evrimsel algoritmalar kullanarak kaynak tahsisi problemini çözmüşlerdir.

Literatürde yer alan algoritmalara bakıldığında çok farklı yöntemler ile OFDMA sistemler için kaynak tahsisi yapılabilmektedir. Fakat genel olarak önerilen algoritmalarda problemin çözümü iki aşamalı olarak gerçekleştirilmektedir. İlk olarak kullanıcılara alt-taşıyıcıların atanması yapılmakta, sonrasında yapılan alt-taşıyıcı atamalarına göre kullanıcı bazında bit ve güç dağıtımı yapılmaktadır. Burada alt-taşıyıcı ve bit-güç dağıtımı birlikte yapılmadığından önerilen yöntemlerin çoğu en iyi çözümü vermemektedir. En iyi çözümü veren algoritmalar ise çok fazla işlem karmaşıklığına sahip oldukları için gerçek zamanlı sistemlerde kullanılmaları çok zor görünmektedir.

Tezin bundan sonraki bölümünde OFDMA sistemler için en iyi çözüme yakın sonuç elde edebilmek için bir çözüm önerilmiştir. Bunun yanında işlem karmaşıklığını azaltmaya yönelik yapılan çalışmalar da anlatılmaktadır.

## 2.2 Problemin Doğrusallaştırılması

Bir önceki bölümde (2.4) 'de verilen MA probleminin amaç işlevinde ve (2.5) ' de verilen RA probleminin ilk kısıtında yer alan güç fonksiyonu  $f_k(c_{k,n})$  doğrusal olmayan bir fonksiyondur. (2.4) ve (2.5)' te verilen problem tanımlarında  $c_{k,n}$  tamsayı değerler almaktadır. Problemin tamsayı kısıtlarının gevşetilmesi ile  $c_{k,n}$  gerçek sayı değerleri almaktadır. Alt-taşıyıcılarda harcanması gereken güç miktarı, verilen  $c_{k,n}$  değerine ve bit hata oranına göre belirlenmektedir. Buna göre verilen tüm  $c_{k,n}$  değerlerine karşılık harcanması gereken güç miktarı  $\rho_{k,n,m}$  hesaplanmaktadır. Tüm  $c_{k,n}$  ve  $\rho_{k,n,m}$  değerleri kullanılarak,  $\mathbf{x}$  alt-taşıyıcı-kullanıcı-bit vektörü ve  $\mathbf{p}$  güç vektörü tanımlanmaktadır [10]. Şekil 2.2' de görüldüğü gibi  $\mathbf{x}^T$  vektörü  $K * N$  adet  $M \times 1$  boyutunda olan alt vektörden meydana gelmektedir.



**Şekil 2.2:**  $\mathbf{x}^T$  bit tahsisi vektörünün yapısı

Bu  $M \times 1$  boyutlu  $\mathbf{x}_{k,n}^T$  vektörü alt-taşıyıcı-bit vektörü olarak adlandırılmakta ve  $k$ -inci kullanıcının,  $n$ -inci alt-taşıyıcı ile taşıyabileceği bit sayısını ifade etmektedir.  $\mathbf{x}_{k,n}$  vektörünün tek bir elemanı “1” değerine sahip olabilmekte, geriye kalan  $(M-1)$  adet eleman ise “0” değerini almaktadır.  $\mathbf{x}_{k,n}$  vektörü içerisinde “1” değerine sahip elemanın indis değeri  $n$ -inci alt-taşıyıcıya yüklenecek bit sayısını vermektedir. Örneğin,  $\mathbf{x}_{k,n}$  vektörünün  $m$ -inci elemanı “1” değerine sahip ise  $n$ -inci alt-taşıyıcıya  $m$  bit atandığı ifade edilmektedir.  $\mathbf{x}_{k,n}$  vektörünün tüm elemanları “0” ise,  $k$ -inci kullanıcı için  $n$ -inci alt-taşıyıcı ile bit taşınmadığı gösterilmektedir. Böylece  $\mathbf{x}_{k,n}$  vektörünün alabileceği değerler kümesi:

$$\mathbf{x}_{k,n} \in \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right\}_{M \times (M+1)}$$

olmaktadır.

Bu şekilde  $n$ -inci alt-taşıyıcı için  $K$  adet  $\mathbf{x}_{k,n}$  vektörü tanımlanmakta ve bu vektörler sıralı bir şekilde bir araya gelerek  $\mathbf{x}_N^n$  alt-taşıyıcı-kullanıcı-bit vektörünü oluşturmaktadır. Tüm  $k \in K$  değerleri için,  $\mathbf{x}_N^n$  vektörünü oluşturan  $\mathbf{x}_{k,n}$  vektörlerinden bir tanesi “1” değerine sahip eleman içeriyorsa, geriye kalan  $K-1$  adet

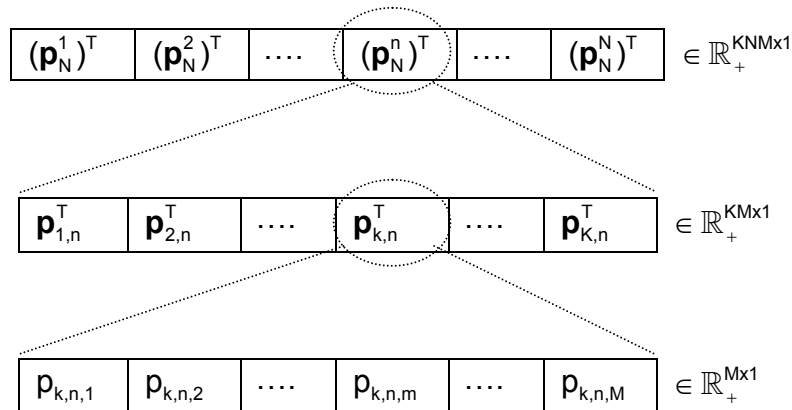
, tüm  $k_i \in \{1,2,\dots,K\}$ ,  $k_i \neq k$  için  $\mathbf{x}_{k_i,n}$  vektörünün elemanları "0" olmaktadır. Yani  $\mathbf{x}_N^n$  vektörü içerisinde en fazla bir adet "1" değerine sahip eleman bulunmaktadır. Bunun sonucunda n-inci alt-taşıyıcı sadece k-ıncı kullanıcıya tahsis edilmiş olmaktadır. Benzer şekilde n-inci alt-taşıyıcının hiçbir kullanıcıya atanmadığı durumda,  $\mathbf{x}_N^n$  vektörünün sahip olduğu tüm elemanlar "0" olmaktadır. Böylece  $\mathbf{x}_N^n$  vektörünün alabileceği değerler kümesi

$$\mathbf{x}_N^n \in \left\{ \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \right\}_{K \times M \times (K \times M + 1)}$$

olmaktadır.

OFDMA sisteminde bulunan tüm  $n \in \{1,2,\dots,N\}$  alt-taşıyıcıları için toplam olarak N adet  $\mathbf{x}_N^n$  vektörü tanımlanmaktadır. N adet  $\mathbf{x}_N^n$  vektörünün birleştirilmesi ile  $\mathbf{x}$  bit-tahsis vektörü oluşturulmaktadır. Olurlu bir çözüm için  $\mathbf{x}$  vektörünün en fazla N adet elemanı "1" değerini alabilmektedir. Bu durumda  $\mathbf{x}$  vektörünün kalan diğer  $(K \times M \times N - N)$  adet elemanı ise "0" değerini almaktadır.

Toplam verici gücünün elde edilebilmesi için  $\mathbf{x}$  bit-tahsis vektörüne karşılık harcanacak güçlerin yer aldığı güç vektörü  $\mathbf{p}$  olarak tanımlanmaktadır.  $\mathbf{p}$  vektörünün genel yapısı Şekil 2.3 ' de verilmektedir.



**Şekil 2.3:** Güç Tahsisi Vektörünün Genel Yapısı

Şekil 2.3' e bakıldığında,  $\mathbf{p}$  vektörünün genel parçalama yapısının Şekil 2.2' de verilen  $\mathbf{x}$  vektörünün genel parçalama yapısı ile aynı olduğu görülmektedir. Burada n-inci alt-taşıyıcı da k-ıncı kullanıcının m bit gönderebilmesi için harcanacak güç  $\rho_{k,n,m}$ ' nin değeri:

$$\rho_{k,n,m} = \frac{f_k(m, P_{e,k})}{\alpha_{k,n}^2} \quad (2.12)$$

$$f_k(m, P_{e,k}) = \frac{N_0}{2} \left[ Q^{-1} \left( \frac{P_{e,k}}{4} \right) \right]^{-1} (2^m - 1)$$

olarak verilmektedir. Burada  $\alpha_{k,n}$  k-ıncı kullanıcının n-inci alt-taşıyıcıda maruz kaldığı kanal transfer fonksiyonunun genlik değeridir. Kanal katsayısı  $\alpha_{k,n}$  değerinin büyük olması iyi bir kanalı işaret eder. Yüksek katsayıya sahip bir kanaldan, daha küçük bir katsayıya sahip kanala göre aynı sayıda bit, daha az güç kullanarak gönderilebilmektedir.

Tüm  $m \in \{1, 2, \dots, M\}$  bit değerlerine göre k-ıncı kullanıcı için n-inci alt-taşıyıcıda harcanması gereken güç değerleri alt-taşıyıcı-kullanıcı güç vektörü  $\mathbf{p}_{k,n}$  içerisinde yer almaktadır. Bu nedenle  $\mathbf{p}_{k,n}$  vektörünü  $\rho_{k,n,m}$  değerlerinden oluşmaktadır. n-inci alt-taşıyıcı için tüm kullanıcıların  $\mathbf{p}_{k,n}$  vektörlerinin birleştirilmesiyle maliyet işlevinin alt-taşıyıcı güç vektörü  $\mathbf{p}_N^n$  elde edilmektedir. N adet alt-taşıyıcı için tanımlı olan tüm  $\mathbf{p}_N^n$  alt-taşıyıcı-güç vektörlerinin bir araya getirilmesi ile maliyet işlevinin güç vektörü  $\mathbf{p}$  oluşturulmaktadır.  $\mathbf{x}$  ve  $\mathbf{p}$  vektörlerinin tanımlanması ile birlikte maliyet işlevi  $\mathbf{p}^T \mathbf{x}$  vektör çarpımı ile elde edilmektedir. Bu şekilde problemin maliyet işlevi doğrusal bir yapıda tanımlanmaktadır. Kaynak tahsisi eniyileme probleminin amaç fonksiyonu tanımlandıktan sonra eniyileme probleminin kısıt kümesinde yer alan eşitsizlikler, matris ve vektörler ile tanımlanacaktır.

(2.10)' da verilen problemin ilk kısıtı Servis Kalitesi (QoS) kriterinden gelen, bir kullanıcının bir OFDMA çerçevesi başına en az  $r_k$  bit göndermesidir. Tanımlanan  $\mathbf{x}$  bit-güç tahsis vektörünün bu kısıtı sağlayabilmesi için  $\mathbf{A}_u$  matrisi tanımlanmaktadır [10]. Aşağıda verilen  $\mathbf{A}_u$  matrisinin her satırı bir kullanıcıyı temsil etmektedir. Bu matrisin tanımlanması ile her kullanıcının en az  $r_k$  bit göndermesi

garanti altına alınmaktadır. Bu nedenle  $\mathbf{A}_u$  matrisi veri hızı kısıtı matrisi olarak adlandırılmaktadır.

$$\mathbf{A}_u = \begin{bmatrix} \mathbf{a}_{K1} & \cdots & \mathbf{a}_{K1} \\ \mathbf{a}_{K2} & \cdots & \mathbf{a}_{K2} \\ \vdots & \ddots & \vdots \\ \mathbf{a}_{KK} & \cdots & \mathbf{a}_{KK} \end{bmatrix} \in \mathbb{Z}^{K \times KM}$$

Burada

$$\begin{cases} \mathbf{a}_{K1} = [\mathbf{a}_u & \mathbf{0}_u & \cdots & \mathbf{0}_u] \in \mathbb{Z}^{1 \times KM} \\ \mathbf{a}_{K2} = [\mathbf{0}_u & \mathbf{a}_u & \cdots & \mathbf{0}_u] \in \mathbb{Z}^{1 \times KM} \\ \vdots \\ \mathbf{a}_{KK} = [\mathbf{0}_u & \mathbf{0}_u & \cdots & \mathbf{a}_u] \in \mathbb{Z}^{1 \times KM} \end{cases}$$

$$\mathbf{a}_u = [1 \ 2 \ \cdots \ M] \in \mathbb{Z}^{1 \times M}$$

$$\mathbf{0}_u = [0 \ 0 \ \cdots \ 0] \in \{0\}^{1 \times M}$$

olarak tanımlanmaktadır. Veri hızı kısıtının sağlanması için tanımlanan  $\mathbf{A}_u$  veri hızı kısıtı matrisinin  $\mathbf{x}$  bit-güç tahsisi vektörü ile çarpımı,  $\mathbf{A}_u \mathbf{x} \geq \mathbf{r}$  olmalıdır. Burada  $\bar{\mathbf{r}}$  veri hızı kısıt vektörü

$$\mathbf{r} = [r_1 \ r_2 \ \cdots \ r_k]$$

olarak tanımlanmakta ve  $\mathbf{r}$  vektörünün her bir elemanı karşılık geldiği kullanıcının veri hızı kısıtından oluşmaktadır.

Kaynak tahsisi probleminin bir diğer kısıtı olan; her bir alt-taşıyıcının en fazla bir kullanıcı tarafından kullanılması koşulu, matris yapıda ifade edildiğinde  $\mathbf{A}_c$  matrisi ile gösterilmektedir.

$$\mathbf{A}_c = \begin{bmatrix} \mathbf{1}_c & \mathbf{0}_c & \cdots & \mathbf{0}_c \\ \mathbf{0}_c & \mathbf{1}_c & \cdots & \mathbf{0}_c \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_c & \mathbf{0}_c & \cdots & \mathbf{1}_c \end{bmatrix} \in \{0,1\}^{N \times KM}$$

$$\mathbf{1}_c = [1 \ 1 \ \cdots \ 1] \in \{1\}^{1 \times KM}$$

$$\mathbf{0}_c = [0 \ 0 \ \cdots \ 0] \in \{0\}^{1 \times KM}$$

olarak verilmektedir. Her bir alt-taşıyıcının en fazla bir kullanıcı tarafından kullanılması için  $\mathbf{A}_c \bar{\mathbf{x}} \leq \bar{\mathbf{c}}$  sağlanmalıdır. Burada  $\bar{\mathbf{c}}$  vektörü:

$$\bar{\mathbf{c}} = [1 \ 1 \ \cdots \ 1]^T \in \{1\}^{N \times 1}$$

olarak tanımlanmaktadır.

Örnek:

İki kullanıcı (K=2), iki alt-taşıyıcı (N=2) ve her bir kullanıcı için alt-taşıyıcı başına en fazla iki bit (M=2) yüklenebildiği, OFDMA çerçevesi başına ilk kullanıcının 2 bit, ikinci kullanıcının 1 bit göndermesinin istendiği senaryoda problemin kısıtları aşağıdaki gibi olmaktadır.

$$\mathbf{A}_u \mathbf{x} \geq \mathbf{r} \Rightarrow \begin{bmatrix} 1 & 2 & 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} x_{1,1,1} \\ x_{1,1,2} \\ x_{2,1,1} \\ x_{2,1,2} \\ x_{1,2,1} \\ x_{1,2,2} \\ x_{2,2,1} \\ x_{2,2,2} \end{bmatrix} \geq \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\mathbf{A}_c \mathbf{x} \leq \mathbf{c} \Rightarrow \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_{1,1,1} \\ x_{1,1,2} \\ x_{2,1,1} \\ x_{2,1,2} \\ x_{1,2,1} \\ x_{1,2,2} \\ x_{2,2,1} \\ x_{2,2,2} \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Bu senaryonun olurlu kümesinin dört elemanı bulunmaktadır.



$\mathbf{x} = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]^T$  : ilk kullanıcıya ilk alt-taşıyıcı ile 2 bit, ikinci kullanıcıya ikinci altşıyıcı ile 1 bit atanmakta.

$\mathbf{x} = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^T$  : ilk kullanıcıya ilk alt-taşıyıcı ile 2 bit, ikinci kullanıcıya ikinci altşıyıcı ile 2 bit atanmakta.

$\mathbf{x} = [0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0]^T$  : ilk kullanıcıya ikinci alt-taşıyıcı ile 2 bit, ikinci kullanıcıya ilk altşıyıcı ile 1 bit atanmakta.

$\mathbf{x} = [0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0]^T$  : ilk kullanıcıya ikinci alt-taşıyıcı ile 2 bit, ikinci kullanıcıya ilk altşıyıcı ile 2 bit atanmakta.

Yapılan matris ve vektör tanımlamalarından sonra MA probleme (2.4) eşdeğer doğrusallaştırılmış eniyileme problemi (2.13) ile tanımlanmaktadır.

$$\begin{aligned} \min \quad & \mathbf{p}^T \mathbf{x} \\ \text{kısıtlar} \quad & \mathbf{A}_u \mathbf{x} \geq \mathbf{r} \\ & \mathbf{A}_c \mathbf{x} \leq \mathbf{c} \\ & x_{k,n,m} \in \{0, 1\} \quad k=1,2,\dots,K, \quad n=1,2,\dots,N \text{ ve } m=1,2,\dots,M \end{aligned} \tag{2.13}$$

Benzer şekilde matris ve vektörler ile RA probleme (2.5) eşdeğer doğrusallaştırılmış RA eniyileme problemi (2.14) ile tanımlanmaktadır.

$$\begin{aligned} \max \quad & \mathbf{r}^T \mathbf{x} \\ \text{kısıtlar} \quad & \mathbf{p}^T \mathbf{x} \leq P_T \\ & \mathbf{A}_u \mathbf{x} \geq \Gamma(\mathbf{r}^T \mathbf{x}) \\ & \mathbf{A}_c \mathbf{x} \leq \mathbf{c} \\ & \Gamma = [\gamma_1 \ \gamma_2 \ \dots \ \gamma_K]^T \\ & x_{k,n,m} \in \{0, 1\} \quad k=1,2,\dots,K, \quad n=1,2,\dots,N \text{ ve } m=1,2,\dots,M \end{aligned} \tag{2.14}$$

## 2.3 Kaynak Tahsis Problemlerinde Olurlu Kümenin Geometrisi

(2.13)' de tanımı verilen kaynak tahsisi probleminde, kısıt kümesinde yer alan kısıt matris ve vektörlerinin problemin kanal katsayılarından bağımsız olduğu görülmektedir. Yani kullanıcı sayısı  $K$ , alt-taşıyıcı sayısı  $N$  ve bir alt-taşıyıcıdan gönderilebilecek en yüksek bit sayısı  $M$  sabit kaldığı sürece problemin olurlu kümesi kanaldan bağımsız olarak sabit kalmaktadır. (2.13)' de verilen problem doğrusal tamsayı programlama problemi olduğundan problemin eniyi çözümleri Dallar-ve-Sınırla, Kesin-Düzlemler gibi algoritmalar ile çözülebilmektedir [4]. Ancak bu algoritmaların işlem karmaşıklıkları çoğu zaman pratik sistemlerde kullanılmayacak kadar yüksektir. Problem (2.13)' yi çözmek için bir diğer yol ise bir önceki bölümde anlatılan Wong [7] 'un önerdiği gibi, problemin tamsayı kısıtlarının  $x_{k,n,m} \in [0,1]$  değerlerine gevşetilerek çözümlenmesidir. Bu gevşetme sonucunda (2.13)' de yer alan doğrusal kısıtlar  $KNM$  boyutlu uzayda her boyutta  $[0,1]$  aralığına hapsedilmiş bir çokyüzlü tanımlanmaktadır. Bölüm 2.5 'de verilen bilgilere göre, doğrusal eniyileme probleminin enküçültülmesi, bu çokyüzlünün bir uç noktasına gidilmesine neden olmaktadır. Yani problemin eniyi çözümü, çokyüzlünün  $KNM$  boyutlu bir uç noktasına eşit olmaktadır. Gevşetilmiş problem için tanımlanan bu çokyüzlünün tüm uç noktaları tamsayı değerlere sahip olmamaktadır. Gevşetilmiş problemin doğrusal programlama algoritmaları (Simpleks Algoritması, İç-Noktalar gibi) ile enküçültülmesi sonucu tamsayı olmayan bu uç noktalar problemin çözümü olabilmektedir.

### 3. DOĞRUSAL TAMSAYI PROGRAMLAMA ve YÜKSELT-ve-İZDÜŞÜR ALGORİTMASI

Doğrusal Tamsayı Programlama problemleri, Doğrusal Programlama problemleri ile aynı yapıdadır. Doğrusal Tamsayı Programlama problemleri, tüm değişkenlerin tamsayı olma zorunluluğu nedeniyle Doğrusal Programlama' dan ayrılmaktadır. Doğrusal Programlama problemlerinin çözümleri sonrasında değişkenler kesirli olabileceği için Tamsayı Programlama' ya ihtiyaç duyulmaktadır. Çıkan kesirli sonucu günlük hayatta kullanılan uygulamalarda kullanabilmek için tüm değişkenlerin tamsayı olması gerektiği durumlarda kesirli çözüm en yakın tamsayı değerlere yuvarlanabilir. Fakat bu durumda problemin çözümü eniyilikten uzaklaşabilir ve hatta yuvarlama sonucu elde edilen sonuç olurlu bölge içerisinde dahi yer almayabilir. Bu nedenle günlük hayatta birçok alanda Doğrusal Tamsayı Programlama' ya ihtiyaç duyulmaktadır. Doğrusal tamsayı programlama algoritmaları ile kaynak tahsisi probleminin eniyi çözümü elde edilebilmektedir. Bu kapsamda doğrusal tamsayı algoritmalarından yükselt ve izdüşür algoritması kullanılarak kaynak tahsisi probleminin eniyi çözümü elde edilmeye çalışılacaktır.

Doğrusal Tamsayı Programlama problemleri aşağıdaki şekilde ifade edilmektedir:

$$\begin{aligned} \text{maliet işlevi, } & \mathbf{c}^T \mathbf{x} & (3.1) \\ \text{kısıtlar, } & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{x} \geq 0, \mathbf{x} \text{ tamsayı} \end{aligned}$$

$x'$  in alacağı tamsayı değerlerinin 0 ve 1 ile sınırlandırılması ile Doğrusal Tamsayı Programlama problemi "**0-1 Programlama**" problemi adını almaktadır ve problem aşağıdaki gibi ifade edilmektedir:

$$\begin{aligned} \text{maliet işlevi, } & \mathbf{c}^T \mathbf{x} \\ \text{kısıtlar, } & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{x} \in \{0,1\} \end{aligned} \quad (3.2)$$

Doğrusal Tamsayı Programlama probleminin tanım kümesinin ayrık doğası nedeniyle çözümü Doğrusal Programlama problemlerinin çözümüne göre daha karmaşık ve zordur.

Tamsayı Programlardaki kısıtların tamsayı olma zorunluluğunun kaldırılmasına “**gevşetme**” denilmektedir. Doğrusal Programlama’ da eniyileme için kullanılan uzay, Tamsayı Programlama’ da eniyileme için kullanılan uzaydan daha geniş bir alanı kapladığı için; Tamsayı Programlamadaki gevşetme sonucu bulunan sonuç, en iyi olurlu tamsayı çözüm için bir alt sınır oluşturmaktadır. Gevşetilmiş 0-1 Programlama (3.3) ‘deki gibi ifade edilmektedir:

$$\begin{aligned} \text{maliet işlevi, } & \mathbf{c}^T \mathbf{x} \\ \text{kısıtlar, } & \mathbf{Ax} \leq \mathbf{b} \\ & 0 \leq \mathbf{x} \leq 1 \end{aligned} \quad (3.3)$$

Burada (3.2)’ de verilen problemden farklı olarak  $\mathbf{x}$  vektörü içindeki değerlerin tamsayı olması gerekirken,  $\mathbf{x}$  vektörünün alabileceği değerler tamsayı olmayan değerlere gevşetilmiştir.

### 3.1 Yükselt-ve-İzdüşür Algoritması

(3.3)’ de verilen gevşetilmiş doğrusal programlama probleminin kısıt kümesinde yer alan eşitsizliklerin oluşturduğu çokyüzlünün tüm uç noktaları tamsayı değerler almamaktadır. Ek-1 ‘de verilen bilgiler doğrultusunda (3.3) ‘de tanımlanan problemin çözüm kümesini, çokyüzlünün uç noktalarının oluşturduğu bilinmektedir. Bu nedenle çokyüzlü için tanımlı olan uç noktaların tamamının tamsayı değerlere sahip olması problemin eniyi çözümünün elde edilebilmesi için gereklidir. Yükselt-ve-izdüşür algoritması ile bu çokyüzlünün tüm uç noktaları tamsayı değerlere çekilebilmektedir.

Gevşetilmiş Doğrusal Programlama’ da gevşetmeyi daha sıkı hale getirip problemin formülasyonunu güçlendirmek için; probleminin daha yüksek boyutlu bir uzaya yükseltilmesi (**lifting**) gerekmektedir. Yüksek boyutlu uzaya yükseltme işlemi sonrasında problemin orijinal uzaya izdüşümü (**projection**) alınmaktadır. Yapılan bu işlemler sonrasında orijinal uzay üzerinde tanımlı olan olurlu bölge üzerin-

de, düzlem kesme (**cutting plane**) [4] metodu uygulanmış gibi görünmektedir. Tüm değişkenler için yükseltme ve izdüşürme adımları yapıldıktan sonra problemin sahip olduğu çokyüzlünün tüm uç noktaları tamsayı değerlere taşınmış olacaktır. Yani çokyüzlünün tanımlı olduğu olurlu küme içerisinde tamsayı noktalarının dışbükey kabukları elde edilecektir.

Gevşetilmiş Doğrusal Program'ın olurlu kümesi şöyle ifade edilebilir:

$$K = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} \leq \mathbf{b}, 0 \leq \mathbf{x} \leq \mathbf{1}\} \quad (3.4)$$

$$K = \{\mathbf{x} \in \mathbb{R}^n \mid \tilde{\mathbf{A}}\mathbf{x} \leq \tilde{\mathbf{b}}\}$$

(3.4) 'de verilen problemin tamsayı kümesi aşağıda verilmektedir [1].

$$K^0 = \{\mathbf{x} \in K \mid x_j \in \{0, 1\}, j = 1, \dots, p\} \quad (3.5)$$

(3.4) 'de verilen Gevşetilmiş Doğrusal Programlama probleminin tanım kümesi içerisinde yer alan eşitsizliklerin tanımladığı çokyüzlünün uç noktalarının tamamı tamsayı değerler almamaktadır. Çokyüzlünün tüm uç noktalarının tamsayı değerlere çekilmesiyle dışbükey  $K^0$  tamsayı kümesi elde edilmektedir. Bundan sonraki kısımda dışbükey  $K^0$  tamsayı kümesinin nasıl elde edileceği anlatılmaktadır.

### 3.1.1 Yükseltme (Lifting)

$\mathbf{x}$  vektörü içerisinde yer alan elemanlar  $x_j$  ise, tüm  $j \in \{1, \dots, p\}$  değerleri için yükseltme işlemi; tanım kümesinde yer alan  $\tilde{\mathbf{A}}\mathbf{x} \leq \tilde{\mathbf{b}}$  eşitsizliklerinin  $x_j$  ve  $(1 - x_j)$  ile ayrı ayrı çarpılması olarak tanımlanmaktadır. (3.4) 'de verilen  $\tilde{\mathbf{A}}\mathbf{x} \leq \tilde{\mathbf{b}}$  eşitsizliklerinin matris ve vektörlerinin genel yapısı aşağıda verilmektedir.

$\tilde{\mathbf{A}} = [\mathbf{A}_1 \quad \mathbf{A}_2 \quad \dots \quad \mathbf{A}_{N-1} \quad \mathbf{A}_N]_{M \times N}$   $\bar{\mathbf{A}}_i$ :  $\tilde{\mathbf{A}}$  matrisinin tüm  $i \in \{1, 2, \dots, N\}$  için, i-inci sütunlarına karşılık gelen sütun vektörü.

$$\mathbf{x}^T = [x_1 \quad x_2 \quad \dots \quad x_{N-1} \quad x_N]$$

$$\tilde{\mathbf{b}}^T = [b_1 \quad b_2 \quad \dots \quad b_{M-1} \quad b_M]$$

olarak verilmektedir.

Sırası ile (i) ve (ii)' de verilen çarpma işlemleri gerçekleştirilmektedir.

$$(i) \quad x_j(\tilde{\mathbf{A}}\mathbf{x} \leq \tilde{\mathbf{b}}) \quad (3.6)$$

$$(ii) \quad (1-x_j)(\tilde{\mathbf{A}}\mathbf{x} \leq \tilde{\mathbf{b}})$$

Burada  $\bar{\mathbf{x}}$  vektörü içerisinde yer alan elemanların  $x_j$  ve  $(1-x_j)$  ile çarpılması sonucu ortaya yeni elemanlar çıkmaktadır.

$$\begin{aligned} x_i x_i &= x_i^2, \quad i = j \text{ için} \\ x_i x_j &= y_{ij}, \quad \text{tüm } i = 1, 2, \dots, N, j = 1, 2, \dots, N \text{ ve } i \neq j \text{ için} \end{aligned} \quad (3.7)$$

Ortaya çıkan yeni elemanlar ile birlikte yükseltilmiş uzayda olurlu bölgeyi tanımlayan eşitsizlik sayısı  $(2M-1)$ ' e, değişken sayısı ise  $(2N-1)$ 'e çıkmıştır. Yükseltme sonrasında hem eşitsizlik sayısı hem de değişken sayısı yaklaşık olarak iki katına çıkmaktadır.

