T.C.
DOKUZ EYLÜL ÜNİVERSİTESİ
SOSYAL BİLİMLER ENSTİTÜSÜ
EKONOMETRİ ANABİLİM DALI
DOKTORA TEZİ

KÜME BÖLME PROBLEMLERİNİN OPTİMİZASYONU VE TÜRKİYE FUTBOL LİGLERİNE UYGULANMASI

İbrahim GÜNGÖR

Danışman Öğretim Üyesi
Prof.Dr.Metin ÇAKICI

İZMİR
1993

Doktora tezi olarak sunduğum "Küme Bölme Problemlerinin Optimizasyonu ve Türkiye Futbol Liglerine Uygulanması" adlı çalışmanın, tarafımdan, bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurulmaksızın yazıldığını ve yararlandığım eserlerin Bibliyografyada gösterilenlerden oluştuğunu, bunlara atıf yapılarak yararlanılmış olduğunu belirtir ve bunu onurumla doğrularım.

.../../1993

İbrahim GÜNGÖR

TUTANAK

Dokuz Eylül Üniversitesi Sosyal Bilimler Enstitüsünün ../../1993 tarih ve ......sayılı toplantısında oluşturulan jüri, Lisansüstü Öğretim Yönetmeliğinin .... maddesine göre Ekonometri Anabilim Dalı doktora öğrencisi İbrahim GÜNGÖR 'ün "Küme Bölme Problemlerinin Optimizasyonu ve Türkiye Futbol Liglerine Uygulanması" konulu tezini incelemiş ve aday .... tarihinde, saat ....'da jüri önünde savunmasından sonra .... dakikalık süre içinde gerek tez konusu, gerekse tezin dayanağı olan anabilim dallarından jüri üyelerince sorulan sorulara verdiği cevaplar değerlendirilirek tezin .... olduğuna oy .... ile karar verildi.

BAŞKAN

ÜYE

ÜYE

İÇİNDEKİLER

Sayfa

I. GİRİŞ

Bu tez çalışmasında ele alınan problem, Yöneylem Araştırması alanında yer alan sorunlardan biridir. Yöneylem araştırması, ilk olarak 1940 `da II. Dünya Savaşı `nda Alman hava hücumlarına daha etkin karşı koyabilmek için İngilizler tarafından geliştirilmiştir. Savaştan sonra askeri yöneylem araştırmalarında çalışan personelden çoğu işletme problemlerine doğrudan doğruya uygulanabilen yeni teknikler geliştirdiler(1). Bu gelişmeler günümüze kadar hem kullanılan modeller hem de uygulama alanları açısından hızla devam etmiştir. Bu gelişmelerden biri de küme bölme problemlerinin optimizasyonuyla ilgili modeller ve uygulama alanları konusunda ortaya çıkmıştır ki, bu konu bu çalışmanın içeriğini oluşturmaktadır. Burada, küme bölme sorunlarının optimum çözüme nasıl ulaştırılabileceği ve Türkiye futbol liglerinde alt kümelerin belirlenmesi sorununun küme bölme problemi olarak nasıl çözülebileceği araştırılmaktadır.

Bu araştırma konusunun fikirsel oluşum aşamaları şöyle gelişmiştir: Bilindiği gibi ülkemizde son yıllarda yeni il merkezlerinin oluşum çalışmaları gündemdedir. Bu oluşumda akla gelen sorunlardan biri, belirlenecek yeni illerin merkezleri ve bu merkezlere bağlanacak yerleşim birimlerinin optimum (en uygun) şekilde belirlenmesidir. Bunun dışında herhangi bir bakanlığın

---

(1) Ahmet Öztürk, "Yöneylem Araştırması", Uludağ Üniversitesi Basımevi, 1987, s.1

veya bir kuruluşun genel müdürlüğüne bağlı bölge müdürlükleri merkezlerinin ve görevlendirildiği coğrafi alanların belirlenmesi sorunu akla gelmektedir. Ayrıca, havayolu mürettebat planlaması ve havayolu filo planlaması sorunu, araç rotalama sorunu, futbol ve diğer spor liglerinde alt grupların belirlenmesi (Örneğin, ikinci futbol liginde yer alan 53 takımdan 5 alt küme oluşturulması gibi) sorunları düşünülmektedir.

Birbirlerine benzer özellikteki bu sorunların hangi optimizasyon teknikleri ile en iyi çözüme ulaştırılabileceğinin araştırılması amacıyla yapılan literatür taraması sonucu bu sorunların küme bölme (set partitioning) problemi olarak ele alındığı ve bu problemlere özel çözüm algoritmalarının geliştirildiği gözlenmiştir.

Bu araştırmada kullanılan metod, 0-1 tamsayılı doğrusal programlama prolemlerinin özel bir hali olarak düzenlenen ve küme bölme problemlerinin optimizasyonu için kullanılan tamsayımlama metodudur.

Araştırmanın ana amacı, Türkiye ikinci ve üçüncü futbol liglerinde alt grupların en uygun şekilde belirlenmesi ve bu tür problemlerin çözümünde kullanılabilecek uygun bir algoritmanın ortaya konulmasıdır. Böylece, benzer sorunların çözümünde de yararlanılabilecek uygun bir modelin ortaya konulması da dolaylı olarak amaçlanmış olmaktadır.

Türkiye birinci futbol ligindeki bütün takımlar aynı grup içinde yeraldıklarından bu ligde alt grupların belirlenmesi sorunu bulunmamaktadır. Bu nedenle birinci lig araştırmaya dahil edilmemiştir.

Türkçe literatürde, küme bölme problemlerini direkt olarak ele alan herhangi bir çalışmaya rastlanılmamıştır. Bu nedenle, konu ile ilgili yabancı dillerdeki terimlerin Türkçe karşılıkları ilk defa belirlenmeye çalışılmış ve bu konuda zaman zaman zorluklarla karşılaşılmıştır.

Araştırmanın I. bölümünde konuyla ilgili genel açıklamalar yeralmaktadır. II. bölümde çalışmanın amacı ve ele alınan problemin ortaya konulması konuları işlenmiştir. III. bölümde, küme bölme problemleri ile ilgili yerli literatürün azlığı nedeniyle, yöntem ve uygulanan model geniş olarak ele alınmış ve kullanılan algoritmanın bilgisayar programına geniş bir şekilde yer verilmiştir. IV. bölümde, Türkiye futbol liglerinde alt grupların belirlenmesi sorununun küme bölme problemi olarak çözümü konusunda yapılan çalışmalar yer almaktadır. V. bölümde ise, bütün bu çalışmaların sonuçları ve konu ile ilgili öneriler rapor edilmiştir.

Bana bu çalışmada her türlü imkanı veren hocam Prof.Dr. Metin Çakıcı 'ya başta olmak üzere; fakültemizde, Dokuz Eylül Üniversitesi Buca İktisadi ve İdari Bilimler Fakültesi ve Endüstri Mühendisliği bölümlerinde ve Ege Üniversitesi Bilgisayar Mühendisliği bölümünde, yurdumuzdaki diğer üniversitelerde ve yabancı üniversitelerde bu çalışmaya emeği geçen bütün öğretim üyelerine teşekkür etmeyi bir görev bilirim.

II. AMAÇ VE PROBLEMİN ORTAYA KONULMASI

A. Amaç

Bu çalışmanın esas amacı; küme bölme problemlerinin (set partitioning problems) kendine has yapısal özelliklerini ortaya koyarak bu problemlerin çözümü için kullanılabilecek algoritmaları araştırmak ve Türkiye ikinci ve üçüncü futbol liglerinde alt grupların belirlenmesi sorununu çözecek uygun bir algoritma önermektir. Ayrıca, 1992-1993 futbol sezonu için ikinci ve üçüncü futbol liglerinde alt grupların belirlenmesinde ulaşım maliyetlerini ve oyuncuların seyahat nedeniyle yorgunluklarını minimize eden optimum bir planın ortaya konulması ve bu planın mevcut durumla karşılaştırmasını yaparak kazançları ortaya koymak, amaçlar arasındadır.

Uygulama problemlerinin büyük boyutlu olması nedeniyle elle çözülmesi uygun olmayacağından, önerilen algoritmanın bilgisayar programının yapılması da bu tez çalışmasının kapsamı içinde yer alacaktır.

Yöneylem araştırması alanında ülkemizde ve yurt dışında birçok konuda araştırmalar yapılmaktadır. Ancak, Türkçe literatürde küme bölme (set partitioning) problemleri konusunda yapılmış bir çalışmaya rastlamak mümkün olmamıştır. Bu nedenle, bu çalışmada küme bölme problemleriyle ilgili terimlerin Türkçe karşılıklarının belirlenmesine de çalışılmış ve böylece ülkemizdeki diğer araştırmacılara bu konuda yardımcı olunacağı düşünülmüştür.

B. Problemin Ortaya Konulması

Bu çalışmada ele alınan problem bir küme bölme (set partitioning) problemidir. Küme bölme problemi, 0-1 tamsayılı doğrusal programlama probleminin özel bir halidir. Küme bölme probleminin bu özel hali:

- a_{ij} katsayı (girdi-çıktı katsayıları) matrisinin 0 ve 1 sayılarından oluşması,

- Bütün kısıtların eşitlik şeklinde olması

- Bütün eşitliklerin sağ tarafının 1 olmasıdır.

Bu problemin yapısal özelliği III. bölümde geniş olarak ele alınmıştır.

Küme bölme problemleri uygulamada genellikle onbinlere kadar varabilen değişken sayısı ve binlere kadar çıkabilen kısıt sayısı içerebildiğinden oldukça büyük boyutlu olarak ortaya çıkmaktadır. Bu problemlerin çözümü için genel 0-1 doğrusal programlama için geliştirilen algoritmaların kullanılması çok sayıda işlem ve uzun bir süre gerektirmektedir. Bu nedenle, bu çalışmada ele alınan esas sorun, küme bölme problemlerinin çözümünde kullanılabilecek yüksek performanslı algoritmaların belirlenmesidir. Küme bölme problemlerinin çözümünde, ele alınan problemin yapısal özelliğine göre, bu konudaki algoritmaların performansı farklı olabilmektedir. Bundan dolayı, ele alınan probleme en uygun küme bölme algoritmasının belirlenmesi halinde, çözüm işlemleri ve süresi daha kısa olabilmektedir. Bu konuyla ilgili çalışmalar da III. bölümün optimizasyon kısmında geniş olarak ele alınmaktadır.

Bu araştırmada uygulama çalışması olarak, Türkiye futbol liglerinde alt grupların belirlenmesi sorunu ele alınmıştır. Bilindiği gibi Türkiye futbol liglerinde, birinci ligteki takımlar tek bir grupta yer almasına karşılık, ikinci ve üçüncü ligteki takımlar birden fazla alt gruplar arasında paylaştırılmaktadır. Bu nedenle, ikinci ve üçüncü ligte bulunan takımların hangi alt grupta yer alması gerektiği bir sorun olarak ortaya çıkmaktadır. Örneğin; yurdumuzun çeşitli yerleşim bölgelerinde dağınık halde yer alan ikinci ligteki 53 takımın 5 ayrı alt kümeye ayrılması gerekmektedir. Futbol maçları, alt kümedeki takımlar arasında ve deplasmanlı olarak yapıldıklarında, 53 takımdan oluşan evrensel kümenin 5 `e bölünmesi olarak kurulabilecek küme bölme probleminde amaç, ulaşım maliyetlerini ve oyuncuların yolculukta karşılaştıkları yorgunluğu minimize edecek alt küme oluşum planını ortaya koymak olmalıdır. Bu sorunun çözümüyle ilgili olarak yapılan çalışma IV. bölümde geniş olarak ele alınmaktadır.

III. MODEL VE YÖNTEM

Bu araştırmada, doğrusal programlama modelinin özel bir hali olan 0-1 tamsayılı doğrusal programlama modelinden yararlanılmıştır. Küme bölme problemlerinin 0-1 tamsayılı doğrusal programlama modeli ile çözülebilmesi için birtakım ön çalışmaların yapılması gerekmektedir. Bu problemlerin çözümünde, genel 0-1 dogrusal programlama ile ilgili algoritmaların işlerliği (yapılan özel düzenlemer sonucunda) çok daha yüksek hale getirilebilmektedir. Bunun nedeni, küme bölme problemlerinde kısıtların ve a_{ij} katsayı matrisinin özel bir yapı göstermesidir. Küme bölme problemleri, bunların çözüm yöntemleri ve uygulamaları ile ilgili olarak yerli literatürde herhangi bir çalışmaya rastlanılmadığından, bu bölümde bu problemlerin özellikleri ve çözüm teknikleri ile ilgili çalışmalara yer verilmmiştir.

A. Küme Bölme Probleminin Tanıtımı

Küme bölme (set partitioning) veya eşitlik kısıtlı küme örtüleme (equality-constrained set covering) problemi, 0-1 tamsayılı doğrusal programlamanın en çok karşılaşılan özel bir halidir. Aşağıdaki gibi formüle edilir(1):

$$\text{Min.} \quad X_0 = \Sigma_{j\in N}\, C_j\, X_j \qquad (3.1)$$

$$\text{Kısıtlar:} \quad \Sigma_{j\in N}\, a_{ij}\, X_j = 1 \qquad (i\in M) \qquad (3.2)$$

$$X_j \in \{0,1\} \qquad (j\in N) \qquad (3.3)$$

Burada,

$$M = \{1,2,\ldots\ldots,m\}$$
$$N = \{1,2,\ldots\ldots,n\}$$

olarak yeralır.

$a_{ij} \in \{0,1\}$ dır. Yani; i. inci eleman, j. inci alt küme içinde yer alıyorsa $a_{ij} = 1$, diğer durumlarda $a_{ij} = 0$ olan mxn boyutunda bir 0-1 matristir.

(3.1) nolu eşitlikte C_j, j' inci alt kümenin maliyetidir ve $C_j \geq 0$ olmalıdır. X_j, j' inci alt küme optimal çözüm için seçilecek alt kümelerden biri ise 1 değerini, değil ise 0 değerini alır.

(3.2) eşitlikleri, her bir elemanın sadece bir tek alt kümede yer almasını sağlar.

Bu problemlere "Küme Bölme Problemi" adı aşağıdaki yorumdan dolayı konmuştur(2):

m adet elemandan oluşan bir $M=\{1,2,\ldots\ldots\ldots,m\}$ kümesinden, p adet alt kümenin oluşması için bölünmesi istenir. Bu amaçla M kümesinin, istenen özelliklere uygun olmak kaydıyla mümkün alt kümeleri

---

(1) Ruy Eduardo Campello, "Updating a Hybrid Algorithm For Set Partitioning Problems", Mat.Aplic.Comp., Vol.4, No:1, Rio de Jenerio, 1985, s.76

(2) Roy E. Marsten, "An Algorithm for Set Partitioning Problems", Management Science 20, 1974, s.774

$F=\{M_1,M_2,..........,M_n\}$ 'nin tamamı $M_j=\{i\in M \mid a_{ij}=1 \ ,j\in N\}$ olarak önceden belirlenir. küme bölme probleminin herhangi bir uygun çözümüyle F 'nin bir alt grubu $\{M_{j(1)}, M_{j(2)},..........M_{j(p)}\}$ elde edilir ki, bu, M kümesinin aşağıdaki eşitliklere uygun olarak bölümlemesidir:

$$M_{j(r)} \cap M_{j(s)} = \emptyset \quad , \ (s \neq r)$$

$$\bigcup_{k=1}^{p} M_{j(k)} = M \qquad (3.4)$$

(3.4) eşitliklerini sağlamayan $M_{j(k)}$ 'ler ayrılabilir durumda değildirler(1).

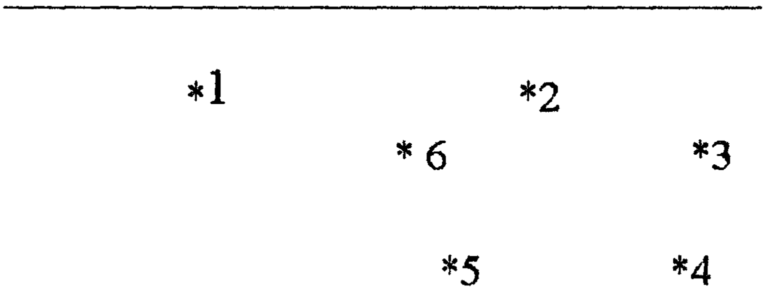
Küme bölme probleminin optimum çözümü sonucunda, $\Sigma_{j\in N}C_jX_j$ 'yi minimize eden bir alt kümeler grubu (3.4) eşitliklerine uygun olarak elde edilir. Burada ; C_j , M_j 'nin maliyetidir ve M_j , seçilen alt kümeler grubunun bir elemanı ise X_j =1 'dir. Değilse X_j = 0 'dır.

Küme bölme problemini uygulamaya dönük küçük bir problem üzerinde ifade etmek daha uygun olabilir.

Örneğin, bir bölgede bulunan 6 şehir merkezinin konumu şekil 1. de görüldüğü gibi olsun. Bu şehirler arasında iki gözetleme uçağı devriye gezecektir. Bu durumda, 6 şehirden oluşan evrensel küme, iki alt kümeye ayrılmalıdır. Bu sorunu bir küme bölme problemi olarak

---

(1) Mustafa Bayraktar, "Soyut Cebir ve Sayılar Teorisi", Atatürk Üniversitesi Yayınları no:650, Erzurum, 1988, s.6



Şekil 1. Örnek Problem İçin Ele Alınan Altı Şehir Merkezinin Yerleşim Durumu

düşünebiliriz. Burada amaç, uçakların bir turu için en az yol katetmelerini sağlayan çözüm planını belirleme, yani, katedilecek yolun minimizasyonudur. Şehir merkezleri arasındaki hava yolu uzaklıkları Tablo 1. de verilmiştir.