Yükseltme işlemi sonrasında ortaya çıkan yeni değişkenlerden  $x_i^2$ , yükseltilmiş uzayda problemin doğrusallığını bozmaktadır ve problem, doğrusal olmayan (**nonlinear**) bir yapıya dönüşmektedir. Burada problemin kısıtları gevşetilmeden önce (3.2) 'de kısıt kümesinde  $x_j \in \{0,1\}$  olduğu için  $x_i^2 = x_i$  olarak değiştirilmektedir. Yapılan bu işlem ile yükseltilmiş uzayda da problemin doğrusal yapısı korunmuş olur.

$x_j$  değişkenine göre yapılan yükseltme sonucunda, yükseltilmiş uzayda problemin olurlu bölgesinin  $x_j$  eksenindeki uç noktaları artık sadece  $\{0,1\}$  tamsayı değerleri almaktadır.

### 3.1.2 İzdüşüm (Projection)

Yükseltme işlemi sonrasında yüksek boyutlu uzaydaki çokyüzlünün yeniden orijinal uzaya indirgenmesi gerekmekte ve bu da izdüşüm (**projection**) ile gerçek-

leştirilmektedir. Orijinal uzaya izdüşüm işleminin yapılması ile yükseltme sonrası ortaya çıkan geçici  $y_{ij}$  değişkenleri ortadan kalkmakta ve çokyüzlü, sadece tüm  $j=1,2,\dots,N$  için  $x_j$  değişkenleri ile tanımlanmaktadır. Yükseltmiş uzayda çokyüzlünün tüm uç noktalarının  $x_j$  eksenindeki değerleri  $\{0,1\}$  tamsayı değerlerine sahip oldukları için; orijinal uzaya izdüşüm sonrasında çokyüzlünün uç noktaları  $x_j$  ekseninde yine  $\{0,1\}$  tamsayı değerler almaktadır.

$x_j$  için yapılan yükseltme ve izdüşüm işlemleri tüm  $j=1,2,\dots,N$  indislerine sahip  $x_j$  değişkenler için sırasıyla tekrar edilmektedir. Tüm değişkenler için yükseltme ve izdüşüm yapıldıktan sonra elde edilecek eşitsizliklerin tanımladığı çokyüzlünün tüm uç noktaları  $\{0,1\}$  tamsayı değerlere sahip olmaktadır. Yükseltme ve izdüşüm işlemi sadece tek bir eksen için değil tanımlanan tüm uzay için tek seferde de yapılabilir. Fakat bu şekilde daha fazla işlem karmaşıklığı ve hafıza gereksinimi ortaya çıkmaktadır. Bu nedenle sonraki bölümlerde her seferde bir eksende yükseltme ve izdüşüm işlemi yapılacak ve problem döngüsel bir yapıda çözülecektir.

“**Lift and Project**” metodu ile çokyüzlünün sahip olduğu uç noktaların  $\{0,1\}$  tamsayı değerlerine kaydırılması yukarıda anlatılmaktadır. Fakat burada merak edilen, uç noktaların nasıl  $\{0,1\}$  tamsayı değerler aldığıdır? Yukarıdaki yükseltme aşamasında problemin kısıt kümesinde yer alan  $\tilde{\mathbf{A}}\mathbf{x} \leq \tilde{\mathbf{b}}$  eşitsizliklerinin  $x_j$  ve  $(1-x_j)$  ile çarpılmasıyla problemin kısıtlarında bir daralma meydana gelmemektedir. Aynı şekilde farklı indislere sahip değişkenlerin çapraz çarpımdan ortaya çıkan  $y_{ij}$  değişkenleri de kısıtlarda daralmaya neden olmamaktadır. Yükseltme işleminin yapıldığı eksen üzerindeki değerlerin tamsayı noktasına kaydırılması ile aynı indeksli değişkenlerin çapraz çarpımından oluşan  $x_j^2$  değişkeninin  $x_j$  ile değiştirilmesi problemin kısıtlarında bir daralmaya sebep olmaktadır. Bu nedenle çokyüzlünün tamsayı olmayan uç noktaları  $\{0,1\}$  tamsayı değerlere taşınmaktadır. Basit bir örnek ile bu durum incelendiğinde:

$$\begin{aligned} \mathbf{a}^T \mathbf{x} &\leq \mathbf{b} \\ 0 &\leq x_i \leq 1, i=\{1,2\} \end{aligned} \tag{3.8}$$

olarak tanımlanan çokyüzlü için  $\mathbf{a}^T = [a_1 \ a_2]$  ve  $a_i, b \in \mathbb{R}^1, i = \{1,2\}$  olduğu durumda, (3.8) 'de verilen eşitsizlikler aşağıdaki şekilde yazılır.

$$\begin{aligned}
 & a_1 x_1 + a_2 x_2 \leq b \\
 & -x_1 \leq 0 \\
 & x_1 \leq 1 \\
 & -x_2 \leq 0 \\
 & x_2 \leq 1
 \end{aligned} \tag{3.9}$$

(3.9)' da verilen eşitsizlikleri  $x_1$  ile çarptığımız zaman elde edilen yeni eşitsizlikler aşağıda yer almaktadır.

$$\left( \begin{array}{l} a_1 x_1 x_1 + a_2 x_1 x_2 \leq b x_1 \\ -x_1 x_1 \leq 0 \\ x_1 x_1 \leq x_1 \\ -x_1 x_2 \leq 0 \\ x_1 x_2 \leq x_1 \end{array} \right) \Rightarrow \left\{ \begin{array}{l} x_1 x_2 = y_{1,2} \\ x_1 x_1 = x_1^2 \end{array} \right\} \text{ için } \Rightarrow \left( \begin{array}{ll} a_1 x_1^2 + a_2 y_{1,2} - b x_1 \leq 0 & \text{(i)} \\ -x_1^2 \leq 0 & \text{(ii)} \\ x_1^2 - x_1 \leq 0 & \text{(iii)} \\ -y_{1,2} \leq 0 & \text{(iv)} \\ y_{1,2} \leq x_1 & \text{(v)} \end{array} \right) \tag{3.10}$$

olmaktadır. Aynı şekilde  $(1-x_1)$  ile çarpıldığı zaman aşağıdaki oluşan yeni eşitsizlikler aşağıda verilmektedir.

$$\left( \begin{array}{l} a_1(1-x_1)x_1 + a_2(1-x_1)x_2 \leq b(1-x_1) \\ -(1-x_1)x_1 \leq 0 \\ (1-x_1)x_1 \leq 1-x_1 \\ -(1-x_1)x_2 \leq 0 \\ (1-x_1)x_2 \leq 1-x_1 \end{array} \right) \Rightarrow \left\{ \begin{array}{l} x_1 x_2 = y_{1,2} \\ x_1 x_1 = x_1^2 \end{array} \right\} \text{ için } \left( \begin{array}{ll} \frac{a_1}{b}(x_1 - x_1^2) + \frac{a_2}{b}x_2 - \frac{a_2}{b}y_{1,2} + x_1 \leq 1 & \text{(vi)} \\ x_1^2 - x_1 \leq 0 & \text{(vii)} \\ 2x_1 - x_1^2 \leq 1 & \text{(viii)} \\ y_{1,2} - x_2 \leq 0 & \text{(ix)} \\ x_1 + x_2 - y_{1,2} \leq 1 & \text{(x)} \end{array} \right) \tag{3.11}$$

(3.10) ve (3.11) 'de yer alan yükseltilmiş uzaya ait on adet eşitsizlik incelendiğinde;  $x_1$ 'in alabileceği en küçük değer "0" iken alabileceği en büyük değer ise (vi)' da verilen eşitsizliğe bağlı durumdadır. (vi)'da verilen eşitsizlik incelendiğinde:

$$\frac{a_1}{b}(x_1 - x_1^2) + \frac{a_2}{b}x_2 - \frac{a_2}{b}y_{1,2} + x_1 \leq 1$$

burada  $x_1$ 'in alabileceği en büyük değer, diğer değişkenlerden bağımsız olarak 1 olabilmesi için :



$$\frac{a_1}{b}(x_1 - x_1^2) = 0 \quad (3.12)$$

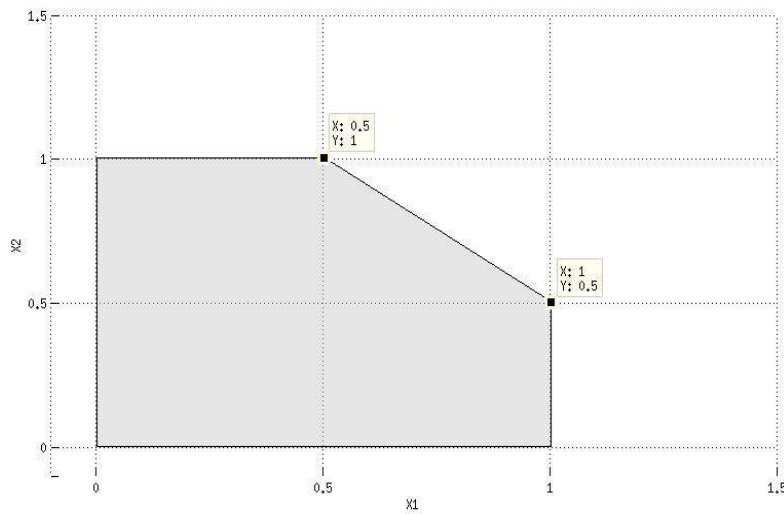
eşitliğinin sağlanması gerekmektedir.  $a_1$  ve  $b$ 'den bağımsız olarak eşitliğin sağlanması için  $x_1^2 = x_1$  olmalıdır.  $x_1^2 = x_1$  olduğunda (3.8)' de verilen eşitsizlerde yer alan tüm katsayı ve değişkenlerden bağımsız olarak  $x_1$  ekseninde uç noktalar  $\{0,1\}$  tamsayı değerler almaktadır.

Bu nedenle yükseltme işlemi sonrasında aynı indisli değişkenlerin çapraz çarpımından gelen  $x_j^2$  değişkeni  $x_j$  ile değiştirildiğinde, yükseltilmiş uzayda çokyüzlünün uç noktaları  $x_j$  ekseninde  $\{0,1\}$  tamsayı değerler almaktadır.

### Örnek 3.1

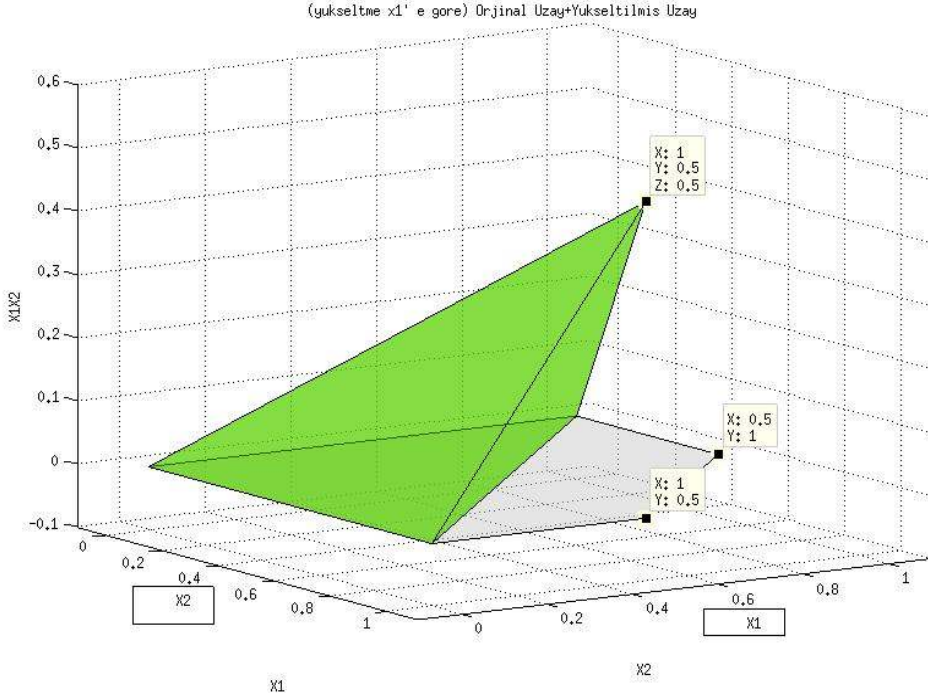
$$\begin{aligned} x_1 + x_2 &\leq \frac{3}{2} \\ 0 &\leq x_1 \leq 1 \\ 0 &\leq x_2 \leq 1 \end{aligned} \quad (3.13)$$

Yukarıda verilen eşitsizlikler ile tanımlanan olurlu bölgenin şekli aşağıda verilmektedir.



**Şekil 3.1:** (3.13)'de verilen Eşitsizliklerin Tanımladığı Olurlu Bölge

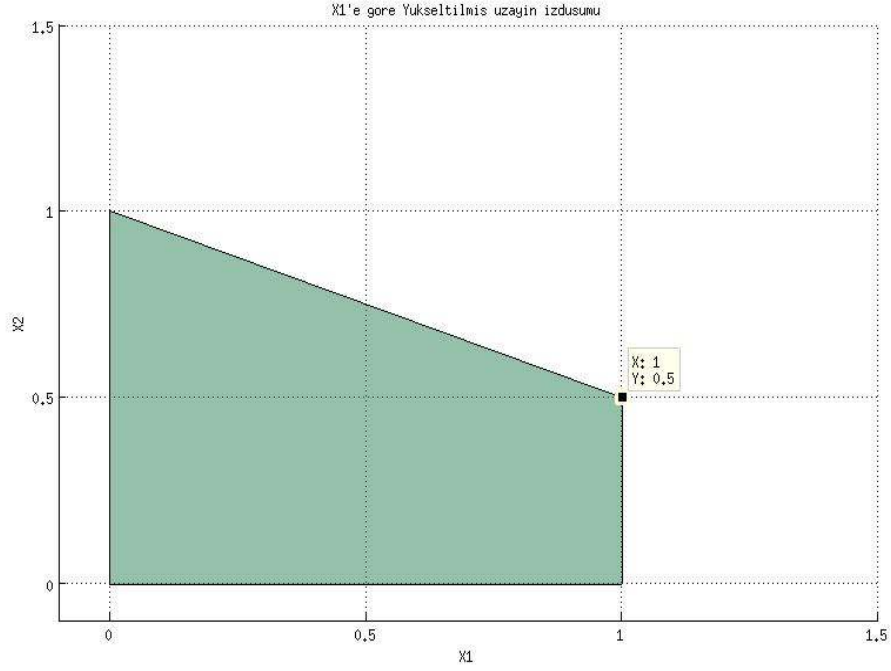
Olurlu bölge 5 adet uç noktaya sahiptir. Fakat bu uç noktalardan ikisi (0.5,1) ve (1,0.5) tamsayı değerler değildir. Olurlu bölgenin uç noktalarını tamsayı değerlere kaydırmak için, ilk olarak  $x_1$  eksenini için olurlu bölge yükseltilmiş uzaya taşınacaktır.



**Şekil 3.2:**  $x_1$  eksenine göre yükseltme sonrası yükseltilmiş uzay

Şekil 3.2 ' de görüldüğü gibi  $x_1$  eksenine göre yükseltme yapıldıktan sonra (0.5,1) noktası yükseltilmiş uzayda olurlu bölge dışında kalmıştır.

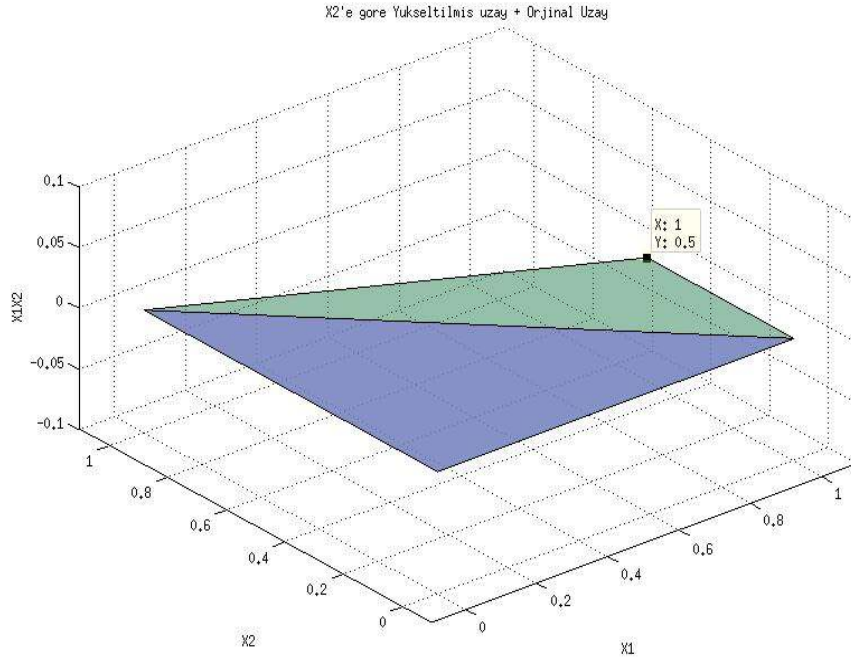
(1, 0.5, 0.5) noktası sadece yükseltilmiş uzay için tanımlı bir noktadır ve bu noktanın  $x_1$  eksenindeki değeri tamsayıdır. Yükseltilmiş uzayın XY düzlemine izdüşümü alındığı zaman Şekil 3.3' te gösterilen olurlu bölge elde edilmektedir.



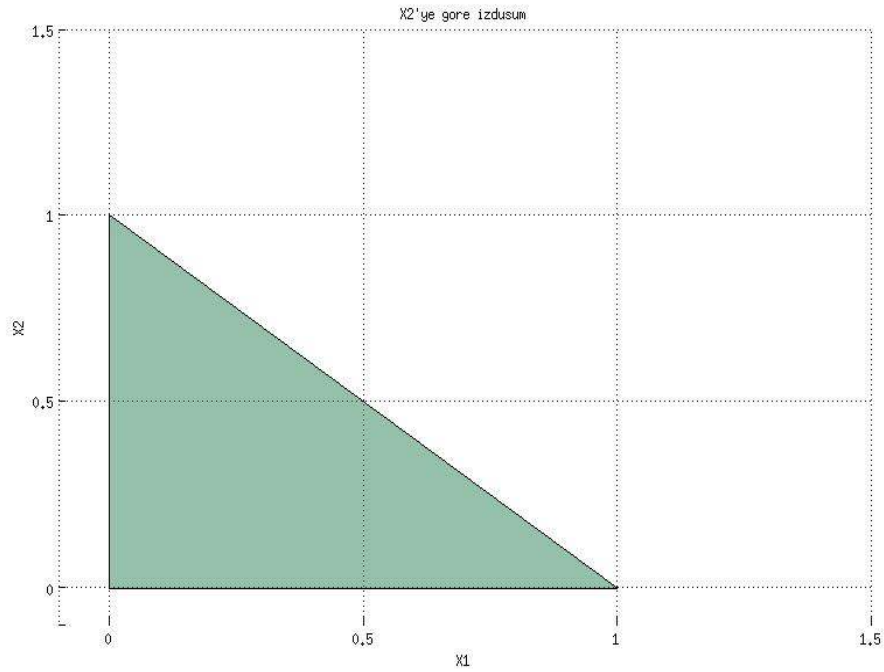
**Şekil 3.3:** Yükseltilmiş Uzayın XY Düzlemine İzdüşümü (Yeni Olurlu Bölge)

Burada görüldüğü gibi  $x_1$  eksenine göre yapılan yükseltme ve sonrasında orijinal düzleme yapılan izdüşüm işlemi sonrasında yeni oluşan olurlu bölgenin tüm uç noktalarının  $x_1$  eksenindeki değerleri tamsayıdır.

Bir sonraki adımda ise, yeni oluşan olurlu bölgenin  $x_2$  eksenine göre yükseltmesi yapılmaktadır. Burada yükseltme yapılırken yeni olurlu bölgeyi tanımlayan eşitsizlikler kullanılmaktadır. Yapılan yükseltme sonrasında yükseltilmiş uzayda  $x_1 x_2 = 0$  değerini almaktadır. Bu nedenle yükseltme sonrasında yeni olurlu bölge Şekil 3.4' de görüldüğü gibi yine aynı düzlem üzerinde kalmıştır. Buna rağmen (1, 0.5) noktası olurlu bölge dışında kalmıştır. Böylece tamsayı olmayan son uç nokta da (1,0) noktasına kaydırılmıştır.



**Şekil 3.4:**  $x_2$  Eksenine Göre Yükseltme Sonrası Olurlu Bölge



**Şekil 3.5:**  $x_2$  Eksenine Göre Yükseltme ve İzdüşüm Sonrası Olurlu Bölge

$x_2$  eksenini için yükseltme yapıldıktan sonra yükseltilmiş uzayın olurlu bölgesi yine aynı düzlemde kaldığı için yükseltme sonrasında olurlu bölgenin orijinal düzleme izdüşümüne gerek kalmamıştır. Şekil 3.5 'te verilen grafikte, tüm değişkenler için yükseltme ve izdüşüm yapıldıktan sonra elde edilen olurlu bölge gösterilmektedir. Burada görüldüğü gibi olurlu bölgenin tüm uç noktaları tamsayı değerlere çekilmiştir.

### 3.2 Lift and Project İçin Önerilen Algoritma

Bölüm 3.1 'de Yükseltme ve izdüşüm algoritmasının genel yapısı verilmektedir. Verilen algoritma eniyileme problemlerine uygulandığı takdirde problemin tanımlı olduğu tüm eksenler için yükseltme ve izdüşüm işlemi döngüsel olarak yapılmaktadır. Algoritmanın uygulanacağı problemin boyutu arttıkça, ortaya çok daha fazla değişken ve eşitsizlik çıkmaktadır. Bundan dolayı yüksek boyutlu problemler için yükseltme ve izdüşüm algoritmasının uygulanması zorlaşmaktadır. Bu bölümde Yükseltme ve İzdüşüm algoritmasından yola çıkarak yeni bir çözüm önerilmektedir.

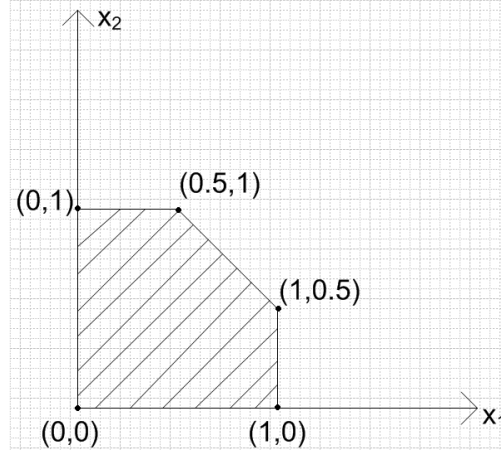
Belirli sayıda doğrusal eşitsizlik tarafından tanımlanan çokyüzlünün uç noktalarının (**vertices**) hesaplanabilmesi doğrusal programlama problemlerinin çözümü için büyük önem teşkil etmektedir. Eğer  $A\bar{x} \leq \bar{b}$  olarak verilen eşitsizliklerin tanımladığı çokyüzlünün tüm uç noktaları hesaplanırsa, bu uç noktaları kullanarak çokyüzlünün olurlu tamsayı uç noktalarının dış bükey kabuğu hesaplanabilir. Bu kapsamda çokyüzlünün tüm uç noktalarını daha kısa sürede tamsayı haline getirmek için aşağıdaki çözüm önerilmektedir.

Özetle,

- 1) Çokyüzlünün  $x^*$  uç noktasının  $x_j$  eksenindeki değeri "0" yada "1" ise  $x_j$  ekseninde yükseltme ve izdüşüm işlemleri sonrasında  $x^*$  uç noktasının  $x_j$  eksenindeki değeri yine sırası ile "0" yada "1" olur. Yani uç noktanın değeri aynı kalır.
- 2) Çokyüzlünün  $x^*$  uç noktasının  $x_j$  eksenindeki değeri "0" yada "1" **olmadığı** durumda  $x_j$  ekseninde yükseltme ve izdüşüm işlemleri sonrasında  $x^*$  uç noktasının  $x_j$  eksenindeki değeri yine "0" yada "1" olur.

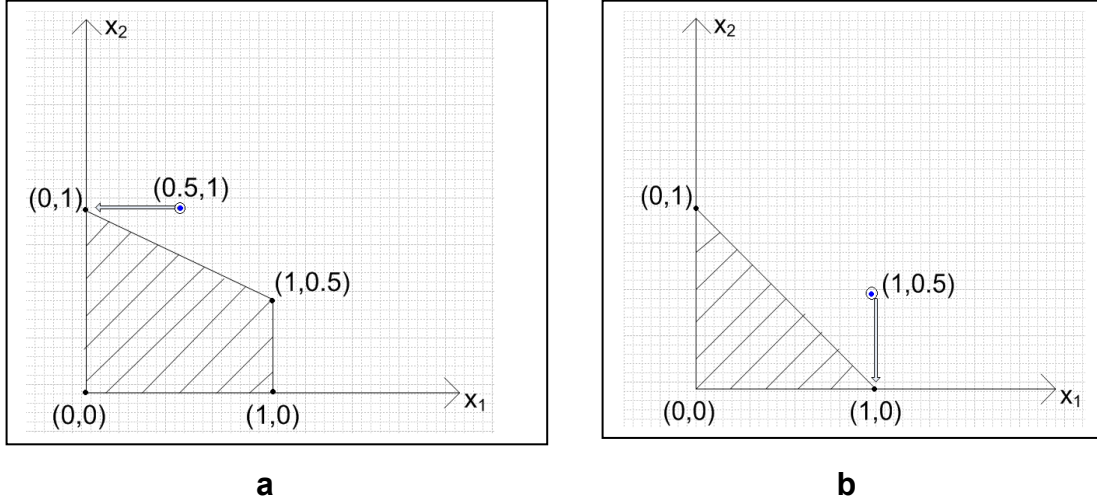
Yukarıda elde edilen sonuçlara göre hiçbir yükseltme ve izdüşüm işlemi gerçekleştirilmeksizin çokyüzlünün olurlu tamsayı uç noktaları hesaplanabilmektedir.

### Örnek 3.2



**Şekil 3.6:** 5 Adet Uç Noktaya Sahip Çokyüzlü

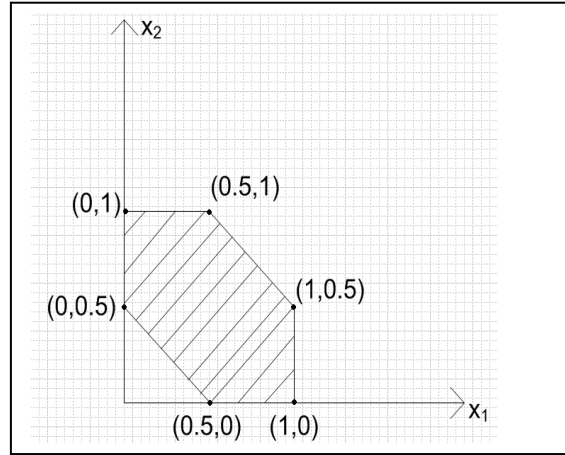
Şekil 3.6' da verilen çokyüzlü toplam 5 adet uç noktaya sahiptir. Bu çokyüzlünün  $(0.5,1)$  ve  $(1,0.5)$  uç noktaları tamsayı değerlere sahip değildir. Çokyüzlüyü tanımlayan eşitsizlikler  $x_1$  eksenine göre yükseltme ve izdüşüm algoritması uygulandığında  $(0.5,1)$  noktası  $(0,1)$  noktasına kaymakta ve çokyüzlünün tüm uç noktalarının  $x_1$  eksenindeki değerleri tamsayı değerler almaktadır. Bu durum Şekil 3.7.a 'da detaylı olarak gösterilmiştir.



**Şekil 3.6: a)  $x_1$  Eksenine Göre Yükseltme ve İzdüşüm Algoritmasının Uygulanması**  
**Sonucu Oluşan Çokyüzlü**  
**b)  $x_2$  Eksenine Göre Yükseltme ve İzdüşüm Algoritmasının Uygulanması**  
**Sonucu Oluşan Çokyüzlü**

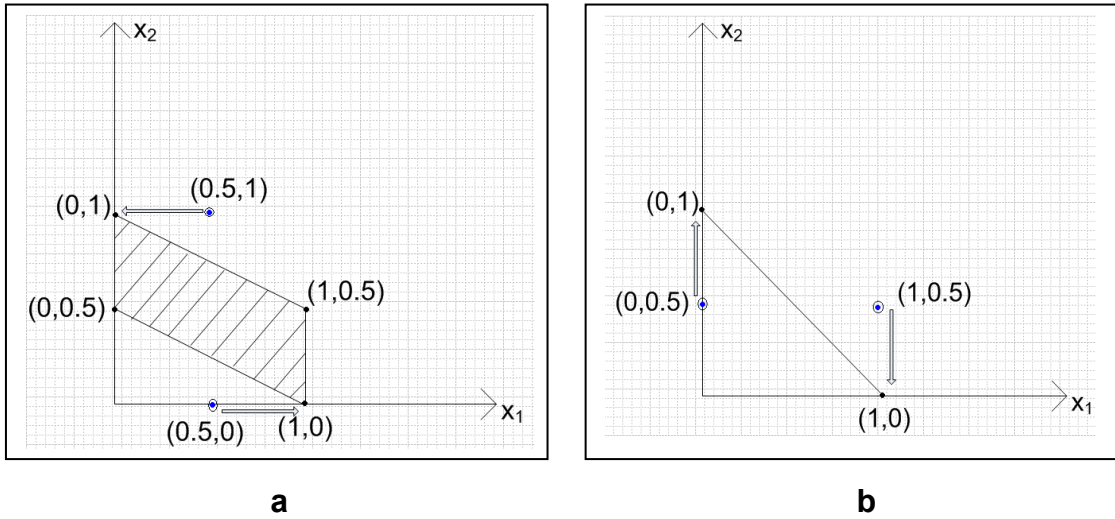
Şekil 3.6.a 'ya bakıldığında tamsayı değerlere sahip olmayan tek uç nokta  $(1,0.5)$ ' tir. Bu uç noktayı da tamsayı değere çekebilmek için, Şekil 3.6.a 'da verilen çokyüzlüyü tanımlayan eşitsizliklere  $x_2$  eksenine göre yükseltme ve izdüşüm algoritması uygulanmaktadır. Bu işlem sonrasında  $(1,0.5)$  uç noktası  $(1,0)$  uç noktasına kaymakta ve tanımlanan yeni çokyüzlünün tüm uç noktaları tamsayı değerler vermektedir. Bu işlemler sonrasında kalan uç noktalar Şekil 3.6.b 'de gösterilmiştir. İlk durumda verilen (Şekil 3.5) çokyüzlünün tamsayı uç noktaları ile son durumda elde edilen çokyüzlünün uç noktaları birbirine eşittir. Başka bir örnek daha incelendiğinde yine benzer bir sonuç elde edilmektedir.

### Örnek 3.3



**Şekil 3.7:** 6 Adet Uç Noktaya Sahip Çokyüzlü

Şekil 3.7 'de verilen çokyüzlünün toplam 6 adet uç noktası vardır. Bu uç noktalardan 2 tanesi tüm eksenler için tamsayı değere sahipken, kalan dört uç noktanın tamsayı değerlere sahip olmadığı görülmektedir. Sırasıyla  $x_1$  ve  $x_2$  eksenine göre yükseltme ve izdüşüm algoritması uygulandığında elde edilen eşitsizlikler ile tanımlanan çokyüzlü Şekil 3.8.b 'de verilmektedir.



**Şekil 3.8: a)**  $x_1$  Eksenine Göre Yükseltme ve İzdüşüm Algoritmasının Uygulanması

Sonucu Oluşan Çokyüzlü

**b)**  $x_2$  Eksenine Göre Yükseltme ve İzdüşüm Algoritmasının Uygulanması

Sonucu Oluşan Çokyüzlü



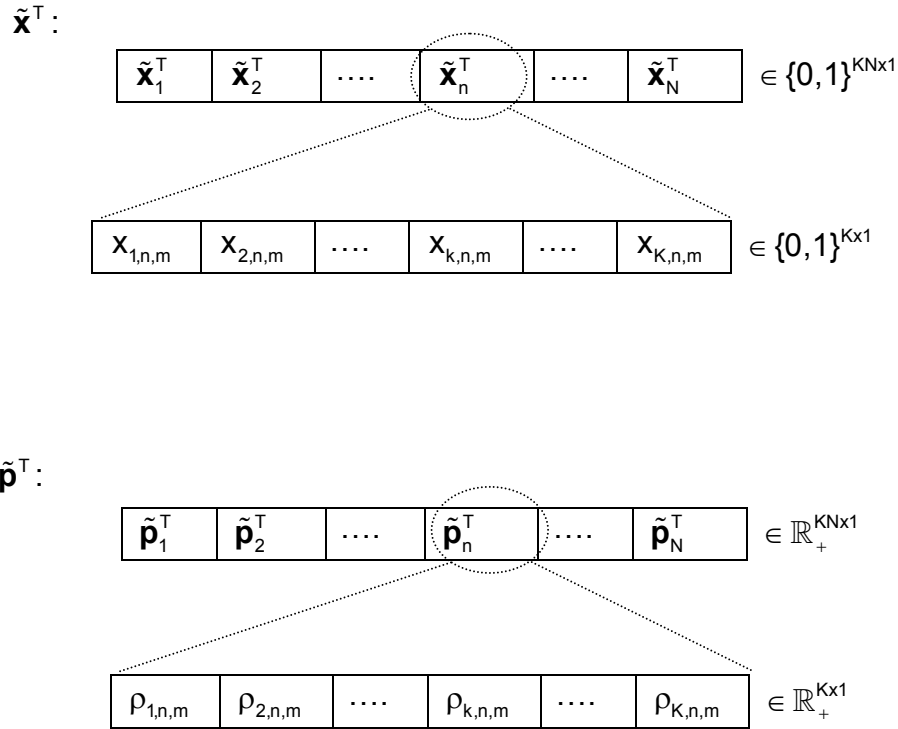
Bir önceki örnekte elde edilen sonuca benzer şekilde  $x_1$  ve  $x_2$  eksenine göre yükseltme ve izdüşüm algoritması sonucunda elde edilen çokyüzlünün iki adet uç noktası bulunmakta ve bu noktalar yine Şekil 3.7 'de verilen çokyüzlünün tamsayı uç noktaları ile aynı değerlere sahip olmaktadır.

Yukarıda verilen iki örnek incelendiğinde görüldü ki, bir çokyüzlünün tüm uç noktaları bulunduktan sonra, bu uç noktalardan herhangi bir boyuttaki değeri tamsayı olmayanlar çıkartılmalıdır. Kalan tamsayı uç noktalar kümesi çokyüzlüyü tanımlayan problemin olurlu çözüm kümesine eşit olmaktadır. Detaylar için Ek-3' e bakınız.

## 4 ÖNERİLEN ENİYYE YAKIN ALGORİTMA

(2.13)' de tanımlı problemin kısıtlarının oluşturduğu çokyüzlünün tüm uç noktaları tamsayı değerler almamaktadır. Bu durumda düşünülmesi gereken durum, gevşetilmiş problemin kısıtları tarafından tanımlanan çokyüzlünün tüm uç noktalarının tamsayı değerlere çekilmesinin (Lift-and-Project Algoritması) sağlanmasıdır. Tamsayı olmayan uç noktalardan kurtulmanın bir diğer yolu ise, problemin doğrusal yapısını bozmadan, kısıt kümesinde yapılacak değişiklikler ile yeni bir kısıt kümesi tanımlanmasıdır. Böylece, yeni kısıt kümesindeki eşitsizlikler tarafından oluşturulan çokyüzlünün tamsayı uç noktalara sahip olması sağlanmaktadır. Bu amaç doğrultusunda bu bölümde yeni bir algoritma önerilmektedir. Önerilen yöntem öncelikle alt-taşıyıcılara sabit sayıda bit atanması daha sonra aç gözlü algoritma ile orijinal problemin kısıtlarının sağlanması üzerine kurulmuştur.

İlk olarak OFDMA sistem içerisinde yer alan tüm alt-taşıyıcılara sabit sayıda bit yüklemesi yapılmaktadır. Tüm alt-taşıyıcılara sabit sayıda “y” bit ( $y \in \{1,2,\dots,M\}$  için) yüklendiği durum için yeni  $\tilde{\mathbf{x}}$  ve  $\tilde{\mathbf{p}}$  vektörleri tanımlanmaktadır.



**Şekil 4.1:**  $\tilde{\mathbf{x}}^T$  ve  $\tilde{\mathbf{p}}^T$  vektörlerinin genel yapısı

Şekil 4.1 incelendiğinde  $\tilde{\mathbf{x}}$  ve  $\tilde{\mathbf{p}}$  vektörlerinin elemanlarının, (2.13)' de verilen  $\bar{\mathbf{x}}$  ve  $\bar{\mathbf{p}}$  vektörlerinin elemanları ile aynı olduğu görülmektedir. Fakat  $\bar{\mathbf{x}}$ ,  $\bar{\mathbf{p}}$  vektörleri KNM boyutuna sahip iken,  $\tilde{\mathbf{x}}$ ,  $\tilde{\mathbf{p}}$  vektörleri ise KN boyutludur.

Tüm alt-taşıyıcılara “y” bit atanması durumu için problemin yeni veri hızı kısıt vektörü  $\tilde{\mathbf{A}}_u$  aşağıdaki gibi tanımlanmaktadır.

$$\tilde{\mathbf{A}}_u = \begin{bmatrix} \tilde{\mathbf{a}}_{K1} & \cdots & \tilde{\mathbf{a}}_{K1} \\ \tilde{\mathbf{a}}_{K2} & \cdots & \tilde{\mathbf{a}}_{K2} \\ \vdots & \ddots & \vdots \\ \tilde{\mathbf{a}}_{KK} & \cdots & \tilde{\mathbf{a}}_{KK} \end{bmatrix} \in \mathbb{Z}^{K \times KN}$$

Burada

$$\begin{cases} \tilde{\mathbf{a}}_{K1} = [y \ 0 \ \cdots \ 0] \in \mathbb{Z}^{1 \times K} \\ \tilde{\mathbf{a}}_{K2} = [0 \ y \ \cdots \ 0] \in \mathbb{Z}^{1 \times K} \\ \vdots \\ \tilde{\mathbf{a}}_{KK} = [0 \ 0 \ \cdots \ y] \in \mathbb{Z}^{1 \times K} \end{cases}$$

olarak verilmektedir.  $\tilde{\mathbf{A}}_u$  matrisi aynı zamanda  $\mathbf{A}_u$  matrisinin ilgili sütun vektörleri ile de tanımlanabilir.

$\mathbf{A}_u^i$ :  $\mathbf{A}_u$  matrisinin i-inci sütun vektörü olduğunda

$$\tilde{\mathbf{A}}_u = \left[ \mathbf{A}_u^y \ \mathbf{A}_u^{y+M} \ \mathbf{A}_u^{y+2M} \ \cdots \ \mathbf{A}_u^{y+(KN-2)M} \ \mathbf{A}_u^{y+(KN-1)M} \right]_{K \times KN}, i \in (y, y+M, \dots, y+(KN-1)M)$$

olarak tanımlanabilmektedir.

$\tilde{\mathbf{A}}_u$ , veri hızı kısıt matrisinin  $\tilde{\mathbf{x}}$ , bit-güç tahsisi vektörü ile çarpımı  $\tilde{\mathbf{A}}_u \tilde{\mathbf{x}} \geq \mathbf{r}$  kısıtını sağlamak zorundadır. Burada kullanılan  $\mathbf{r}$  veri hızı kısıt vektörü

$r_k = y \cdot s_k$  olduğu durumda

$$\mathbf{r} = y [s_1 \ s_2 \ \cdots \ s_K] \in \mathbb{Z}_+^{K \times 1}$$

olarak tanımlanmaktadır. Burada verilen  $s_k$ , tüm  $k=1,2,\dots,K$  için, her bir kullanıcıya tahsis edilecek olan alt-taşıyıcı sayısını belirtmektedir. Bu nedenle, kullanıcılara verilen  $r_k$  bit veri hızı kısıtı için

$s_k = \frac{r_k}{y}$  ifadesi tamsayı değerler almadığı durumda, tüm  $k \in (1,2,\dots,K)$  değerleri için,

$s_k$  değeri en yakın bir sonraki tamsayı değere çekilmektedir.

$\mathbf{s} = [s_1 \ s_2 \ \dots \ s_K] \in \mathbb{Z}_+^{K \times 1}$  için  $\mathbf{r} = \mathbf{y}\mathbf{s}$  olarak verilmektedir.

Alt-taşıyıcılara sabit sayıda bit ataması yapıldığı durum için yeni  $\tilde{\mathbf{A}}_c$  kısıt matrisi aşağıdaki gibi tanımlanmaktadır.

$$\tilde{\mathbf{A}}_c = \begin{bmatrix} \bar{1}_c & \bar{0}_c & \dots & \bar{0}_c \\ \bar{0}_c & \bar{1}_c & \dots & \bar{0}_c \\ \vdots & \vdots & \ddots & \vdots \\ \bar{0}_c & \bar{0}_c & \dots & \bar{1}_c \end{bmatrix} \in \{0,1\}^{N \times KN}$$

burada

$$\bar{1}_c = [1 \ 1 \ \dots \ 1] \in \{1\}^{1 \times K}$$

$$\bar{0}_c = [0 \ 0 \ \dots \ 0] \in \{0\}^{1 \times K}$$

olarak verilmektedir. Her bir alt-taşıyıcının en fazla bir kullanıcı tarafından kullanılması için  $\tilde{\mathbf{A}}_c \tilde{\mathbf{x}} \leq \bar{\mathbf{c}}$  sağlanmalıdır. Burada kullanılan  $\bar{\mathbf{c}}$  vektörü (2.13)'de verilen ile aynı yapıdadır.

Yukarıda tanımlanan yeni vektör ve matrisler ile kaynak tahsisi problemi aşağıdaki gibi oluşturulmaktadır.

$$\begin{aligned} & \min \quad \tilde{\mathbf{p}}^T \tilde{\mathbf{x}} \\ & \text{kısıtlar} \quad \tilde{\mathbf{A}}_u \tilde{\mathbf{x}} \geq \bar{\mathbf{r}} \\ & \quad \quad \tilde{\mathbf{A}}_c \tilde{\mathbf{x}} \leq \bar{\mathbf{c}} \\ & \quad \quad x_{k,n,m} \in \{0,1\} \quad k=1,2,\dots,K, \ n=1,2,\dots,N \text{ ve seçilen } m=y \in \{1,2,\dots,M\}, \text{ sabit} \\ & \quad \quad \text{değerleri için.} \end{aligned} \tag{4.1}$$

Gösterim kolaylığı olması için (4.1)' de kısıt kümesinde yer alan eşitsizlikler aşağıda bir araya getirilmektedir.

$$\begin{pmatrix} \tilde{\mathbf{A}}_u \tilde{\mathbf{x}} \geq \bar{\mathbf{r}} \\ \tilde{\mathbf{A}}_c \tilde{\mathbf{x}} \leq \bar{\mathbf{c}} \end{pmatrix} = \begin{pmatrix} -\tilde{\mathbf{A}}_u \tilde{\mathbf{x}} \leq -\bar{\mathbf{r}} \\ \tilde{\mathbf{A}}_c \tilde{\mathbf{x}} \leq \bar{\mathbf{c}} \end{pmatrix} = \tilde{\mathbf{A}} \tilde{\mathbf{x}} \leq \bar{\mathbf{b}} \text{ ve } \bar{\mathbf{b}} = \begin{bmatrix} -\bar{\mathbf{r}} \\ \bar{\mathbf{c}} \end{bmatrix} \tag{4.2}$$

olmaktadır.

Böylece (4.1)' te verilen kaynak tahsis problemi son olarak aşağıdaki yapıda gösterilmektedir.

$$\begin{aligned}
& \min \quad \tilde{\mathbf{p}}^T \tilde{\mathbf{x}} \\
& \text{kısıtlar} \quad \tilde{\mathbf{A}} \tilde{\mathbf{x}} \leq \bar{\mathbf{b}} \\
& \quad x_{k,n,m} \in \{0,1\} \quad k=1,2,\dots,K, \quad n=1,2,\dots,N \text{ ve seçilen } m=y \in \{1,2,\dots,M\}, \text{ sabit} \\
& \quad \text{değerleri için.}
\end{aligned} \tag{4.3}$$

Verilen  $\tilde{\mathbf{A}} \tilde{\mathbf{x}} \leq \bar{\mathbf{b}}$  ifadesinde matris ve vektörlerin elemanları detaylı olarak Şekil 4.2' de verilmektedir.

$$\begin{bmatrix}
-1 & 0 & \dots & 0 & -1 & 0 & \dots & 0 & \dots & -1 & 0 & \dots & 0 \\
0 & -1 & \dots & 0 & 0 & -1 & \dots & 0 & \dots & 0 & -1 & \dots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \dots & -1 & 0 & 0 & \dots & -1 & \dots & 0 & 0 & \dots & -1 \\
1 & 1 & \dots & 1 & 0 & 0 & \dots & 0 & \dots & 0 & 0 & \dots & 0 \\
0 & 0 & \dots & 0 & 1 & 1 & \dots & 1 & \dots & 0 & 0 & \dots & 0 \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \dots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 & \dots & 1 & 1 & \dots & 1
\end{bmatrix}_{(K+N) \times KN}
\begin{bmatrix}
x_{1,1,y} \\
x_{2,1,y} \\
\vdots \\
x_{K,1,y} \\
x_{1,2,y} \\
x_{2,2,y} \\
\vdots \\
x_{K,2,y} \\
\vdots \\
x_{1,N,y} \\
x_{2,N,y} \\
\vdots \\
x_{K,N,y}
\end{bmatrix}_{1 \times KN}
=
\begin{bmatrix}
-s_1 \\
-s_2 \\
\vdots \\
-s_K \\
1 \\
1 \\
\vdots \\
1
\end{bmatrix}_{1 \times (K+N)}$$

Şekil 4.2:  $\tilde{\mathbf{A}} \tilde{\mathbf{x}} \leq \bar{\mathbf{b}}$  eşitsizliklerinin genel yapısı

( $\tilde{\mathbf{A}}_u$  matrisinin ve  $\bar{\mathbf{r}} = y\bar{\mathbf{s}}$  vektörünün tüm elemanları “y” ile sadeleştirilmiştir.)

Şekil 4.2 'de genel yapısı verilen  $\tilde{\mathbf{A}}$  matrisi aynı zamanda bir tümüyle birimsel matristir (**Totally Unimodular Matrix, TUM**)(Bkz. EK-1.g) . Verilen  $\bar{\mathbf{b}}$  vektörünün tüm elemanları tamsayı değerler aldığı takdirde ( $\tilde{\mathbf{A}}, \bar{\mathbf{b}}$ ) matris-vektör eşitsizlik ikilisi tarafından tanımlanan çokyüzlünün tüm uç noktaları (Birimsel Matrisin özelliğinden dolayı) tamsayı değerler almaktadır [3] [4].

OFDMA kaynak tahsis problemlerinde alt-taşıyıcılara sabit sayıda bit yüklenmesi sonucunda:

- Problemin kısıt kümesi içerisinde yer alan eşitsizlikler tarafından tanımlanan çokyüzlünün, tüm uç noktaları tamsayı değerler almaktadır.

- Kısıt kümesi tarafından tanımlanan çokyüzlünün tüm üç noktaları tamsayı değerler aldığı için, problemin doğrusal programlama algoritmaları ile çözümü sonucu,  $\vec{x}$  alt-taşıyıcı-bit tahsis vektörü tamsayı değerler almaktadır.
- (2.13)' de problemin  $K*N*M$  olan boyutu (4.1)' de  $K*N$  'ye düşürüldüğü için işlem karmaşıklığı azalmaktadır.

Yukarıdaki verilen bilgiler doğrultusunda tez kapsamında iki aşamalı bir çözüm önerilmektedir.

- İlk aşamada tüm alt-taşıyıcılara, seçilen sabit  $y$ -bit değeri atanarak, kaynak tahsis problemi Simpleks Algoritması ile çözülmektedir. Böylece, her bir kullanıcıya veri hızı kısıtını sağlayacak şekilde alt-taşıyıcı ve sabit sayıda bit tahsisi yapılmaktadır.
- İkinci aşamada, yapılan alt-taşıyıcı ve sabit sayıda bit tahsisi için (2.11)' de detayları verilen Açgözlü Algoritma çalıştırılarak kullanıcılara verilen fazla bitler alınarak eniyiye yakın çözüm elde edilmektedir.

Burada en önemli noktalardan biri alt-taşıyıcılara sabit sayıda bit ataması yapmak için gereken  $y$ -bit değerinin nasıl bulunacağıdır. Pratik senaryolar içerisinde kullanıcılara atanacak bit sayıları farklılık gösterebildiği için çeşitli senaryolar için kullanılacak olan sabit sayıda  $y$ -bit değeri aynı olmayacaktır. Tez kapsamında problemin kısıt kümesi içerisinde yer alan veri hızı kısıtı vektörü kullanılarak  $y$ -bit değeri hesaplanmakta ve sabit sayıda  $y$ -bit değerinin kullanılması '**akıllı yükleme**' olarak adlandırılmaktadır. Bu amaç doğrultusunda tez kapsamında farklı senaryolar için kullanılacak sabit  $y$ -bit değeri, aşağıdaki şekilde hesaplanmaktadır:

- 1) Tüm kullanıcılara ataması yapılacak olan en düşük bit sayıları kullanılarak, sistemde iletimi yapılacak en az toplam bit sayısı hesaplanır.
- 2) Hesaplanan en az toplam bit sayısı sistemde tanımlı toplam alt-taşıyıcı sayısına bölünerek, her bir alt-taşıyıcıda taşınacak ortalama bit sayısı hesaplanır.
- 3) Elde edilen ortalama bit sayısı bir üst tamsayı değere yuvarlanarak, hesaplaması yapılan senaryo için sabit  $y$ -bit değeri elde edilir.

## Önerilen Algoritma RA Problem Çözümü

Bölüm 2.1.2' de genel tanımı verilen hız uyarlamalı kaynak tahsis problemi tez kapsamında güç uyarlamalı problemde olduğu gibi yine iki aşamalı olarak çözülmektedir. Hız uyarlamalı problem çözümü için (4.3) ' de genel yapısı verilen güç uyarlamalı kaynak tahsis probleminden yararlanılmıştır. Hız uyarlamalı kaynak tahsisi problemi aslında alt-taşıyıcılara tam yükleme yapıldığı durumda güç limiti kısıtı,  $P_T$  göz ardı edildiği durum için güç uyarlamalı problem ile aynı sonucu vermektedir.

RA problem için önerilen yöntemde ilk olarak kullanıcıların orantılı oranları,  $\gamma_k$  ve güç limiti kısıtı,  $P_T$  kullanılarak ilgili senaryo için sabit y-bit değeri hesaplanmaktadır. Hesaplanan sabit y-bit değeri kullanılarak ilk aşamada akıllı yükleme olarak adlandırılan alt-taşıyıcı tahsisi ve bit yüklemesi MA problem ile aynı olacak şekilde gerçekleştirilmektedir. İkinci aşamada ise öncelikle ilk aşama sonrası atanmayan veya boşta kalan alt-taşıyıcılar en fazla iletim gücü harcanan kullanıcılara dağıtılmaktadır.

Yapılan dağıtım sonrasında sistemde kullanılacak olan toplam iletim gücü hesaplanmaktadır:

- Eğer toplam iletim gücü güç limiti kısıtından fazla ise ters aç gözlü algoritma (2.11) kullanılarak her bir kullanıcıda en fazla güç harcanan alt-taşıyıcılardan bitler sırası ile alt-taşıyıcıdan çıkartılmaktadır. Bu işlem güç limiti kısıtı sağlanana kadar devam etmektedir. Burada kullanıcılar arası adalet korunacak şekilde bit çıkarma işlemi gerçekleştirilmektedir.
- Fakat eğer toplam iletim gücü güç limiti kısıtından az ise bu durumda her bir kullanıcı için alt-taşıyıcılara adaleti bozmayacak şekilde aç gözlü algoritma ile bit eklenmesi işlemi gerçekleştirilmektedir. Bu işlem sistemin toplam iletim gücü, güç limiti kısıtına ulaşıncaya kadar devam etmektedir. Böylece verilen güç limiti kısıtı altında sistemde iletimi yapılan toplam bit sayısı enbüyüktür.

Burada, yine en önemli nokta akıllı yüklemenin yani sabit y-bit değerinin nasıl seçildiğidir. Çünkü seçilebilecek isabetli bir sabit y-bit değeri ile ilk aşama sonrasında

eniye çözüme yakın bir nokta elde edilebilir ve sonrasında çok kısa süre için çalışacak aç gözlü algoritma ile problemin çözüm süresinin kısaltılması planlanmaktadır.

RA problem için sabit y-bit değeri aşağıdaki anlatıldığı gibi hesaplanmaktadır:

- 1) Verilen güç limiti kısıtı,  $P_T$  için; tüm kullanıcılar ve alt-taşıyıcıların kanal katsayılarının,  $\alpha_{k,n}^2 = 1$  olduğu varsayılarak, tüm alt-taşıyıcılara  $P_T$  güç limitini geçmeyecek şekilde bir ortalama güç değeri hesaplanır.
- 2) Hesaplanan ortalama iletim gücü için  $\alpha_{k,n}^2 = 1$  kullanılarak iletimi yapılabilecek ortalama bit sayısı değeri hesaplanır.
- 3) Hesaplanan ortalama bit sayısı değeri bir sonraki en yakın tamsayı değere yuvarlanarak sabit y-bit değeri elde edilir.

Tezin bundan sonraki kısmında yer alacak benzetim sonuçları için klasik OFDMA sistemi yerine LTE benzeri bir sistem kullanılmıştır. Kaynak tahsisi problemi açısından iki sistem düşünüldüğünde birbirlerine çok benzer yapıda oldukları görülmektedir. LTE sisteminin tez kapsamında kullanılan genel parametreleri Bölüm 4.1' de anlatılmaktadır.

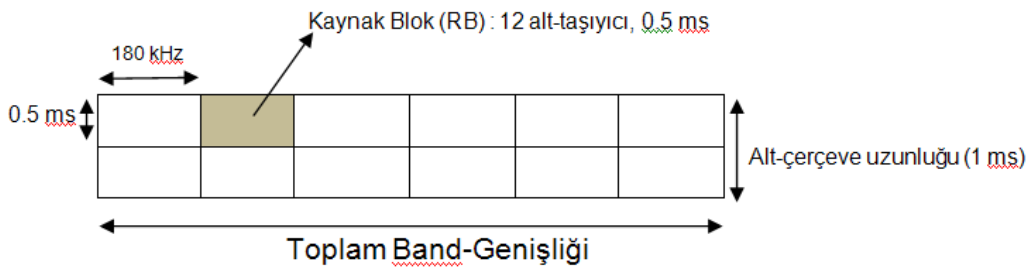


## 4.1 LTE Sistem Yapısı

Günümüzde mobil veri kullanım ihtiyacının artması, mobil TV kullanımı, çevrimiçi oyunlar gibi çok yüksek bant genişliği taleplerinin karşılanabilmesi için **Uzun Vadeli Evrim** yani **LTE (Long Term Evolution)** standartları oluşturulmuş ve günümüzde aktif olarak geliştirilmeye devam edilmektedir. LTE' nin yaklaşık olarak 2020'li yıllarda aktif bir şekilde mobil haberleşme sistemi olarak kullanılması planlanmaktadır.

LTE' de aşağı yönlü iletimde OFDM, yukarı yönlü iletimde Tek Taşıyıcılı-Frekans Bölmeli Çoklu Erişim (Single-Carrier Frequency Division Multiple Access, SC-FDMA) kullanılmasına karar verilmiştir. LTE' de kullanılan bant genişliği 1.4, 3, 5, 10, 15, 20 MHz olarak sağlanabilmektedir. Seçilen bant genişliği 15 kHz' lik bant genişliğine sahip alt-taşıyıcılara bölünmektedir. Arka arkaya gelen 12 adet alt-taşıyıcının 7 OFDM sembol süresi **Kaynak Blok (Resource Block, RB)** olarak adlandırılmaktadır. 1.4 MHz bant genişliği için 6 adet RB, 20 MHz bant genişliğinde ise 100 adet RB bulunmaktadır. Benzetimler 25 adet RB' nin bulunduğu 5 MHz' lik bant genişliği için yapılmıştır.

RB' ler LTE sisteminde frekans ve zaman alanında tahsisi yapılacak en küçük birimlerdir (Klasik OFDMA sistemde planlaması ve tahsisi yapılan en küçük birim alt-taşıyıcılarıdır.).

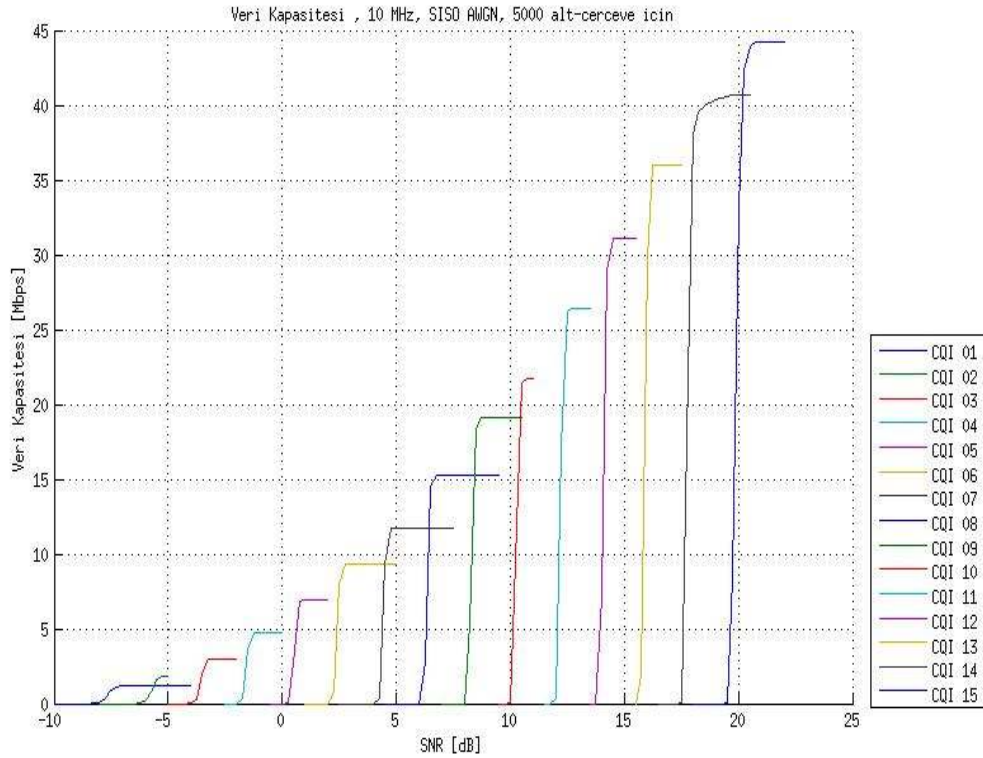


**Şekil 4.3:** 1.4 MHz Bant Genişliğinde RB Genel Yapısı

Verilen bant genişliği içerisinde yer alan tüm RB' lerin kanal katsayıları, 12 adet alt-taşıyıcının kanal katsayılarının aritmetik ortalamasının alınmasıyla elde edilmektedir [35]. Burada

$\alpha_{k,n}$  : k-inci kullanıcının n-inci RB' ye sahip olduğu kanal katsayısını ifade etmektedir.