Tablo 1. Altı Şehir Merkezinin Maliyet Matrisi

	1	2	3	4	5	6
1	0	20	32	30	28	18
2	20	0	12	22	36	11
3	32	12	0	10	23	13
4	30	22	10	0	12	11
5	28	36	23	12	0	15
6	18	11	13	11	15	0

Bu sorunu küme bölme problemi olarak formüle edebilmek için, öncelikle, altı şehirden oluşan M={1,2,3,4,5,6} kümesinin birbirinden farklı üç ayrı şehir merkezini içeren olası bütün alt kümelerini ve bu alt kümelerin oluşum maliyetlerini belirlemek gerekir.

Olası alt kümelerin toplam sayısı = $C_{6\;3}$ =6!/(3!)(3!)=20

olmalıdır. Bu alt kümeler ve oluşum maliyetleri Tablo 2.'de verilmiştir.

Tablo 2. Örnek Problem İçin Oluşturulan Alt Kümeler

j	M_j	C_j
1	1,2,3	20 + 12 + 32 = 64
2	1,2,4	20 + 22 + 30 = 72
3	1,2,5	20 + 36 + 25 = 84
4	1,2,6	20 + 11 + 18 = 49
5	1,3,4	32 + 10 + 30 = 72
6	1,3,5	32 + 23 + 28 = 83
7	1,3,6	32 + 13 + 18 = 63
8	1,4,5	30 + 12 + 28 = 70
9	1,4,6	30 + 11 + 18 = 59
10	1,5,6	28 + 15 + 18 = 61
11	2,3,4	12 + 10 + 22 = 44
12	2,3,5	12 + 23 + 36 = 71
13	2,3,6	12 + 13 + 11 = 36
14	2,4,5	22 + 12 + 36 = 70
15	2,4,6	22 + 11 + 11 = 44
16	2,5,6	36 + 15 + 11 = 62
17	3,4,5	10 + 12 + 23 = 45
18	3,4,6	10 + 11 + 13 = 34
19	3,5,6	23 + 15 + 13 = 51
20	4,5,6	12 + 15 + 11 = 38

Tablo 2. de verilen alt kümeler dikkate alınarak, i. inci şehir j. inci alt kümede yer alıyorsa $a_{ij} = 1$, yeralmıyorsa $a_{ij} = 0$ değerleri verildiğinde a_{ij} katsayı matrisi ve C_j maliyet vektörü Tablo 3. deki gibi oluşacaktır. Bu tabloya "küme bölme tablosu" denir(1).

Tablo 3. Örnek Problem İçin Elde Edilen 0-1 Katsayılar Matrisi

	C_j katsayıları																				
	j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
i	C_j	64	72	84	49	72	83	63	70	59	61	44	71	36	70	44	62	45	34	51	38
1		1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
2		1	1	1	1	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0
3		1	0	0	0	1	1	1	0	0	0	1	1	1	0	0	0	1	1	1	0
4		0	1	0	0	1	0	0	1	1	0	1	0	0	1	1	0	1	1	0	1
5		0	0	1	0	0	1	0	1	0	1	0	1	0	1	0	1	1	0	1	1
6		0	0	0	1	0	0	1	0	1	1	0	0	1	0	1	1	0	1	1	1

Bu aşamadan sonra, örnek problem için küme bölme formülasyonu aşağıdaki gibi yazılabilir:

(1) Hein Fleuren, "A Computational Study of the Set Partitioning Approach For Vehicle Routing and Scheduling Problems", Twente Üniversity, 1988, s.12

$$\text{MİN. } X_0 = 64X_1 + 72X_2 + , \ldots\ldots , 51X_{19} + 38X_{20}$$

Kısıtlar:

$$X_1 + X_2 + X_3 + X_4 + X_5 + X_6 + X_7 + X_8 + X_9 + X_{10} = 1$$
$$X_1 + X_2 + X_3 + X_4 + X_{11} + X_{12} + X_{13} + X_{14} + X_{15} + X_{16} = 1$$
$$X_1 + X_5 + X_6 + X_7 + X_{11} + X_{12} + X_{13} + X_{17} + X_{18} + X_{19} = 1$$
$$X_2 + X_5 + X_8 + X_9 + X_{11} + X_{14} + X_{15} + X_{16} + X_{17} + X_{20} = 1$$
$$X_3 + X_6 + X_8 + X_{10} + X_{12} + X_{14} + X_{16} + X_{17} + X_{19} + X_{20} = 1$$
$$X_4 + X_7 + X_9 + X_{10} + X_{13} + X_{15} + X_{16} + X_{18} + X_{19} + X_{20} = 1$$

$$X_j \in \{0,1\} \qquad (j \in N)$$

$$N = \{1,2,3,\ldots\ldots,19,20\}$$

Bu problem , 0-1 tamsayılı doğrusal programlama ile ilgili çözüm tekniklerinden biri ile çözüldüğünde optimum çözüm değerleri şöyle elde ediler:

$X_0 = 94$

$X_4 = X_{17} = 1$

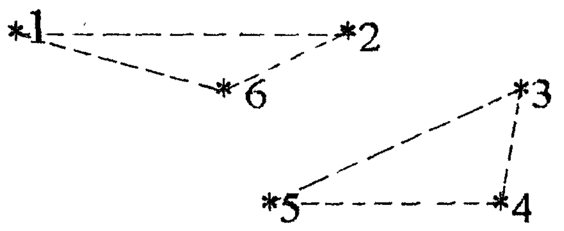
$X_j = 0$ (X_4 ve X_{17} hariç diğer X_j değerleri)

Bu sonuca göre, $M_4 = \{1,2,6\}$ ve $M_{17} = \{3,4,5\}$ alt kümelerinden oluşan bir bölme işlemi gerçekleşmiştir. Bu bölme işlemi (3.4) eşitliklerine uygundur.

$$\{1,2,6\} \cap \{3,4,5\} = \varnothing$$

$$\{1,2,6\} \cup \{3,4,5\} = \{1,2,6,3,4,5\} = M$$

Optimum çözüm sonuçlarına göre, sözkonusu altı şehir merkezi Şekil 2.deki gibi iki alt küme halinde gösterilebilir. İki uçaktan biri 1,2 ve 6 nolu şehirler arasında, diğeri ise 3,4, ve 5 nolu şehirler arasında devriye gezecektir. Bir şehirden diğer herhangi bir şehire gidiş veya geliş arasında maliyet açısından bir fark olmadığı kabul edilerek Tablo 1. deki matris simetrik olarak yeralmıştır. Bu yüzden, devriye gezecek uçakların gidiş yönü önemli olmadığından optimum planda yön belirtilmemiştir.



Şekil 2. Örnek Problem İçin Optimum Çözüm İle Elde Edilen Alt Kümeler

Bütün bu işlemlerden anlaşılacağı gibi, küme bölme problemlerinin çözümü iki aşamada gerçekleşmektedir(1):

I. Aşama : Olası alt kümelerin ve bu alt kümelerin maliyetlerinin belirlenmesi, yani "küme bölme tablosunun oluşturulması" aşaması,

II. Aşama : Birinci aşamada elde edilen verilerle kurulan küme bölme probleminin optimum çözümünün bulunması, yani "optimizasyon" aşaması.

Evrensel kümedeki eleman sayısı büyüdükçe, olası alt kümelerin sayısı çok büyük olacağından bilgisayar kullanımıyla dahi böyle bir problemin çözümü için çok uzun

(1) Martin Rohde, "Set Partitioning Mit Liniaren Randbedingungen", OR Spektrum.1, 1979, s.76

zaman gerekeceği ortadadır. Bu sebepten dolayı, olası alt kümelerin tamamı yerine (optimum çözümde yer alamayacağı kesin olarak belirlenenlerin dışlanarak), bir kısmının elde edilmesi yolunda çeşitli çalışmalar yapılmıştır. Bu konu Bölüm 3.1 'de ele alınacaktır.

Küme bölme probleminin optimum çözümüne 0-1 tamsayılı doğrusal programlamaya ilişkin algoritmaların kullanımıyla ulaşmak mümkündür. Ancak, bu şekilde yapılacak çözüm işleminde bilgisayar çözüm zamanı(run time) problemin hacmi büyüdükce çok uzun olmaktadır. küme bölme probleminin özel yapısını dikkate alarak yapılan çeşitli araştırmalar sonucunda , çözüm zamanını önemli ölçüde kısaltan çözüm algoritmaları geliştirilmiştir. Bu konu da Bölüm 3.2 'de ele alınacaktır.

B. Küme Bölme Probleminin Uygulama Alanları

Küme bölme problemleri, ilginç ikili 0-1 (binary) yapıları nedeniyle üzerinde çok sayıda çalışmalar ve bir çok konuda uygulama imkanları bulmuştur. Bununla ilgili olarak literatürde aşağıda belirtilen konularda yapılan uygulama çalışmaları görülmektedir:

-Hava yolu mürettebat planlaması (airline crew scheduling)
-Hava yolu filo planlaması (airline fleet scheduling) siyasal bölgeleme (political districting)
-Bilginin geri toplanması (information retrieval)
-Stok kesme (stock cutting)
-Hat dengeleme (line balancing)
-Kapasite dengeleme (capacity balancing)

-Boyama problemleri (coloring problems)
-Denizdeki sondaj platformlarının yerleşimi (location of offshore drilling platforms)
-Yer belirleme sorunu (facilities location) (1)
-Kutu paketleme (bin packing)
-Araç rotalama (vehicle ruting) (2)
-Frekans planlaması (3)
-Kan analiz cihazlarının performansının optimizasyonu (4)
-Demiryolu katar personeli planlaması (railroad crew scheduling)
-Kamyon dağıtımları (truck deliveries)
-Tanker rotalama (tanker routing) (5).

Bütün bu uygulamalardan anlaşılacağı gibi, bir bütünün parçalara ayrılması sorununda optimizasyon amacı olan hemen her konuda küme bölme problemi uygulanabilmektedir. Konuyla ilgili olarak, E. Balas - M.W. Padberg 'in ve S. Anıly - A.F. Gruen 'in çeşitli makalelerinde(2 ve 5) çok sayıda uygulama örnekleri yeralmaktadır.

---

(1) Roy E. Marsten, a.g.e., s.774

(2) S.Anıly ve A.Federgruen, "Structured Partitioning Problems", Operations Research, 39/1, 1991, s.130

(3) H.Thuve, "Frequency Planning as a Set Partitioning problem", European Journal of Operational Research. Vol:6, 1981, s.29-37

(4) W.M.Nawıjn, "Optimizing the Performance of a Blood Analyser : Application of the Set Partitioning Problem", European Journal of Operational Research, 36, 1988, s.167-173

(5) Egun Balas ve Manfred W.Padberg, "Set Partitioning : A Survey", SIAM Review, 18/4, 1976, s.712

C. Küme Bölme Problemlerinin Çözümü

Küme bölme problemlerinde çözüm işlemlerinin iki aşamada gerçekleştirildiği daha önce belirtilmişti. Bu aşamaların birincisi, "küme bölme tablosunun oluşturulması", ikincisi ise "küme bölme probleminin optimizasyonu" 'dur.

Küme bölme problemlerinin çözümünde, hem küme bölme tablosunun oluşturulması hem de optimizasyon aşamaları ile ilgili çeşitli algoritmalar geliştirilmiştir. Bu çalışmaların esas amacı; çözüm kalitesi (solution quality), kullanım alanı (utilization domain) ve bilgisayar çözüm zamanı (run time) açısından en iyi durumu gerçekleştirmektir (1). Yani; kesin optimal çözümü veya yaklaşık optimal çözümü bulabilmesi, çok sayıda küme bölme sorunları için uygulanabilmesi ve kabul edilebilir bir sürede çözüme ulaşabilmesi konuları, bu algoritmaların hazırlanmasında dikkate alınan temel özelliklerdir.

1. Küme bölme Tablosunun oluşturuması

Bir küme bölme probleminde, evrensel kümeden elde edilmesi mümkün olan alt kümelerin tamamını içeren bir küme bölme tablosu hazırlanırsa ve verilerin tamamı dikkate alınarak optimum çözüm elde edilirse, bulunan çözüm kesin optimum çözüm olacaktır. Çünkü, bulunan çözüm değerini daha da iyileştirebilecek durumda olup da dikkate alınmayan (optimizasyon işlemi içinde yeralmayan) herhangi bir alt kümenin bulunması ihtimali sıfırdır.

---

(1) Hein Fleuren, a.g.e., s.40

Ancak, uygulamada evrensel kümede eleman sayısının çok az olduğu çok küçük problemlerde olası bütün alt kümeler belirlenebilir ve böylece elde edilen küme bölme tablosundan optimum çözümü araştırılabilir. Çünkü, Tablo 4. de görüldüğü gibi, evrensel kümedeki eleman sayısı büyüdükçe olası alt kümelerin sayısı üstel olarak artmaktadır. Küme bölme problemlerinde her alt küme bir değişkenle ifade edilmektedir. Onbinlerce alt küme için aynı sayıda değişken yeralacağından bilgisayarla bile çözüm zamanın çok uzun olacağı ortadadır.

Olası alt kümelerin sayısında ele alınan sorunun yapısına göre farklılıklar gözlenir. Bu farklılıklar esas itibariyle; alt kümelerdeki eleman sayısının farklı (az veya çok) ve/veya evrensel kümedeki elemanlar arasındaki ilişkileri (örneğin birbirlerine olan uzaklıkları) belirten maliyet matrisinin simetrik olup olmamasından ve alt kümelerdeki eleman sayısının her alt küme için eşit veya belli değerler arasında serbest veya tamamen serbest olmasından v.s. kaynaklanmaktadır.

Eğer, maliyet matrisi simetrik ($C_{a,b} = C_{b,a}$) ise ve alt kümelerdeki eleman sayısı belli bir sayı olarak belirtilmişse olası alt kümelerin sayısı aşağıdaki formülle hesaplanır(1).

m : Evrensel kümedeki eleman sayısı
k : Her bir alt kümedeki eleman sayısı
$|F|$:Olası alt kümelerin toplam sayısı

olarak tanımlandığında,

---

(1) Metin Çakıcı, "Temel İstatistik", Dokuz Eylül Üniversitesi -Manisa İktisadi ve İdari Bilimler Fakültesi, Manisa, 1991, s.64

$$|F| = {}_{m}C_{k} \tag{3.5}$$

olur.

Yukarıdaki açıklamalara uygun problemler için, çeşitli verilere göre hesaplanan olası alt küme sayıları Tablo 4. 'de verilmiştir.

Tablo 4. Farklı Verilere Göre Olası Alt Küme Sayıları

	Alt kümedeki eleman sayısı (k)			
m	2	5	10	20
10	45	252	1	0
25	300	$5,31*10^{4}$	$3,27*10^{6}$	$5,31*10^{4}$
50	1225	$2.12*10^{6}$	$1,03*10^{10}$	$4,71*10^{13}$
75	2775	$1,73*10^{7}$	$8,29*10^{11}$	$8,03*10^{17}$
100	4950	$7,53*10^{7}$	$1,73*10^{13}$	$2,43*10^{18}$
200	$1,99*10^{4}$	$2,54*10^{9}$	$2,25*10^{16}$	$1,61*10^{27}$
500	$1,25*10^{5}$	$2,55*10^{11}$	$2,46*10^{20}$	$2,67*10^{35}$

Eğer, maliyet matrisi simetrik ve alt kümelerdeki eleman sayısı için bir üst sınır(L) belirlenmiş ($1 \leq k \leq L$) ise olası alt kümelerin sayısı şöyle hesaplanır :

$$|F| = \sum_{k=1}^{L} {}_{m}C_{k} \tag{3.6}$$

Eğer, maliyet matrisi simetrik değil ($C_{a,b} \neq C_{b,a}$) ve alt kümelerdeki eleman sayısı belli bir sayı olarak belirlenmiş ise olası alt kümelerin sayısı :

$$|F| = {}_{m}P_{k} \tag{3.7}$$

olur.

Eğer, maliyet matrisi simetrik değil ve alt kümelerdeki eleman sayısı için bir üst sınır belirlenmiş ise:

$$|F| = \sum_{k=1}^{L} {}_{m}P_{k} \tag{3.8}$$

olur.

Bu olası durumlar için kurulacak küme bölme probleminin boyutlarını, aşağıdaki sayısal sonuçlar çarpıcı bir şekilde ortaya koymaktadır.

Evrensel kümedeki eleman sayısı $M = 50$, alt kümelerdeki eleman sayısı $k = 10$ ve maliyet matrisi simetrik ise olası alt kümelerin sayısı :

$$|F| = {}_{50}C_{10} = 50! \,/\, [(50-10)! * 10!] = 10272278170$$ olur.

Alt kümelerdeki eleman sayısı $1 \leq k \leq 10$ ise :

$$|F| = \sum_{k=1}^{10} {}_{50}C_{10} = \sum_{k=1}^{10} [50! \,/\, (50-k)! * k!] = 13432735555$$ olur.

Maliyet matrisi simetrik değil ise ve $k = 10$ ise :

$$|F| = {}_{50}P_{10} = 50! \,/\, (50-10)! = 3{,}727 * 10^{16}$$ olur.

Maliyet matrisi simetrik değil ve $1 \leq k \leq 10$ ise :

$$|F| = \sum_{k=1}^{10} {}_{50}P_{k} = \sum_{k=1}^{10} [50! \,/\, (50-k)!] = 3{,}729 * 10^{16}$$ olur.

Bu verilerden anlaşıldığı gibi, maliyet matrisinin simetrik olmaması ve/veya alt kümedeki eleman sayısının belli bir sayı olmaması hallerinde olası alt kümelerin sayısı daha fazla olarak ortaya çıkar.