LTE sisteminde kullanıcılara tahsis edilen RB'lerde kullanıcının sinyal gürültü oranına (SNR) göre farklı modülasyon ve kodlama yöntemleri (Modulation-and-Coding Scheme, MCS) kullanılmaktadır. Kullanıcıların farklı RB'lerde sahip oldukları kanal kalitesi, Kanal Kalite Göstergesi (Channel Quality Indicator, CQI) ile ifade edilmektedir. LTE standartlarında tanımlanmış olan 15 farklı CQI değeri bulunmaktadır. Tanımlanmış olan her bir CQI değeri için farklı kiplenim çeşidi ve kodlama<sup>(1)</sup> oranları kullanılmaktadır. [23]'te verilen LTE Simülasyonu'nda, AWGN kanal koşullarında tek kullanıcılu LTE sistemi için farklı SNR değerlerinde kullanılacak olan CQI değerleri ve buna karşılık olarak sistemin veri kapasitesinin grafiği Şekil 4.4' de verilmiştir.



**Şekil 4.4:** Farklı CQI değerlerinde SNR' a karşılık Veri Kapasitesi Grafiği

Şekil 4.4' de verilen grafikte her bir CQI değerinde veri kapasitesinin en yükseğe ulaştığı ilk noktaya karşılık gelen SNR değeri, ilgili CQI değerinin kullanılması için eşik değer kabul edilmiştir. Örneğin grafiğe göre RB' ta CQI 7 'nin kullanılabilmesi için kullanıcının sinyal gürültü oranının yaklaşık olarak 5 dB olması gerekmektedir.

<sup>1</sup> LTE sisteminde kodlama yöntemi olarak Turbo Kodlama Kullanılmaktadır.

Eğer kullanıcının bir RB için SNR değeri 4 dB ise RB' ta artık en fazla CQI 6 kullanılabilir. Bu şekilde tüm CQI değerlerinin, kullanılabilmesi için gerekli olan SNR değeri ve ilgili CQI' in kullanıldığı durumda RB' de taşınabilecek veri hızı elde edilmektedir.

Tablo 4.1' de tüm CQI değerlerine karşılık gelen kiplenim yöntemi, etkin kod oranı (**Effective Code Rate, ECR**), SNR değeri ve veri hızı değerleri bir arada toplanmıştır. Tablo 4.1' de verilen değerler Blok Hata Oranı' nın (Block Error Rate, BLER)  $10^{-3}$  olduğu durum için geçerlidir.

CQI	KİPLENİM	ECR	SNR (dB), $f_k(m_{CQI})$	Max. Veri Hızı, $m_{CQI}$ (bits/RB)
1	4QAM	0.0762	-6.6	12
2	4QAM	0.1172	-4.9	18
3	4QAM	0.1885	-3.05	29
4	4QAM	0.3008	-1.1	47
5	4QAM	0.4385	0.8	69
6	4QAM	0.5879	2.7	93
7	16QAM	0.3691	4.75	117
8	16QAM	0.4785	6.8	151
9	16QAM	0.6016	8.7	191
10	64QAM	0.4551	10.6	216
11	64QAM	0.5537	12.6	263
12	64QAM	0.6504	14.5	309
13	64QAM	0.7539	16.25	358
14	64QAM	0.8525	18.3	405
15	64QAM	0.9258	20.65	440

**Tablo 4.1:** BLER= $10^{-3}$  için Tüm CQI değerlerinde Kiplenim, Etkin Kod Oranı (ECR), SNR ve max. Veri Hızı değerleri

Benzetimlerde yukarıda genel yapısı anlatılan LTE sistemine göre kaynak tahsisi problemi incelenmiştir. Bölüm 4.1’ de genel yapısı verilen OFDMA kaynak tahsisi problemi ile LTE kaynak tahsisi problemi birbirine benzer yapıdadır. Sadece LTE Kaynak Tahsisi Problemi için aşağıdaki farklılıklar bulunmaktadır.

$N_{RB}$  : Sistemde tanımlı toplam Kaynak Blok sayısı

$M_{CQI}$  = tüm  $CQI \in \{1,2,\dots,15\}$  değerleri için Tablo 4.1’ de verilen veri hızı değerlerinin kümesi.

$m_{CQI} \in M_{CQI}$ , tüm  $cqi=\{1,2,\dots,15\}$  değerleri için RB’ lara yüklenecek veri hızı.

Örneğin, CQI 7 için veri hızı  $m_7 = 117$  bits/RB, CQI 10 için veri hızı  $m_{10} = 216$  bits/RB dir.

(4.1)’ de tanımlanan OFDMA kaynak tahsisi problemi LTE kaynak tahsisi problemine kolayca dönüştürülebilmektedir. LTE kaynak tahsisi problemi için  $\bar{\mathbf{x}}$  bit-güç tahsisi vektörü, (4.1)’ de verilen  $\tilde{\mathbf{x}}$  bit-güç tahsisi vektörünün yapısı değişmeden olduğu gibi kullanılmaktadır.  $\bar{\mathbf{p}}$  güç tahsis vektörü ise, parçalama yapısı Şekil 4.1 verildiği gibi olacak, fakat  $\tilde{\mathbf{p}}$  vektörünün elemanları  $\rho_{k,n,m}$ , Tablo 4.1’ de verilen SNR değerlerinden elde edilecektir. Öyle ki,

$$\rho_{k,n,m} = \frac{f_k(m_{CQI})}{\alpha_{k,n}^2} \quad (4.4)$$

olarak verilmekte ve  $f_k(m_{CQI})$  değeri Tablo 4.1 ‘de seçilen CQI değerine karşılık gelen SNR değerine eşit olmaktadır. Örneğin,  $BLER=10^{-3}$  ve  $\alpha_{k,n}^2=1$  olduğu durumda,  $m_9 = 191$  bit gönderebilmek için  $f_k(m_9) = 8.7$  dB güce ihtiyaç vardır.

Farklı kanal koşulları için Tablo 4.1 den alınacak olan SNR değeri (4.4) ‘ da  $f_k(m_{CQI})$  yerine konularak, ilgili kanalın katsayısı  $\alpha_{k,n}$ ’ye göre gerekli olan güç hesaplanmaktadır. Sistemde yer alan tüm kullanıcıların  $N_{RB}=25$  RB’ de seçilen sabit CQI değerine göre  $\bar{\mathbf{p}}$  güç tahsis vektörü yukarıda anlatıldığı gibi oluşturulacaktır.

LTE kaynak tahsisi problemi için yeni oluşturulacak  $\mathbf{A}_u$  matrisi, (4.1)’ de verilen  $\tilde{\mathbf{A}}_u$  matrisi, verilen kullanıcı sayısı ve RB sayısına göre, sabit sayıda bit ataması yapılabilmesi için seçilen  $m_{CQI}$  değerine,  $m=m_{CQI}$  olacak şekilde oluşturulacaktır.

LTE kaynak tahsisi problemi için tanımlanacak  $\mathbf{A}_c, \bar{\mathbf{r}}$  ve  $\bar{\mathbf{c}}$  matris ve vektörleri ise (4.1)'de verilen  $\tilde{\mathbf{A}}_c, \bar{\mathbf{r}}$  ve  $\bar{\mathbf{c}}$  matris ve vektörleri ile aynı yapıda kullanılacaktır.

## 5. BENZETİM SONUÇLARI

Bu bölümde yapılan tüm benzetimler LTE sistemi parametreleri ile gerçekleştirilmiştir. LTE sisteminin 5 MHz 'lik bant genişliği ve  $N_{RB}=25$  (kaynak blok sayısı) olduğu durum kullanılmıştır. Sistemde yer alan kullanıcıların ortalama hızlarının 10m/s olduğu varsayılmaktadır. Bu durum altında kullanılan kanal VehA olarak seçilmiştir. Benzetimlerde kullanılan tüm kanallar [23]' te verilen LTE Simülatörü kullanılarak elde edilmiştir.

Kaynak Tahsis problemi için önerilen algoritmanın performansı Bölüm 2.1' de detayları verilen doğrusal programlama ile yapılan kaynak tahsisi sonrasında Aç Gözlü Algoritma ile bit ve güç yüklemesinin yapıldığı metoda göre karşılaştırma yapılarak ortaya çıkartılmıştır. Sonraki kısımlarda yer alan grafikler üzerinde bu yöntem '**LP+Greedy**' ( Doğrusal Programlama + Aç Gözlü Algoritma ) olarak adlandırılmaktadır. Bu yöntemin performansının gevşetilmiş doğrusal programlamaya oldukça yakın olduğu bilinmektedir [13]. Sürekli uzayda elde edilen gevşetilmiş doğrusal programlama çözümü, Tamsayı Programlama problemleri için iyi bir kestirim olmaktadır. Bu nedenle bu yöntem, önerilen algoritmanın performans kıyaslaması için iyi bir referanstır. Ayrıca, LP+Greedy ve önerilen algoritma dışında problem serpiştirmeli kaynak tahsisi (interleaved resource allocation) yöntemi ile de çözülmüştür. Bu yöntemde göre alt-taşıyıcılar, kanal bilgisine bakılmaksızın önceden belirlenmiş bir sıraya göre kullanıcılara atanmaktadır. Benzetimlerde hem Güç Uyarlamalı (MA) hem de Hız Uyarlamalı (RA) probleme göre önerilen algoritmanın performansı incelenmiştir. Yapılan benzetimler için  $N_0=1$  olarak kabul edilmiştir.  $N_0=1$  kabul edildiği için grafiklerde ifade edilen toplam güç aslında SNR değerine eşit olmaktadır. Bu nedenle tüm grafiklerde toplam güç için dB birimi kullanılmıştır.

Benzetimler Intel Core İ7 2.2 GHz işlemciye sahip bilgisayarda, Linux İşletim sistemi altında, MATLAB ortamında gerçekleştirilmiştir. Benzetimlerde, problemin çözümünde kullanılan doğrusal programlama algoritması olarak kullanılan Simpleks Algoritması [2],[3],[4] seçilmiştir. Tüm benzetimlerde Simpleks Algoritması için GLPK/MEX (GNU Doğrusal Programlama Kiti) [28] doğrusal programlama kiti kullanılmıştır.

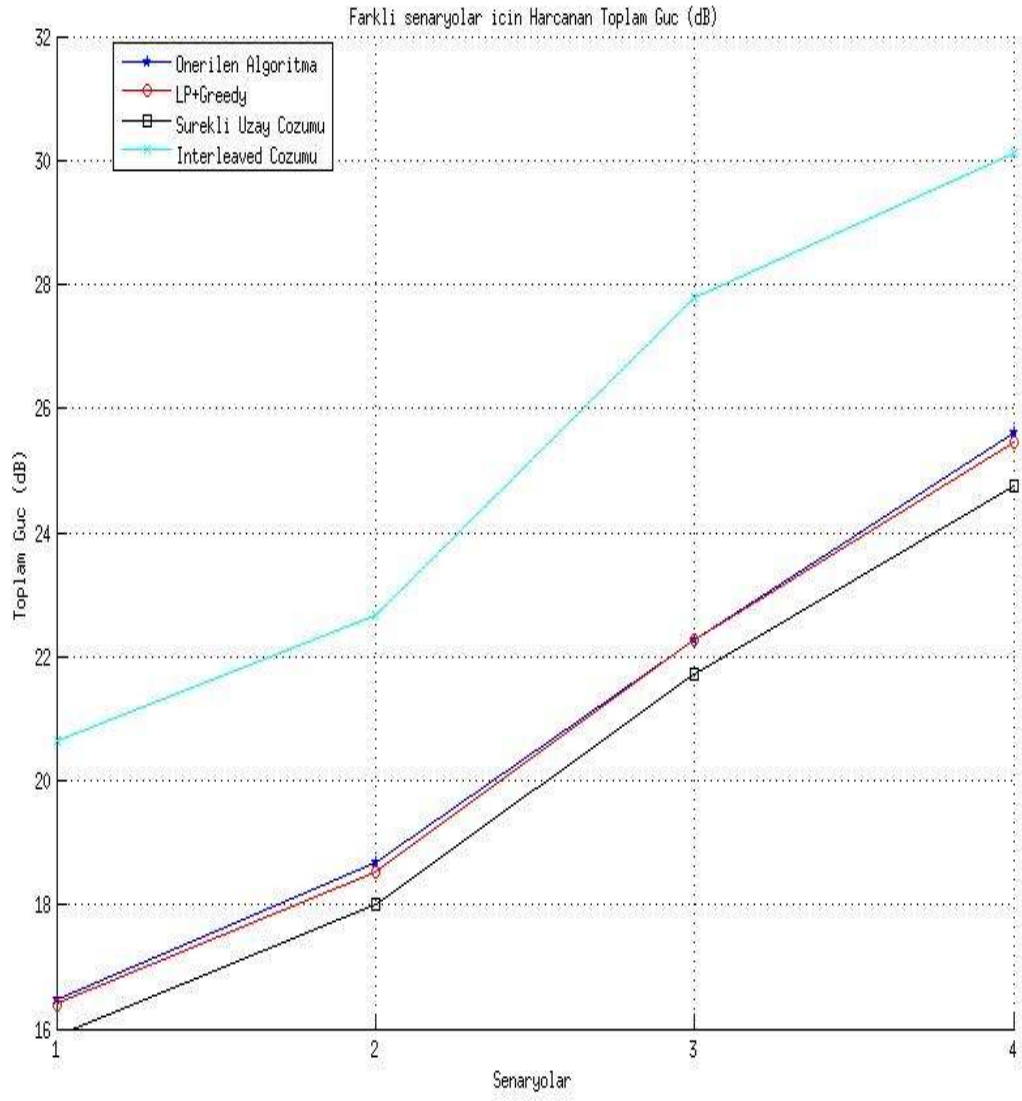
## 5.1 MA Problem Benzetimleri

İlk olarak güç uyarlamalı kaynak tahsis problemi (MA) aşağıda verilen kullanıcı bit değerlerine göre çözülmüştür. Tablo 5.1' de K=8 kullanıcı bir LTE sistem için dört farklı senaryo tanımlanmıştır. Her senaryonun yanında kullanıcıların her birine en az tahsis edilmesi gereken bit sayıları yer almaktadır. Verilen dört farklı senaryo için MA problemi önerilen algoritmaya, LP+Greedy algoritmasına, sürekli uzaya ve Interleaved yani serpiştirme yöntemlerine göre çözülmüştür.

Kullanıcı	1	2	3	4	5	6	7	8	Toplam Bit Sayısı
Senaryo 1	200	200	200	200	400	400	400	400	2400
Senaryo 2	400	400	400	400	400	400	400	400	3200
Senaryo 3	400	400	400	400	800	800	800	800	4800
Senaryo 4	800	800	800	800	800	800	800	800	6400

**Tablo 5.1:** K=8 Kullanıcı için Verilen Farklı Bit Yükleme Senaryoları

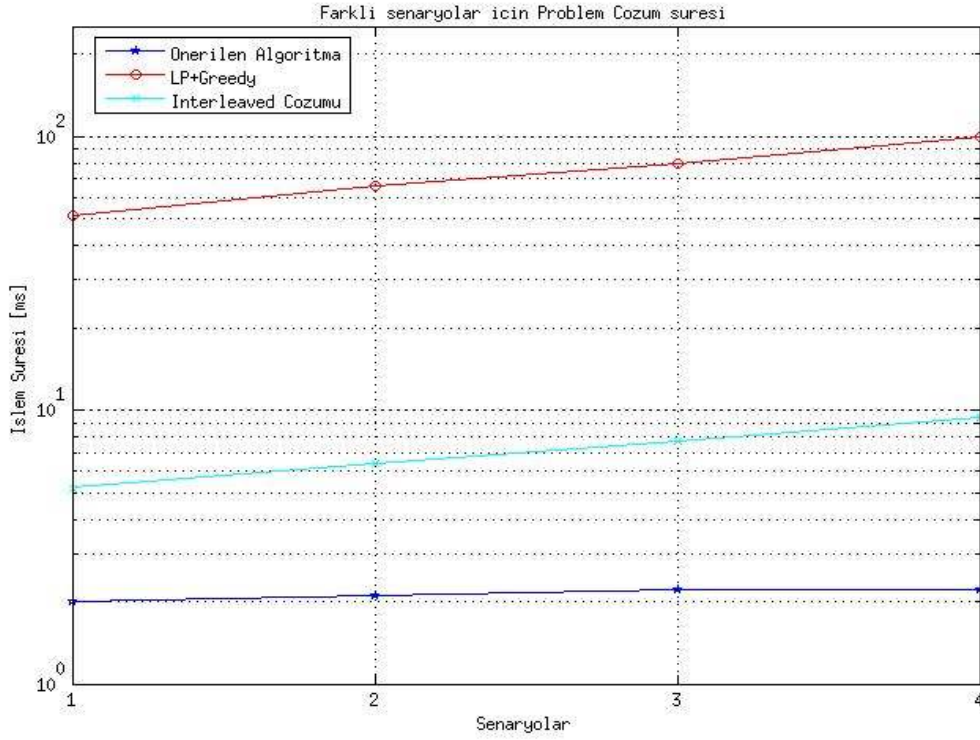
Şekil 5.1 'de Tablo 5.1' de verilen senaryolar için toplam verici güç miktarları karşılaştırılmıştır. Burada, önerilen algoritmanın diğer yöntemlere oldukça yakın sonuçlar elde ettiği görülmektedir. Önerilen algoritmanın, [13]' te verilen LP+Greedy yöntemi ile neredeyse aynı sonuçları elde ettiği görülmektedir. Ayrıca Interleaved çözümün önerilen yöntemden ortalama ~5 dB daha kötü sonuç verdiği görülmektedir. Verilen senaryolar için problemin eniyi çözümü grafik üzerinde sürekli uzay çözümü ile önerilen LP+Greedy yöntemi arasında yer aldığı bilinmektedir. Bu nedenle önerilen algoritma ile elde edilen çözüm eniyi çözüme oldukça yakın bir sonuç vermektedir.



**Şekil 5.1:** Tablo 5.1' de Verilen Senaryolar için Toplam Verici Gücü



Şekil 5.2' de Şekil 5.1' de elde edilen çözümler için işlem süreleri karşılaştırılmaktadır. Burada önerilen algoritmanın LP+Greedy metodundan oldukça hızlı olduğu görülmektedir. Önerilen algoritma, LP+Greedy ile neredeyse aynı sonucu vermesine rağmen, işlem süresinin önerilen algortmada çok düşük olduğu gözlenmektedir. Önerilen algoritma ile problemin tanımlı olduğu uzayın boyutu LP+Greedy algoritmasına göre M kat daha düşük olduğundan ve ilk aşamada hem kaynak tahsisi hem de sabit sayıda bit yüklemesi yaptığı için problemin işlem süresi oldukça kısaltılmıştır.



**Şekil 5.2:** Tablo 5.1'de Verilen Senaryolar için Ortalama İşlem Süresi

Şekil 5.2' de elde edilen problem çözüm süreleri(ms) Tablo 5.2'de verilmiştir.

		Senaryo 1	Senaryo 2	Senaryo 3	Senaryo 4
<b>Önerilen Algoritma</b>	LP (ms)	1.5	1.5	1.5	1.5
	Greedy (ms)	0.8	0.6	0.7	0.7
	Total (ms)	2.3	2.1	2.2	2.2
<b>LP+Greedy</b>	LP (ms)	30.4	40.6	50.8	63.9
	Greedy (ms)	3.1	4.1	5.5	6.0
	Total (ms)	33.5	44.7	56.3	69.9
<b>Interleaved Cozum</b>	Greedy (ms)	3.2	4.2	5.2	6.2

**Tablo 5.2:** Tanımlanan Dört Farklı Senaryo İçin İşlem Süreleri (ms)

Tablo 5.1 'de verilen dört farklı senaryo için kullanıcı sayısı sabit kaldığından problem boyutu da değişmemektedir. Bu nedenle, Tablo 5.2 'de verilen süreler bakıldığında önerilen algoritma için ilk aşama doğrusal programlama için çözüm sürelerinin dört farklı senaryoda da aynı olduğu görülmektedir. Tablo 5.1 'e göre senaryo 1' den senaryo 4' e doğru iletilen toplam bit sayısı artmaktadır. Bu durumda Tablo 5.2' de üç algoritmanın aç gözlü algoritma süreleri incelendiğinde; LP+Greedy ve Interleaved yönteminin toplam bit sayısı arttıkça aç gözlü algoritmanın süresinin arttığı fakat önerilen algoritma da ise aç gözlü algoritma süresinin neredeyse değişmediği görülmektedir. Bu durum önerilen algoritma içerisinde kullanılan akıllı yükleme nedeni ile oluşmaktadır. Burada akıllı yükleme ile eniyiye yakın bir noktaya gidilmekte bu nedenle aç gözlü algoritma ile çok fazla bit eklenmesine gerek kalmadan problemin çözümü elde edilmektedir.

	Senaryo 1		Senaryo 2		Senaryo 3		Senaryo 4	
	Greedy Süre (ms)	Greedy'de atanan toplam bit sayısı	Greedy Süre (ms)	Greedy'de atanan toplam bit sayısı	Greedy Süre (ms)	Greedy'de atanan toplam bit sayısı	Greedy Süre (ms)	Greedy'de atanan toplam bit sayısı
<b>Akıllı Yükleme</b>	0.8	290	0.6	271	0.7	304	0.7	760
<b>0' dan Başlama</b>	2.4	2516	2.7	3328	3.3	4910	4.1	6592
<b>Yarım Yükleme</b>	0.8	290	1.3	546	0.8	2072	1.9	3840
<b>Tam Yükleme</b>	3.3	8041	3	7205	2.3	5677	2.3	3904

**Tablo 5.3:** Tablo 5.1' de Tanımlanan Dört Farklı Senaryo İçin Önerilen Algoritma Farklı Yükleme Sonrası Kaynak Tahsis Problemi Çözüm Süresi (ms) (LP işlem süresi tüm senaryolar için aynı ve 1,5 ms 'dir.)

Tablo 5.1' de verilen dört farklı senaryo için önerilen algoritma da akıllı yükleme yerine RB' lere ilk aşamada bit yüklenmemesi, tam yükleme yani CQI=15 seçilmesi ve yarım yükleme, CQI=7 için, aç gözlü algoritma süreleri ve ne kadar bit eklenip çıkartıldığı incelenmiş ve sonuçlar Tablo 5.3' de verilmiştir. 25 RB 'nin tamamı ilk aşamada atandığı durumda sistemde gönderilebilecek toplam bit sayısı 11000 bit/altçerçeve olmaktadır. Fakat verilen senaryolarda veri hızı kısıtlarına göre problemin doğrusal programlama çözümü ile tüm alt-taşıyıcılar RB 'lara atanmadığından dolayı aç gözlü algoritma ile atanan bit sayısı ile senaryo için ataması yapılacak bit sayısı toplamı 11000 değerine eşit olmamaktadır. Burada açgözlü algoritma ile en az bit ekleme veya çıkarma işleminin akıllı yükleme ya-

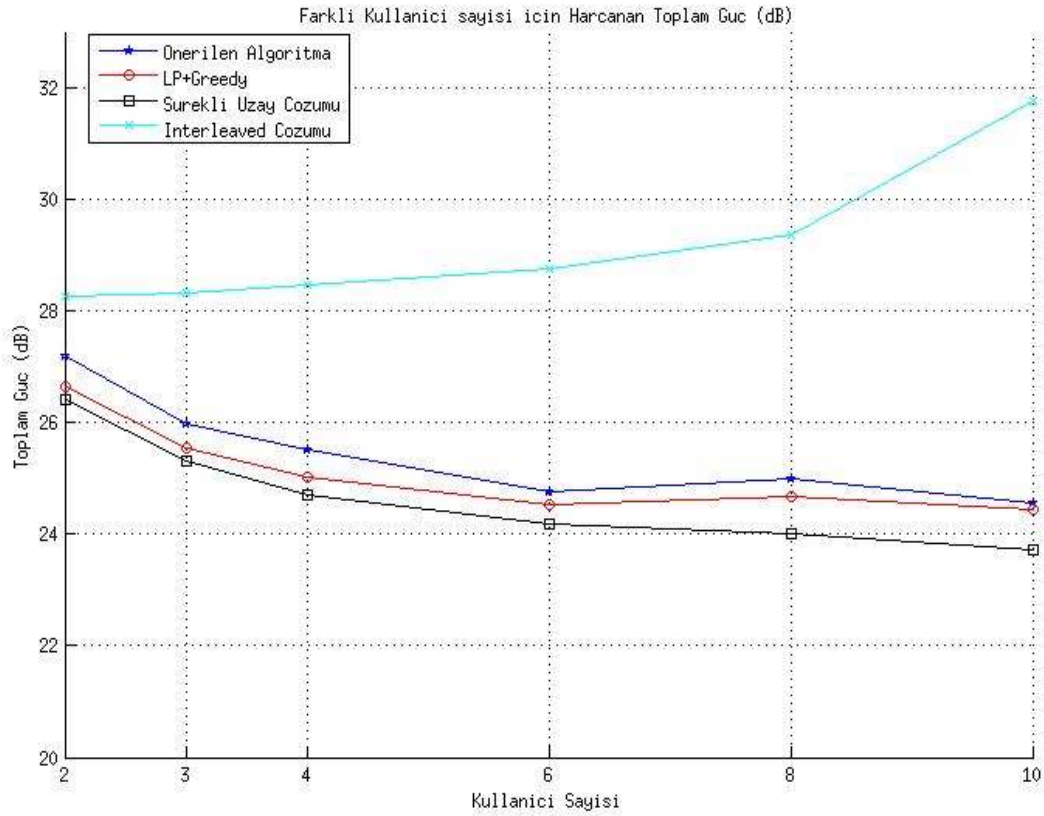
pılması sonucu elde edildiği görülmektedir. RB' lara yarım yükleme yapılması ise verilen senaryolar için başlangıç için bit yüklemesi yapılmaması ve tam yükleme yapılmasına göre az bit ekleme çıkarma işleminin gerçekleştirildiği görülmektedir.

Tablo 5.4' de tanımlanmış olan altı farklı senaryo bulunmaktadır. Burada her bir senaryoda sistemde bulunan kullanıcı sayısı ve kullanıcılara yüklenecek bit miktarları görülmektedir. Her bir senaryo için kullanıcı sayıları ve kullanıcılara yüklenecek bit miktarları değişse bile, toplam bit miktarının sabit olduğu görülmektedir.

Senaryo 1	K=2	3000 3000
Senaryo 2	K=3	2000 2000 2000
Senaryo 3	K=4	1500 1500 1500 1500
Senaryo 4	K=6	1000 1000 1000 1000 1000 1000
Senaryo 5	K=8	750 750 750 750 750 750 750 750
Senaryo 6	K=10	600 600 600 600 600 600 600 600 600 600

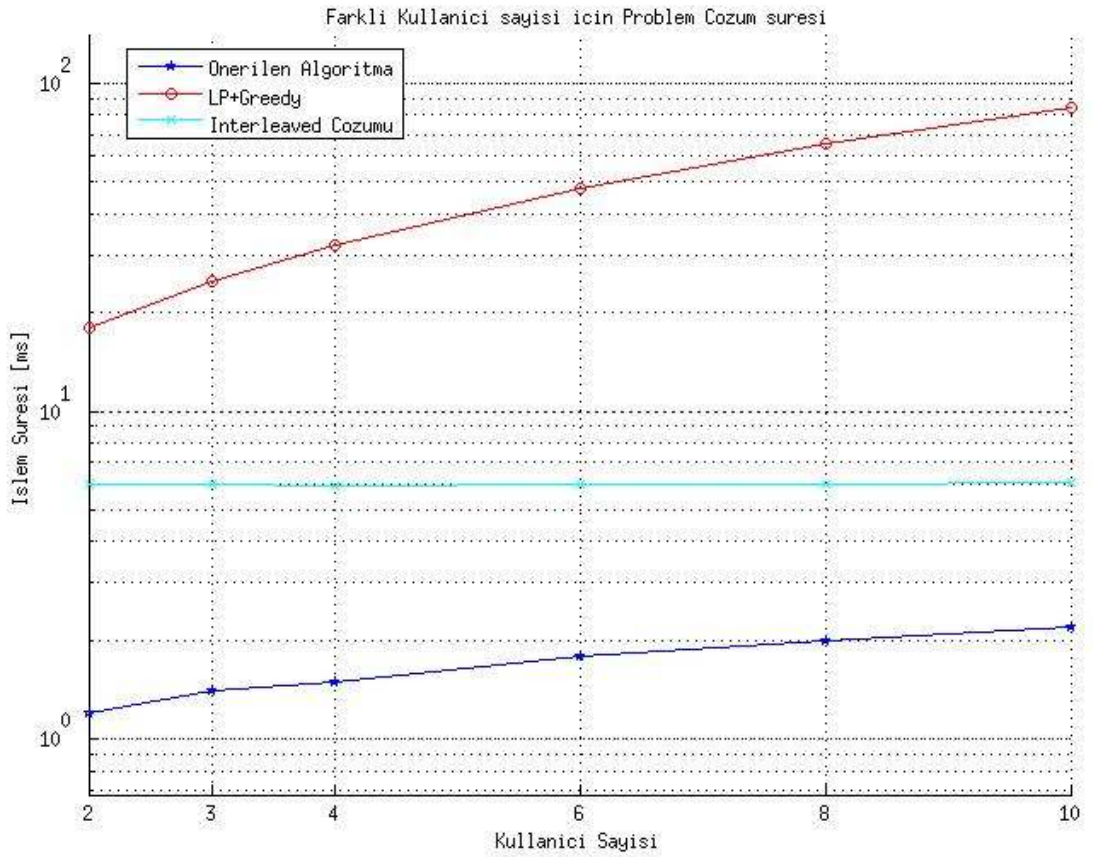
**Tablo 5.4:** Tanımlanan Altı Farklı Senaryo İçin Kullanıcı Sayısı ve Herbir Kullanıcıya Yüklenecek Bit Miktarı

Şekil 5.3' te verilen grafik Tablo 5.4' de verilen farklı kullanıcı sayıları için harcancak toplam verici gücünü göstermektedir. Önerilen algoritma sürekli uzay çözümüne göre ~0.6 dB daha kötü performans göstermektedir. Buna rağmen LP+Greedy metoduna yine oldukça yakın bir performans elde etmektedir. Burada dikkat çeken bir diğer nokta ise, toplamda gönderilecek bit miktarı sabit kaldığı halde toplam verici gücünün kullanıcı sayısı arttıkça azalmasıdır. Bunun nedeni kullanıcı sayısının artması ile sistemin çeşitliliğinin (diversity) artmasıdır. Serpiştirme metodu ile çözümde ise, kullanıcı sayısı arttıkça toplam iletim gücünün de paralel olarak arttığı görülmektedir. Serpiştirme metodunda daha az kullanıcı olması durumunda yapılan statik kaynak tahsisi sonucunda kullanıcıların iyi kanal katsayılarına sahip RB'ları kullanma olasılıkları daha yüksek olmaktadır. Kullanıcı sayısı arttıkça bu olasılık azalmaktadır. Bu nedenle, serpiştirme metodun performansı, sistemdeki kullanıcı sayısı arttıkça azalmaktadır.