Bütün bu sebeplerden dolayı, büyük problemlerde olası alt kümelerin tamamının belirlenmesi durumunda küme bölme probleminin çözümünde çözüm zamanı açısından önemli zorluklar ortaya çıktığından, olası bütün alt kümelerin oluşturulması yerine, çözüm kalitesine zarar vermeyecek şekilde bir kısmının oluşturulması konusu, bu konuda çalışan hemen hemen bütün araştırmacılar tarafından önemli görülmektedir. Ancak, küme bölme problemlerinin tamamı için kullanılabilecek şekilde ortaya konulmuş bir alt küme oluşturma algoritmasına rastlanılmamıştır.

Esas olarak uygulamaya dönük çalışmalarda ortaya çıkan bu soruna, ya problemin kendi yapısına uygun şekilde özel çözüm yaklaşımları geliştirilerek ya da olası bütün alt kümeler oluşturulurken belli bir maliyetten daha fazlasını gerektiren alt kümelerin iptal edilmesi gibi yollarla çözüm bulunmaya çalışılmaktadır.

2. Küme Bölme Problemlerinin Optimizasyonu

Küme bölme problemlerinin, 0-1 tamsayılı doğrusal programlama problemlerinin özel bir yapısı olduğu daha önce belirtilmişti.

Değişkenlerin sadece 0 ya da 1 değerini alabileceği doğrusal karar modellerinde değişken sayısı az ise, tüm seçenekler taranarak en iyi çözüm kolaylıkla bulunabilir. Böyle bir problemde, her değişken için iki değer sözkonusu olduğundan, n adet değişken için uygun çözüm sayısı en fazla 2^n kadar olur. Ancak, değişken sayısı artırıldığında, tüm seçenekleri taramak güç ve zaman alıcı olur(1). Örneğin, 10 değişkenli bir problemde seçenek sayısı $2^n = 2^{10} = 1024$ iken, 30 değişkenli bir problemde ise $2^{30} = 1073741824$ olur. Bu nedenle genellikle çok sayıda değişkenin bulunduğu küme bölme problemlerinde tüm seçeneklerin taranması yolu ile en iyi çözümün bulunması uygun görülmemektedir.

Başka bir yol olarak, 0-1 tamsayılı doğrusal modeli bilinen dal ve sınır yöntemi ile çözmek de mümkündür. Ancak bu yöntemle her dal için, ele alınan doğrusal programlama probleminin çözümü gerektiğinden, oldukça uzun işlemle karşılaşılır. Karşılaşılan işlem yükünü azaltmak amacıyla,modelin yapısından hareketle, 0-1 tamsayılı modelin çözümü için özel teknikler geliştirilmiştir. Bu konuda ilk özel tekniği 1965 'de E. Balas ortaya koymuştur. Daha sonra Glover 1965 'de ve Geoffrion 1967 'de kısmi sayımlama ile çözüm için özel algoritmalar geliştirmişlerdir. Bugün, 0-1 doğrusal programlamada geniş uygulama ve kullanım gören tekniğin temeli

---

(1) İmdat Kara, "Yöneylem Araştırması, Doğrusal Olmayan Modeller", Anadolu Üniversitesi Basımevi, Eskişehir, 1986, s.149

Balas'ın algoritmasıdır. Tekniğin uygulanmasında, X_j'lerin 0 veya 1 oluşları gözönüne alınarak, yalnız toplama veya çıkarma işlemleri yapıldığından, yaklaşıma "Toplamlı Algoritma" , "Tamsayımlama Algoritması" veya "Balas 'ın Algoritması" denmektedir. Bu algoritma 2^n sayıdaki seçeneklerin tümünü işleme tabi tutmadan ve fakat göz önüne alarak en iyi çözümü araştırdığından oldukça önemli kolaylıklar sağlar(1).

Küme bölme problemlerinin çözümünde, yukarıda belirtilen yöntemler içinde en uygun olanının "toplamlı algoritma" olduğu anlaşılmaktadır. Ancak, küme bölme problemlerinin, genel 0-1 tamsayılı doğrusal modele göre bazı olumlu özelliklerinin olması, araştırmacıları bu problemler için özel algoritmalar üretmeye sevketmiştir.

Küme bölme problemleri için geliştirilen algoritmaların analizine geçmeden önce, küme bölme problemlerinin, genel 0-1 tamsayılı modellerden farkını ortaya koymanın uygun olacağı düşünülmektedir.

0-1 Doğrusal model, C_j 'ler tamsayı değer olmak üzere,

$$\text{MİN.} \quad X_0 = \sum_{j \in N} C_j X_j \qquad (3.9)$$

$$\text{Kısıtlar:} \quad \sum_{j \in N} a_{ij} X_j \geq b_i \qquad i \in M \qquad (3.10)$$

$$X_j \in \{0,1\} \qquad j \in N \qquad (3.11)$$

$$M = \{1,2,\ldots\ldots\ldots,m\}$$
$$N = \{1,2,\ldots\ldots\ldots,n\}$$

(1) İmdat Kara, a.g.e., s.149

Şeklinde formüle edilebilir(1).

Küme bölme problemi ise,

$$\text{MİN. } X_0 = \sum_{j\in N} C_j X_j$$

$$\text{Kısıtlar: } \sum_{j\in N} a_{ij} X_j = 1 \qquad i\in M$$

$$X_j \in \{0,1\} \qquad j\in N$$

$$M = \{1,2,\ldots,m\}$$
$$N = \{1,2,\ldots,n\}$$

$$a_{ij} \in \{0,1\}$$

$$C_j \geq 0$$

şeklinde formüle edilebilir. Her iki durumda da C_j 'lerin tamamı veya bir kısmı tamsayı değer değil ise, amaç fonksiyonu üzerinde yapılacak uygun matematiksel işlemlerle tamamı tamsayı durumuna dönüştürülebilir.

Yukarıdaki formüllerde de görüldüğü gibi, küme bölme problemlerinde kısıtların tamamı eşitlik şeklinde ve kısıtların sağ tarafı 1 ' e eşit durumdadır. Ayrıca, $a_{ij} \in \{0,1\}$ olarak yeralmaktadır. Yani, modelde amaç fonksiyonunun katsayıları olan C_j 'ler dışında bütün değerler 0 veya 1 'lerden oluşmaktadır ve kısıtların tamamı eşitlik şeklindedir.

Küme bölme problemlerinin bu özellikleri dikkate alınarak, bu problemlerin çözümünde, genel 0-1 doğrusal

---

(1) Hamdy A. Taha, "Operations Research - An Introduction", Mac Millan Publishing Company, Inc., 1987, s.323

tamsayılı modelin çözüm algoritmalarına göre daha kısa sürede çözüm sonucuna ulaşan algoritmalar geliştirilmiştir. Bu algoritmalardan başlıcaları şunlardır(1) :

- Tam sayımlama (implicit enumeration) algoritmaları
- Simpleks 'e dayalı kesme düzlemi metodları (simlex-based cutting plane methods)
- Kolon oluşturma algoritması (a column generating algorithm)
- Primal kesme düzlemi/Tam sayımlama çapraz algoritması (a hybrid primal cutting plane / implicit enumeration algorithm)

a. Tam Sayımlama (Implicit Enumeration) algoritmaları

Küme bölme problemlerinin optimum çözümü için geliştirilen tam sayımlama algoritmalarının temeli, 0-1 doğrusal tamsayılı nodelin çözümü için ilk önce Balas (1964) tarafından ortaya konan ve daha sonra Glover(1965) ve Geoffrion(1967) tarafından geliştirilen "Tam Sayımlama Yaklaşımı" na dayanır. Pierce(1968), Garfinkel-Nemhauser(1969), Pierce-Lasky(1973) ve Marsten(1974) küme bölme problemlerinin çözümü için birer tam sayımlama (bundan sonra tam sayımlama kısaca, sayımlama olarak ifade edilecektir) algoritmaları ortaya koymuşlardır(1).

Bu kısımda, Garfinkel-Nemhauser(1969) ve Marsten(1974) 'in algoritmalarına yer verilecektir. Çünkü, sayımlama algoritmalarının içinde bir birinden önemli ölçüde farklılık gösterenler bu iki algoritma olup, diğer sayımlama algoritmaları bu algoritmalara benzer durumdadır. Ayrıca, pratikte en fazla kullanılanların bu iki algoritma olduğu literatürden anlaşılmaktadır.

---

(1) Egon Balas ve M.W. Padberg, a.g.e., s.738-755

aa. R.S.Garfinkel ve G.L.Nemhauser'in Tam Sayımlama Algoritması

Bu algoritma, olurlu çözümlerin tamamını dal-sınır yöntemi ile tam sayımlayan belirli sayıdaki adımlar sonucunda optimal çözümü veya çözümsüzlüğü ortaya çıkarır. Algoritma, küme bölme problemlerinin özel yapısından yararlanılarak ortaya konulmuştur. Aşağıdaki açık yazılımından da görüleceği gibi oldukça basit ve sade olmakla birlikte performansı yüksektir(1).

Algoritmanın tanımlanmasına geçmeden önce $A = (a_{ij})$ matrisinde hangi indirimlerin (küme bölme tablosunda yeralan gereksiz satır ve/veya sütunların tablodan çıkarılması) yapılabileceği konusunun incelenmesi, algoritmanın tanımlanmasına kolaylık getirmesi bakımından yararlı olacaktır.

$A = (a_{ij})$ Matrisinden bazı satırların veya sütunların, problemin yapısını bozmayacak şekilde matristen çıkarılması mümkün olabilir. İndirimler (Reductions) olarak isimlendirilen bu işlemler sonucunda katsayı matrisinin olabildiğince küçültülmüş olması, optimizasyon safhası için önemli kolaylıklar sağlayacaktır.

$A = (a_{ij})$ matrisinin sütunları $A = (A_1, \ldots\ldots\ldots, A_n)$ ve satırları $A = (R_1, \ldots\ldots\ldots, R_m)^T$ olarak tanımlansın.

İNDİRİM-1 : Eğer R_i , her hangi bir (i) için boş vektör ise yani, bu vektördeki bütün elemanlar sıfır ise, problemde çözümsüzlük vardır. Bu durum, $A = (a_{ij})$ matrisi oluşturulurken i. inci elemanın hiç hesaba katılmaması veya gerçekte yok iken suni olarak boş bir i elemanının

---

(1) R.S. Garfinkel ve G.L. Nemhauser, "The Set Partitioning Problem : Set Covering With Equality Constraints", Operations Research, 17, 1969, s.848-856

ilave edilmesi ile ortaya çıkar. Eğer, i elemanın hesaba katılması unutulmuş ise $A = (a_{ij})$ matrisi bu elemanın da hesaba katılması ile yeniden düzenlenmelidir. Her hangi bir sebepten bilerek hesaba katılmamış ise bu eleman $A = (a_{ij})$ matrisinden çıkarılması gerekir. Suni olarak boş bir i elemanı ilave edilmiş ise yine çıkarılmalıdır. Örneğin,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

böyle bir matriste, son satır boş bir vektör olduğundan, bu satır fazladan ise silinmeli, değilse A matrisi, bu son elemanı da dikkate alacak şekilde yeniden düzenlenmelidir.

İNDİRİM-2 : Eğer R_k, sadece t sütunu içindeki 1 ile bir birim vektör oluşturuyor ise, bütün çözümlerde $X_t = 1$ olmak zorundadır. Bu yüzden, $a_{it} = 1$ olan bütün R_i satırları silinebilir. Aynı zamanda, $a_{it} = a_{ip} = 1$ ($p \neq t$ olmalı) olan her A_p sütunu da silinebilir. Örneğin,

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

bu matriste, $X_3 = 1$ olmak zorundadır. Bu yüzden $a_{2,3} = 1$ olan R_2 satırı silinebilir. Ayrıca, $a_{it}=a_{ip}=1$ (burada, $t \neq p$ 'dir) olan A_1, A_2 ve A_4 sütunları da silinebilir. Çünkü A_3 sütunuyla örtülenmesi(1) kesin olan satırları bu sütunlar tekrar örtülememelidir. Bu işlemler sonunda A matrisi aşağıdaki gibi olur.

$$A = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix}$$

İNDİRİM-3 : $R_k \geq R_i$ (bir vektör anlamında) ise, bu durumda hem R_k hem de $a_{kt}=1$ ve $a_{it}=0$ olan A_t sütunu silinebilir. Çünkü, R_i örtülemesi yani R_i satırında bulunan 1 'ler, R_k örtülemesindeki fazla 1 'leri hükümsüz kılar. Bu sebepten dolayı, R_i 'yi örtmeksizin R_k 'yi örten herhangi bir A_t sütunu herhangi bir uygun çözüm içinde yer alamaz. Örneğin,

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

bu matriste, $R_2 > R_1$ olduğundan R_2 satırı silinebilir. Bununla birlikte, $a_{2,5} = 1$ ve $a_{1,5} = 0$ olan A_5 sütunu da silinebilir ve A matrisi,

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

olarak indirgenmiş olur.

İNDİRİM-4 : Eğer,
$A_t = \sum_{j \in Y} A_j$ (burada, $Y \subset N = \{1,2,\ldots\ldots,n\}$ ve $t \notin Y$ 'dir) ve
$C_t \geq \sum_{j \in Y} C_j$ ise, bu durumda A_t silinebilir.

---

(1) Bir A_j sütununda bulunan 1 'lerin yeraldığı satırlara, A_j sütunu tarafından örtülenen satırlar denmektedir.Bir A_j sütununun i. inci satırı örtülemesi için $a_{ij} = 1$ olması gerekir.

Burada ele alınan 5 indirim kuralı içinde, C_j 'lere bağli olarak işlem göreni sadece indirim-4 'dür. Örneğin aşağıdaki katsayı matrisinde ve maliyet vektöründe,

$$C = [15 \quad 6 \quad 3 \quad 5 \quad 8]$$

$$A = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1 = A_2+A_3+A_4 = \begin{matrix} 1 \\ 1 \\ 1 \\ 0 \end{matrix} + \begin{matrix} 1 \\ 0 \\ 0 \\ 0 \end{matrix} + \begin{matrix} 0 \\ 1 \\ 0 \\ 0 \end{matrix} + \begin{matrix} 0 \\ 0 \\ 1 \\ 0 \end{matrix}$$ olduğundan ve

$C_1 > C_2+C_3+C_4$ yani, $15>6+3+5=14$ olduğundan A_1 sütunu silinebilir ve A matrisi,

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

olarak indirgenmiş olur.

İNDİRİM-5 : A matrisinde, $R_k \geq R_i$ veya $R_k \leq R_i$ durumunun olmadığı iki satırın var olduğunu kabul edelim. Burada, $K = \{j \mid a_{kj} > a_{ij}\}$ ve $I = \{j \mid a_{kj} < a_{ij}\}$ olsun. Bütün $j \in k$ için $a_{sj} = 1$ olan bir R_s varsa ve en az bir $j \in I$ için $a_{sj} = 1$ varsa, $j = t$ ve $j \in \{I \cap L\}$ ve $L = \{ R_s$ satırını örten sütunlar $\}$ olarak tanımlandığında, $X_t = 0$ olur ve bundan dolayı A_t sütunları silinebilir. Çünkü, bütün $j \in k$ için $X_j = 0$ olursa, $R_k \leq R_i$ olur ve indirim-3 'den dolayı bütün $j \in I$ için $X_j = 0$ olur. Diğer taraftan, eğer bir tane $j \in k$ için $X_j = 1$ olursa, bu durumda, $a_{sj} = 1$ olmasından dolayı R_s örtülenir. Bu yüzden $a_{st} = 1$ olması, $X_t = 0$ olması anlamına gelir. Örneğin,

$$A = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

matrisinde, $R_1 \geq R_2$ ve $R_1 \leq R_2$ değildir. Burada,

$$K = \{j \mid a_{1j} > a_{2j}\} = \{3\} \text{ ve } I = \{j \mid a_{1j} < a_{2j}\} = \{1,2,5\}$$

olduğuna göre, K kümesinin belirlediği bütün sütunları ve I kümesinin belirlediği sütunlardan en az birisini örten R_3 satırı bulunduğundan, $L = \{2,3,5\}$, $I \cap L = \{2,5\}$ olduğundan A_2 ve A_5 sütunları silinebilir ve a_{ij} matrisi,

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

olarak indirgenmiş olur.

R.S.Garfinkel ve G.L.Nemhauser tarafından ortaya konan bu indirim işlemleri, problemde değişken sayısını ve kısıt sayısını azalttığından, optimizasyon işlemi hangi algoritmayla yapılırsa yapılsın küme bölme tablosunda bu işlemlerin yapılması halinde önemli kolaylıklar sağlanabileceği ortadadır.

R.S.Garfinkel ve G.L.Nemhauser 'in algoritmasında dikkate alınacak kümeler şöyle tanımlanabilir.

Q = (i | $i \in M$ ve indirim işlemleriyle silinmeyen R_i)
P = (j | indirim işlemleriyle silinmeyen A_j)
P_1= (j | indirim işlemleriyle $X_j = 1$ olduğu belirlenen ve silinen A_j)

($X_j = 0$ olduğu belirlendiği için silinen sütunlar artık dikkate alınmaz)

Eğer, $a_{1,j} = , \ldots\ldots\ldots, = a_{i-1,j} = 0$ ve $a_{ij} = 1$ durumu mümkünse, bu durumda A_j 'nin, i listesinde(1) olduğu kabul edilerek, ($j \in P$) olan A_j sütunları listelerde düzenlenir. Bu düzenlemeden önce, $i \leq k$ dikkate alınarak

$\sum_j a_{ij} \leq \sum_j a_{kj}$ olacak şekilde i satırları ($i \in Q$ 'dır) sıralanırsa avantaj sağlar. Yukarıdaki düzenleme yapıldığında, her bir sütunun eksiksiz olarak bir listeye kaydedilmesi ile oluşan m listelerine sahip olunur. Ayrıca, her bir listenin karşısına C_j maliyetleri de kaydedilir.

Algoritma, mevcut olan en iyi çözümü bulmak için listeleri başından sonuna kadar uygun bir şekilde araştırır.

---

(1) a_{ij} matrisinde bulunan bir A_j sütununda ilk 1 'in bulunduğu satır numarası A_j sütununun listede kayıtlı olduğu numaradır. Yani, listedeki numaralar, karşılık geldiği sütunda ilk 1 'in hangi satırda yeraldığını gösterir.

Küme bölme problemi için kısmi bir çözüm;

$\sum_{j \in P} a_{ij} X_j \leq 1$ ($i \in Q$ 'dır), ve $X_j = 0$ veya 1

($j \in P$ 'dir) kısıtlarını sağlar.

$W = (j \mid j \in P$ ve kısmi çözümde $X_j=1) =$ (kısmi çözümde çözüme giren değişkenler)

$$Z(W) = \sum_{j \in W} c_j$$

$T = (i \mid i \in Q$ ve $\sum_{j \in W} a_{ij} = 1) =$ (kısmi çözüm ile örtülenmiş satırlar)

$V = Q - T$
$S_j = \{ j$ kolonuyla örtülenen satırlar $\}$

Bütün bu tanımlamalar dikkate alınarak R.S.Garfinkel ve G.L.Nemhauser 'in küme bölme problemlerinin çözümü için ortaya koydukları algoritmanın adımları bu yazarlar tarafından aşağıdaki gibi açıklanmaktadır(1):

I. ADIM (Başlama) : Q, P ve P_1 kümelerini elde etmek için, mümkün olan indirimler yapılır. Eğer, $Q=\emptyset$ ise ve bir çözüm mevcut ise, optimal çözüm P_1 kümesinden belirlenir. $Q \neq \emptyset$ ise, A_j sütunları ($j \in P$ 'dir) , yukarıda tanımlandığı gibi m listelerinin içinde düzenlenir.

---

(1) R.S.Garfinkel ve G.L.Nemhauser, a.g.e., s.852-853

$W = T = \emptyset$, $Z = 0$, $V = Q$ ve $Z_0 = \infty$ (Z_0 = en iyi çözüm değeri) olarak başlanır.

II. ADIM (Sonraki listeyi seç): $V = Q - T$ ve $i^* = \mathrm{Min}(i \mid i \in V)$ olsun. i^* satırını örten sütunların için den en düşük C_j maliyetine sahip olan sütun belirlenir ve Adım III. 'e gidilir.

III. ADIM (Kısmi çözümün belirlenmesi) : i^* listesinde işaretlenen durumdan başlanır ve artan maliyet sırası içinde listenin sütunları incelenir. Eğer, $T \cap S_j = \emptyset$ olan j kolonu bulunursa ve $Z(W)+C_j \leq Z_0$ ise Adım V. 'e gidilir, değilse Adım VI. 'ya gidilir.

ADIM IV (Aynı yoldan geri dönüş) : Mevcut kısmi çözümdeki sütunları içeren başka optimal çözümler yoktur. Eğer, $W = \emptyset$ ise çözüm işlemlerine son verilir. Bulunan en iyi çözüm veya çözümler optimaldir. En iyi çözüm bulunamamışsa uygun çözüm yoktur.

Eğer, $W \neq \emptyset$ ise k, W içindeki son eleman olsun. $W = W - k$, $Z(W) = Z(W) - C_k$ ve $T = T - S_k$ olarak hesaplanır. i^* = (k sütununun bulunduğu listenin numarası) olsun. Bu i^* listesi içinde k sütunu altındaki duruma bir gösterge belirlenir ve Adım III. 'e gidilir.

ADIM V (Çözümün testi) : $W = W \cup (j)$,

$Z(W) = Z(W) + C_j$ ve $T = T \cup S_j$ olarak kaydedilir. Eğer, $T = Q$ ise Adım VI. 'ya gidilir, değilse Adım II. 'ye gidilir.

ADIM VI (Çözüm bulundu) : Yeni bir en iyi çözüm bulunur. $Z_0 = Z(W)$ olarak belirlenir ve W saklanır. Adım IV. 'e gidilir.

Garfinkel ve Nemhauser bu algoritmalarını aşağıdaki sayısal örnek üzerinde şöyle açıklamışlardır.

$$A = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{cccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

$$C = \begin{bmatrix} 18 & 22 & 14 & 36 & 17 & 14 & 8 & 24 & 14 & 7 \end{bmatrix}$$

Adım I: Bu problemde ilk üç indirim kuralına uygun bir durum yoktur. Veriler, aşağıdaki tabloda olduğu gibi düzenlenir.

$P_1 = \varnothing$, $Q = (1,2,3,4,5,6)$, $P = (1,2,....,10)$,
$W = \varnothing$, $Z_0 = \infty$

Adım II: $V = (1,2,3,4,5,6)$, $i^* = 1$
Adım III: $S_j = S_3 = (1,4)$
Adım V: $W = (3)$, $Z(W) = 14$, $T = (1,4)$
Adım II: $V = (2,3,5,6)$, $i^* = 2$
Adım III: $S_j = S_6 = (2,6)$
Adım V: $W = (3,6)$, $Z(W) = 28$, $T = (1,2,4,6)$
Adım II: $V = (3,5)$, $i^* = 3$
Adım III: $S_j = S_9 = (3,5)$
Adım V: $W = (3,6,9)$, $Z(W) = 42$, $T = (1,2,3,4,5,6) = Q$

Sütun	Sj	Cj	Listede Sıralama
1	(5,6)	18	5
2	(3,5,6)	22	3
3	(1,4)	14	1
4	(2,3,4,6)	36	2
5	(1,2)	17	1
6	(2,6)	14	2
7	(5)	8	5
8	(3,4,6)	24	3
9	(3,5)	14	3
10	(4)	7	4

Adım VI: Çözümü kaydet : $W=(3,6,9)$, $Z_0=42$

Adım IV: $W=(3.6)$, $Z(W)=28$, $T=(1,2,4,6)$, $i^*=3$
Adım III: S_j bulunamadı

Adım IV: $W=(3)$, $Z(W)=14$, $T=(1.4)$, $i^*=2$
Adım III: S_j bulunamadı

Adım IV: $W=\varnothing$, $Z(W)=0$, $T=\varnothing$, $i^*=1$
Adım III: $S_j=S_5=(1,2)$

Adım V: $W=(5)$, $Z(W)=17$, $T=(1,2)$
Adım II: $V=(3,4,5,6)$, $i^*=3$
Adım III: $S_j=S_9=(3,5)$

Adım V: $W=(5,9)$, $Z(W)=31$, $T=(1,2,3,4,5)$
Adım II: $V=(4,6)$, $i^*=2$
Adım III: $S_j=S_{10}=(4)$

Adım V: $W=(5,9,10)$, $Z(W)=38$, $T=(1,2,3,4,5)$
Adım II: $V=(6)$, $i^*=6$
Adım III: S_j bulunamadı

Adım IV: $W=(5,9)$, $Z(W)=31$, $T=(1,2,3,5)$, $i^*=4$
Adım III: S_j bulunamadı

Adım IV: $W=(5)$, $Z(W)=17$, $T=(1,2)$, $i^*=3$
Adım III: $S_j=S_2=(3,5,6)$

Adım V: $W=(5,2)$, $Z(W)=39$, $T=(1,2,3,5,6)$
Adım II: $V=(4)$, $i^*=4$
Adım III: S_j bulunamadı

Adım IV: $W=(5)$, $Z(W)=17$, $T=(1,2)$, $i^*=3$
Adım III: $S_j=S_8=(3,4,6)$
Adım V: $W=(5,8)$, $Z(W)=41$, $T=(1,2,3,4,6)$
Adım II: $V=(5)$, $i^*=5$
Adım III: S_j bulunamadı

Adım IV: $W=(5)$, $Z(W)=17$, $T=(1,2)$, $i^*=3$
Adım III: S_j bulunamadı

Adım IV: $W=\varnothing$, $Z(W)=0$, $T=\varnothing$, $i^*=1$
Adım III: S_j bulunamadı
Adım IV: ***DUR***. Optimal çözüm :
$X_3=X_6=X_9=1$,
$X_1=X_2=X_4=X_5=X_7=X_8=X_{10}=0$ ve
$Z_0=42$ dir.

Bu algoritma, yine Garfinkel ve Nemhauser tarafından, IBM 7094 bilgisayarında, Fortran IV dili kullanılarak programlandı ve 20 ayrı problem bu programla çözüldü. Elde edilen sonuçlar Tablo 7. de verilmiştir.

Garfinkel ve Nemhauser 'in bu çözüm tekniğinde önemli noktalar, a_{ij} matrisinde çözüme başlamadan önce mümkün indirimler yapılarak problemin daha küçük hale getirilmesi ve sayımlama yapılırken gereksiz dallandırmalardan bazılarının (sütunların listelerde düzenlenmiş olmalarından yararlanılarak) önceden görülebilmesiyle bu işlemlerin yapılmasından kaçınılabilmesidir. Balas 'ın toplamlı algoritmasına göre üstünlüğü de buradadır. Problemin çözümünün daha kısa sürede elde edilmesini sağlayan bu avantajlar, küme bölme problemlerinde a_{ij} matrisinin 0 ya da 1 değerlerinden oluşması ve bütün

Tablo 7. Garfinkel ve Nemhauser 'in Sayımlama Algoritmasının Uygulama Sonuçları.

Problem No:	m	n	Problemin oluşum şekli	Yoğunluk(*)	Bilgisayar çözüm zamanı (saniye)
1-9	20	100	Rastgele	0.1 - 0.5	2 - 5
10	37	200	"	0.3	5
11	37	200	"	0.1	257
12	26	385	"	0.16	25
13	26	777	"	0.16	39
14	100	1400	"	0.15	895
15	37	1790	"	0.25	(>900)(**)
16	26	385	Siyasal Bölgeleme	0.16	7
17	26	777	"	0.16	16
18	37	1790	"	0.25	49
19	104	133	Ekip planlaması	0.03	2
20	40	400	Atama	0.05	(>300)(**)

(*)
$$\text{Yoğunluk} = \frac{\sum_{i \in M} \sum_{j \in N} a_{ij}}{(m)\,(n)}$$

Yani, $a_{ij} = 0$ ve 1 ' lerden oluşan $A = (a_{ij})$ matrisindeki 1 'lerin toplam sayısının, 1 'lerin ve 0 'ların toplam sayısına oranı *yoğunluk* olarak ifade edilir.

(**) Belirtilen zaman içinde bu problemlerin çözümü tamamlanamadı. Fakat her iki problem için de verilen sürede uygun çözümler bulundu

kısıtların eşitlik şeklinde olmasından kaynaklanmaktadır. Ancak, Garfinkel ve Nemhauser 'in bu özelliklerin getireceği avantajların tamamını kullanamadıkları bu konuda yapılan diğer çalışmalardan anlaşılmmaktadır.

bb. Roy E. Marsten 'in Sayımlama Algoritması (1)

Küme bölme problemi aşağıdaki şekilde yeniden formüle edilebilir.

$M = \{1, \ldots\ldots\ldots\ldots, m\}$ sıra indeks kümesi ve

$N = \{1, \ldots\ldots\ldots\ldots, n\}$ sütun indeks kümesi

olsun

ve j sütunu tarafından örtülen sıraları içeren M 'nin alt kümesi,

$$A_j \equiv \{i \in M \mid a_{ij} = 1\} \qquad (j \in N) \qquad (3.12)$$

olsun. Eğer,

$$\bigcup_{j \in J} A_j = M \qquad (J \subseteq M) \qquad (3.13)$$

ise ve $j, k \in J$ $(j \neq k)$ durumunda

$$A_j \cap A_k = \emptyset \quad \text{ise,} \qquad (3.14)$$

"M 'in herhangi bir örtülemesi" (any covering of M) olan J 'ye(2) M 'in bölümlenmiş hali veya uygun bir çözüm denir.

---

(1) R.E.Marsten, "An Algorithm For Large Set Partitioning Problems", Management Science 20, 1974, s.774-787

(2) M kümesinin her bir elemanının (a_{ij} matrisindeki her bir sıranın) sadece bir defa olmak kaydıyla mutlaka örtülendiği herhangi bir sütunlar (alt kümeler) grubuna J (yani, bir uygun çözüm) denmektedir. $a_{ij} = 1$ olması, i. inci satırın j. inci sütun tarafından örtülendiği anlamına gelir.

A matrisinin sütunlarından elde edilmesi mümkün olan, uygun çözümler kümesinin tamamının F ile gösterilmesi halinde,

$F = \{J \subseteq N \mid J,$ (3.13) ve (3.14) 'ü sağlamalı$\}$ olur. Bu tanımlamalar doğrultusunda küme bölme problemi;

$\sum_{j \in J} c_j$ 'yi minimize eden $J \in F$ 'in seçilmesi olarak ifade edilebilir.

Marsten'in ortaya koyduğu algoritma, F kümesinin (uygun çözümler kümesinin) bir tamsayımlaması esasına dayanır.

Bir dal ve sınır algoritmasını tanımlamak için, uygun çözümler kümesinin nasıl ayrıldığını ve amaç fonksiyonu değerinde alt sınırların nasıl hesaplandığını göstermek yeterlidir. Bu amaçla, algoritmayı tanımlamadan önce aşağıdaki konuları incelemenin yararlı olacağı düşünülmüştür.

Marsten 'in sayımlama algoritmasının en önemli özelliği, a_{ij} matrisinin merdiven şeklinde düzenlenmiş olmasıdır.

$A=(a_{ij})$ matrisinin merdiven şeklinde (staircase form) düzenlenmesi problemin çözümünde önemli kolaylıklar sağlar. Bu durum ilk önce Pierce tarafından ortaya konulmuştur(1). Merdiven formunda düzenleme aşağıdaki şekilde yapılır:

---

(1) J.F.Pierce, "Application of Combinatorial Programming to a Class of All 0-1 Integer Programming Problems", Management Science, 15, 1968, s.195

$t=1,2,\ldots,m$ olarak tanımlayalım. $A=(a_{ij})$ matrisinde $i < t$ sıralarındaki elemanları sıfır, $i = t$ elemanı bir olan sütunlar t. inci blokta (B_t blokunda) toplanır. Sonra, her blok içindeki sütunlar sözlük düzenleme mantığıyla sıralanır. Tablo 8. 'de verilen matris bu şekilde düzenlenmiştir.

Tablo 8. Merdiven Şeklinde Düzenlenmiş Matris.

	B_1					B_2			B_3	
	1	2	3	4	5	6	7	8	9	10
1	1	1	1	1	1	0	0	0	0	0
2	0	0	0	1	1	1	1	1	0	0
3	0	0	1	0	0	0	1	1	1	1
4	0	1	1	0	1	1	0	1	0	1

F 'nin herhangi bir alt kümesindeki (herhangi bir uygun çözüm içindeki) sütunlardan her biri, ayrı ayrı bloklarda bulunmak zorundadır. Aksi halde, herhangi bir B_t blokunda bulunan sütunlardan birden fazlası aynı çözüm içinde yer alırsa , t satırı birden fazla sütun tarafından örtülmüş olur ki, bu bir uygun çözüm olamaz.

Tablo 8. 'deki gibi düzenlenen bir A matrisinde her B_t 'nin son sütun numarasına n_t dersek,

$$B_t = \{j \in \mathbf{N} \mid n_{t-1} + 1 \leq j \leq n_t\}$$ olur.

Burada, $n_0 = 0$, $n_m = n$ dir. B_t boş ise, $n_{t-1} = n_t$ olur. Tablo 8. 'deki matriste; $n_1 = 5$, $n_2 = 8$, $n_3 = 10$ ve $n_4 = 10$ 'dur.

Marsten 'in algoritmasında, uygun çözümlerin ayırımı (belirlenmesi) esas olarak merdiven formundan yararlanılarak yapılmaktadır.

Bu ayırma metodunda, hangi satırların hangi sütunlar için değil, hangi bloklar için ayrılacağı mantığı vardır. Bloklar içindeki sütunların, yukarıda anlatıldığı şekilde belli kurallara göre düzenlenmiş olmasından dolayı bu mantıktan yararlanılarak hareket etmek önemli kolaylıklar sağlamaktadır.

(1,.....,r) sıralarının, sıra ile $g(1),......,g(r)$ bloklarına atanacaklarını düşünelim. (i) satırının ($i=1,...,r$), $g(i)$ bloku içinde bulunan bir sütun tarafından örtülenmesi halinde oluşturulması mümkün olan uygun çözümler kümesi F^g ile gösterilsin. $F^g \subseteq F$ dir. t bloku ($t=1,.....,m$) için ayrılmış olan sıralar kümesi ise , $g^{-1}(t)$ ile gösterilsin.