### Şekil 5.3: Farklı Kullanıcı Sayısı (bkz. Tablo 5.4) İçin Harcanacak Toplam Güç

Şekil 5.4' teki grafikte bir önceki grafikte elde edilen çözümler için işlem süreleri yer almaktadır. Burada verilen tüm senaryolarda önerilen algoritmanın işlem süresi LP+Greedy metodunun işlem süresinden oldukça az olduğu görülmektedir. K=2 kullanıcıli durumda bile önerilen algoritmanın işlem süresi LP+Greedy metodundan ortalama 15 kat daha azdır. Burada bir diğer önemli nokta ise kullanıcı sayısı arttıkça LP+Greedy metodunun işlem süresinin hızla artmasıdır. Fakat önerilen algoritmanın işlem süresinde çok fazla bir değişim gözlenmemektedir.



Şekil 5.4: Farklı Kullanıcı Sayısı için (bkz. Tablo 5.4) İşlem Süresi Grafiği

Şekil 5.4' de elde edilen problem çözüm süreleri(ms) detaylı olarak aşağıda verilmiştir.

		Senaryo 1	Senaryo 2	Senaryo 3	Senaryo 4	Senaryo 5	Senaryo 6
<b>Önerilen Algoritma</b>	LP (ms)	0.7	0.8	1	1.2	1.5	1.5
	Greedy (ms)	0.5	0.6	0.5	0.6	0.5	1.1
	Total (ms)	1.2	1.4	1.5	1.8	2.0	2.6
<b>LP+Greedy</b>	LP (ms)	15.2	21.2	28.3	43.9	61.9	80.8
	Greedy (ms)	2.8	3.7	3.7	3.7	3.8	3.7
	Total (ms)	18	24.9	32.0	47.6	65.7	84.5
<b>Interleaved</b>	Greedy (ms)	6	6	5.9	6	6	6.1

**Tablo 5.5:** Tablo 5.4' de Tanımlanan Altı Farklı Senaryo İçin İşlem Süreleri

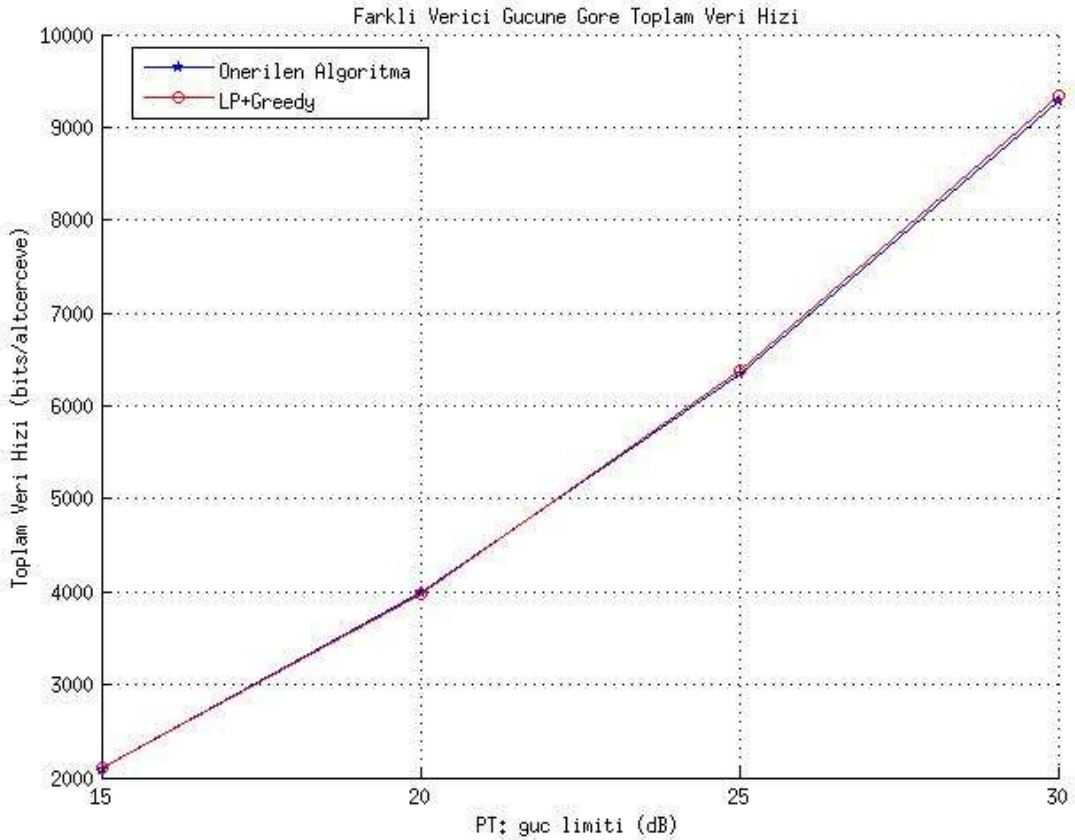
Her bir senaryoda sistemde gönderilen toplam bit sayısı sabit olduğu için serpiştirme metodunun işlem süresinin farklı senaryolar için neredeyse değişmediği görülmektedir. Burada önerilen algoritma ve LP+Greedy metodunun işlem sürelerini kullanıcı sayısının belirlediği açıkça görülmektedir. Senaryoların toplam iletim bit sayıları sabit olduğundan problem çözümlerinin aç gözlü algoritma süreleri çok fazla değişkenlik göstermemektedir. Fakat doğrusal programlama (LP) aşamalarının süreleri kullanıcı sayısı ile doğru orantılı olarak artmaktadır. Önerilen algoritma için problemin boyutu 15 kat düşürüldüğü için önerilen algoritmanın kullanıcı sayısının değişimine daha az hassasiyet göstermektedir.

Sonuç olarak, MA problemi için yapılan benzetimlerde önerilen algoritma karşılaştırma yapılan Doğrusal Programlama-Açgözlü algoritma (LP+Greedy) metoduna oldukça yakın değerler elde etmektedir. Bununla birlikte önerilen algoritmanın çalışma süresi hem LP+Greedy hemde serpiştirme metoduna göre oldukça kısa olduğu görülmektedir. Böylece MA problemleri için önerilen algoritma ile çok kısa sürede en iyiye yakın çözüm elde edilebilmektedir.

## 5.2 RA Problem Benzetimleri

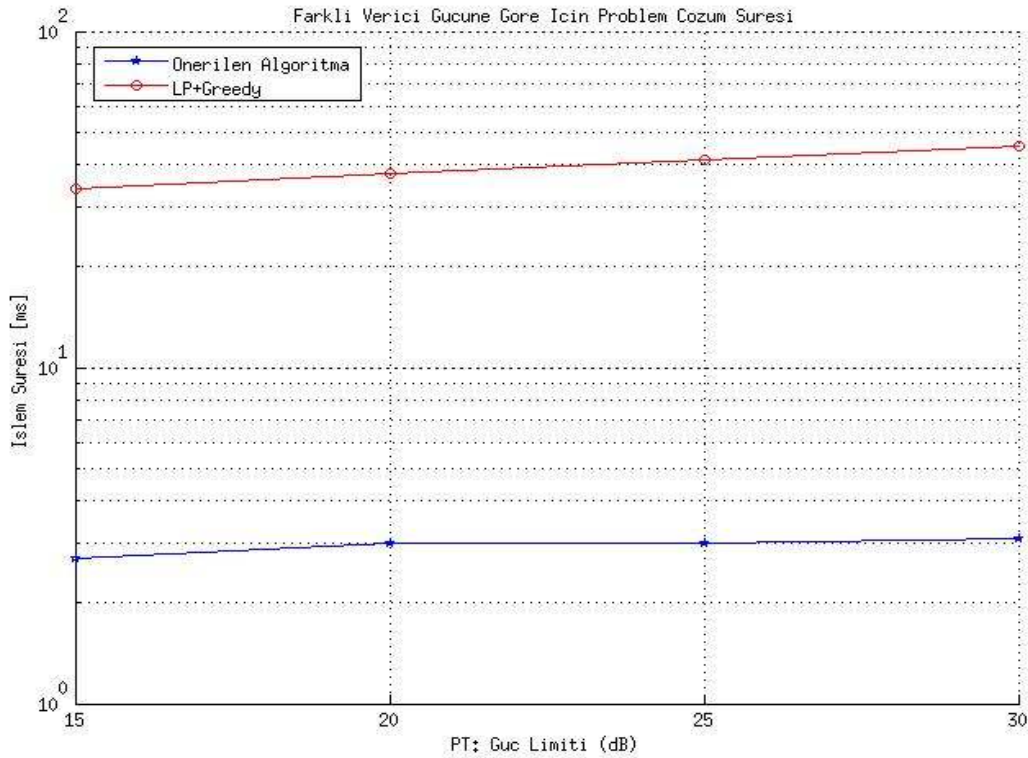
Bölüm 2.1.2' de tanımlanan RA problemi için farklı senaryolar oluşturularak benzetimler yapılmaktadır.

Şekil 5.5 ' de K=8 kullanıcının yer aldığı ve kullanıcılar arası adaletin sağlandığı bir sistem için benzetim yapılmıştır. Burada kullanıcılar arası orantılı oranın 1:1:1:1:1:1:1 olduğu koşulda, farklı güç kısıtları altında bir alt çerçeve süresinde iletimi yapılan toplam veri hızı değerleri görülmektedir. Burada önerilen algoritma ile LP+Greedy yönteminin birbirlerine çok benzer sonuçlar elde ettiği görülmektedir.



**Şekil 5.5:** K=8 ve Kullanıcıların Orantılı Oranlarının 1:1:1:1:1:1:1 Olduğu Durumda Farklı Güç Kısıtları Altında Toplam Veri Hızı

Şekil 5.5' te elde edilen sonuçlar için, önerilen algoritma ile LP+Greedy metodunun çalışma süreleri Şekil 5.6 'da karşılaştırılmaktadır. Burada önerilen algoritma farklı güç limitleri için ortalama 3 ms süresinde kaynak tahsisini yapabilirken, LP+Greedy metodu ile problemim çözüm süresi ortalama 30-40 ms sürmektedir. Önerilen algoritma veri hızı bakımından LP+Greedy metoduna göre yaklaşık olarak aynı performansı göstermesine karşılık ortalama 10-15 kat daha kısa sürede kaynak tahsisini yapabilmektedir.



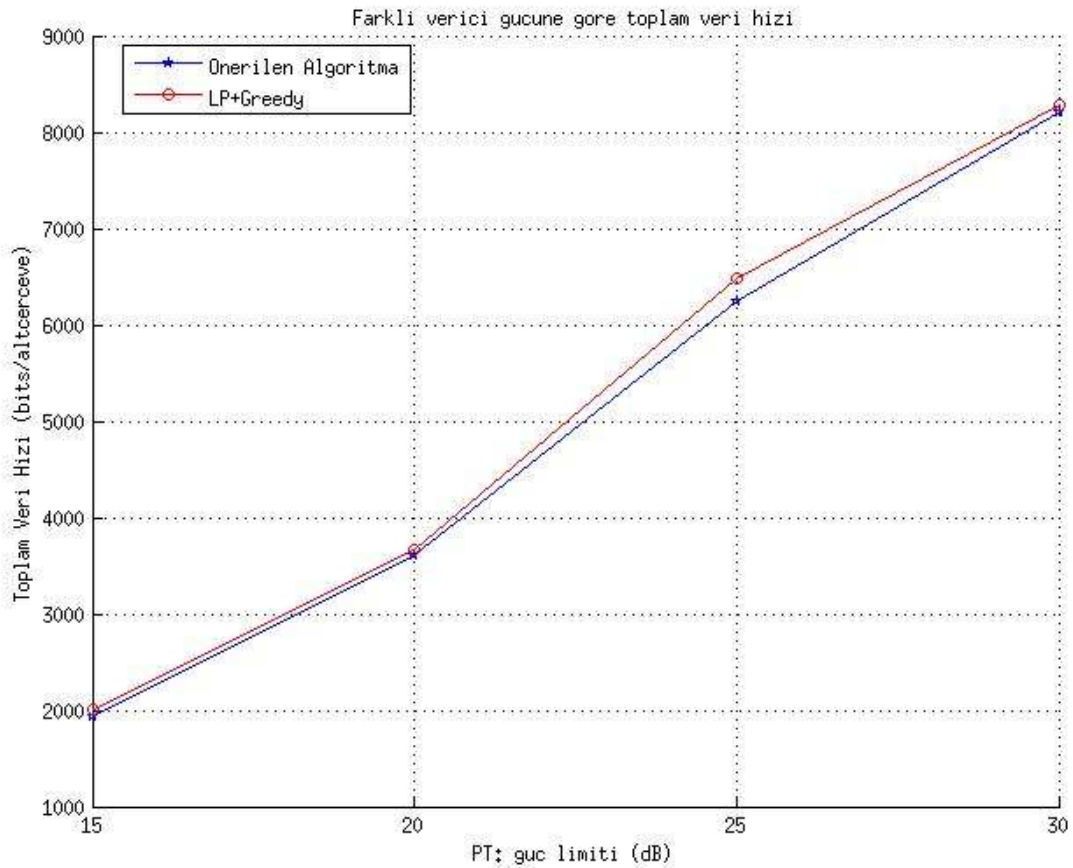
**Şekil 5.6:** K=8 ve Kullanıcıların Orantılı Oranlarının 1:1:1:1:1:1:1 Olduğu Durumda Farklı Güç Kısıtları Altında Kaynak Tahsisi Süresi

		PT=15 dB	PT=20 dB	PT=25 dB	PT=30 dB
<b>Önerilen Algoritma</b>	LP (ms)	1.5	1.6	1.5	1.5
	Greedy (ms)	1.2	1.4	1.5	1.6
	Total (ms)	2.7	3.0	3.0	3.1
<b>LP+Greedy</b>	LP (ms)	27.3	27.2	37.6	27.2
	Greedy (ms)	6.8	10.4	13.7	17.6
	Total (ms)	34.1	37.6	41.3	45.6

**Tablo 5.6:** Şekil 5.6' da Verilen Grafik İçin Detaylı İşlem Süreleri

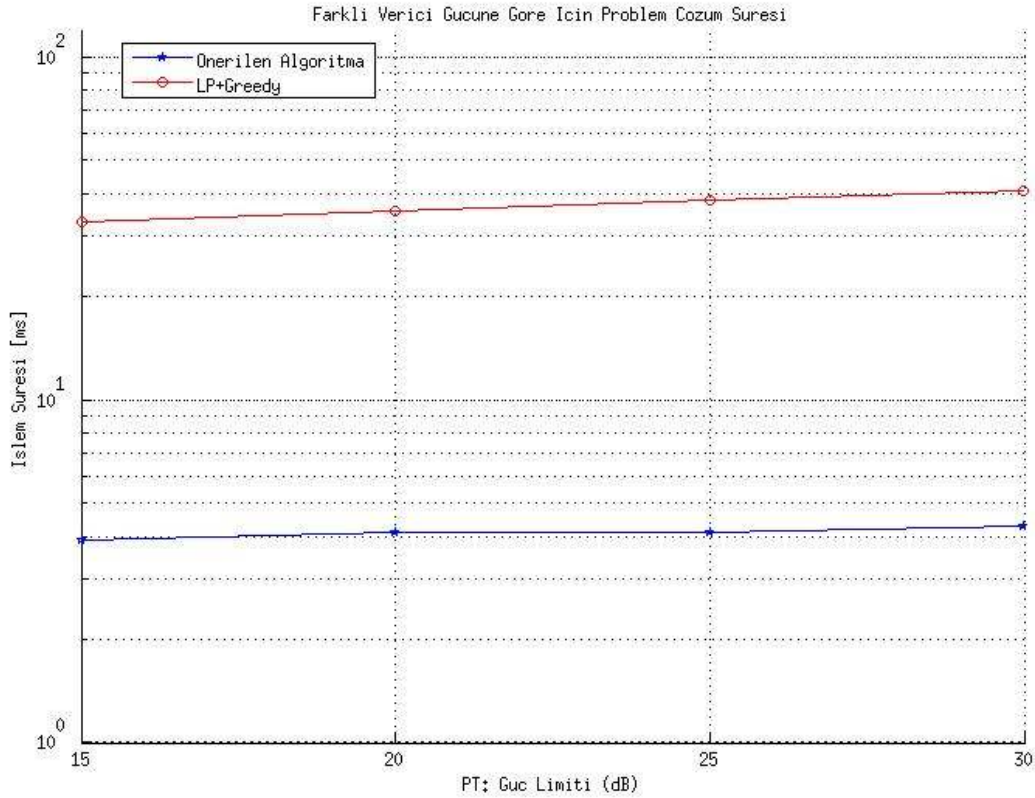


Şekil 5.7’ de K=8 kullanıcının olduğu bir sistemde kullanıcıların orantılı oranlarının 4:4:2:2:1:1:1:1 olarak verildiği koşulda farklı iletim gücü kısıtı altında gerçekleştirilen benzetim sonuçları verilmektedir. Kullanıcılar arası orantılı oranın farklı olması durumunda önerilen algoritma ile LP+Greedy algoritması sonucu bir altçerçeve süresinde iletilen toplam veri hızlarının oldukça yakın değerlere sahip olduğu görülmektedir. Şekil 5.5 ‘de verilen eşit orantılı oran koşulunda  $PT=30$  dB için, sistemin toplam veri hızı kapasitesi ~9300 bite kadar çıkabilirken, farklı orantılı oran koşulunda yaklaşık olarak ~8300 bit olduğu görülmektedir. Buna göre kullanıcıların farklı orantılı oranlara sahip olmaları sistemin toplam veri hızı kapasitesini düşürebilmektedir.



**Şekil 5.7:** K=8 ve Kullanıcıların Orantılı Oranlarının 4:4:2:2:1:1:1:1 Olduğu Durumda Farklı Güç Kısıtları Altında Toplam Veri Hızı

Şekil 5.8 'de, Şekil 5.7 için yapılan benzetimlerin işlem süresi yer almaktadır. Burada LP+Greedy yöntemi ile kaynak tahsisi yapılması ortalama olarak 35-40 ms civarında gerçekleşmektedir. Fakat önerilen algoritma için bu süre ortalama 4 ms civarında olmaktadır. Kullanıcıların orantılı oranlarının farklı olduğu durum içinde önerilen algoritma LP+greedy metoduna göre ortalama 10 kat daha az sürede benzer performansa sahip olmaktadır.

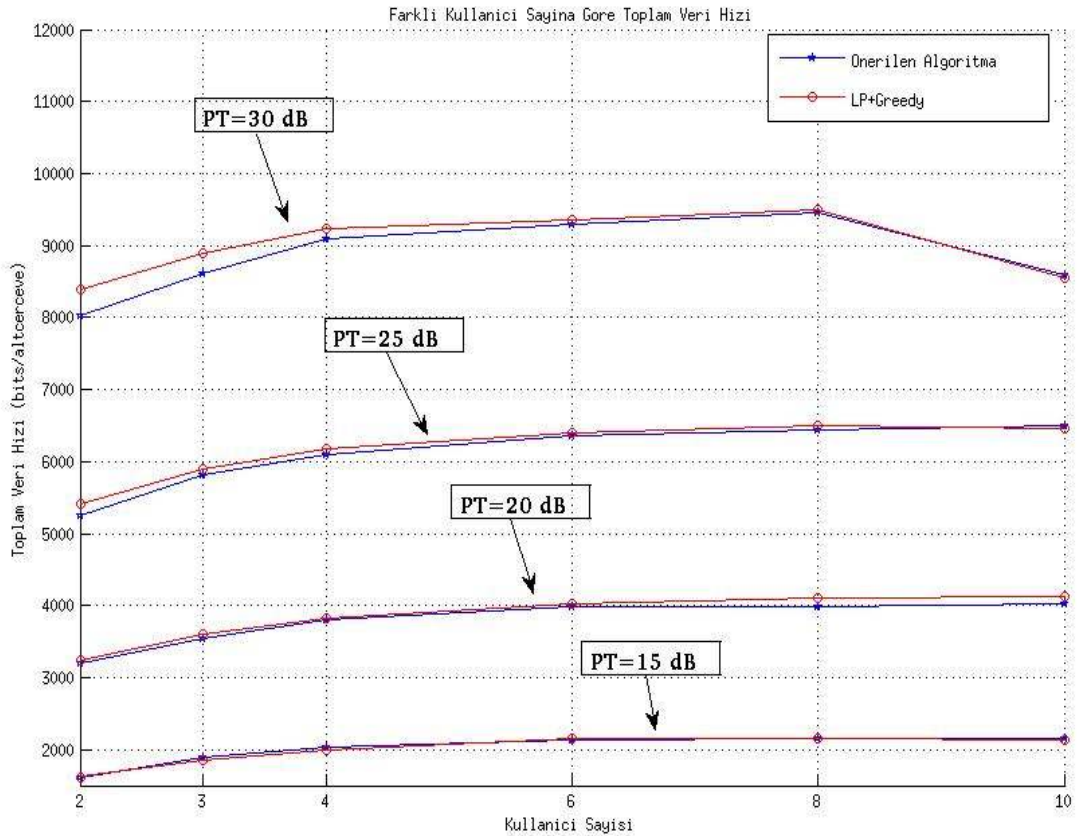


**Şekil 5.8:** K=8 ve Kullanıcıların Orantılı Oranlarının 4:4:2:2:1:1:1:1 Olduğu Durumda Farklı Güç Kısıtları Altında Kaynak Tahsisi Süresi

		PT=15 dB	PT=20 dB	PT=25 dB	PT=30 dB
<b>Önerilen Algoritma</b>	LP (ms)	1.2	1.6	1.5	1.6
	Greedy (ms)	2.7	2.5	2.6	2.7
	Total (ms)	3.9	4.1	4.1	4.3
<b>LP+Greedy</b>	LP (ms)	27.8	27.8	27.8	27.8
	Greedy (ms)	5.2	7.8	10.4	13.0
	Total (ms)	33.0	35.6	38.2	40.8

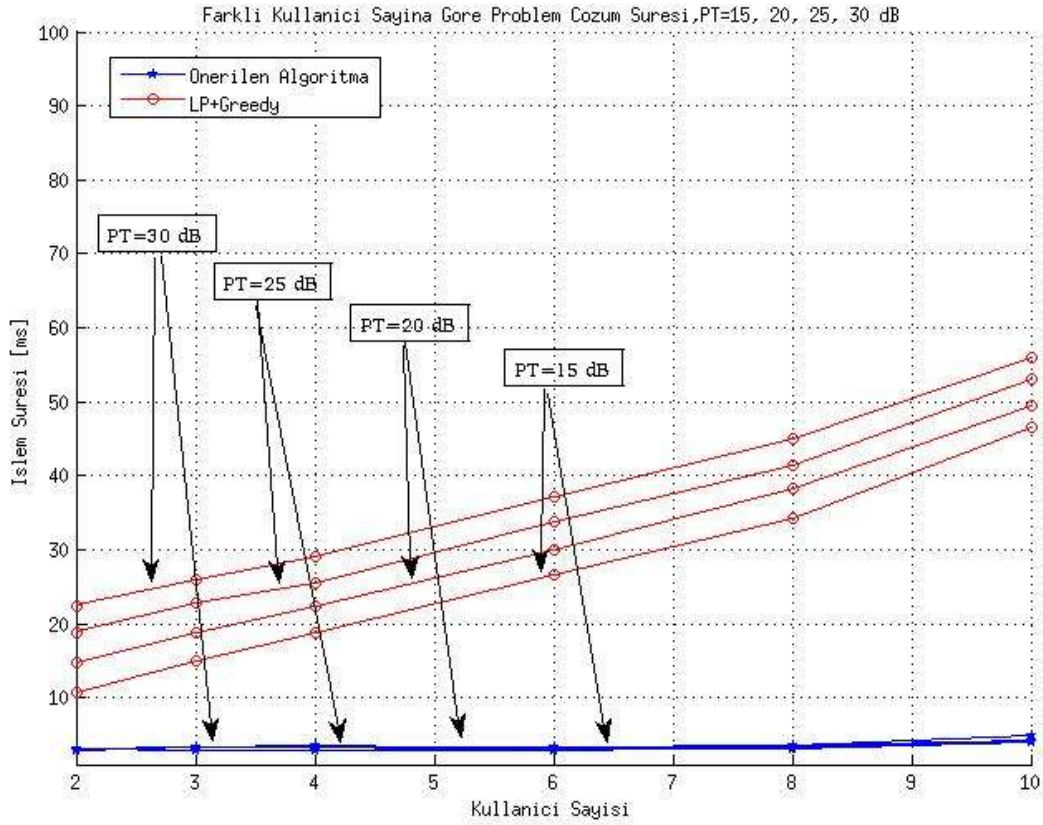
**Tablo 5.7:** Şekil 5.8' de Verilen Grafik İçin Detaylı İşlem Süreleri

RA probleminin eşit orantılı oran koşulunda farklı güç limiti kısıtları altında kullanıcı sayısına bağlı olarak sistemin toplam veri hızının incelenmesi Şekil 5.9' da verilmiştir. Burada sistemin güç limiti kısıtı  $P_T=15, 20, 25, 30$  dB olduğu durumlarda sistemde yer alan farklı kullanıcı sayısına göre ( $K=2, 3, 4, 6, 8, 10$  için) bir altçerçeve boyunca sistemin toplam iletim veri hızı değişimi incelenmiştir. Önceki senaryolara benzer şekilde önerilen algoritma ile LP+Greedy metodu ile yaklaşık olarak aynı performansı göstermektedir.  $P_T=30$  dB iletim güç kısıtı altında  $K=8$  için sistem toplam veri hızı kapasitesi eniyi değerine sahip iken diğer iletim güç kısıtları altında, toplam veri hızı kapasitesi  $K=10$  kullanıcı için eniyi değerine sahip olmaktadır.  $P_T=30$  dB kısıtında  $K=8$  kullanıcı için her bir kullanıcıya ortalama 3 RB tahsis edilebilmekteyken,  $K=10$  olduğu durumda her bir kullanıcıya atanan RB sayısının 2'ye düşmektedir. Bu nedenle bu farklılık ortaya çıkmaktadır.



**Şekil 5.9:** Eşit Orantılı Oran Koşulunda ve Farklı Güç Limiti Kısıtlarında Kullanıcı Sayısına göre Bir Altçerçeve Süresinde Sistemin Toplam Veri Hızı

Şekil 5.9 'da benzetimleri yapılan senaryolar için karşılaştırması yapılan iki algoritmanın işlem süreleri Şekil 5.10 'da verilmektedir. LP+Greedy metodu kullanıldığı zaman işlem süresi kullanıcı sayısına ve güç limiti kısıtının değerine göre değişiklik göstermektedir. Örneğin  $K=2$  ve  $P_T=20$  dB olduğu durumda işlem süresi  $\sim 14.7$  ms seviyelerinde fakat  $K=10$  ve  $P_T=20$  dB için işlem süresi  $\sim 47$  ms seviyelerinde olmaktadır. Diğer taraftan önerilen algoritma kullanıldığında işlem süresinin kullanıcı sayısına ve güç limiti kısıtına bağlı olarak çok fazla değişmediği görülmektedir.



**Şekil 5.10:** Eşit Orantılı Oran Koşulunda ve Farklı Güç Limiti Kısıtlarında Kullanıcı Sayısına göre İşlem Süresi

Şekil 5.10' da elde edilen problem çözüm süreleri (ms) detaylı olarak aşağıda verilmiştir.