Bu tanımlamalara bağlı olarak,

$$A_j \cap \{1,...,r\} = g^{-1}(t) \qquad (3.15)$$

olmadıkça, $j \in B_t$ sütununun F^g 'nin herhangi bir üyesine dahil edilemeyeceği kolayca görülebilir. Bu durum şöyle de ifade edilebilir: j sütunu t bloku için önceden atanan her bir sırayı örtmeli, fakat, başka bir blok için önceden ayrılan sırayı örtmemelidir. Herhangi bir blokta bu şartları taşıyan birden fazla sütunun bulunabilmesinin sebebi atama yapılmamış blokların varlığıdır. (3.15) Eşitliğindeki $\{1,2,....,r\}$ kümesi çeşitli bloklara ataması yapılmış satırlar kümesidir.

g 'nin kısmi atamasından dolayı her bloktaki sütunların bazıları elenebilir.

$$B^g{}_t \equiv \{ j \in B_t \mid A_j \text{ , (3.15) 'i sağlamalı} \} \qquad (3.16)$$

olarak tanımlandığında $B^g{}_t$, B_t 'nin sütunlarından F^g 'nin üyelerine dahil edilmesi uygun olanları ifade eder. $r = m$ ise (yani, satırların tamamı atanmışsa), $B^g{}_t$ 'nin ya boş ya da tek bir kolondan ibaret olacağı görülebilir. $B^g{}_t$ için sol taraf ve sağ taraf sınırları,

$$\text{Sol}.B^g{}_t \equiv \text{Min}\{ j \in B_t \mid A_j \text{ , (3.15) 'i sağlamalı} \}$$

$$\text{Sağ}.B^g{}_t \equiv \text{Max}\{ j \in B_t \mid A_j \text{ , (3.15) 'i sağlamalı} \}$$

olarak tanımlandığında,

$$B^g{}_t \equiv \{ j \in B_t \mid \text{Sol}.B^g{}_t \leq j \leq \text{Sağ}.B^g{}_t \} \qquad (3.17)$$

olur.

Algoritmanın yürütülmesi esnasında, her blok için sol ve sağ taraf sınırları olmalıdır. Bu sınırlar, veri olarak dikkate alınan g 'nin değiştirilmesi ile ayarlanır ve bu durumda (3.17) eşitliği daima sağlanır. B_t 'nin sözlük düzeninde sıralanmış olmasından dolayı, bu ayarlama işlemi çok basit olarak yapılabilir. Tablo 8. 'deki problem için; $g(1) = 1$, $g(2) = 2$, ve $g(3) = 2$ (yani, birinci satır birinci blok, ikinci satır ikinci blok ve üçüncü satır ikinci blok için) atamaları yapıldığında, sol ve sağ sınırlar Tablo 9. 'daki gibi ortaya çıkar.

Tablo 9. $g(1)=1$, $g(2)=2$, ve $g(3)=2$ Olması Halinde Sol ve Sağ Taraf Sınır Durumları.

	Sol Sınır	Sağ Sınır
B_1	1	2
B_2	7	8
B_3	9	10

Bu aşamada, $0 \leq r \leq m$ olduğunda, $g(r+1)$ için mümkün seçimlerin kümesini T^g ile tanımlayalım. $A=(a_{ij})$ matrisinin merdiven formunda olmasından dolayı, $g(r+1) \in \{1,......,r+1\}$ olarak gösterilebilir. Burada, $t \leq r$ $(t=1,......,m)$ iken $g(r+1) = t$ 'nin iki durumu sözkonusudur:

I. Durum : $g(t) \neq t$ ise, $B^g{}_t$ boştur ve $(r+1)$ sırası t bloku için atanamaz.

II. Durum : $g(t) = t$ durumunda,

A) $B^g{}_t$ 'nin hiç bir sütunu $(r+1)$ sırasını örtmüyor ise, $(r+1)$ sırası t bloku için atanamaz.

B) $B^g{}_t$ 'nin en az bir sütunu $(r+1)$ sırasını örtüyor ise, $(r+1)$ sırası t blokuna atanabilir.

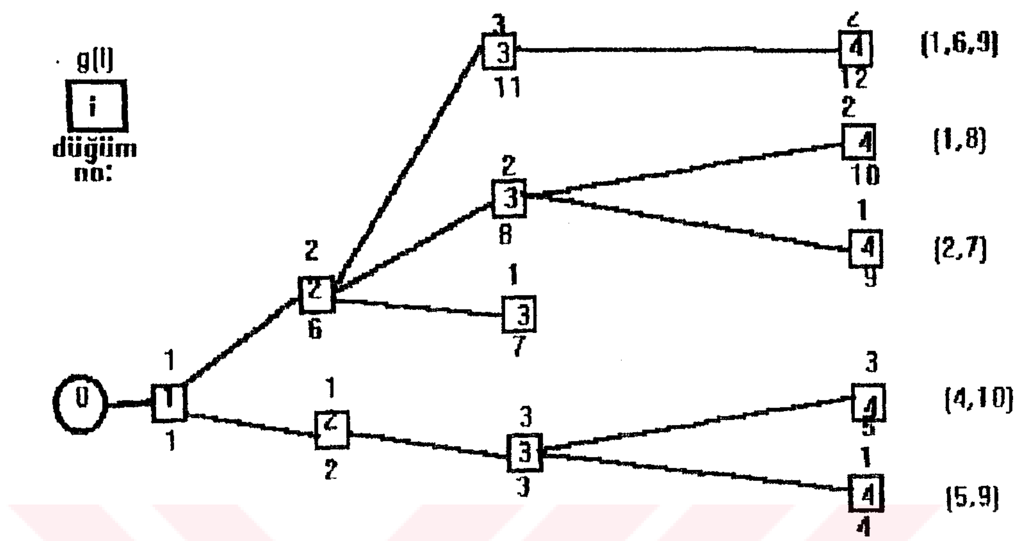
C) $B^g{}_t$ boş değilse ve onun içindeki bütün sütunlar $(r+1)$ sırasını örtüyor ise, $(r+1)$ sırası t bloku için mutlaka atanmalıdır.

İki veya daha çok $t \leq r$ durumunda II-C sağlanırsa, $(r+1)$ sırasını atamak imkansız olur ve $T^g = \emptyset$ dir. II-C durumunu sağlayan bir tane $t^* \leq r$ olursa, bu durumda, $T^g = \{ t^* \}$ olarak belirlenir. Diğer durumlarda, II-B 'yi sağlayan her $t \leq r$ 'yi T^g kümesine dahil ederiz ve B_{r+1} boş değil ise $(r+1)$ 'i de T^g kümesine dahil ederiz. Eğer, $t \leq r$, I veya II-A durumuna rastlarsa ve B_{r+1} = Ø olursa, T^g =Ø olacağı görülebilir.

Tablo 10. 'da, Tablo 8.'deki problem için, buraya kadar belirlenen kurallara uygun şekilde yapılan bir tamsayımlamanın sonuçları sunulmuştur. Bu sayımlamada 13 düğüm vardır. Ana düğüm sıfır ile ifade edilir. "Önceki Düğüm" sütunu, her düğümün bağlı olduğu ön düğümü verir. (i) ve g(i) sütunları, bir önceki düğüm ile bu düğümü birleştiren dalı işaret eder. Her terminal düğüm (bir dalın sonundaki düğüm) çözüm yerine karşılık gelir. Uygun çözümler son sütunda veriliyor. Tablo 9. 'daki örnek sınır durumları Tablo 10. 'daki 8. inci düğümde elde edilen sınırlara karşılık gelir. Tablo 10. 'daki bilgilere göre oluşturulan dallar, Şekil 3. 'deki gibi ilerlemektedir.

Tablo 10. Tablo 8. 'de Verilen Problemin Tamsayımlama Sonuçları.

Düğüm	Önceki Düğüm	i	g(i)	Çözüm
1	0	1	1	
2	1	2	1	
3	2	3	3	
4	3	4	1	(5,9)
5	3	4	3	(4,10)
6	1	2	2	
7	6	3	1	
8	6	3	2	
9	8	4	1	(2,7)
10	8	4	2	(1,8)
11	6	3	3	
12	11	4	2	(1,6,9)



Şekil 3. Tablo 10. 'daki Verilerin Dallandırma Formunda Gösterimi

Marsten 'in algoritmasında alt sınır değerleri; problemin, tamsayılı olmayan doğrusal programlama problemi çözümü sonucunda elde edilen amaç fonksiyonu değeriyle elde edilir.

Yukarıda tanımlanan F^g kümesi, küme bölme problemi (KBP) 'nin aşağıdaki şekilde uygulanması halinde uygun çözümler kümesini ifade eder.

$$(KBP^g) \qquad \text{Min} \quad X_o = \sum_{j \in N(g)_a} c_j x_j$$

$$\text{Kısıtlar:} \quad \sum_{j \in N(g)_a} a_{ij} x_j = 1 \qquad (i \in M)$$

$$x_j = 0 \text{ veya } 1 \qquad (j \in N_{(g)a})$$

Bu durumda,

$$N_{(g)a} \equiv \bigcup_{t \in M} B^g{}_t \;, \qquad (3.18)$$

(g) belli iken bütün mevcut (kullanılmaya hazır) sütunlar kümesidir. (DP^g), yukarıdaki (KBP^g) modelinde $X_j=0$ veya 1 kısıtı yerine $X_j \geq 0$ kısıtının yeralması durumundaki doğrusal programlama modelidir. (KBP^g) 'nin minimizasyonunda amaç fonksiyonu için bulunabilecek en alt sınırını , (DP^g) 'nin optimal çözümündeki amaç fonksiyonunun değeri belirler. X^g, (DP^g) 'nin optimal çözümü olsun ve $N_{(g)b}$=(X^g içindeki temel değişkenler kümesi) olsun. Şimdi, (h) 'nin aşağıdaki gibi belirlendiğini kabul edilsin:

$$h(\,i\,) = \begin{cases} g(\,i\,) & (\,i = 1,\ldots\ldots,r) \\ t & (\,i = r+1) \qquad (t \in T^g) \end{cases} \qquad (3.19)$$

$N_{(h)a} \subseteq N_{(g)a}$ olduğunda, bundan dolayı (X^g), (DP^h)'nin dual uygun çözümüdür. Eğer,

$$N_{(g)b} \subseteq N_{(h)a} \qquad (3.20)$$

ise, bu durumda X^g , (DP^h) 'nin optimal çözümüdür. Eğer, (3.20) sağlanmazsa,

$(N-N_{(h)a}) \cap N_{(g)b} = \{j_1,\ldots\ldots,j_k\} \neq \varnothing$ olur. Yani, (g) 'ye göre temel olan sütunlardan bazıları (h) 'ye göre temel değildir. Bu durumda, $X_{j_1}+\ldots\ldots+X_{j_k} = 0$ ilave sınırını eklemekle ortaya çıkan (DP^g) amaç fonksiyonundan (gittikçe artmakta olan) alt sınır hesaplanabilir. Bu alt sınır CEZA (h) diyoruz. (3.20) ifadesi sağlanırsa CEZA (h)=0 olur. Her iki durumda da, (DP^g) nin optimal tablosundan direkt olarak hesaplanabilen (DP^h) 'nin değeri ile bir alt sınıra sahip olunur.

$v(DP)$, (DP)'nin optimal değeri olarak tanımlandığında,

$$SINIR(h) = v(DP^g) + CEZA(h) \quad \text{olur.}$$

(3.19) ifadesinden anlaşıldığı gibi, algoritma her $t \in T^g$ için SINIR(h) 'yi hesaplar. T^g kümesi, SINIR 'daki değişikliğe uygun olarak ayrılır. (g) 'deki her değişme durumunda öncelikle alt sınır araştırılır.

Buraya kadar elde edilen özel sonuçlar ve tanımlamaları dikkate alan R.E.Marsten, küme bölme problemlerinin çözümü için ortaya koyduğu algoritmasının adımlarını aşağıdaki şekilde açıklamaktadır(1).

Adım 0: $i = 0$, $g = \varnothing$, $SINIR(\varnothing) = 0$, $T^{\varnothing} = \{1\}$ ve $V = \infty$ olarak belirlenir.

Adım 1: Eğer, (DP^g) çözümü uygun değil ise (değişken değerleri 0 veya 1 olarak bulunmamışsa), Adım 10. 'a gidilir.

Adım 2: $v(DP^g) \geq V$ ise, Adım 10. 'a gidilir.

Adım 3: (DP^g) 'nin optimal çözümü tamsayı (0 veya 1) ise Adım 9. 'a gidilir.

Adım 4: $i = m$ ise Adım 9. 'a gidilir.

Adım 5: $i = i+1$

(1) R. E. Marsten, a.g.e., s. 780-781

Adım 6: $T^g = \varnothing$ ise Adım 13. 'e gidilir.

Adım 7: Her $t \in T^g$ için SINIR hesaplanır ve sınırlara uygun olarak artan düzen içinde T^g seçilir.

Adım 8: $T_i=T^g$ olarak belirlenir ve Adım 11.'e gidilir.

Adım 9: Çözüm kaydedilir. $V=Min\{V , v(DP^g)\}$ belirlenir.

Adım 10: $T_i=\varnothing$ ise Adım 13. 'e gidilir.

Adım 11: t^* = (T_i içinde ilk eleman) olsun. $g(i)=t^*$ ve $T_i = T_i-t^*$ olarak belirlenir.

Adım 12: $SINIR(g) \geq V$ ise, Adım 10.'a gidilir. Değilse Adım 1. 'e gidilir.

Adım 13: $i = 0$ ise DURULUR.

Adım 14: $i = i -1$ olarak belirlenir ve Adım 10.'a gidilir.

Bu algoritmanın kullanılması ile çözülen beş ayrı problemle ilgili sonuçlar Tablo 11. de verilmiştir. Çözüm için IBM 360/91 marka bilgisayardan yararlanılmıştır.

1. inci problemin gerçek büyüklüğü 147x544 iken, yapılan indirgemelerle 90x303 olarak, aynı şekilde 3. üncü problem de 117x4845 'den 111x4826 olarak küçültülebilmiştir. 4. üncü problem, 5. inci problemin bir alt kümesinden elde edilmiştir.

Tablo 11. Marsten 'in Algoritmasıyla Yapılan Çözümlerin Özeti

Prob No:	m	n	I	II	III	IV	V	VI	VII
1	90	303	3.53	42,719.50	32.11	42,855.00	56	1	7
2	63	1641	84.50	60,902.50	84.26	60,990.00	12	2	11
3	111	4826	314.93	217,351.00	165.84	217,687.00	19	3	5
4	200	2362	418.86	81,730.00	0.00	81,730.00	0	1	1.5
5	419	21585	>3 saat	>119,800	?	?	?	?	2

Tablo 11. 'deki sütunlarda :

m : satır sayısını,

n : sütun sayısını,

I : Doğrusal programlama probleminin çözümü için geçen süre (saniye),

II : Doğrusal programlama probleminin çözümüyle bulunan minimum maliyet,

III: Sayımlama safhasında geçen zaman (saniye),

IV : Tamsayılı çözümde minimum maliyeti,

V : Sayımlama safhasında herhangi bir dönemde atanan maksimum sıra sayısı,

VI : Bulunan tamsayılı optimum çözüm planı sayısı,

VII: Katsayı matrisinin yoğunluğu (%) 'nu,

ifade ederler.

Marsten 'in bu algoritmasını, küme bölme problemleri üzerinde çalışan araştırmacıların çoğu tarafından ya aynen ya da ele aldıkları sorunun özel yapısına göre bazı değişiklikler yaparak kullandıkları literatürden anlaşılmaktadır. Bu algoritmanın en önemli özelliği, büyük küme bölme problemlerinde dahi kesin optimum çözümü uygun bir sürede bulabilme ihtimalinin yüksek olmasıdır. Algoritmada, simpleks yöntemi ve sayımlama yöntemi iç içe kullanılmaktadır. Bazı küme bölme problemlerinin simpleks yöntemle çözümü çok uzun sürebilmektedir. Tablo 2. 'deki 5 nolu problemde bu durum görülmektedir. Marsten 'in algoritması bir bütün olarak ele alındığında bu dezavantajının bazı küçük problemlerde dahi sorun olma olasılığı mevcut gözükmektedir.

Marsten 'in bu çalışmasıyla ortaya koyduğu en önemli fikir; küme bölme problemlerinin tamsayımlama metoduyla çözümünde, merdiven formunda hazırlanmış katsayı matrisinin kullanılmasıdır. Katsayı matrisinin merdiven formunda düzenlenmiş olmasının avantajı, sayımlama yaparken uygun bir çözüme çıkmayacak dalların çok büyük bir kısmı çok önceden (ilk ana dalların oluşumu sırasında) bilinebilmekte ve böylece gereksiz işlemleri yapmak zorunda kalınmamaktadır.

b. Simpleks'e Dayalı Kesme Düzlemi Metodları

Tamsayılı doğrusal programlama problemlerinin çözümünde kullanılan kesme düzlemi metodu,ilk defa R.E.Gomory tarafından ortaya konulmuştur. Kesme düzlemi metodunda çözüme doğrusal programlama çözümü ile başlanır. Daha

---

(1) Bu tez çalışmasının uygulama kısmında ele alınan, Türkiye ikinci futbol ligi ile ilgili olarak düzenlenen m=53 ve n=168 boyutundaki küme bölme probleminin simpleks çözümü de 3 saate yakın sürmüştür.)

sonra sistematik olarak, değişkenlerin tamsayı olma özelliğini kapsayan kısıtlayıcılar çözüme dahil edilir. Bu işlemlere optimal tamsayılı çözüm bulununcaya kadar devam edilir. İlave edilen her tamsayı kısıtı, uygun çözüm alanının tamsayılı çözüm bulundurmayan bir bölgesini keserek çözüm dışı bırakır. Yönteme, kesme düzlemi yöntemi denmesinin sebebi de budur. İlave edilen kısıtlarla yapılan kesmeler, uygun çözüm alanının konveks set olma özelliğini bozmamalı ve çözüm alanı dışında bırakılacak bölgede tamsayılı çözüm verecek nokta bulunmamalıdır(1).

Tamsayılı programlama problemlerinin çözümü için kullanılan yöntemlerin birbirine üstünlüklerinde esas kıstas, çözüm zamanının kısalığı olmaktadır. Hangi yöntemin konuyla ilgili bütün problemlerde daha kısa zamanda çözüme veya çözümsüzlüğe ulaştığı konusunda kesin bir şey söylemek mümkün görülmemektedir. Ancak, bir tamsayılı doğrusal programlama probleminin, tamsayılı olmayan doğrusal programlama problemi olarak çözümü sonunda temel değişkenlerin çok az bir kısmının değerleri kesirli sayılar olursa, bu durumda, kesme düzlemi metodunun daha verimli olması olasılığının daha yüksek olacağı söylenebilir.

Küme bölme problemlerinin doğrusal programlama çözümünde genellikle, temel değişkenlerin çok büyük bir kısmının değerleri tamsayı (0 veya 1) olabilmektedir. Bu oran rasgele düzenlenmeyen gerçek problemlerde daha da yüksek olabilmektedir. Hatta, gerçek problemlerin bazılarında tüm temel değişkenlerin tamsayı değer aldıkları da görülür. Bundan dolayı, küme bölme problemlerinin kesme düzlemi metoduyla çözümünde avantajlar olabilir(2).

---

(1) Atilla Sezgin-Erhan Ada, "İşletmeciler İçin Yöneylem Araştırması", Gazi Üniversitesi, İ.İ.B.F., İşletme Bölümü, Ankara, 1990, s.174

(2) Egon Balas ve Manfred W.Padberg, "Set Partitioning : A Survey", a.g.e., s.227-247

Gomory (1963) 'nin kesme dözlemi metodunu yeniden tanımlayan Martin(1963), 1969 yılında yaptığı fakat yayınlamadığı çalışmasında, büyük küme bölme problemlerinin (hava yolu mürettebat planlaması problemleri) çözümü için yeni bir çözüm metodu kullandı. Bu çalışmayla çözülen 300-500 kısıtlı ve 1500-4000 değişkenli küme bölme problemleri çok büyük olmasına rağmen kesmelerin sayısı 10 'dan az olarak ortaya çıktı. Sadece bir tanesinde 16 olarak gerçekleşti. Bununla birlikte, 300 değişkenden daha az olan bazı problemler uygun bir süre içinde çözülemedi(1).

Kesme düzlemi metodları ile küme bölme problemlerinin çözümü konusunda 1974 'de Delorme tarafından da bir çalışma yapılmıştır(2). Bu çalışmada ortaya konan kesme düzlemi metodu; Garfinkel-Nemhauser 'in sayımlama algoritmasının Heurgon(1972) tarafından ortaya konulan yeni bir versiyonu ile, çeşitli büyüklükteki küme bölme problemlerinin çözümünde karşılaştırmalar yapıldığı ve kesme düzlemi metodu kullanılmasıyla daha az ortalama süre içinde çözüme ulaşıldığı aynı çalışmada rapor edilmiştir.

Bu konuda yapılan bilgisayar uygulamaları için Delorme-Heurgon (1975) 'ın adı geçen çalışmasına bakılabilir. Bu çalışmalarda kullanılan kesmeler, en büyük kesirli sabit terimin bulunduğu sıradan yapılır. Her bir iterasyondan sonra kesirli sayı varsa yeni bir kesme yapılır. Bu işlemlere bütün temel değişkenlerin tamsayı değer almasına kadar devam edilir. Tamsayılı olmayan

---

(1) E. Balas ve M. W. Padberg, a.g.e., s. 242

(2) J. Delormme ve E. Heurgon, "Problemes de Partitionnement: Exploration Arborescente Ou Methodes de Troncatures?", Rev.Francaise Autom., Inf. Rech. Oper., 9, v-2, 1975, s. 53-65

doğrusal programlama problemi olarak bulunan optimum çözümden sonra kesmelerle ilgili iterasyonlar yapılırken tablonun tamamının kullanılması şart değildir. Bir çok problemde kesme sayısı 3-4 'ü nadiren geçer. Kesme düzlemi metoduyla çözülen çeşitli küme bölme probemleri ile ilgili özet sonuçlar Tablo 12. 'de verilmiştir(1).

Tablo 12. Kesme Düzlemi Metoduyla Yapılan Çözümlerin Özet Sonuçlar.

Prob. No:	m	n	Toplam kesme sayısı	Geçen süre (saniye)	İteras-yon sayısı	I	II	III
1	111	2186	6	152	59	4	3320	3327
2	77	969	7	46	27	122	3240	3271
3	78	693	5	35	66	12	2942	2945
4	91	1019	3	28	42	4	3161	3162
5	51	913	3	15	49	3	450	479
6	88	953	1	2	1	68	3305	3316
7	58	405	5	23	35	24	2870	2930
8	185	1043	1	12	5	3	3061	3080

Tablo 12. 'de ;

I Sütunu, Küme bölme probleminin doğrusal programlama olarak çözümü sonucunda ortaya çıkan kesirli temel değişken sayısını,

II Sütunu, Doğrusal programlama çözümünde amaç fonksiyonunun optimum değerini,

---

(1) E. Balas ve M. W. Padberg, a.g.e., s. 242

III Sütunu, Kesme düzlemi metodunun uygulanmasıyla bulunan optimum tamsayılı çözümün amaç fonksiyonu değerini ifade etmektedirler.

Yukarıda adı geçen çalışmaların sonuçlarından da anlaşıldığı gibi, küme bölme problemlerinin simplekse dayalı kesme düzlemi metoduyla çözümü, tamsayımlama metoduna göre bazen daha az sürede olabilmektedir. Bunun için, problemin ilk simpleks çözümünde, simpleks çözüm süresinin kısa olması ve bu çözümün sonucunda tamsayılı değer almayan değişken sayısının çok az olması gerekmektedir.Ancak, ele alınan bir küme bölme probleminin bu sonuçları vereceğini önceden görmek mümkün olmayabilmektedir. Çükü, bu problemlerde değişken sayısı ve kısıt sayısı fazladır.

Bu durumda, bir küme bölme probleminin kesme düzlemi metoduyla çözümüne başlandığında simpleks çözüm zamanı ve kesirli değer alan değişken sayısı fazla olursa, bu metodla çözüm yapmaktan vazgeçmek gerekecektir.

Bu tez çalışmasında, Türkiye ikinci futbol ligiyle ilgili olarak hazırlanan (53x168) boyutundaki küme bölme probleminin, "QSB" isimli yöneylem paket programında simpleks çözümü yapıldı. Çözüm süresi 2.48 saat ve optimum çözümde kesirli değer alan değişken sayısı 8 olarak ortaya çıktı. Buna karşılık aynı problemin, uygun bir tamsayımlama algoritmasıyla 163 saniyede optimum çözümü elde edildi. Bu nedenlerden dolayı, bu araştırmada yapılacak uygulama çalışmalarında simplekse dayalı kesme düzlemi metodunun kullanılması uygun görülmemektedir.

D. Araştırmada Kullanılan Model ve Veriler

1. Uygulama Probleminin Özellikleri

Küme bölme problemlerinin ele alınan soruna göre farklı özellikler gösterdiği III. bölümde genel olarak ele alınmıştı. Bu özelliklerden biri, alt kümelerdeki eleman sayısının bütün alt kümeler için eşit veya farklı olması durumudur. Eşit olması halinde, optimum çözümde kaç tane değişkenin 1 değerini alacağı, yani kaç tane alt kümenin oluşacağı önceden kesin olarak bilinebilmektedir. Eşit olmaması halinde ise, bu sayı kesin olarak bilinememektedir.

Örneğin; 100 elemandan oluşan bir Evrensel kümenin 20 elemanlık alt kümelere ayrılması istendiğinde, optimum çözümde, 100/20=5 alt kümenin bulunacağı önceden hesaplanabilir. Ancak, alt kümelerin eleman sayısı 1-25 arasında değişebilir özellikte olan bir problemin optimum çözümünde bulunacak alt küme sayısı; 100/25=4 ile 100/2=50 arasında bir rakam olacağı önceden belirlenmekle birlikte bu verilerden alt küme sayısını veren tek bir rakamın bilinmesi mümkün görülmemektedir.

Buna karşılık bazı küme bölme problemlerinde alt kümelerdeki eleman sayısı farklı olmasına rağmen optimum çözümdeki alt küme sayısının kaç olması gerektiği önceden kısıt olarak verilmiş olabilir.

Örneğin, 1992-1993 futbol sezonunda Türkiye ikinci futbol ligindeki durum böyledir. Bu sezonda, ikinci ligte

53 takım yer almakta ve bunların 5 alt gruba ayrılması istenmektedir. Alt gruplardaki takım sayısı 10 veya 11 olacaktır. Yani, 53 elemandan oluşan bir Evrensel kümenin, eleman sayıları 10 veya 11 olan 5 alt kümeye ayrılması sorunuyla ilgili bir küme bölme problemi söz konusudur.

1992-1993 Türkiye üçüncü futbol ligindeki durum ise, alt kümelerdeki eleman sayılarının eşit olduğu şeklindedir. Bu sezonda üçüncü ligte 160 takım yer almakta ve bunların 16 'şar takımdan oluşan 10 alt gruba ayrılması gerekmektedir. Yani, 160 elemandan oluşan bir Evrensel kümenin 16 'şar elemanlı 10 alt kümeye ayrılması sorunuyla ilgili bir küme bölme problemi söz konusudur.

Yukarıda bahsedilen iki örnek problem, bu çalışmada uygulama olarak seçilen problemlerdir. Bu problemlerin kendine has özelliklerinden en önemlileri, alt kümelerdeki eleman sayıları ya eşit ya da eşite çok yakın olması ve alt küme sayısının kesin olarak bilinmesidir.

Bu özelliklerin çözüm algoritmasında dikkate alınmasıyla çözüm işlemlerinde ve zamanında önemli indirimler yapılması düşünülmektedir. Bu özellikteki problemler için, III. bölümün küme bölme problemlerinin optimizasyonu ile ilgili kısmında ele alınan algoritmalardan en uygununun Marsten 'in sayımlama algoritması olduğu söylenebilir. Çünkü, Garfinkel ve Nemhauser 'in sayımlama algoritmasına göre, Marsten 'in sayımlama algoritması merdiven formundan yararlanmasından dolayı çok hızlıdır.

Ancak, Marsten 'in algoritmasında defalarca simpleks çözüm de gerekmektedir. Simpleks algoritması genelde çok hızlı olmasına rağmen küme bölme problemlerinin çözümünde çok uzun zaman gerekebilmektedir. İkinci lig ile ilgili problemin "QSB" paket programıyla simpleks çözümü yapıldığında çözüm süresi 2.48 saat ve çözüm sonunda kesirli değer alan değişken sayısı da 8 olarak ortaya çıkmıştır. Çözüm süresinin uzun olması nedeniyle bu problemlerde Marsten 'in algoritmasını olduğu gibi kullanmak uygun olmamaktadır.

Aynı şekilde, hem simpleks çözüm zamanının uzun olması hem de kesirli değer alan değişken sayısının fazla olması nedeniyle "simpleks 'e dayalı kesme düzlemi metodu" nun kullanılması da uygun olmamaktadır.

Marsten 'in sayımlama algoritmasındaki merdiven formunun kullanılmasıyla geliştirilebilecek saf bir sayımlama (simpleks yönteminin hiç kullanılmadığı) algoritmasıyla bu problemlerin uygun bir sürede çözülebileceği düşünülmektedir.

2. Küme Bölme Tablosunda Yapılacak Düzenlemeler

Küme bölme tablosunun; A_j alt kümelerinin nasıl oluştuğunu gösteren a_{ij} matrisi ve bu alt kümelerin oluşum maliyetlerini (bir alt kümedeki elemanların bu alt kümede yeralmaları nedeniyle ortaya çıkan maliyetlerin toplamı) gösteren C_j maliyet vektöründen oluştuğu daha önceki bölümlerde belirtilmişti.

Problemin optimizasyonu safhasına geçmeden yani algoritmanın uygulanmasına başlamadan önce, küme bölme tablosunda aşağıdaki düzenlemelerin yapılması problemin çözümünde çok büyük kolaylıklar sağlayacaktır. Bu işlemler:

- a_{ij} matrisinde mümkün indirimlerin yapılması,

- a_{ij} matrisinin merdiven formunda düzenlenmesi,

- Merdiven formunda düzenlenmiş matriste oluşan bloklar içindeki alt kümelerin, alt küme maliyetlerine göre sıralandırılması şeklinde yapılacak düzenlemelerdir.

Bu işlemler, aşağıda geniş olarak ele alınmaktadır.

a_{ij} matrisinde yeralan satır ve/veya sütunlardan bazıları gereksiz olarak yeralıyor olabilirler. Yani, bunların problemin optimum çözümünü etkilemeyecekleri önceden belirlenebilir. Bu durumdaki satır ve sütunlar bölüm III.C.'de belirtilen kurallara uygun olarak tablodan çıkarılır. Böylece a_{ij} matrisinde uygun indirimler yapıldığında, matrisin boyutları küçültülebilir ve problemin çözümünde önemli kolaylıklar sağlanabilir.

Küme bölme tablosunda yapılacak ikinci düzenleme ise 0 veya 1 değerlerinden oluşan a_{ij} matrisinin merdiven formunda oluşturulmasıdır. Merdiven formu:

$t=1,2,\ldots\ldots\ldots,m$

$i=1,2,\ldots\ldots\ldots,m$

$j=1,2,\ldots\ldots\ldots,n$

m= satır sayısı

n= sütun sayısı

$a_{ij} \in \{0,1\}$

olarak tanımlandığında, a_{ij} matrisinde $i<t$ sıralarındaki elemanları 0 ve $i=t$ elemanı 1 olan sütunların t.'inci blokta toplanmasıyla oluşturulur. Merdiven formunun

oluşturulmasıyla ilgili olarak III.C bölümünde daha geniş açıklamalar yapılmıştı.

Bu çalışmada kullanılacak tamsayımlama algoritmasının en önemli özelliği, merdiven formunda düzenlenmiş küme bölme tablosuna uygulanmasıdır. Bu sayede gereksiz dallandırmanın çok önemli bir kısmı önceden görülebildiğinden algoritmanın performansı çok yüksek olmaktadır. Bu durum bir sonraki alt bölümde incelenecektir.

Küme bölme tablosunda yapılacak üçüncü düzenleme için yukarda açıklandığı şekilde merdiven formunda düzenlenen a_{ij} matrisinde ortaya çıkan her bir blok içindeki sütunlar, alt küme oluşum maliyetleri (alt kümelerdeki eleman sayısı her alt küme için eşit değil ise birim eleman başına düşen maliyetleri) dikkate alınarak soldan sağa ve küçükten büyüğe doğru sıralanır. Bu şekilde düzenlenmiş bir küme bölme tablosunda tamsayımlama işlemleri, birinci blokun ilk sütunundan (tablonun birinci sütunu) başlayarak diğer blokların sol sınırına mümkün olduğunca en yakın sütunlarından sağa doğru devam edilmesi halinde (yani, ilk önce ilgili bloklardaki sütunların mümkün olduğunca en küçük maliyetli olanı dikkate alındığında) optimum çözüme ilk işlemlerde ulaşma ihtimalinin oldukça yüksek olacağı söylenebilir.

Bu düşünceyi, tablo 3. deki küme bölme problemi üzerinde gözlemek mümkündür. Merdiven formundaki bu tabloda, bloklar içindeki sütunlar yukarıda belirtildiği şekilde düzenlenirse Tablo 13. ortaya çıkar. Bu şekilde düzenlenmiş tabloya bundan böyle, *"blokları maliyet sıralı küme bölme tablosu"* denilecektir.

Tablo 13. Blokları Maliyet Sıralı Küme Bölme Tablosu

i	j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
	C_j	49	59	61	63	64	70	72	72	83	84	36	44	44	62	70	71	34	45	51	38
1		1	1	1	1	1	1	1	1	1	1										
2		1				1			1		1	1	1	1	1	1	1				
3					1	1		1		1		1		1			1	1	1	1	
4			1				1	1	1				1	1		1		1	1		1
5				1			1			1	1				1	1	1		1	1	1
6		1	1	1	1							1	1		1			1		1	1

Tablo 13. \`deki $a_{ij}$ matrisinde sadece $a_{ij}$=1 olan değerler yeralmaktadır. Bu problem, birinci blokun ilk sütunu ve sırayla diğer sütunları ele alınarak ($X_j$=1 kabul edilerek) tamsayımlama çözümü yapıldığında uygun çözüm sonuçları Tablo 14. \`deki gibi ortaya çıkar.

Tablo 14. tablo 13. \`deki Problemin Çözüm Sonuçları

Uygun Çözüm Sıra No:	Uygun Çözüm Sonuçları	
1	$X_1 = X_{18} = 1$,	z = 94
2	$X_2 = X_{16} = 1$,	z = 130
3	$X_3 = X_{13} = 1$,	z = 105
4	$X_4 = X_{15} = 1$,	z = 133
5	$X_5 = X_{20} = 1$,	z = 102
6	$X_6 = X_{11} = 1$,	z = 106
7	$X_7 = X_{14} = 1$,	z = 134
8	$X_8 = X_{19} = 1$,	z = 123
9	$X_9 = X_{12} = 1$,	z = 127
10	$X_{10} = X_{17} = 1$,	z = 118

Tablo 14. `deki sayımlama sonuçlarından görüldüğü gibi, optimum çözüm, ilk uygun çözümlerden biri hatta bu örnekte birincisidir. Buna rağmen, sayımlama işlemleri devam ederken hangi çözümün optimum olduğunu kesin olarak bilmek yine de mümkün görünmemektedir. Kesin sonuç, ancak bütün sayımlama işlemleri bittikten sonra görülebilir.

Bütün bunlara rağmen, burada şöyle bir sonuç çıkarılabilir: İlk uygun çözümlerden en iyisinin optimum çözüme yakın bir değer olması ihtimali oldukça yüksektir ve uygun çözümler arttıkça bu ihtimal daha da yükselir.