PT=15	K=2	K=3	K=4	K=6	K=8	K=10
Önerilen Algoritma (ms)	3.2	3.4	3.2	3.2	3.6	5.0
LP+Greedy (ms)	10.8	15.1	18.7	26.7	34.2	46.5

PT=20	K=2	K=3	K=4	K=6	K=8	K=10
Önerilen Algoritma (ms)	3.1	2.8	3	3.3	3.5	4.3
LP+Greedy (ms)	14.7	18.9	22.4	30.1	38.3	49.6

PT=25	K=2	K=3	K=4	K=6	K=8	K=10
Önerilen Algoritma (ms)	3.0	3.3	3.5	3.2	3.1	4.0
LP+Greedy (ms)	18.9	22.8	25.6	33.8	41.4	53.0

PT=30	K=2	K=3	K=4	K=6	K=8	K=10
Önerilen Algoritma (ms)	3.1	3.0	2.9	3.0	3.4	4.4
LP+Greedy (ms)	22.5	26.0	29.2	37.2	45.0	55.9

**Tablo 5.8:** Şekil 5.10' da Verilen Grafik İçin Detaylı İşlem Süreleri

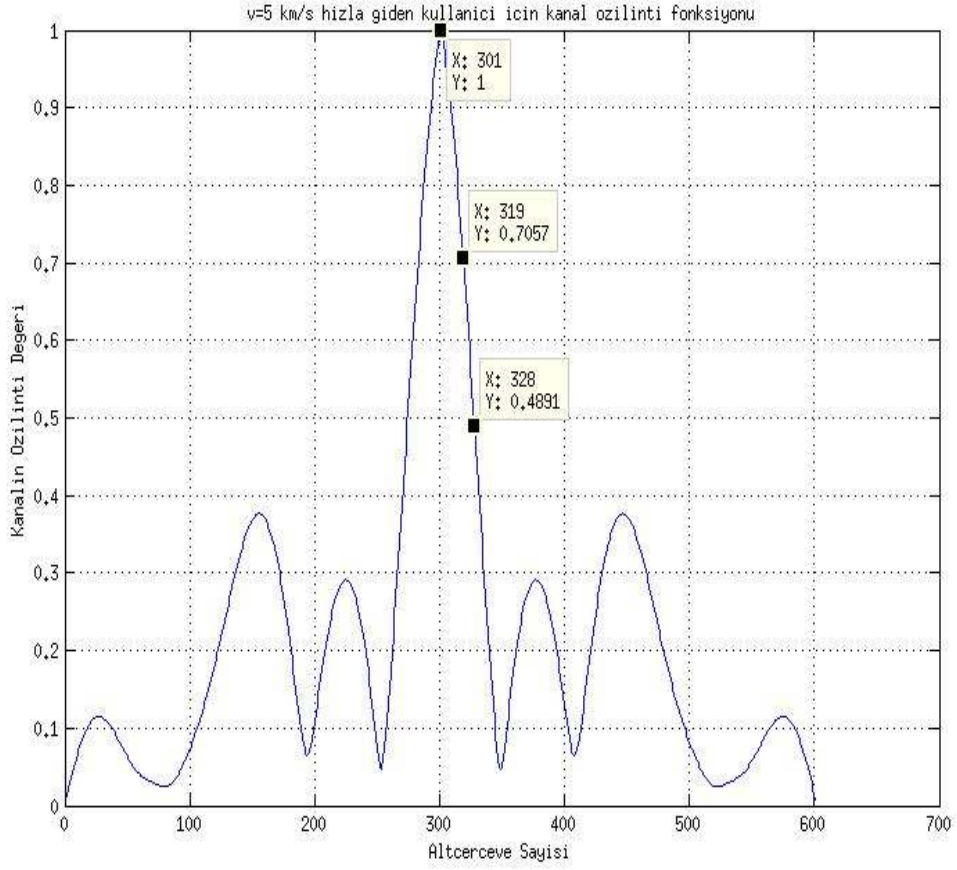
RA problemi için iki farklı algoritma karşılaştırıldığında, önerilen algoritma ile LP+Greedy metoduna göre toplam veri hızı olarak neredeyse aynı performansa sahip olmaktadır. Bununla birlikte iki algoritma işlem süresine göre karşılaştırıldığında, önerilen algoritmanın işlem süresi neredeyse LP+Greedy algoritmasının işlem süresinin yaklaşık %10 'una karşılık gelmektedir. Bu nedenle önerilen algoritma hem veri hızı açısından hemde işlem süresi açısından LP+Greedy metoduna üstünlük sağlamaktadır.

### 5.3 Kanal Özilinti İşlevine Göre Kaynak Tahsisi Benzetimleri

Kaynak Tahsisi problemlerinde işlem süresinin azaltılabilmesi için kanalın bir önceki altçerçeve zamanına olan bağımlılığı kullanılarak yeni bir yöntem önerilmektedir. Sistemden faydalanan kullanıcı sayısı sabit kaldığı sürece kaynak tahsisi probleminin kısıt kümesinde yer alan eşitsizliklerin tanımladığı çokyüzlüde bir değişiklik olmamaktadır. Bu durumda kaynak tahsisi probleminin maliyet işlevi kanalın değişimine bağlı olarak farklı sonuçlar vermektedir. Bölüm 4.3' te anlatıldığı gibi kaynak tahsisi probleminin çözümü bu çokyüzlünün tamsayı olan uç noktalarından biri olmaktadır. Kanalın zamanla değişiminin yavaş olduğu durumu düşünelim.  $t_0$  altçerçeve zamanı için yapılan kaynak tahsisi probleminin çözümü çokyüzlünün bir uç noktası olan  $x^1$  olarak verildiğinde,  $t_1$  altçerçeve zamanı için kaynak tahsisi probleminin çözümü çokyüzlünün başka bir uç noktası olan  $x^2$  olsun. Kanalın zamanla değişimi yavaş olduğu durumda, çokyüzlü üzerinde  $x^2$  uç noktasının  $x^1$  uç noktasına yakın bir yerde bulunacağı varsayılmaktadır. Eğer kanalın zamana göre değişimi oldukça yavaş ise  $x^1$  ve  $x^2$  uç noktaları aynı yüzlem üzerinde bile yer alabilmektedir. Bu durumda kaynak tahsisi problemini  $t_1$  anı için çözerken  $t_0$  zamanında elde edilen çözümü, yani  $x^1$  değerini çözüm için başlangıç noktası olarak kullandığımızda, kaynak tahsisi probleminin çok daha hızlı sürede çözülebileceği düşünülmektedir.

Tez kapsamında kaynak tahsisi problemi için kanalın özilinti işlevi dikkate alınarak bir analiz gerçekleştirilmiştir. Sistemde kullanılan kanalın değişim hızı azaldığında, kanalın bir önceki alt-çerçeve zamanındaki kanala olan bağımlılığı artmaktadır.

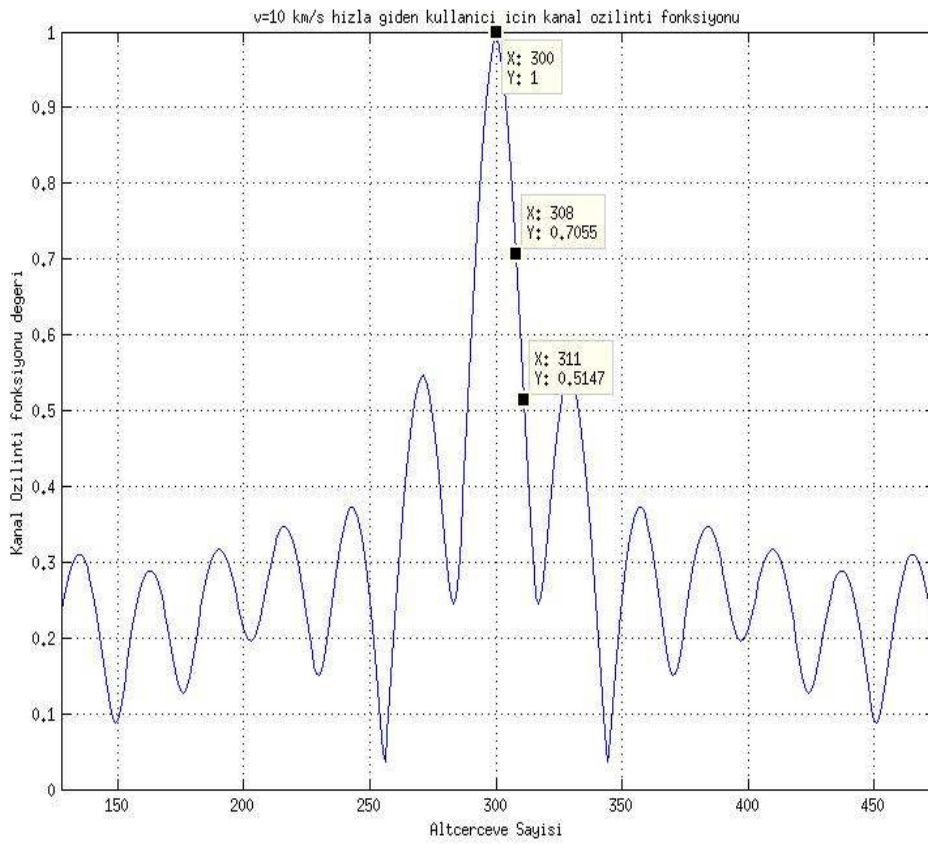
Kullanıcının farklı hızlara sahip olduğu durumlar için [23]' de verilen LTE simülasyonu kullanılarak, zamanla değişimi farklı olan kanallar oluşturulmuştur.



**Şekil 5.11:** 5 km/saat hızla giden bir kullanıcının kanal özilinti fonksiyonu

Şekil 5.11’ de kullanıcının hızının 5 km/saat olduğu durum için sahip olduğu kanalın özilinti işlevinin zamanla değişimi gösterilmektedir. Burada verilen grafiğe göre kanalın en az 19 altçerçeve zamanı boyunca ilintili olduğu görülmektedir.

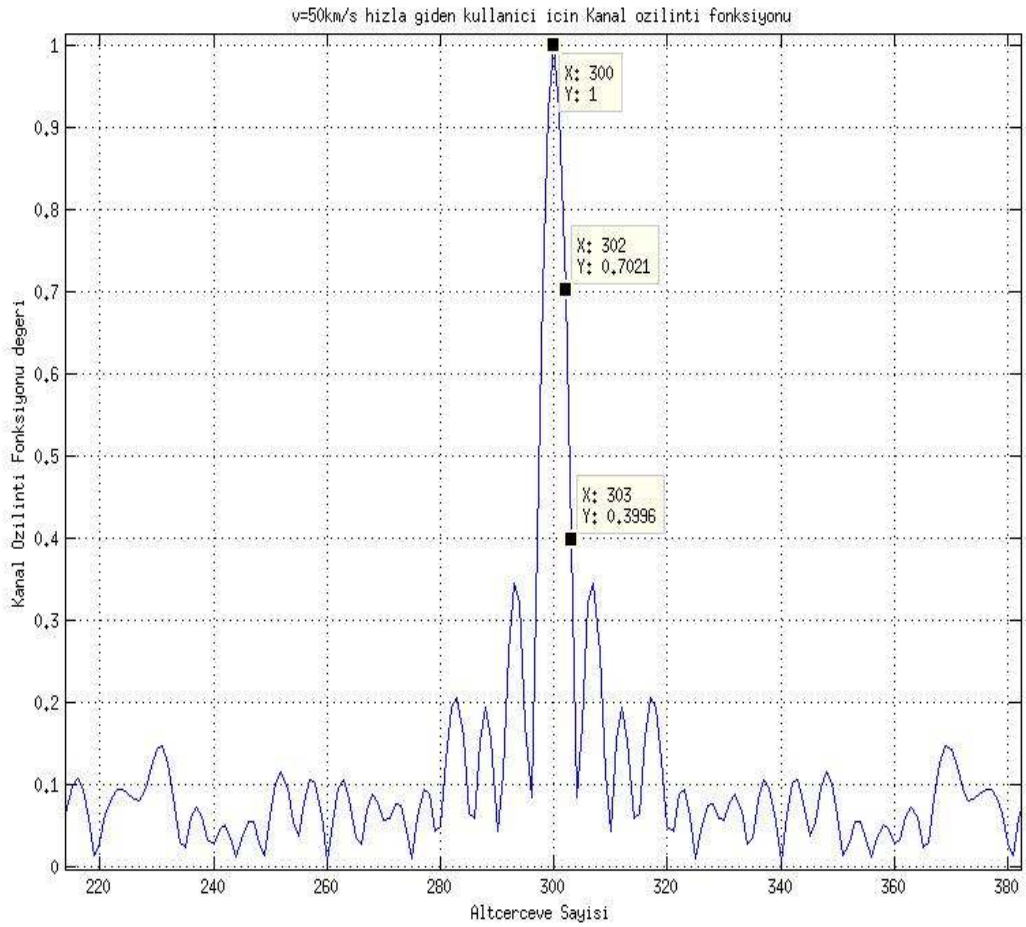
Şekil 5.12' de kullanıcının hızının 10 km/saat olduğu durumda sahip olduğu kanalın özilinti işlevinin zamanla değişimi gösterilmektedir. Bu durumda kanalın en az 9 altçerçeve zamanı boyunca ilintili olduğu görülmektedir. Kullanıcının hızı arttıkça sahip olduğu kanalın zamanla değişimi arttığından kanalın önceki altçerçeve zamanlarındaki kanal ile olan ilintisi azalmaktadır.



**Şekil 5.12:** 10 km/saat hızla giden bir kullanıcının kanal özilinti fonksiyonu

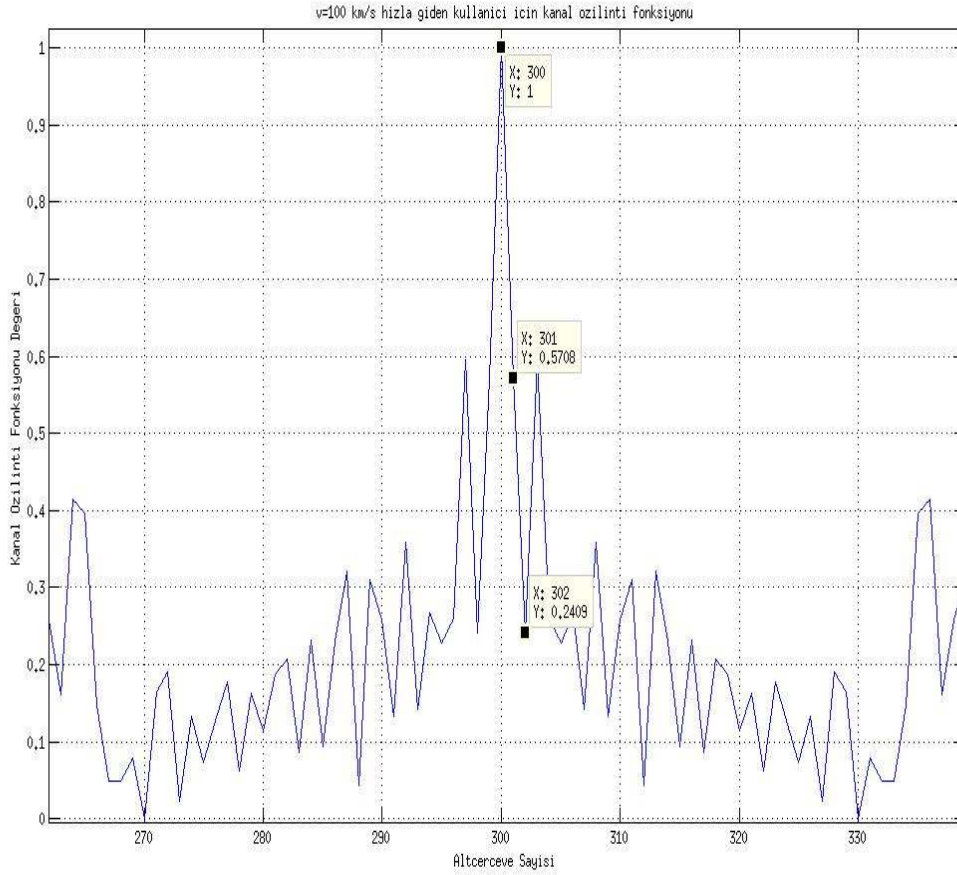


Şekil 5.13' de 50 km/saat hızla giden bir kullanıcının sahip olduğu kanalın özilinti değeri 300 altçerçeve zamanı için verilmektedir. Burada kanalın en az 2 altçerçeve (subframe) zamanı boyunca bir önceki altçerçeve zamanına ilintili olduğu görülmektedir.



**Şekil 5.13:** 50 km/saat hızla giden bir kullanıcının kanal özilinti fonksiyonu

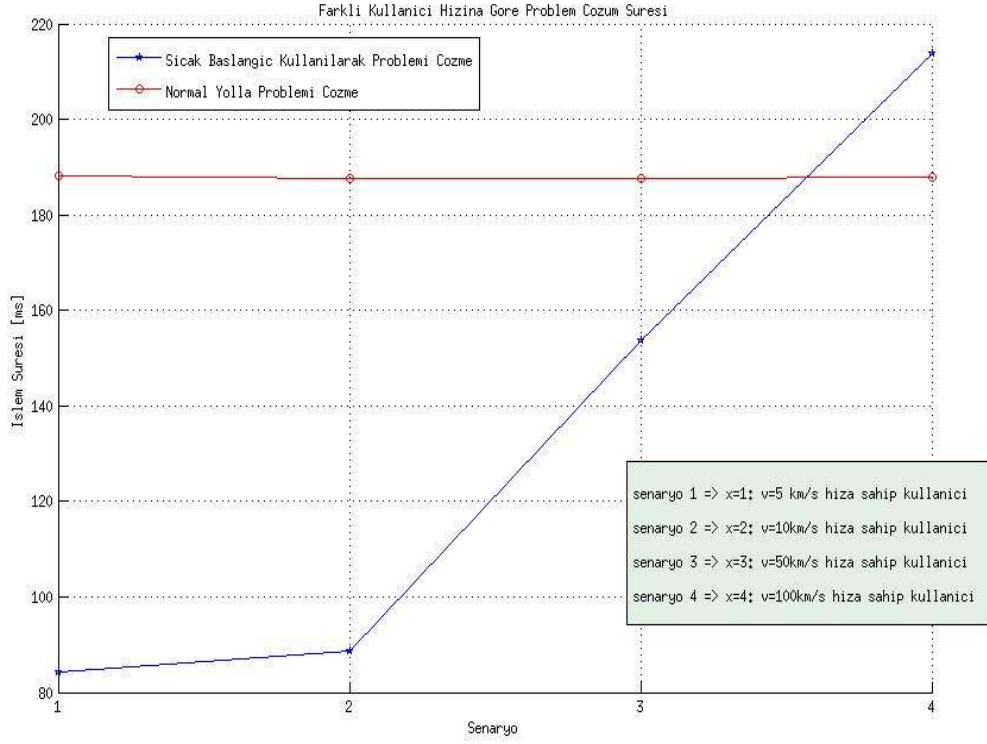
Son olarak Şekil 5.14' de 100 km/saat hızla giden bir kullanıcının sahip olduğu kanalın özilinti değeri 300 altçerçeve zamanı için verilmektedir. Burada kanalın artık bir önceki altçerçeve zamanından tamamen ilintisiz hale geldiği görülmektedir



**Şekil 5.14:** 100 km/saat hızla giden bir kullanıcının kanal özilinti fonksiyonu

Şekil 5.11 ile Şekil 5.14 arasında kanal özilinti fonksiyonu verilen kanallar kullanılarak kaynak tahsisi problemi çözülmektedir. Burada kanalın  $t_0$  altçerçeve zamanında kaynak tahsisi yapılması ile elde edilen  $x^1$  değeri bir sonraki  $t_1$  altçerçeve zamanında problemin çözümü için başlangıç noktası olarak kullanılmaktadır. Yapılan bu işlem sıcak başlangıç olarak adlandırılmaktadır.

Sıcak başlangıç yapılarak kaynak tahsisi problemini çözebilmek için [29]' da verilen TOMLAB program paketi kullanılmıştır. Kaynak tahsisi probleminin sıcak başlangıç yapılmadan elde edilen çözümü de yine TOMLAB ile yapılmıştır. Bu şekilde elde edilen iki farklı metodun çözüm süreleri Şekil 5.15' de karşılaştırılmıştır.



**Şekil 5.15:** Kanalın Değişim Hızına Göre Kaynak Tahsisi Probleminin Çözüm Süresi

Şekil 5.15' de verilen grafik incelendiğinde kanalın bir önceki altçerçeve zamanına bağımlılığı arttıkça problemin çözüm süresinin yaklaşık 2 kata kadar azaldığı görülmektedir. Fakat kanalın bağımlılığı ortadan kalktıkça sıcak başlangıç ile problemin çözüm süresi, normal çözüm süresinin de üzerine çıkabilmektedir. Bu nedenle, kanal bağımsız bir kanal ise önceki altçerçevde elde edilen çözümü bir sonraki altçerçeve için başlangıç noktası olarak kullanmak gereksiz hale gelmektedir. Sonuç olarak kanalın zamanla değişiminin yavaş olduğu durumlarda sıcak başlangıç kullanılarak kaynak tahsisi probleminin çözüm süresi azaltılmaktadır.

## 5.4 Yükselt ve İzdüşür Algoritması Benzetimleri

Kaynak tahsisi problemi 3. Bölümde anlatılan Yükselt ve İzdüşür (Lift and Project) algoritması kullanılarak çözülmeye çalışılmıştır. Bu kapsamda Fukuda' nın Double Description Algoritması uygulanmıştır. Double Description Algoritması için cdd/cdd+ [33] program paketi kullanılmıştır. Bu algoritmanın işlem karmaşıklığı problem boyutları ile polinom zamanla artmaktadır. Gösterim zorluğu nedeniyle sadece önemli parametrelere değinilecektir.

K=2 kullanıcı, N=2 alt-taşıyıcı ve bir kullanıcı için alt-taşıyıcı başına en fazla M=2 bit atanabildiği bir sistem düşünüldüğünde:

OFDM sembolü başına bit her bir kullanıcının en az bir bit gönderebildiği yani  $R_k = 1$  bit tüm  $k=1,2$  değerleri için olduğu durum kullanılmıştır. Burada verilen parametreler ile (2.13) 'de verilen problemin oluşturulmuştur. Problem boyutları ( $K*N*M=8$ ) küçük olduğu için kaba kuvvet yoluyla problemin çözüm kümesinde yer alan  $x$  vektörleri kolayca hesaplanabilmektedir. Çözüm kümesi içerisinde yer alan  $x$  vektörleri aşağıda sıralanmıştır:

$$\begin{aligned} & [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]^T \\ & [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^T \\ & [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0]^T \\ & [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1]^T \\ & [0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0]^T \\ & [0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0]^T \\ & [0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0]^T \\ & [0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0]^T \end{aligned} \tag{5.1}$$

Orijinal problem kümesinde toplam 20 adet eşitsizlik bulunmaktadır. Bu eşitsizliklerin tanımladığı çokyüzlü ise toplam 44 adet uç noktaya sahiptir. Bu uç noktalar Tablo 5.3 'de gösterilmektedir. Burada toplam 36 adet uç noktanın tamsayı değerler almadığı görülmektedir.

1	0	0	0	1	0	0.5	0	0
2	0	1	0	0	0	0	0	0.5
3	0	0.5	0	0.5	0	0	0	0
4	0	0.5	0.5	0	0	0	0	0.25
5	0	0.5	0.5	0	0	0	0.5	0
6	0.5	0	0	0.5	0.5	0	0	0
7	0.5	0	0	0.5	0	0.25	0	0
8	0	0.5	0	0	0	0	0	0.5
9	0	0	0	0.5	0	0.5	0	0
10	1	0	0	0	0	0	0	0.5
11	0	0	1	0	0	0.5	0	0
12	0.5	0	0	0	0.5	0	0	0.5
13	0	0.5	0	0	0	0	0	1
14	0	0	0	0.5	0	1	0	0
15	0	0.25	0	0	0.5	0	0	0.5
16	0	0	0.5	0	0	0.5	0.5	0
17	0	0	0	0.25	0	0.5	0.5	0
18	0	0.5	0.5	0	0	0	1	0
19	0	0.5	0	0.5	0	0	1	0
20	0	0.5	0.5	0	0	0	0	1
21	0	0.5	0	0.5	0	0	0	1
22	0.5	0	0	0.5	1	0	0	0
23	0	0.5	0	0.5	1	0	0	0
24	0.5	0	0	0.5	0	1	0	0
25	0	0.5	0	0.5	0	1	0	0
26	1	0	0	0	0	0	0	1
27	1	0	0	0	0.5	0	0	0.5
28	1	0	0	0	0	0.5	0	0.5
29	0	1	0	0	0	0	0	1
30	0	1	0	0	0.5	0	0	0.5
31	0	1	0	0	0	0.5	0	0.5
32	0	0	1	0	0	0.5	0.5	0
33	0	0	1	0	0	0.5	0	0.5
34	0	0	1	0	0	1	0	0
35	0	0	0	1	0	0.5	0.5	0
36	0	0	0	1	0	0.5	0	0.5
37	0	0	0	1	0	1	0	0
38	0	0	1	0	1	0	0	0
39	1	0	0	0	0	0	1	0
40	0	0	0	0	0	0.5	0	0.5
41	0	0	0	0.5	1	0	0	0
42	0	0.5	0	0	0	0	1	0
43	0	1	0	0	0	0	1	0
44	0	0	0	1	1	0	0	0

**Tablo 5.3:** K=2, N=2, M=2 olduğu durumda çokyüzlünün sahip olduğu uç noktalar kümesi (Tablonun her bir satır  $\mathbf{x}^T$  vektörünün elemanlarını vermektedir.)

Tamsayı olmayan bu uç noktaları tamsayı değerlere çekebilmek için yükselt ve izdüşür algoritması kullanılmıştır. Böylece elde edilen yeni eşitsizliklerin tanımladığı çokyüzlünün toplam 8 adet uç noktası aşağıdaki değerlere sahip olmaktadır.

Uç Nokta								
1	1	0	0	0	0	0	1	0
2	1	0	0	0	0	0	0	1
3	0	1	0	0	0	0	1	0
4	0	1	0	0	0	0	0	1
5	0	0	1	0	1	0	0	0
6	0	0	1	0	0	1	0	0
7	0	0	0	1	1	0	0	0
8	0	0	0	1	0	1	0	0

**Tablo 5.4:** K=2, N=2, M=2 olduğu durumda yükselt ve izdüşür algoritması sonucu elde edilen çokyüzlünün sahip olduğu uç noktalar kümesi

Tablo 5.4' de verilen toplam 8 adet tamsayı uç noktanın (5.1)' de verilen  $x$  çözüm kümesi vektörleri ile birebir aynı olduğu görülmüştür.

Tez kapsamında yükselt ve izdüşür algoritması ile dallan ve sınırla (branch and bound) algoritmasının çözüm süreleri karşılaştırılmıştır.  $K*N*M= 32$  boyutlu bir kaynak tahsisi problemi için dallan ve sınırla algoritması ile ortalama 1.6 ms sürede problem çözülebilmekteyken, yükselt ve izdüşür algoritması ortalama 7 s gibi bir sürede problemi çözebilmektedir. Bu sonuca bakarak yükselt ve izdüşür algoritmasının oldukça yavaş çalıştığı ve yüksek boyutlu problemlerin çözümünde yetersiz kaldığı görülmüştür. Yükselt ve izdüşür algoritmasında işlem karmaşıklığı polinom zamanla artmasına rağmen problem boyutu arttıkça problemin çözümü için geçen süresi oldukça uzamaktadır. Bunların sonucunda, yükselt ve izdüşür algoritmasının kullanılan bilgisayar işlemcisi ile ancak düşük boyutlu ( $K*N*M=32$  boyuta kadar) problemlerde istenilen işlemi yerine getirebildiği görülmüştür. Problem boyutları arttıkça ( $K*N*M > 32$ ) yükselt ve izdüşür algoritmasının pratik senaryolarda karşımıza çıkabilecek problemleri çözmede yavaş kaldığı gözlenmiştir.

## 6. SONUÇLAR

Bu çalışmada OFDMA sistemler için kaynak tahsisi problemi incelenmiştir. OFDMA sistemler için kaynak tahsisi probleminde kullanıcıya tahsis edilecek alt-taşıyıcı ve bit sayısı tamsayı değerler almaktadır. Bu nedenle problem tamsayı programlama problemi olarak değerlendirilmektedir. Fakat tamsayı programlama algoritmalarının işlem karmaşıklığı yüksek olduğundan dolayı mevcut sistemlerde kullanılmaları oldukça zordur. Bu nedenle tez kapsamında farklı yöntemler önerilmiştir.

İlk olarak yükselt-ve-izdüşür algoritması ile kaynak tahsisi problemini tanımlayan çokyüzlünün tüm uç noktaları tamsayı değerlere çekilerek problemin tamsayı dışbükey kabuğu bulunmuştur. Tamsayı uç noktalara simpleks algoritması uygulanarak problemin çözümü gerçekleştirilmektedir. OFDMA sisteminden faydalanan kullanıcı sayısı sabit kaldığı sürece, bulunan tamsayı uç noktalardan biri kanalın değişimine göre problemin çözümünü vereceği için bu yöntemin kullanılabilmesi önemlidir. Fakat yapılan benzetimler sonrasında bu yöntem ile düşük kullanıcı sayısı ve alt-taşıyıcı sayısına sahip sistemler için problem çözülebilmektedir. Yükselt ve izdüşür algoritması ile kaynak tahsisi probleminin eniyi çözümü elde edilmektedir. Fakat mevcut imkânlar ile pratik sistemlerde bu yöntemin kullanılması çok zor olduğu görülmüştür.