Bu tesbitin, uygulamada iki ayrı yararı olacağı düşünülmektedir.

1) Tamsayımlama yapılırken araştırılan amaç fonksiyonunun en küçük değeri (amaç fonksiyonu değeri için üst sınır) daha ilk uygun çözümlerde küçük bir rakama ulaşacağından devam edilen işlemlerde bulunacak uygun çözümlerin büyük bir kısmında amaç fonksiyonu değeri önceden bulunan üst sınır değerinden yüksek olacaktır. Bu durumda, birçok uygun çözümün optimum çözüm olamayacağı daha erken bir aşamada belli olacağından daha az işlem(1) ve zaman sonucunda optimum sonuca ulaşılabilir.

2) Çok büyük boyutlu küme bölme problemlerinde çözüm zamanı çok uzun süreceğinden, "belli bir süre sonunda bulunacak en iyi uygun çözüm", yaklaşık optimum çözüm olarak kabul edilebilir.

---

(1)İmdat Kara, a.g.e., s.160

3. Algoritma İle İlgili Ön Bilgiler

Bu çalışmada kullanılacak algoritma ile ilgili olarak aşağıdaki tanımlamalar kullanılacaktır:

$A_{t,k}$ = t blokunda bulunan sütunlar (alt kümeler)
$k \in \{NI_t, NI_t+1, \ldots, NS_t\}$

NI_t = t blokunun ilk sütun numarası

NS_t = t blokunun son sütun numarası

N_t = t blokundaki sütun sayısı

A_j = j. inci sütun (alt birim) $j \in \{1, \ldots, n\}$

n = a_{ij} matrisinde sütun sayısı

m = a_{ij} matrisinde satır sayısı

B_t = t. inci blok $t \in \{1, \ldots, m\}$

ÜS = Minimum maliyet için üst sınır. Yani, optimum çözüm sonunda bulunacak minimum maliyet bu sınırın üstünde olamaz.

Bu sayımlama algoritmasında uygun çözümler şöyle araştırılır: İlk ana dalın oluşumu, birinci blokta bulunan alt kümelerden ilkine karşılık gelen X_1 değişkenine $X_1 = 1$ değeri verilerek oluşturulacak ilk kısmi çözüm(1) ile başlar.

---

(1) Kısmi çözüm: Bir optimizasyon probleminde değişkenlerden en az birinin çözüm değerinin bilinmesi (veya bu değerin 1 olduğunun varsayılması) durumuna karşılık olarak kullanılmaktadır.

$X_1 = 1$ olarak ele alınan kısmi çözüme $X_k = 1$ olarak ilave edilecek "en küçük indisli" ikinci değişken ve A_k alt kümesi (sütunu) şöyle seçilir: A_1 sütununda ilk sıfırın yer aldığı satır (A_1 alt kümesiyle örtülenmeyen ilk satır) numarası t olarak belirlenir. Yukarıda aranan A_k alt kümesi, ancak ve ancak B_t bloku içinde yer alan alt kümelerden biri olabilir. Çünkü, a_{ij} matrisinin merdiven formunda olmasından dolayı daha küçük indisli B_l blokundaki (l<t) bütün sütunlar, A_1 sütunuyla örtülenen satırlardan bir kısmını ikinci kez örtülemek zorundadır ki bu durumda uygun çözüm şartlarından biri yerine getirilmemiş olacaktır.

Bu sonuca göre, $X_1 = 1$ ile başlatılan daldan ayrılarak devam edecek olan ($Nİ_t$ - j) adet dal, daha başlamadan kurutulmuş olmaktadır. Yani, bu dallandırmaların sonunda uygun bir çözüme çıkılamayacağı önceden kesin olarak bilinebilmektedir. B_t bloku içindeki alt kümelerden, $A_1 \cap A_k = \emptyset$ (burada $Nİ_t \leq k \leq NS_t$'dir) olanları ve kaç tane olduğu belirlenir (hiç yoksa, $X_1 = 1$ ile başlatılan dalın ilerlemesine son verilir) ve $X_1 = 1$ ile başlatılan daldan bunların adedi kadar yeni dallandırmalar yapılır. Seçilen alt kümelerden en küçük maliyetli olanına karşılık gelen değişken $X_k = 1$ alınarak $X_1 = 1$ ile başlatılan kısmi çözüme ilave edilir. Bloklar içindeki alt kümeler oluşum maliyetleri küçükten büyüğe olacak şekilde sıralanmış olduğundan, en küçük maliyet katsayılı değişken kolayca belirlenebilir. Çünkü, bu değişken seçilenlerin içinde en küçük indisli olanıdır.

$X_1 = 1$ ve $X_k = 1$ olarak genişletilen kısmi çözüme üçüncü değişkenin belirlenerek dahil edilmesi işlemi de, $X_k = 1$ olarak alınan değişkenin belirlenmesi mantığına paralel olarak şöyle yapılır: Daha önce kısmi çözüme girmiş olan değişkenlere karşılık gelen alt kümelerin bileşim kümesinde ($A_1 \cup A_k$) ilk sıfırın yeraldığı satır numarası (t) olarak belirlenir. Yeni kısmi çözüme girebilecek değişkenler B_t blokundaki alt kümelerden $(A_1 \cup A_k) \cap A_p = \emptyset$ şartını taşıyan A_p alt kümelerine (burada $NI_t \leq p \leq NS_t$ 'dir) karşılık gelen değişkenlerdir. Belirlenen bu değişkenler, $X_1 = 1$ ve $X_k = 1$ olarak devam eden daldan yapılabilecek yeni dallandırmalara karşılık gelirler. Kısmi çözüme ilave edilecek her yeni değişken bu mantıkla belirlenerek devam ettirilen kısmi çözümün uygun bir çözüme ulaşıp ulaşmayacağı araştırılır.

Bulunan çözümlerin uygun çözüm olarak kabul edilmesi için 3.13 ve 3.14 eşitliklerini sağlaması gerekir. Yani, uygun çözümde bir değeri alan değişkenlere karşılık gelen alt kümelerin bileşimleri, $M = \{1,2,....,m\}$ kümesine eşit, kesişimleri ise boş küme olmalıdır.

$X_1 = 1$ alınarak başlanan ana daldan yapılabilecek bütün dallandırmalar tamamlandıktan sonra, $X_j = 1$ alınarak (burada $1<j\leq NS_1$ 'dır) yeni ana dallar başlatılır ve $X_1 = 1$ durumunda yukarıda yapılan işlemler burada da tekrarlanır.

Sayımlama işleminde ana dal sayısı NS_1 kadar olacaktır. Çünkü, merdiven formundan dolayı bir nolu satırı örtüleyen sütunlar sadece B_1 blokunda yer alan sütunlardır. Bu nedenle, $1\leq j\leq NS_1$ olan X_j değişkenlerinden sadece bir tanesinin değeri 1 'e eşit olmalı ve diğerleri sıfır olmalıdır. Aksi durumda, bir nolu satır ya hiç örtülenmez ya da birden çok örtülenir ki bu da uygun çözüm şartına uymayan bir durumdur.

Bu tanımlamalar ve açıklamalar doğrultusunda işleyen algoritmanın adımları aşağıda verilmektedir.

4. Algoritma

Küme bölme problemi olarak düzenlenen Türkiye ikinci ve üçüncü futbol liglerinde alt grupların belirlenmesi problemlerinin optimizasyonunda kullanılması uygun görülen tamsayımlama algoritmasının adımları, yukarıda yapılan açıklamalar da dikkate alındığında şöyle özetlenebilir:

Adım 0: Küme bölme tablosunda a_{ij} matrisi merdiven formuna getirilir ve bloklardaki sütunlar, alt küme maliyetleri küçükten büyüğe doğru sıralanacak şekilde düzenlenir.

Adım I: Üst sınır değeri $ÜS = +\infty$ ve $j=1$ olarak belirlenir.

Adım II: $X_j = 1$ $(j \in \{1,...,N_1\})$ olduğu durumdaki bütün uygun çözümler bulunur. Bunların içinde ÜS 'den daha küçük maliyetli uygun çözüm varsa belirlenir ve "şimdiye kadarki en iyi uygun çözüm" olarak saklanır ve ÜS bu uygun çözüm maliyetine eşitlenir.

Adım III: $j < N_1$ ise $j=j+1$ alınarak Adım II 'ye gidilir. $j = N_1$ ise Adım IV 'e gidilir.

Adım IV: "Şimdiye kadarki en iyi uygun çözüm" (ÜS 'ye karşılık gelen) optimum çözümdür. $ÜS = +\infty$ ise (hiçbir uygun çözüm bulunamamışsa) problem çözümsüzdür.

5. Veriler

Küme bölme problemlerinin çözümü için gerekli olan veriler aşağıdaki gibi sıralanabilir:

- Evrensel küme içinde yeralacak elemanların ve sayısının belirlenmesi,

- Problemin amacı,

- Evrensel küme içinde yeralan elemanlar arasındaki ilişkilerde amacı etkileyen maliyet katsayılarından oluşan maliyet matrisinin belirlenmesi,

- Alt küme sayısının belirlenmesi,

- Alt kümelerdeki eleman sayılarının belirlenmesi,

- Diğer kısıtlar.

Bu çalışmada ele alınan, 1992-1993 sezonunda ikinci ve üçüncü futbol liglerinde alt grupların belirlenmesi sorunu, küme bölme problemi olarak düzenlendi ve optimize edildi. Bu sorunların küme bölme problemi olarak düzenlenmesinde kullanılan verilerden çoğu Türkiye Futbol Federasyonu 'ndan elde edildi. Bu sezonda ikinci ligde 53 takım, üçüncü ligde 160 takım yeralmaktadır. Futbol federasyonunun kararına göre; ikinci ligdeki takımların 10 veya 11 'er takımdan oluşan 5 alt kümeye, üçüncü ligdeki takımların ise 16 'şar takımdan oluşan 10 alt kümeye ayrılması gerekmektedir(1).

---

(1) Türkiye Futbol Federasyonu, "1992-1993 Türkiye Profesyonel Birinci-İkinci Ligleri Kupa Müsabakaları Statüsü ve Üçüncü Lig Fikstürü", Ankara, 1992, s.26-37

Alt grupların oluşturulmasında hangi amacın dikkate alınacağı, federasyonun yayınladığı fikstürde belirtilmemektedir. Federasyon yetkilileri buradaki amacın, alt gruplar içinde yeralan takımların birbirlerine olabildiğince yakın mesafelerde olması yani, alt gruptaki takımların deplasman maçları için birbirlerinin bulunduğu yerleşim merkezlerine gidip-gelmelerinden dolayı ortaya çıkan ulaşım maliyetlerinin en aza indirilmesi olduğunu ifade etmişlerdir.

Takımların birbirlerine ulaşım maliyetlerini veren maliyet matrisi, futbol federasyonundan temin edilemedi. Bu nedenle, Turizm Bakanlığı ve Karayolları Teşkilatı 'nın 1993 yılı yol durumuna göre hazırladıkları "Türkiye karayolları haritası"(1) kullanıldı. Bu sezonda ikinci ve üçüncü futbol liginde yeralan takımların bulundukları yerleşim merkezleri arasındaki asfalt karayolu uzunlukları belirlenerek elde edilen matris, maliyet matrisi olarak dikkate alındı.

Ayrıca, bu sezonda "herhangi iki takımın mutlaka aynı alt grupta yeralması (veya yeralmaması) gerektiği" gibi herhangi bir özel şartın bulunmadığı da futbol federasyonu tarafından ifade edilmiştir.

E. Bilgisayar Programı

Küme bölme problemlerinin genellikle büyük boyutlu problemler olduğu daha önceki bölümlerde belirtilmişti. Bu çalışmada yeralan problemlerde a_{ij} katsayı matrisinin

---

(1) Turizm Bakanlığı - Karayolları Teşkilatı(TCK),"Türkiye Karayolları Haritası", Ankara, 1993

boyutları birinci problem için 53x168 , ikinci problem için ise 160x347 olduğundan yukarıda belirlenen algoritmayla (veya başka bir algoritmayla) bu problemlerin elle çözülmesi uygun görünmemektedir. Bu nedenle, bilgisayar kullanılması gerekmektedir. Literatürde karşılaşılan küme bölme problemleriyle ilgili bütün uygulama çalışmalarında da bilgisayar kullanıldığı gözlenmiştir.

Yukarıda ortaya konan algoritmaya uygun olacak veya buna uyarlanabilecek bir bilgisayar paket programı bulunamadığından bu algoritmanın bilgisayar programı yapılmıştır. Basıc diliyle hazırlanan bu program EK-5 ve EK-6 `de verilmiştir. EK-5 ve EK-6 `deki programların birbirinden farkı; birincisinin ikinci ligle ilgili, ikincisinin ise üçüncü ligle ilgili verilere göre düzenlenmiş olmasıdır.

Bir küme bölme probleminin (veya başka problemlerin) kısa sürede çözüme ulaştırılabilmesi için kullanılan algoritmanın hızlı bir şekilde çözüme ulaştıracak işlerlikte olmasının yanında, algoritmanın bilgisayar programının da bu amaca uygun şekilde hazırlanması gerektiğinden, bu bilgisayar programında gereksiz döngülerden ve işlemlerden olabildiğince kaçınılmaya çalışıldı.

EK-5 ve EK-6 `de verilen bilgisayar programları küçük düzenlemeler sonucunda, gelecek sezonlardaki futbol ligleri ile ilgili alt grupların oluşturulması sorunlarının çözülmesinde ve başka küme bölme problemlerinin çözümünde kullanılmaları sağlanabilir.

IV. MODELİN TÜRKİYE FUTBOL LİGLERİNE UYGULANMASI

Ülkemizde deplasmanlı maçlar yapan (yani takımların karşılıklı olarak birbirinin bulunduğu şehirde maçlar yapması) futbol takımlarının yer aldığı ligler; birinci futbol ligi, ikinci futbol ligi, üçüncü futbol ligi ve deplasmanlı amatör ligleridir. Bunlardan sadece birinci futbol liginde bulunan takımlar tek grup içinde yer alırlar ve bu ligteki bütün takımlar birbirleriyle maçlar yaparlar. Diğer üç ligde ise alt gruplar söz konusudur ve ligteki bütün takımlarla değil sadece bulunduğu alt grup içindeki takımlarla maçlar yaparlar.

Bu çalışmada ele alınan sorun, alt grupların söz konusu olduğu futbol liglerinde alt grupların belirlenmesinde optimum bir fikstür programının ortaya konulması olduğundan araştırmanın kapsamına birinci lig girmemektedir. Bu nedenle sadece ikinci ve üçüncü futbol ligleri ele alınmıştır. Esasen, deplasmanlı amatör liginde de aynı çalışmanın yapılması mümkündür. İkinci ve üçüncü ligin seçilmesindeki esas neden, alt gruplardaki eleman sayısıyla ilgili olarak iki farklı durumun bu liglerde söz konusu olmasıdır. İkinci futbol liginde alt gruplardaki takım sayısı eşit olmazken (10 - 11 arasında), üçüncü ligte alt gruplardaki takım sayısı eşit (hepsi 16 takımlı) olarak yer almaktadır.

1992-1993 Futbol sezonunda ikinci ligte 53 takım yer almıştır. Bu takımlar ve bulundukları yerleşim merkezleri EK-1 `de verilmiştir. Takımlardan her biri 5 ayrı gruptan birine yerleştirilmekte ve sezon boyunca her takım kendi alt grubundaki diğer takımların bulunduğu yerleşim merkezlerine giderek maçlar yapmaktadırlar. Sezon sonuna kadar her bir takım, alt grubunda bulunan diğer bütün takımlarla bu takımların bulunduğu yerleşim merkezlerine giderek maç yapmak zorundadır. Sonuçta her bir takım diğer bir takımla bir deplasmanda ve bir de kendi sahasında olmak üzeri iki maç yapmış olacaktır.

İkinci ligde bulunan 53 takım 5 alt gruba bölüneceğinden, 3 alt grupta 11 `er takım, 2 alt grupta da 10 `ar takım bulunması gerekmektedir. Buna göre, 11 takımın bulunduğu alt grupta 11x10=110, 10 takımın bulunduğu alt grupta ise 10x9=90 defa deplasmana gidiş söz konusu olacaktır.

Buradaki alt grupların (C_j) maliyetlerini, yukarıda belirtilen deplasmana gidiş maliyetleri toplamı oluşturmaktadır. Örneğin, on takımdan oluşan bir alt grup için (C_j) maliyeti şöyle hesaplanabilir. Her bir takımın, diğer dokuz takımın bulunduğu şehir merkezine gidiş ve geri dönüş maliyetleri bulunarak toplanır. Bu rakam, ilgili takımın j. inci alt grup içinde yeralması halinde üzerine (payına) düşen ulaşım maliyeti olacaktır. j' inci alt kümede bulunan on takımın her biri için yukarıda belirtilen ulaşım maliyetleri hesaplanarak on takımın toplam ulaşım maliyeti bulunursa, elde edilen rakam (C_j) maliyetini verecektir.

Burada ele alınan sorun, ikinci ligdeki bütün takımların deplasmana gidiş-geliş maliyetleri toplamının ve bundan dolayı oyuncularda ortaya çıkacak seyahatten kaynaklanan yorgunluğun en az olacağı alt grupların belirlenmesidir. Yani, hangi takımlar 5 alt gruptan hangisinde yer almalıdır ki, kendi alt grubu içindeki diğer bütün takımlara ulaşım maliyetleri toplamı ligteki bütün takımlar için toplandığında minimum olsun.

Aynı durum üçüncü lig takımları için de söz konusudur. Üçüncü ligte 160 takım bulanmaktadır. Bunların 16 `şar takımlı 10 alt grupta yerleştirilmesi gerekmektedir. Bu takımlar ve bulunduğu merkezler EK-2 `de verilmiştir.