Literatürde önerilen kaynak tahsisi yöntemlerinin birçoğu eniyilikten ödün vermek pahasına, işlem karmaşıklığı düşürmeyi ve hızlı bir şekilde çözüme ulaşabilmeyi hedeflemiştir. LP+Greedy yönteminde ilk olarak gevşetilmiş problem doğrusal programlama ile çözülmekte ve bunun sonucuna göre kullanıcılara alt-taşıyıcı tahsisi yapılmaktadır. Sonrasında kullanıcılara iletilecek bitler bu alt-taşıyıcılara aç gözlü algoritma ile yüklenerek bir çözüm elde edilmiştir. Bu yöntem ile eniyeye oldukça yakın bir çözüm elde edilmektedir.

Tez kapsamında önerilen iki aşamalı yöntemde ilk olarak tüm alt-taşıyıcılara sabit sayıda bit yüklemesi yani akıllı ilkeme yapılarak kullanıcılara sabit sayıda bit yüklenmiş alt-taşıyıcılar tahsis edilmektedir. Sonrasında her bir kullanıcının kendi

ihtiyacına göre aç gözlü algoritma ile kullanıcıların ihtiyaçları olan bit sayıları alt-taşıyıcılara yüklenmektedir. Benzetim sonuçlarına bakıldığında bu yöntem ile hem eniyiye yakın kaynak tahsisi gerçekleştirilmekte hem de kaynak tahsisi çok kısa sürede gerçekleştirilmektedir. Böylece başlangıçta alt-taşıyıcılara sabit sayıda bit yüklemesi yapılması ile hem eniyiye yakın bir sonuç elde edilmekte hem de çok kısa sürelerde ( $\sim 2,3$  ms) kaynak tahsisi problemi çözülebilmektedir.

Ayrıca tezde kanalın bir önceki altçerçeve zamanına olan ilintisi kullanılarak bir çözüm önerilmiştir. Burada kaynak tahsisi problemi için bir önceki alt-çerçeve zamanında elde edilen çözümün, bir sonraki alt-çerçeve zamanında, çözümün başlangıç noktası olarak kullanılması ile yavaş değişen bir kanal için yaklaşık olarak çözüm süresi yarıya düşürülebilmektedir.

Sonuç olarak, bu tezde yapılan çalışmalar ile kaynak tahsisi probleminin eniyiye yakın çözümleri çok kısa sürelerde ( $\sim 2,3$  ms) elde edilebilmesi, önerilen algoritma ile sağlanmaktadır. Bir PC üzerinde koşturulan MATLAB ortamında elde edilen bu sürenin, tümüyle kaynak tahsisine ayrılmış bir işlemci ile daha da düşürülebileceği açıktır. Bu nedenle önerilen algoritmanın gerçek zamanlı ve eniyiye yakın başarımla sağladığı söylenebilir.



## **KAYNAKLAR**

- [1] E. Balas, S. Ceria, and G. Cornu'ejols, 1993, A lift-and-project cutting plane algorithm for mixed 0-1 programs. *Mathematical Programming*, 58:295–324.
- [2] D. Bertsimas and J. N. Tsitsiklis, 2008, *Introduction to Linear Optimization Dynamic Ideas*, Athena Scientific, Belmont, Massachusetts.
- [3] C. Papadimitriou and K. Steiglitz, 1998, *Combinatorial Optimization : Algorithms and Complexity*. Mineola, New York: Dover Publications, Inc.
- [4] A. Schrijver, 1986, *Theory of Linear and Integer Programming*, Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons Ltd., Chichester, A Wiley-Interscience Publication.
- [5] G. Nemhauser and L. Wolsey, 1999, *Integer and Combinatorial Optimization*, Wiley-Interscience Series in Discrete Mathematics and Optimization, John Wiley & Sons Inc., New York, Reprint of the 1988 original, A Wiley-Interscience Publication.
- [6] G. Dahl, 1997, *An Introduction to Convexity, Polyhedral Theory and Combinatorial Optimization*, University of Oslo, Department of Informatics.
- [7] C. Wong, R. S. Cheng, K. B. Lateief, R. D. Murch, Oct 1999, Multiuser OFDM with adaptive subcarrier, bit, and power allocation, *IEEE Journal on Selected Areas in Communications*, vol.17, no.10, pp.1747-1758.
- [8] I. Kim, H. Lee, N. Kim, Y. H. Lee, 2001, On the use of linear programming for dynamic subchannel and bit allocation in multiuser OFDM, *IEEE Global Telecommunications Conference* pp.3648-3652.

- [9] K. Choe, Y. Lim, S. Park, 2004, Subcarrier adaptation for multiuser OFDM systems, IEEE Global Telecommunications Conference, vol.2, no. pp. 1230-1233.
- [10] Z. Mao, X. M. Wang, J. Lin, 2005, Fast optimal radio resource allocation in OFDMA system based on branch-and-bound method, IEEE Pacific Rim Conference on Communications, Computers and Signal Processing pp.348-351.
- [11] Y. Ma, 2008, Rate maximization for downlink OFDMA with proportional fairness, IEEE Transactions on Vehicular Technology, vol. 57, no. 5, pp. 3267-3274.
- [12] I. Wong, Z. Shen, B. Evans, and J. Andrews, 2004, A low complexity algorithm for proportional resource allocation in OFDMA systems, in IEEE Workshop on Signal Processing Systems (SIPS 2004), pp. 1-6.
- [13] C. Turgu and C. Toker, 2009, A low complexity resource allocation algorithm for OFDMA systems, in IEEE/SP 15th Workshop on Statistical Signal Processing (SSP '09), pp. 689-692.
- [14] I. C. Wong , Z. Shen , B. L. Evans , J. G. Andrews, 2004, A low complexity algorithm for proportional resource allocation in OFDMA systems,” SiPS 2004 – IEEE International Signal Processing Systems Workshop, pp.1-6
- [15] J. Jang, K. Lee, 2003, Transmit power adaptation for multiuser OFDM systems, IEEE Journal on Selected Areas in Communications, vol.21, no.2, pp. 171-178.
- [16] R. S. Cheng and S. Verdú, 1993 , “Gaussian multiaccess channels with ISI: capacity regions and multiuser water-filling,” IEEE Transaction on Information Theory, vol. 39, pp. 773–785.
- [17] S. Boyd and L. Vandenberghe, Convex Optimization. Cambridge University Press, 2004.

- [18] G. Zhang, 2004, Subcarrier and bit allocation for real-time services in multiuser OFDM systems, IEEE International Conference on Communications , vol.5, pp. 2985-2989.
- [19] N. Ermolova, B. Makarevitch, 2007, Low complexity adaptive power and subcarrier allocation for OFDMA, IEEE Transactions on Wireless Communications, vol.6, no.2, pp.433-437.
- [20] D. Dardari, 2004, Ordered subcarrier selection algorithm for OFDM-based high-speed WLANs, IEEE Transactions on Communications, Vol 3, No:5.
- [21] W. Rhee; J. M. Cioffi, 2000, Increase in capacity of multiuser OFDM system using dynamic subchannel allocation, IEEE Vehicular Technology Conference vol.2, pp.1085-1089.
- [22] D. Kivanc, G. Li, H. Liu, 2003, Computationally efficient bandwidth allocation and power control for OFDMA, IEEE Transactions on Wireless Communications, vol.2, no.6, pp. 1150-1158
- [23] LTE Simulator [Online]. <http://www.nt.tuwien.ac.at/ltesimulator/> [8/6/2012 tarihinde geçerli]
- [24] G. Munz, S. Petschinger, J. Speidel, 2002, An efficient water-filling algorithm for multiple access OFDM, IEEE Global Telecommunications Conference, vol.1, no., pp. 681-685.
- [25] Y. Wang, F. Chen, G. Wei, 2005, Adaptive subcarrier and bit allocation for multiuser OFDM system based on genetic algorithm," International Conference on Communications, Circuits and Systems, vol.1, no., pp. 242-246.
- [26] J. C. Ikuno, M. Wrulich, M. Rupp, 2009 ,Performance and modeling of LTE H-ARQ; ITG Workshop on Smart Antennas, Berlin, Germany; International ITG Workshop on Smart Antennas (WSA 2009), pp. 1-6.

[27] Third Generation Project Partnership, The Mobile Broadband Standard.  
<http://www.3gpp.org/>

[28] GLPK, "GNU linear programming kit". <http://www.gnu.org/software/glpk>  
[8/6/2012 tarihinde geçerli]

[29] Tomlab. <http://tomopt.com/tomlab/> [8/6/2012 tarihinde geçerli]

[30] J. C. Ikuno, M. Wrulich, M. Rupp, 2010, System level simulation of LTE networks; Proc. 2010 IEEE 71st Vehicular Technology Conference: VTC2010-Spring, 5 pages.

[31] S. Haykin, 2001, Communication Systems, John Wiley and Sons.

[32] A. Goldsmith, 2005, Wireless Communications, Cambridge University Press.

[33] cdd/cdd+ , [http://www.ifor.math.ethz.ch/~fukuda/cdd\\_home/](http://www.ifor.math.ethz.ch/~fukuda/cdd_home/) [8/6/2012 tarihinde geçerli]

[34] A. Molisch, 2005, Wireless Communications, IEEE Press, John Wiley and Sons.

[35] J. C. Ikuno, Vienna LTE simulators link level simulator documentation, v1.7r1089 [Online].

Available: <http://www.nt.tuwien.ac.at/fileadmin/topics/simulators/LTElinkDoc.pdf>

## EKLER

### EK-1: Çokyüzlüler ve Doğrusal Programlama

Çokyüzlüler, eniyileme problemlerinin çözümünde önemli bir yere sahiptirler. Çokyüzlüler özellikle doğrusal programlamada çözüm kümesi yani problemin olurlu kümesi olarak bilinmektedir. Aynı zamanda çokyüzlüler tamsayı programlama problemlerinin çözümünde de önemli bir yere sahiptir.

#### a) Doğrusal Programlama

Doğrusal programlama, bir çokyüzlü üzerinde doğrusal bir fonksiyonu enküçültme veya enbüyütme problemi olarak çözmektir. Doğrusal programlama problemi:

$$\begin{array}{ll} \text{maliyet işlevi} & \left\{ \begin{array}{l} \min \\ \max \end{array} \right\} \mathbf{c}^T \mathbf{x} \\ \text{kısıtlar} & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{array} \quad (\text{Ek1.1})$$

$\mathbf{c} \in \mathbb{R}^n$ ,  $\mathbf{x} \in \mathbb{R}^n$ ,  $\mathbf{A} \in \mathbb{R}^{m \times n}$ ,  $\mathbf{b} \in \mathbb{R}^m$  olacak şekilde ifade edilir.

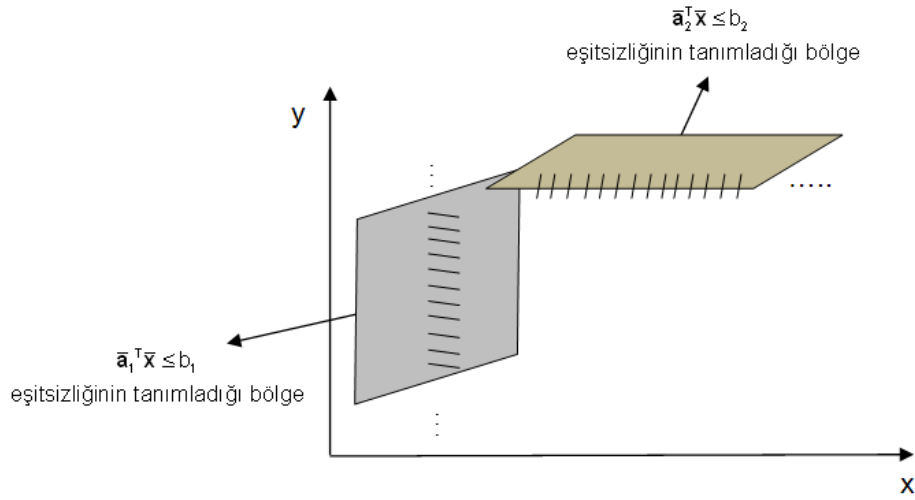
Doğrusal programlama problemleri birden fazla birbirine eşdeğer gösterim ile ifade edilebilir [4].

$\mathbf{Ax} \leq \mathbf{b}$  doğrusal eşitsizliklerini sağlayan herhangi bir  $\mathbf{x}$  vektörü olurlu çözüm (**feasible solution**) olarak tanımlanmaktadır. Çokyüzlü üzerinde tanımlı olan olurlu çözümler kümesi olurlu bölge (**feasible region**) olarak tanımlanır. Doğrusal programlamada olurlu bölge aynı zamanda bir çokyüzlüdür. Bir doğrusal programlamada olurlu bölge boş değil ise, “problem olurludur” denir. Eğer olurlu bölge boş ise problem olursuz olarak tanımlanır.

#### b) Çokyüzlüler

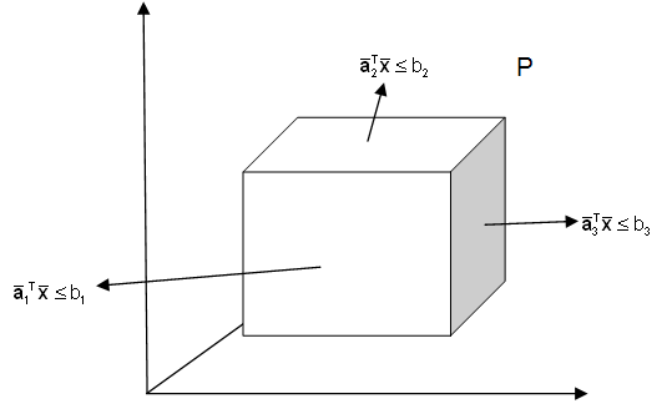
Sonlu sayıda doğrusal eşitsizlik kısıtlarının ve/veya eşitlik kısıtlarının oluşturduğu kümeye **çokyüzlü** denir. Çokyüzlü  $P \subseteq \mathbb{R}^n$ , sınırlı sayıda doğrusal eşitsizliği sağlayan noktalar kümesinden oluşur ve  $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$  olarak ifade edi-

lır. Burada  $\mathbf{A}$   $m \times n$  boyutlu bir matris ve  $\bar{\mathbf{b}}$   $n \times 1$  boyutlu bir vektördür.  $\mathbf{A}$  matrisi ve  $\mathbf{b}$  vektörü içerisinde yer alan elemanların;  $a_{i,j}$ ,  $b_i \in \mathbb{R}$  tüm  $i=\{1,2,\dots,m\}$  ve tüm  $j=\{1,2,\dots,n\}$  için tanımlandıkları varsayılacaktır.  $\mathbf{A}$  matrisinin ve  $\mathbf{b}$  vektörünün her bir satırı  $\mathbf{a}_i^T \mathbf{x} \leq b_i$  şeklinde bir eşitsizlik tanımlamaktadır.



**Şekil 6.1:**  $\mathbf{a}_i^T \mathbf{x} \leq b_i$  eşitsizliklerinin tanımladıkları bölgeler

Şekil 6.1 'de  $\mathbf{a}_i^T \mathbf{x} \leq b_i$  eşitsizliklerinin tanımladığı bölgeler gösterilmektedir.  $\mathbf{a}_i^T \mathbf{x} \leq b_i$  şeklindeki sonlu sayıda eşitsizliğin tanımladığı yarım uzayların kesişmesiyle oluşan bölge bir çokyüzlüdür. Şekil 6.2 'de sonlu sayıda eşitsizliğin oluşturduğu bir  $P$  çokyüzlüsü verilmektedir.  $P$  çokyüzlüsünün her bir yüzü farklı bir eşitsizlik tarafından tanımlanmaktadır.



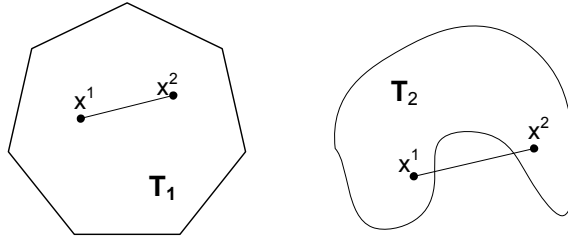
**Şekil 6.2:**  $\mathbf{a}_i^T \mathbf{x} \leq b_i$ ,  $i=1,2,\dots,6$  için  $\mathbb{R}^3$ ' te tanımlı P Çokyüzlüsü.

Doğrusal programlama problemleri  $\mathbf{Ax} \leq \mathbf{b}$  formatında kısıt eşitsizlikleri ile tanımlanır ve bu tanım aynı zamanda çokyüzlünün de tanımıdır. Yine  $\{\mathbf{x} \in \mathbb{R}^n, \mathbf{Ax} = \mathbf{b}, \mathbf{x} \geq 0\}$  şeklinde tanımlanmış olan kısıt eşitlikleri de çokyüzlü olarak tanımlanmaktadır. [5]

Bir çokyüzlü  $P \subseteq \mathbb{R}^n$ , kapsadığı tüm noktalar sınırlı ise,  $\omega, \gamma \in \mathbb{R}$  ve  $\gamma < \omega$  için  $P \subseteq \{\bar{\mathbf{x}} \in \mathbb{R}^n : \gamma \leq x_j \leq \omega \text{ ve } j=1,2,\dots,n\}$  olarak ifade edilir ve bu çokyüzlüye **politop** denir.

### c) Dışbükey küme (Convex Set)

$T \subseteq \mathbb{R}^n$  ve her bir  $\mathbf{x}^{(1)}, \mathbf{x}^{(2)} \in T$  için  $\lambda \mathbf{x}^{(1)} + (1-\lambda)\mathbf{x}^{(2)} \in T$  ve  $0 \leq \lambda \leq 1$  olduğu zaman T bir dışbükey kümedir [3]. Yani verilen bir küme içerisinde herhangi bir noktadan başka bir noktaya uzanan tüm doğruların tamamı küme içerisinde kalıyor ise bu kümeye dışbükey küme denir. Şekil 6.3' te  $T_1$  içerisinde yer alan tüm  $\mathbf{x}^{(1)}$  noktasından  $\mathbf{x}^{(2)}$  noktasına uzanan doğrular tanımlı küme içinde kalacağı için  $T_1$  dışbükey kümedir.  $T_2$  ise dışbükey olmayan bir kümedir [4].



**Şekil 6.3:** Dışbükey ve Dışbükey Olmayan Küme

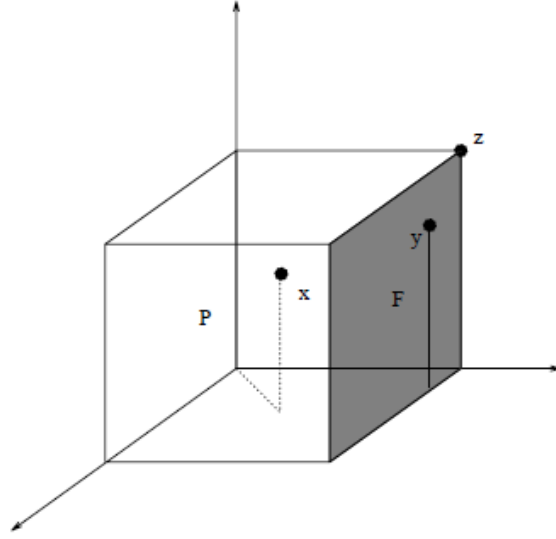
$\mathbb{R}^n$ ' de tanımlı olan  $P = \{\mathbf{Ax} \leq \mathbf{b}\}$  çokyüzlüsü sınırlı sayıda yarım uzayın kesişmesiyle oluşmaktadır ve kapalı dışbükey bir kümedir.

#### d) Örtük eşitlikler, İç Nokta ve Yüzlemsel Nokta

$P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$  çokyüzlüsünün tanım kümesindeki eşitsizlikler eşitlik ve eşitsizlik olarak ayrılmış olduğu durumda  $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{A}^= \mathbf{x} = \mathbf{b}^=, \mathbf{A}^< \mathbf{x} \leq \mathbf{b}^<\}$  şeklinde tanımlanır. Çokyüzlünün tanımında da yer aldığı gibi  $(\mathbf{A}^=, \mathbf{b}^=)$  eşitlikle sağlanan eşitsizlikler için kullanılmakta ve çokyüzlünün **örtük eşitlikleri (implicit equality)** denilmektedir.  $(\mathbf{A}^<, \mathbf{b}^<)$  ise geriye kalan tüm eşitsizlikler için kullanılmaktadır. Çokyüzlüyü tanımlayan tüm eşitsizliklerin indisleri  $M = \{1, \dots, m\}$  ise  $(\mathbf{A}^=, \mathbf{b}^=)$  eşitliklerini gösteren indis kümesi  $M^=$  ve  $(\mathbf{A}^<, \mathbf{b}^<)$  eşitsizliklerini gösteren indis kümesi  $M^<$  ile gösterilir.  $M^=, M^< \in M$  ve  $M^= + M^< = M$  olarak tanımlanır.  $(\mathbf{A}^<, \mathbf{b}^<)$  ve  $(\mathbf{A}^=, \mathbf{b}^=)$ ,  $(\mathbf{A}, \mathbf{b})$ ' deki ilgili satırlara karşılık gelmektedir. Çokyüzlü içinde her iki sistemi de sağlayan bir  $\mathbf{x}$  noktası bulunmaktadır.

$\mathbf{x} \in P$  noktası;  $\mathbf{A}^< \mathbf{x} \leq \mathbf{b}^<$  eşitsizliklerini kesin eşitsizlik (strict inequality,  $<$ ) ile sağlıyor ve  $\mathbf{A}^= \mathbf{x} \leq \mathbf{b}^=$  eşitliklerini de sağlıyorsa **yüzlemsel nokta** (Inner Point) olarak adlandırılır [6]. Şekil 6.4 'te gösterilen  $F$  yüzlemi üstünde yer alan  $\mathbf{y}$  noktası çokyüzlü için bir yüzlemsel noktadır.  $\mathbf{x} \in P$  noktası; eğer tüm  $i \in M$  için  $\mathbf{a}^i \mathbf{x} < \mathbf{b}_i$  eşitsizliklerini kesin eşitsizlik ile sağlıyor ise iç nokta olarak (interior point) adlandırılır. Şekil 6.4 'te gösterilen  $\mathbf{x}$  noktası  $P$  çokyüzlüsü için bir iç noktadır.





**Şekil 6.4 :** Çokyüzlünün iç noktası,yüzlemi ve yüzlemsel noktası [6]

Yukarıda verilen bilgilere göre çokyüzlüler için aşağıdaki tanımlamalar yapılmaktadır [4].

- Boş olmayan bir çokyüzlü en az bir iç noktaya sahiptir.
- Her bir çokyüzlü ile doğal olarak ilişkili olan bir dış bükey koni (**convex cone**) bulunmaktadır.  $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$  çokyüzlüsünün karakteristik konisi,  $P = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{Ay} \leq 0\}$  şeklinde tanımlanmaktadır.
- $P \subseteq \mathbb{R}^n$  'de tanımlı bir çokyüzlünün boyutu  $\dim(P) = n - \text{rank}(A^-)$  ile hesaplanır.  $P \subseteq \mathbb{R}^n$  çokyüzlüsünün boyutu "n" ise, çokyüzlü tam boyutlu olarak adlandırılır. Bir çokyüzlünün tam boyutlu olması için hiçbir örtük eşitliğe sahip olmaması gerekmektedir.
- $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$  olarak verilen çokyüzlü eğer tam boyutlu ise, en küçük sayıdaki doğrusal eşitsizlikler tarafından tek bir gösterimi vardır.
- Bütün çokyüzlüler sınırlı sayıda yüze sahiptir.

## e) Yüzlemsel Olarak Çokyüzlüler

$P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$  ile tanımlanan bir çokyüzlünün, tanım kümesindeki eşitsizliklerden  $\mathbf{a}_i^T \mathbf{x} \leq b_i$ , hangilerinin çokyüzlüyü tanımlamak için yeterli olduğu ve hangilerinin artık (gereksiz) eşitsizlik olduğunun bilinmesi eniyileme problemlerinin karmaşıklığının azaltılması için önem teşkil etmektedir. Artık eşitsizlikler, işlem karmaşıklığını arttırmak dışında probleme bir katkı sağlamamaktadır. Bu nedenle, artık eşitsizliklerin eşitsizlik kümesinden çıkartılması gerekmektedir. Bu bölümde, çokyüzlüyü verilen ilk tanım kümesindeki ile eşdeğer biçimde tanımlayabilecek en az sayıda eşitsizliğin nasıl bulunacağı anlatılmaktadır. En az sayıdaki bu eşitsizlikler aynı zamanda çokyüzlünün yüzlemleri (**facet**) olarak da bilinmektedir.

$\bar{\pi}^T \bar{\mathbf{x}} \leq \pi_0$  eşitsizliği,  $P$  çokyüzlüsündeki tüm noktalar tarafından sağlanıyor ise geçerli eşitsizlik olarak adlandırılır.  $P$  çokyüzlüsü için geçerli eşitsizlik olan  $\bar{\pi}^T \bar{\mathbf{x}} \leq \pi_0$ , eğer  $F = \{\bar{\mathbf{x}} \in P : \bar{\pi}^T \bar{\mathbf{x}} = \pi_0\}$  olarak sağlanıyorsa;  $F$ ,  $P$  çokyüzlüsünün bir yüzü (**face**) olarak tanımlanır [5]. Çokyüzlünün kendisi ve boş kümesi de aynı zamanda bir yüz olarak tanımlanmaktadır.

Bir geçerli eşitsizlik  $\bar{\pi}^T \bar{\mathbf{x}} \leq \pi_0$ , çokyüzlü için eşitlik olarak sağlanıyor ise, bu eşitsizlik aynı zamanda çokyüzlünün bir yüzünü tanımlar. Başlangıç olarak tanımlanan çokyüzlünün tanım kümesindeki artık eşitsizlik kısıtları atıldığında geriye kalan eşitsizlikler bundan sonra tüm  $i \in M$  için  $\mathbf{a}_i^T \mathbf{x} \leq b_i$  olarak ifade edilecektir.  $\mathbf{a}_i^T \mathbf{x} \leq b_i$  eşitsizliklerinin her biri boş olmayan yüzleri ifade etmektedir. Buna göre bir çokyüzlü sadece yüzlerini oluşturan eşitsizlikler tarafından da tanımlanabilmektedir.  $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$  çokyüzlüsü için eşitlik kümesi  $M^= \subseteq M$  olsun ve  $F \neq P$  ve  $F \neq \emptyset$  olduğu durumda  $F$  yüzü aynı zamanda bir çokyüzlü olarak tanımlanabilmektedir.

$$F = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}_i^T \mathbf{x} = b_i, \forall i \in M_F^= \text{ için ve } \mathbf{a}_i^T \mathbf{x} \leq b_i, \forall i \in M_F^{\leq} \text{ için}\} \quad (\text{Ek1.2})$$

$$M_F^= \supseteq M^=, M_F^{\leq} = M \setminus M_F^= \text{ olarak tanımlanır.}$$

Çokyüzlünün sınırında yer alan tüm noktaların oluşturduğu küme, çokyüzlünün yüzlerini oluşturmaktadır. Çokyüzlünün yüz kümesi, doğrusal programlama problemleri açısından bakıldığında, eniyileme probleminin en iyi çözüm kümesi olarak tanımlanmaktadır. Çokyüzlünün en küçük boyutlu yüzü başka hiçbir yüzü kapsamamaktadır. Benzer şekilde çokyüzlünün en büyük boyutlu yüzü ise çokyüzlünün başka hiçbir yüzü tarafından kapsanamamalıdır. Çokyüzlünün en büyük boyutlu yüzü **yüzlem (facet)** olarak adlandırılmaktadır.  $F$ ,  $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$  çokyüzlüsünün bir yüzü olduğunda,  $F$ ' in bir yüzlem olması için (Ek1.3) 'de verilen denkleğin sağlanması gerekmektedir.

$$\dim(F) = \dim(P) - 1 \quad (\text{Ek1.3})$$

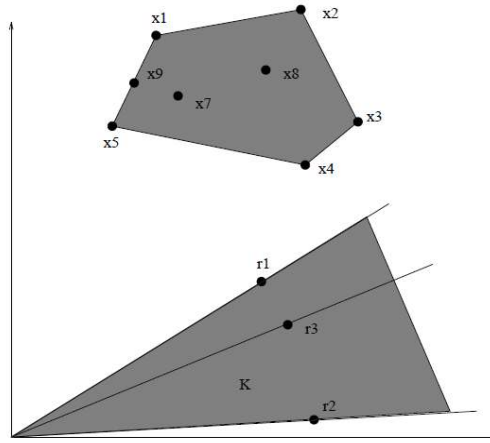
Yani  $P$   $n$  boyutlu bir çokyüzlü ise,  $F$ ' in yüzlem olması için boyutu  $(n-1)$ ' e eşit olmalıdır. Şekil 6.4 'de gösterilen birim küp üzerinde yer alan  $F$  yüzü, küpün bir yüzlemidir. Burada verilen küp üç boyutta,  $F$  yüzlemi ise iki boyutta tanımlanmaktadır.

Tam boyutlu  $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$  çokyüzlüsünün, tanım kümesi içerisinde artık eşitsizlik kalmadığı durumda,  $\mathbf{Ax} \leq \mathbf{b}$  içerisinde yer alan eşitsizlikler ile çokyüzlünün yüzlemleri arasında bire bir ilişki bulunmaktadır. Bu nedenle her bir  $F$  yüzlemine temsil eden eşitsizlikler,  $P$  çokyüzlüsünün tanımlanması için gereklidir. Böylece bir çokyüzlü  $P$ , sadece sahip olduğu yüzlemleri temsil eden eşitsizlikler ile tanımlanabilmektedir. Başka bir deyişle her bir yüzlemi temsil eden eşitsizlikler  $\mathbf{a}_i^T \mathbf{x} \leq b_i$  olarak verildiğinde; çokyüzlü  $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{a}_i^T \mathbf{x} \leq b_i, \text{ tüm } i = 1, 2, \dots, t \text{ için}\}$  olarak tanımlanmaktadır.

#### f) Uç Nokta (Extreme Point) ve Uç Işın (Extreme Ray)

$n$  boyutlu bir çokyüzlünün, en küçük boyuta sahip yüzlerinin boyutu  $n - \text{rank}(\mathbf{A})$  olarak hesaplanır.  $P = \{\mathbf{x} \in \mathbb{R}^n, \mathbf{Ax} \leq \mathbf{b}\} \neq \emptyset$  ve  $\mathbf{A}$  matrisinin kertesini  $n - k$  olduğu durumda  $P$  'nin en küçük boyutlu yüzü  $k$  boyutlu olmaktadır. Eğer çokyüzlü, sivri uçlu (**pointed**) ise çokyüzlünün en küçük boyutlu yüzü sadece bir noktadan oluşmaktadır ve bu nokta çokyüzlünün uç veya köşe (**vertex**) noktası olarak adlandırılmaktadır. Boş olmayan ve sivri uçlu bir çokyüzlü en az bir uç noktaya sahiptir.  $\mathbf{x}_0 \in P$  noktası çokyüzlünün uç noktası (**extreme point**) ise

$\mathbf{x}_0 = (1/2)\mathbf{x}_1 + (1/2)\mathbf{x}_2$  denkleğini sağlayan  $\mathbf{x}_1, \mathbf{x}_2 \in P$  noktaları olmamalı ve  $\mathbf{x}_0$  noktasının boyutu sıfır olmalıdır [5]. Şekil 6.5 'te verilen çokyüzlünün uç noktaları  $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4$  ve  $\mathbf{x}_5$ 'dir. Kalan diğer noktalar ise çokyüzlü için tanımlı birer uç nokta değildir.



**Şekil 6.5:** Uç noktalar (Vertices) ve Uç Işınlr (Extreme Rays) [6]

$P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\}$  çokyüzlüsünün her bir köşe noktası  $n$  adet  $\mathbf{a}_i^T \mathbf{x} = b_i$  olarak verilen bağımsız doğrusal eşitlik tarafından tanımlanmaktadır. Şekil 6.5 'te verilen iki boyutlu düzlemde tanımlı çokyüzlünün tüm uç noktalarının birbirinden bağımsız iki farklı doğrusal eşitlik tarafından tanımlandığı görülmektedir.

Bir çokyüzlünün ayrıtı veya kenarı (**edge**), çokyüzlünün tek boyutlu sınırlı yüzüdür. Yani çokyüzlünün kenarı bir ışın değildir. Çokyüzlünün her bir kenarı, yine çokyüzlünün iki farklı komşu uç noktasını birleştiren doğru parçasıdır.

$P^* = \{\mathbf{r} \in \mathbb{R}^n : \mathbf{Ar} \leq 0\}$  olsun ve  $P = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{Ax} \leq \mathbf{b}\} \neq \emptyset$ ,  $\bar{\mathbf{r}}$  vektörü sıfır vektöründen farklı ve  $\mathbf{r} \in P^*$  koşulunu sağlıyor ise  $\mathbf{r}$  vektörü çokyüzlü  $P$  'nin ışını (ray) olarak adlandırılmaktadır [5]. Bir vektörün çokyüzlünün ışını olması için (Ek1.4)' de verilen koşulu sağlaması gerekmektedir.

$$\mathbf{r} \in \mathbb{R}^n, \mathbf{x} \in P \text{ ve } \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y} = \mathbf{x} + \lambda \mathbf{r}, \lambda \in \mathbb{R}_+\} \subseteq P \quad (\text{Ek1.4})$$

Çokyüzlü için tanımlı bir  $r_0$  ışının, uç ışın (*extreme ray*) olması için;  $r_1 \neq \lambda r_2$ ,  $\lambda \in \mathbb{R}_+^1$  için  $r_0 = (1/2)r_1 + (1/2)r_2$  eşitliğini sağlayan  $r_1, r_2 \in P^*$  ışınların olmaması gerekmektedir. Şekil 6.5 'te K çokyüzlüsü için verilen  $r_1$  ve  $r_2$  uç ışın,  $r_3$  ise sadece bir ışındır. Eğer bir çokyüzlü tanımlı olduğu tüm boyutlarda sınırlandırılmış ise, bu çokyüzlü herhangi bir ışın içermemektedir.

Genel olarak sınırlı bir çokyüzlü, sonlu sayıda uç noktaya sahiptir ve bu uç noktalar sınırlı çokyüzlüyü tanımlamak için yeterlidir. Sınırlı olmayan çokyüzlü ise sonlu sayıda uç noktalara ve uç ışınlara sahiptir. Bu uç nokta ve uç ışınlar sınırlı olmayan çokyüzlüyü tanımlamak için yeterlidir.

### **g) Çokyüzlülerin Doğrusal Programlama Açısından Önemi**

(Ek-1.1)' de verilen doğrusal programlama probleminde kısıt kümesi içindeki doğrusal eşitsizlikler, aynı zamanda bir çokyüzlü tanımlamaktadır. Verilen maliyet işlevinin eniyilenmesi için tanımlanan çokyüzlünün uç noktalarından birine gidilerek temel olurlu çözüm elde edilmektedir. Yani verilen maliyet işlevinin eniyilenmesini sağlayacak çözüm, çokyüzlünün uç noktalarından birine eşit olmaktadır. Eğer tanımlanan çokyüzlü uç ışınlara sahip ise eniyileme probleminin çözümü  $\pm\infty$  değerler alabilmektedir. Fakat eniyileme problemi tarafından tanımlanan çokyüzlü ışına sahip değil ise, eniyileme problemi sonlu bir çözüme sahiptir ve çözüm çokyüzlünün uç noktalarından birine eşit olmaktadır. Bu nedenle bir çokyüzlünün uç noktalarının bulunabilmesi doğrusal programlama problemlerinin çözümü olarak görülmektedir.

## h) Tümüyle Birimsel Matris

Kare bir **B** matrisinin belirteni (determinant),  $\det(\mathbf{B}) = \pm 1$  ise **B** matrisi bir '**birimsel matris**'tir (**Unimodular Matrix, UM**). Eğer tamsayı **A** matrisinin tanımlanabilecek tekil olmayan (nonsingular) tüm kare alt-matrisleri birimsel matris ise, **A** matrisi **tümüyle birimsel bir matris (Totally Unimodular Matrix, TUM)** 'tir.[3],[4]

Başka bir ifadeyle, tüm elemanları  $a_{ij} = 0, \pm 1$  olan bir tamsayı **A** matrisinin satırları iki ayrışık kümeye  $I_1$  ve  $I_2$  'ye ayrılabilirdiğinde;

- **A** matrisinin tüm sütunlarında, ikiden fazla sıfırdan farklı tamsayı yer almıyorsa,
- **A** matrisinin sıfırdan farklı eleman içeren sütunları, aynı işaretli eleman içerdiğinde, bu sütunların satırlarından biri  $I_1$  kümesine diğeri ise  $I_2$  kümesine ayrılabiliriyorsa,
- **A** matrisinin sıfırdan farklı eleman içeren sütunları, farklı işaretli eleman içerdiğinde, bu elemanlara sahip sütunların satırları aynı  $I_1$  veya  $I_2$  kümesine ayrılabiliriyorsa,

**A** matrisi Tümüyle Birimsel Matris olur.[3]

**A** tümüyle birimsel matris ise  $\mathbf{P} = \{\mathbf{x} | \mathbf{Ax} \leq \mathbf{b}\}$  ve **b**' nin tüm elemanları tamsayı olacak şekilde verilen eşitsizliklerin tanımladığı çokyüzlünün tüm uç noktaları tamsayı değerler almaktadır[4].

## EK-2: Yükselt ve İzdüşür Algoritması Kılavuz

Yükseltme işlemi iki aşama halinde anlatılmaktadır. İlk olarak  $x_i$  ile çarpım sonrasında elde edilecek olan eşitsizliklerin düzenlenmesi ve genel matris yapısı gösterilecektir.

Sonrasında  $1-x_i$  ile çarpım sonrasında elde edilecek olan eşitsizlikler düzenlenip, matris yapıda kolay bir biçimde elde etmenin yöntemi gösterilecektir.

$$\mathbf{A}\bar{\mathbf{x}} \leq \bar{\mathbf{b}} \quad (\text{Ek2.1})$$

$\mathbf{A}$ : NxM boyutlu matris,  $\bar{\mathbf{x}}$ : Mx1 boyutlu vektör,  $\bar{\mathbf{b}}$ : Nx1 boyutlu vektör ve  $i = 1, 2, \dots, M$

$x_i$  ve  $(1-x_i)$  ile çarpım sonu elde edilecek eşitsizlikler düzenlenerek yeni  $\mathbf{A}$  matrisi  $\mathbf{A}_{\text{ift}}$  matrisi olarak, yeni  $\bar{\mathbf{b}}$  vektörü ise  $\bar{\mathbf{b}}_{\text{ift}}$  olarak adlandırılacaktır.

### a) $\mathbf{A}\bar{\mathbf{x}} \leq \bar{\mathbf{b}}$ eşitsizliklerini $x_i$ ile çarpımı

(1) 'de genel matris yapısı verilen çokyüzlüyü tanımlayan eşitsizlikler  $x_i$  ile çarpıldığında yeni oluşacak olan  $\mathbf{A}$  matrisi :  $\mathbf{A}_{x_i}$  olarak tanımlanacaktır.  $x_i$  ile çarpma işlemi sonrasında eşitsizliklerin matris yapısı aşağıda anlatıldığı gibi olacaktır.

$$\begin{bmatrix} \bar{\mathbf{A}}_1 & \bar{\mathbf{A}}_2 & \dots & \bar{\mathbf{A}}_i & \dots & \bar{\mathbf{A}}_M \end{bmatrix} \begin{bmatrix} x_1 x_i \\ x_2 x_i \\ \dots \\ x_i x_i \\ \dots \\ x_M x_i \end{bmatrix} - x_i \bar{\mathbf{b}} \leq 0 \quad (\text{Ek2.2})$$

-  $x_i x_i = x_i$  olduğundan  $x_i$  değişkeninin katsayısı  $\bar{\mathbf{A}}_i - \bar{\mathbf{b}}$  olduğu görülmektedir.

( $\bar{\mathbf{A}}_i = \mathbf{A}_{\text{NxM}}$  matrisinin i' nci sütunu)

-  $x_i$  ile çarpım sonucunda  $x_i$  dışında kalan  $x_j$ , ( $j = 1, 2, \dots, i-1, i+1, \dots, M$ ) değişkenlerinin kaybolduğu ve gösterim kolaylığı olması açısından kaybolan M-1 adet de-

ğişkenin katsayıları "0" olarak yazılacaktır. Dolayısı ile  $\mathbf{A}_{x_i}$  matrisi oluşturulurken

$$\bar{\mathbf{A}}_j = 0 \text{ olur ve } j = 1, 2, \dots, i-1, i+1, \dots, M$$

- Çarpım sonrası farklı indeksli değişkenlerin çarpımı ile oluşan yeni değişkenleri  $x_{j,i}$  olarak göstereceğiz ( $j = 1, 2, \dots, M$  ve  $j \neq i$ ) ( $M-1$  adet yeni değişken ortaya çıkmaktadır.)

$x_{j,i} = x_j x_i$  ve  $x_{j,i}$ 'li terimlerin katsayılarınının (Ek2.2)'de  $\bar{\mathbf{A}}_j$  olduğu görülmektedir.

$$\bar{\mathbf{A}}_j = \mathbf{A}_{N \times M} \text{ matrisinin } j\text{-inci sütun vektörü ve } j = 1, 2, \dots, i-1, i+1, \dots, M$$

Yeni oluşan değişkenlerle birlikte yeni  $x$  vektörü aşağıdaki gibi yazabiliriz.

$$\bar{\mathbf{x}} = \left[ x_1 \ x_2 \ \dots \ x_M \ x_{1,i} \ x_{2,i} \ \dots \ x_{i-1,i} \ x_{i+1,i} \ \dots \ x_{M,i} \right]^T \quad (\text{Ek2.3})$$

Yukarıda verilen tüm değişkenler ve katsayıları düzenlediğimiz zaman yeni oluşan  $\mathbf{A}_{x_i}$  matrisi ve  $\bar{\mathbf{b}}_{x_i}$  vektörünü aşağıdaki gibi gösterebiliriz.

$$\bar{\mathbf{A}}_{x_i} = \left[ 0_{(N \times (i-1))} \ \bar{\mathbf{A}}_i - \bar{\mathbf{b}} \ 0_{(N \times (M-i))} \ \bar{\mathbf{A}}_1 \ \bar{\mathbf{A}}_2 \ \dots \ \bar{\mathbf{A}}_{i-1} \ \bar{\mathbf{A}}_{i+1} \ \dots \ \bar{\mathbf{A}}_M \right]_{(N \times (2M-1))} \quad (\text{Ek2.4})$$

$$\bar{\mathbf{b}}_{x_i} = 0_{N \times 1} \quad (\text{Ek2.5})$$

$$\bar{\mathbf{x}} = \left[ x_1 \ x_2 \ \dots \ x_M \ x_{1,i} \ x_{2,i} \ \dots \ x_{i-1,i} \ x_{i+1,i} \ \dots \ x_{M,i} \right]^T_{((2M-1) \times 1)} \quad (\text{Ek2.6})$$

### b) $\mathbf{A}\bar{\mathbf{x}} \leq \bar{\mathbf{b}}$ eşitsizliklerini $1-x_i$ ile çarpımı

Yukarıda anlatılan tüm işlemler bu defa (Ek2.1)'deki matris ve vektörlerin  $1-x_i$  ile çarpımı sonucu yeniden oluşturulacaktır.  $1-x_i$  çarpımı sonucu yeni oluşacak  $\mathbf{A}$  matrisi  $\mathbf{A}_{1-x_i}$  olarak,  $\bar{\mathbf{b}}$  vektörü ise  $\bar{\mathbf{b}}_{1-x_i}$  olarak adlandırılacaktır.

$$(1-x_i)\mathbf{A}\bar{\mathbf{x}} - \bar{\mathbf{b}} \leq 0$$

$$\mathbf{A}\bar{\mathbf{x}} - \bar{\mathbf{b}} - x_i\mathbf{A}\bar{\mathbf{x}} + x_i\bar{\mathbf{b}} \leq 0$$

$$\left[ \bar{\mathbf{A}}_1 \ \bar{\mathbf{A}}_2 \ \dots \ \bar{\mathbf{A}}_i \ \dots \ \bar{\mathbf{A}}_M \right] \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_i \\ \dots \\ x_m \end{bmatrix} - \left[ \bar{\mathbf{A}}_1 \ \bar{\mathbf{A}}_2 \ \dots \ \bar{\mathbf{A}}_i \ \dots \ \bar{\mathbf{A}}_M \right] \begin{bmatrix} x_1 x_i \\ x_2 x_i \\ \dots \\ x_i x_i \\ \dots \\ x_m x_i \end{bmatrix} + x_i \bar{\mathbf{b}} \leq \bar{\mathbf{b}} \quad (\text{Ek2.7})$$



$\bar{\mathbf{A}}_j$  :  $\mathbf{A}$  matrisinin j-inci sütunu

- $x_j x_i = x_i$  olduğundan  $x_i$  değişkenin katsayısı  $\bar{\mathbf{A}}_i - \bar{\mathbf{A}}_i + \bar{\mathbf{b}} = \bar{\mathbf{b}}$  olmaktadır.
- $x_j$  'li terimlerin katsayıları  $\bar{\mathbf{A}}_j$  olmaktadır. ( $i \neq j$  ve  $j = 1, 2, \dots, i-1, i+1, \dots, M$ ).
- Yeni oluşan  $x_{j,i} = x_j x_i$  terimlerin katsayıları ise yukarıda kolayca görülebildiği gibi  $-\bar{\mathbf{A}}_j$  olmaktadır. ( $i \neq j$  ve  $j = 1, 2, \dots, i-1, i+1, \dots, M$ ).

Yukarıda verilen değişkenler ve katsayılar düzenlendiği zaman yeni  $\mathbf{A}$  matrisi, yeni  $\mathbf{b}$  vektörü aşağıdaki gibi oluşmaktadır.

$$\mathbf{A}_{1-x_i} = \left[ \bar{\mathbf{A}}_1 \ \bar{\mathbf{A}}_2 \ \dots \ \bar{\mathbf{A}}_{i-1} \ \bar{\mathbf{b}} \ \bar{\mathbf{A}}_{i+1} \ \dots \ \bar{\mathbf{A}}_M \ -\bar{\mathbf{A}}_1 \ -\bar{\mathbf{A}}_2 \ \dots \ -\bar{\mathbf{A}}_{i-1} \ -\bar{\mathbf{A}}_{i+1} \ \dots \ -\bar{\mathbf{A}}_M \right]_{(N \times (2M-1))} \quad (\text{Ek2.8})$$

$$\bar{\mathbf{b}}_{1-x_i} = \bar{\mathbf{b}} \quad (\text{Ek2.9})$$

$$\bar{\mathbf{x}} = \left[ x_1 \ x_2 \ \dots \ x_M \ x_{1,i} \ x_{2,i} \ \dots \ x_{i-1,i} \ x_{i+1,i} \ \dots \ x_{M,i} \right]_{((2M-1) \times 1)}^T \quad (\text{Ek2.10})$$

**a)** ve **b)** sonunda elde edilen  $\bar{\mathbf{x}}$  vektörünün değişkenlerinin aynı olduğu görülmektedir. Yükseltilmiş uzayda elde edilen  $\mathbf{A}_{\text{lift}}$  matrisi ve  $\bar{\mathbf{b}}_{\text{lift}}$  vektörünü, yukarıda verilen (Ek2.4), (Ek2.5), (Ek2.6), (Ek2.8), (Ek2.9) ve (Ek2.10)' da verilen bilgiler ile aşağıdaki gibi oluşturulmaktadır.

$$\mathbf{A}_{\text{lift}} = \begin{bmatrix} \mathbf{A}_{x_i} \\ \mathbf{A}_{1-x_i} \end{bmatrix} \text{ ve } \bar{\mathbf{b}}_{\text{lift}} = \begin{bmatrix} \bar{\mathbf{b}}_{x_i} \\ \bar{\mathbf{b}}_{1-x_i} \end{bmatrix} \quad (\text{Ek2.11})$$

Sistemin (Ek2.11)'deki  $\mathbf{A}_{\text{lift}}$  matrisinin ve  $\bar{\mathbf{b}}_{\text{lift}}$  vektörlerin detaylı son hali aşağıdaki yapıda olmaktadır.

$$\begin{bmatrix} \bar{0}_{N \times 1} & \bar{0}_{N \times 1} & \dots & \bar{0}_{N \times 1} & \bar{\mathbf{A}}_1 - \bar{\mathbf{b}} & \bar{0}_{N \times 1} & \dots & \bar{0}_{N \times 1} & \bar{\mathbf{A}}_1 & \bar{\mathbf{A}}_2 & \dots & \bar{\mathbf{A}}_{i-1} & \bar{\mathbf{A}}_{i+1} & \dots & \bar{\mathbf{A}}_M \\ \bar{\mathbf{A}}_1 & \bar{\mathbf{A}}_2 & \dots & \bar{\mathbf{A}}_{i-1} & \bar{\mathbf{b}} & \bar{\mathbf{A}}_{i+1} & \dots & \bar{\mathbf{A}}_M & -\bar{\mathbf{A}}_1 & -\bar{\mathbf{A}}_2 & \dots & -\bar{\mathbf{A}}_{i-1} & -\bar{\mathbf{A}}_{i+1} & \dots & -\bar{\mathbf{A}}_M \end{bmatrix}_{(2N \times (2M-1))} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_{i-1} \\ x_i \\ x_{i+1} \\ \dots \\ x_M \\ x_{1,i} \\ x_{2,i} \\ \dots \\ x_{i-1,i} \\ x_{i+1,i} \\ \dots \\ x_{M,i} \end{bmatrix}_{((2M-1) \times 1)} = \begin{bmatrix} \bar{0}_{N \times 1} \\ \bar{\mathbf{b}}_{N \times 1} \end{bmatrix}_{(2N \times 1)}$$

Yukarıda (Ek2.11)' de verilen  $\mathbf{A}_{\text{lift}}$  matrisi ve  $\bar{\mathbf{b}}_{\text{lift}}$  vektörü elde edildikten sonra. Yükseltme işlemi tamamlanmaktadır. Sonrasında  $\mathbf{A}_{\text{lift}}$  ve  $\bar{\mathbf{b}}_{\text{lift}}$  'den oluşan çokyüzlünün uç noktaları bulunacak ve bulunan uç noktalar yükseltilmiş uzaydan orijinal uzaya izdüşümü alınarak tamamlanacaktır. Orijinal uzayda tanımlı çokyüzlünün tüm uç noktalarının  $\{0,1\}$  tamsayı değerlere sahip olabilmesi için  $i = 1,2,\dots,M$  olarak verilen tüm eksenler için M defa (a) ve (b)' de anlatılan işlemler tekrar edilecektir. Sonuçta elde edilen çokyüzlünün tüm uç noktaları tamsayı değerler almaktadır.

### EK-3: Yükselt ve İzdüşür Algoritması İçin Önerilen Yöntem İspatı

K çokyüzlüsü aşağıdaki şekilde tanımlanmaktadır.

$$K = \{\bar{\mathbf{x}} \in \mathbb{R}^n \mid \mathbf{A}\bar{\mathbf{x}} \leq \bar{\mathbf{b}}, \bar{\mathbf{x}} \geq 0, x_j \leq 1, j = 1, \dots, p\}$$
$$= \{\bar{\mathbf{x}} \in \mathbb{R}^n \mid \tilde{\mathbf{A}}\bar{\mathbf{x}} \leq \tilde{\mathbf{b}}\}$$

$$K^0 = \{\bar{\mathbf{x}} \in K \mid x_j \in \{0, 1\}, j = 1, \dots, p\}$$

Burada

K : tamsayı gevşetmesi sonucu her boyutta (0, 1) aralığına hapsedilmiş çokyüzlü

$K^0$  : n-değişkenli karışık tamsayı kümesinde, p adet değişkenin {0, 1} tamsayı değerler alan çözüm kümesini,

$M_j(K)$  : yükseltmiş uzayda tanımlanan çokyüzlü

$P_j(K)$  : İzdüşüm işlemi sonrası orijinal uzayda tanımlanan çokyüzlü olarak adlandırılmaktadır.

$$P_j(K) \subseteq \text{conv}(K \cap \{\bar{\mathbf{x}} : x_j \in \{0, 1\}\}) \text{ olarak verildiğinde}$$

1.  $K \cap \{\bar{\mathbf{x}} : x_j = 0\} = \emptyset$  olduğu durumda

$\varepsilon > 0$  için  $x_j \geq \varepsilon$  ve  $x_j - \varepsilon \geq 0$  olur.  $x_j - \varepsilon \geq 0$  eşitsizliği  $(1 - x_j)$  ile çarpıldığında:

$$(i) \quad (1 - x_j)(x_j - \varepsilon) \geq 0$$

(i) eşitsizliği düzenlendiğinde aşağıdaki gibi yazılır.

$$(ii) \quad x_j - x_j^2 - \varepsilon + \varepsilon x_j \geq 0$$

Burada (ii) içinde  $x_j^2 = x_j$  için eşitsizlik yeniden yazıldığında

$$(iii) \quad \varepsilon(x_j - 1) \geq 0$$

$\varepsilon > 0$  başta verildiğinden,  $\varepsilon(x_j - 1) \geq 0$  eşitsizliğinin sağlanabilmesi için  $x_j \geq 1$  olmalıdır.  $x_j \geq 1$  değeri hem  $M_j(K)$  için hem de  $P_j(K)$  için geçerli bir çözümdür. Başlangıçta  $x_j$  değeri (0, 1) aralığına sıkıştırıldığı için,  $x_j = 1$  olur.

Sonuç olarak  $P_j(K) \subseteq K \cap \{\bar{\mathbf{x}} : x_j = 1\}$  elde edilmiş olur.

2.  $K \cap \{\bar{\mathbf{x}} : x_j = 1\} = \emptyset$  olduğu durumda.

$\varepsilon > 0$  için  $x_j + \varepsilon \geq 1$  olmaktadır.  $x_j + \varepsilon \geq 1$  eşitsizliği  $x_j$  ile çarpıldığında:

$$(i) \quad x_j(1-x_j-\varepsilon) \geq 0$$

(i) eşitsizliği düzenlendiğinde aşağıdaki gibi yazılır.

$$(ii) \quad x_j - x_j^2 - \varepsilon x_j \geq 0$$

Burada (ii) içinde  $x_j^2 = x_j$  için eşitsizlik yeniden yazıldığında

$$(iii) \quad \varepsilon x_j \leq 0$$

elde edilir.  $\varepsilon > 0$  başta verildiği için  $x_j \leq 0$  çözümü elde edilir.

$x_j \leq 0$  değeri hem  $M_j(K)$  için hem de  $P_j(K)$  için geçerli bir çözümdür. Başlangıçta  $x_j$  değeri  $(0,1)$  aralığına sıkıştırıldığı için,  $x_j = 0$  olur. Bu nedenle  $P_j(K) \subseteq K \cap \{\bar{\mathbf{x}} : x_j = 0\}$  elde edilmiş olur.

3.  $K \cap \{\bar{\mathbf{x}} : x_j = 0\} \neq \emptyset$  ve  $K \cap \{\bar{\mathbf{x}} : x_j = 1\} \neq \emptyset$  olduğu durumda

$\text{conv}(K \cap \{\bar{\mathbf{x}} : x_j \in \{0,1\}\})$  için geçerli bir eşitsizlik  $\bar{\alpha}^T \bar{\mathbf{x}} \geq \beta$  olsun.

$$(i) \quad K \cap \{\bar{\mathbf{x}} : x_j = 0\} \text{ ise } \lambda \geq 0 \text{ için } \bar{\alpha}^T \bar{\mathbf{x}} + \lambda x_j \geq \beta \text{ olur.}$$

$$(ii) \quad K \cap \{\bar{\mathbf{x}} : x_j = 1\} \text{ ise } \mu \geq 0 \text{ için } \bar{\alpha}^T \bar{\mathbf{x}} + \mu(1-x_j) \geq \beta \text{ olur.}$$

(i) ve (ii) 'de verilen durumlar  $K$  için geçerli olduğundan (i)' de elde edilen eşitsizlik  $(1-x_j)$  ile, (ii)' de elde edilen eşitsizlik  $x_j$  ile çarpılarak aşağıdaki eşitsizlikler elde edilir.

$$(iii) \quad (1-x_j)(\bar{\alpha}^T \bar{\mathbf{x}} + \lambda x_j) \geq \beta$$

$$(iv) \quad x_j(\bar{\alpha}^T \bar{\mathbf{x}} + \mu(1-x_j)) \geq \beta$$

(iii) ve (iv)' de verilen eşitsizlikler alt alta toplandığında:

$$(v) \quad \bar{\alpha}^T \bar{\mathbf{x}} + (\lambda + \mu)(x_j - x_j^2) - \beta \geq 0$$

elde edilir. Burada  $x_j = x_j^2$  için  $\bar{\alpha}^T \bar{\mathbf{x}} \geq \beta$  eşitsizliği yeniden elde edilmiş olur.

Bu nedenle  $\text{conv}(K \cap \{\bar{\mathbf{x}} : x_j \in \{0,1\}\})$  için geçerli bir eşitsizlik  $\bar{\alpha}^T \bar{\mathbf{x}} \geq \beta$ ,  $M_j(K)$  ve  $P_j(K)$  içinde geçerli bir eşitsizlik olmaktadır.

Özetle,

- 1' de ispatı verilen durumda çokyüzlünün uç noktasının  $x_j$  eksenindeki değeri  $(0,1]$  aralığında ise yükseltme işlemi sonrası çokyüzlünün tüm uç noktalarının  $x_j$  eksenindeki değeri "1" tamsayı değerine kaydırılmaktadır.

- 2' de ispatı verilen durumda çokyüzlünün uç noktasının  $x_j$  eksenindeki değeri  $[0,1)$  aralığında ise yükseltme işlemi sonrası çokyüzlünün tüm uç noktalarının  $x_j$  eksenindeki değeri "0" tamsayı değerine kaydırılmaktadır.
- 3' de ispatı verilen durumda çokyüzlünün uç noktasının  $x_j$  eksenindeki değeri  $\{0,1\}$  ise yükseltme işlemi sonrası çokyüzlünün tüm uç noktalarının  $x_j$  eksenindeki değerleri yine  $\{0,1\}$  olmaktadır.

## ÖZGEÇMİŞ

Adı Soyadı : Hasan AKBULUT

Doğum Yeri : Sivas

Doğum Yılı : 1985

Medeni Hali : Evli

Eğitim ve Akademik Durumu:

Lise 1999 - 2003: Sivas Selçuk Anadolu Lisesi

Lisans 2003 - 2008: Hacettepe Üniversitesi Elektrik ve Elektronik Mühendisliği bölümü

Y. Lisans 2008 - 2012: Hacettepe Üniversitesi

Yabancı Dil: İngilizce

İş Tecrübesi:

2009 - ... ASELSAN