Ulaşım maliyeti ve seyahatten kaynaklanan yorgunluğun, kara yolu uzaklıklarıyla doğru orantılı olduğu kabul edilmektedir. Bu nedenle, maliyet matrisi olarak, takımların birbirlerine olan karayolu uzaklıklarını kilometre cinsinden veren uzaklık matrisi dikkate alınacaktır. Bir yerden bir yere gidiş veya geliş arasında uzaklık farkı olmayacağından bu matris simetrik olarak ortaya çıkmaktadır.

Takımlar arasındaki karayolu uzaklıkları, Karayolları Teşkilatı 'nın 1993 yılında hazırladığı karayolları haritasından ve il merkezleri arasındaki uzaklıklar matrisinden yararlanılarak hesaplanmıştır. Karayolu olarak sadece asfalt yollar dikkate alınmıştır. İkinci lig ile ilgili uzaklık matrisi EK-7 `de ve üçüncü lig ile ilgili uzaklık matrisi de EK-8 `de verilmiştir.

Bu matrislerde bazı değerlerin boş bırakılmasının nedeni, optimum çözümde birbirleriyle aynı alt grupta yer alması mümkün olmadığı önceden görülebilen takımlar arasındaki uzaklıkların M gibi büyük bir sayı olarak dikkate alınabileceği düşüncesidir. Bu düşünce ve sonuçları, alt küme oluşturma işlemleri için de sözkonusu olacaktır. Birbirleriyle aynı alt gruba düşmeyecek takımların nasıl belirlendikleri, "küme bölme tablosunun hazırlanması" başlığı altında bir sonraki alt bölümde geniş olarak ele alınmıştır.

A. Problemin Çözümünde Kullanılabilecek Modeller

Yukarıda tanımlanan problemlerin çözümü için en uygun yolun bu sorunun küme bölme problemi olarak hazırlanıp çözülmesi olduğu söylenebilir. Bunun için, önce olası alt kümeler ve bunların oluşum maliyetleri (bu alt kümede yaralan takımların deplasmana gidiş-geliş maliyetleri toplamı) hazırlanır ve bu verilerle kurulacak 0-1 tamsayılı doğrusal programlama probleminin uygun bir algoritmayla optimum çözümü bulunur. Bu husus devam eden alt bölümlerde geniş olarak açıklanacaktır.

İkinci ve üçüncü futbol liglerinde alt grupların en az maliyetle oluşturulması sorununun çözüme ulaştırılmasında hangi optimizasyon tekniklerinin kullanılabileceği araştırılmış, ancak, araştırma sonucunda bu sorunun Dinamik Programlama, Ulaştırma, Atama ve benzeri modellerden biriyle çözülebilmesi için herhangi bir yol bulunamamıştır.

Bu tür problemlerin çözümünde en uygun yolun küme bölme modeli olmasındaki temel mantık, olası alt kümeleri ve bunların oluşum maliyetlerini içeren bir tablonun (küme bölme tablosu) hazırlanması sonucunda 0-1 tamsılı doğrusal modelin kurulabilmesidir. Bu aşamadan sonra, Küme bölme sorunlarıyla ilgili problemlerin optimum çözümlerinin bulunması hem mümkün hem de kolay olmaktadır.

B. Küme Bölme Tablosunun Hazırlanması

Problemde yer alabilecek bütün alt kümelerin ve bunların oluşum maliyetlerinin bir arada yer aldığı tabloya küme bölme tablosu adı verildiği daha önce de belirtilmişti. Ele alınan problemin optimum çözümü bu tablodan yararlanılarak bulunacağı için tablonun eksiksiz ve hatasız hazırlanmasına azami dikkatin gösterilmesi gerekmektedir.

Eksiksiz ve hatasız bir küme bölme tablosu oluşturulmasının bir yolu evrensel kümeden elde edilmesi mümkün olan bütün alt kümeleri ve bunların oluşum maliyetlerini bilgisayar yardımıyla hesaplamaktır. Ancak, aşağıdaki hesaplama sonuçlarından da görüleceği gibi bu durumda tabloda yer alacak alt küme sayısı çok büyük olmakta ve böyle bir tablodan problemin çözümü çok uzun süre almaktadır.

İkinci ligde bulunan 53 takımdan oluşan evrensel kümenin, takım sayısı (eleman sayısı) 10 - 11 arasında olacak 5 alt kümeye bölünmesi gerektiğinden ve takımlar arasındaki uzaklığı veren matrisin simetrik olmasından dolayı (3.6) nolu formülün uygulanmasıyla bulunabilecek mümkün alt kümelerin sayısı:

$$|F| = \sum_{k \in \{10,11\}} {}_{53}C_k = {}_{53}C_{10} + {}_{53}C_{11}$$

$$|F| = 95722852660$$

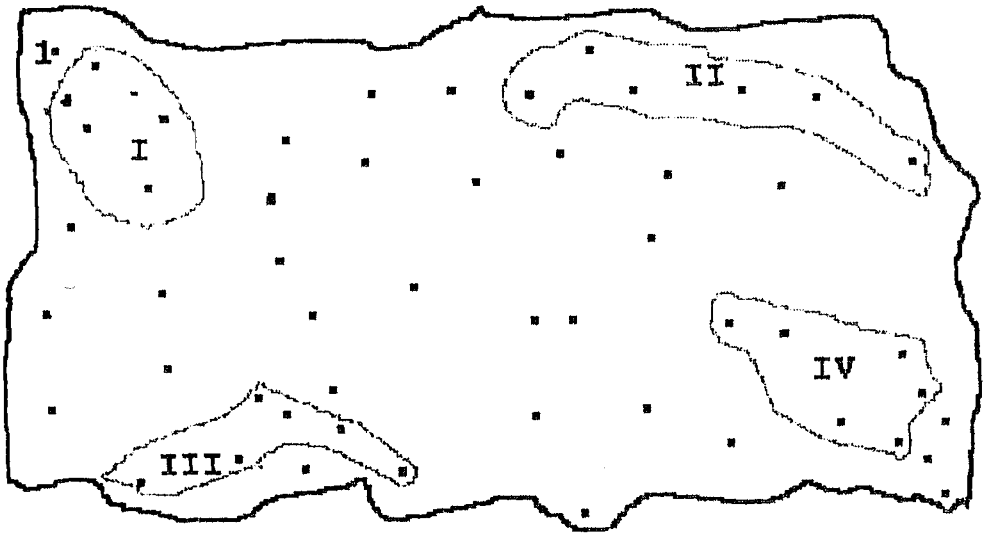
95 milyardan fazla olmaktadır.

Üçüncü ligde bulunan 160 takımdan oluşan evrensel kümenin 16 `şar takımdan oluşan alt kümelere bölünmesi probleminde ise mümkün alt küme sayısının yukarıdaki rakamdan daha da fazla olacağı ortadadır. Böyle bir problemin çözümü neredeyse imkansız görünmektedir. Bundan dolayı bu yolla küme bölme tablosunun oluşturulması büyük boyutlu problemler için uygun olmamaktadır.

Küme bölme tablosunun oluşturulmasında ikinci bir yol ise, ele alınan her problemin kendi özel yapısı dikkate alınarak optimum çözümde yer alması olasılığı bulunan alt kümelerin tabloya dahil edilmesi, diğerlerinin ise dahil edilmemesidir. Bu şekilde yapılacak bir tabloda alt küme sayısı oldukça azaltılabilmektedir. Ancak, bu yolla küme bölme tablosu oluşturulması halinde, uygulayıcının ele aldığı soruna ve modele yeterince hakim olması ve özen göstermesi gerekmektedir. Çünkü, optimum çözümde yer alacak bir alt kümenin küme bölme tablosunda yer almaması halinde bu alt kümenin optimum çözümde yer alması mümkün değildir.

Bu araştırmada ikinci yol uygulanmıştır. Burada, alt küme içindeki takımların birbirlerine olan uzaklıkları toplamının minimum olması amaçlandığına göre bu amaca uygun olmayacağı kesinlikle belli olan alt kümeler Şekil 4. 'deki gibi gösterilebilir.

I. nolu alt kümenin bulunduğu bölgede, 1 nolu takımın dışarda kalması kesinlikle uygun değildir. Çünkü, bu durumda 1 nolu takım kendisinden çok uzaklarda (daha yakın olanları var iken) olan takımlarla aynı alt kümede yer almak durumunda kalacak ve böylece daha uzak yerleşim merkezlerine gidip geleceği için böyle bir alt küme oluşumu minimizasyon amacına uygun olmayacaktır. III ve IV nolu alt kümeler de aynı nedenden dolayı hatalı oluşturulmuştur. Alt kümedeki elemanların ip gibi sıralanmasına göre, olabildiğince yuvarlak bir alan içinde



Şekil 4. Küme Bölme Tablosunda Yeralmayacağı Önceden Kesin Olarak Bilinen Alt Küme Örnekleri

yer alması, elemanların birbirlerine olan uzaklıkları toplamını daha az yapacağından II nolu alt küme de hatalıdır.

İkinci ve üçüncü lig ile ilgili problemlerde küme bölme tablosunda yer alacak alt kümelerin oluşturulması bu kurallar dikkate alınarak yapılmıştır. Yani, yukarıda belirtilen hatalı alt kümelere tabloda yer verilmemiştir. Ayrıca, ikinci lig probleminde alt kümelerdeki takım sayısının iki alt kümede 10, üç alt kümede ise 11 olması gerektiğinden, takımların (birim alanda) yoğun olduğu bölgelerde oluşturulan alt kümelerin sayısı hem 10 `arlı hem de 11`erli, yoğun olmadığı bölgelerde ise 10 `arlı olarak oluşturulması gerektiği düşüncesinden de yararlanılmıştır.

İkinci lige ilişkin problemde bu kurallar dikkate alınarak oluşturulan küme bölme tablosunda toplam alt küme sayısı: 168, üçüncü ligde ise 347 olarak ortaya çıkmıştır.

İkinci lig ile ilgili küme bölme tablosu EK-3 `de, üçüncü lig ile ilgili küme bölme tablosu ise EK-4 `de yer almaktadır. Bu tablolarda, $a_{ij}=0$ olan değerler yeralmamıştır. Sadece $a_{ij}=1$ değerlerinin yerine i rakamları yazılarak tabloların daha az yer tutması amaçlanmıştır.

C. Problemlerin Optimizasyonu - Bulgular ve Sonuçların Tartışılması

Yukarıda tanımlanan algoritma, EK-5 ve EK-6 `da verilen bilgisayar programları kullanılarak, 1992-1993 sezonu Türkiye ikinci ve üçüncü futbol ligi ile ilgili olarak hazırlanan EK-3 ve EK-4 `deki küme bölme tabloları verilen problemlerin optimizasyonu için uygulandı. Problemlerin çözümünde, hızı 33 Mhz olan "Intel" marka 80486 Dx-33 model bilgisayardan yararlanıldı.

1992-1993 Sezonundaki Türkiye ikinci futbol ligi ile ilgili küme bölme probleminin optimum çözümü;

$$X_4 = X_{54} = X_{86} = X_{105} = X_{155} = 1$$

$$X_j = 0 \quad (X_j=1 \text{ olmayan } j' \text{ ler için})$$

$$Z = 300072 \text{ Km.}$$

olarak elde edildi. Bu problemin optimum çözümünde yeralan alt kümeler ise aşağıdaki şekilde olmaktadır:

$$S_4 = \{1,2,3,4,5,6,7,8,9,10,11\}$$

$$S_{54} = \{12,13,14,15,16,17,18,19,20,21,22\}$$

$$S_{86} = \{23,24,31,32,33,38,39,40,41,42,43\}$$

S_{105} = {25,26,27,28,29,30,50,51,52,53}

S_{155} = {34,35,36,37,44,45,46,47,48,49}

Bu sonuçlar; optimum çözümde yeralan alt kümelerin kesişimleri boş kümeye, bileşimleri ise M evrensel kümesine eşit olması gerektiği şartını sağlamaktadır:

$$S_4 \cap S_{54} \cap S_{86} \cap S_{105} \cap S_{155} = \emptyset$$

$$S_4 \cup S_{54} \cup S_{86} \cup S_{105} \cup S_{155} = M$$

$M = \{1,2,3,.....,53\}$

Bu çözüm sonucuna göre, takımların yeralacağı alt gruplar tablo 14. 'de verilmiştir.Alt grupların harita üzerindeki görünümleri şekil 5. 'deki gibi ortaya çıkmaktadır.

1992-1993 Sezonundaki üçüncü futbol ligiyle ilgili küme bölme probleminin optimum çözümü ise;

$$X_4=X_{16}=X_{52}=X_{109}=X_{162}=X_{191}=X_{221}=X_{265}=X_{282}=X_{330} = 1$$

$X_j = 0$ (X_j=1 olmayan j 'ler için)

Z = 991364 Km.

olarak bulundu. Bu çözüme göre oluşan alt kümeler aşağıdaki şekilde ortaya çıkmıştır:

S_4 ={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16}

S_{16}={17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32}

S_{52}={33,34,35,36,37,151,152,153,154,155,156,157,158,159,160}

Tablo 14. 1992-1993 Futbol Sezonunda İkinci Ligdeki Takımların Optimum Plana Göre Yerleştirilecekleri Alt Gruplar

1. Grup

1 Eyüpspor	2 Gaziosmanpaşaspor	3 İstanbulspor
4 Kartalspor	5 Küçükçekmecespor	6 Üsküdar Anadolu
7 Zeytinburnuspor	8 Yalovaspor	9 Bandırmaspor
10 İnegölspor	11 Sakaryaspor	

2. Grup

12 Ayvalıkgücü	13 Balıkesirspor	14 Bucaspor
15 Göztepe	16 İzmirspor	17 Y.Salihlispor
18 Sökespor	19 Denizlispor	20 Manisaspor
21 Muğlaspor	22 Y.Nazillispor	

3. Grup

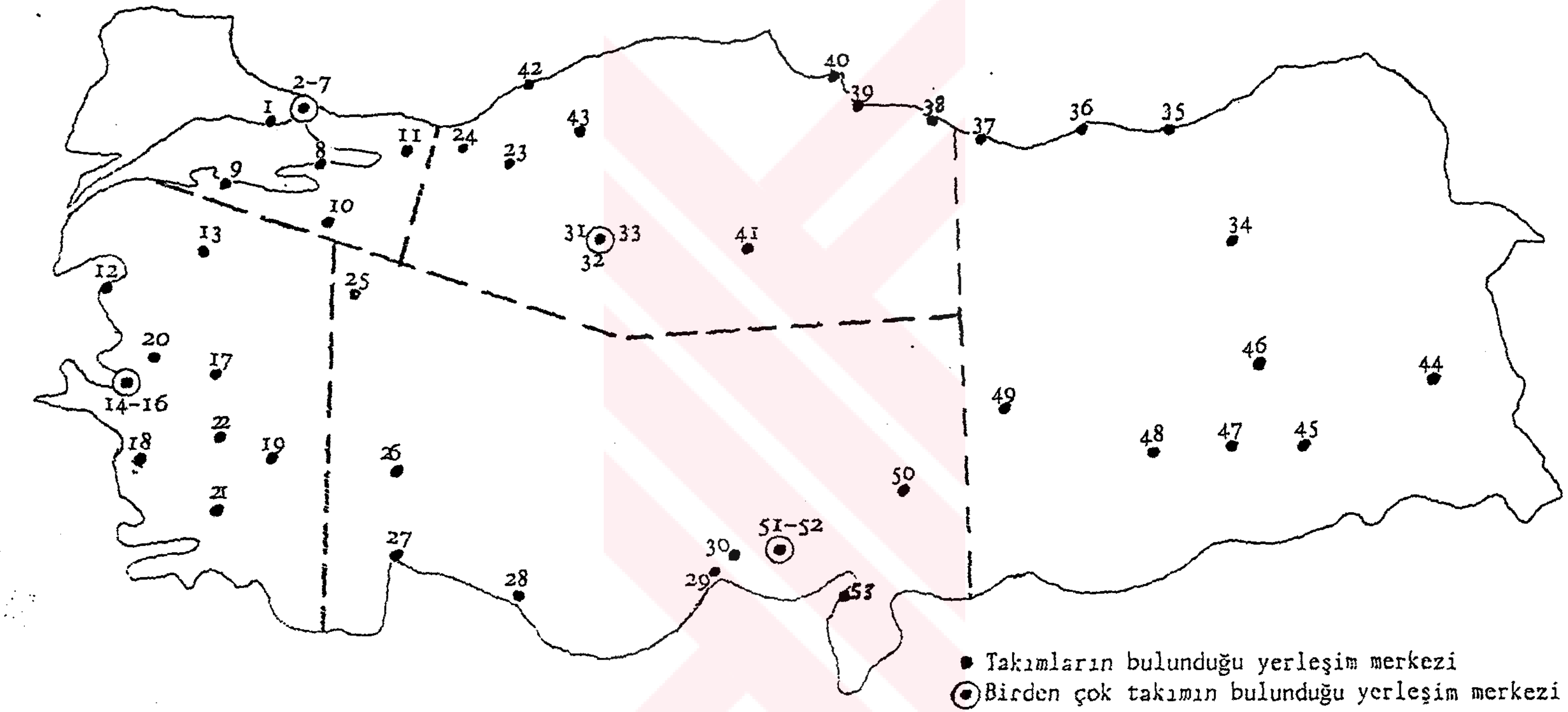
23 Boluspor	24 Düzce Doğsanspor	31 Keçiörengücü
32 P.T.T	33 Petrolofisspor	38 Ünyespor
39 Samsunspor	40 Bafraspor	41 Y.Yozgatspor
42 Zonguldakspor	43 D.Ç.Karabükspor	

4. Grup

25 Kütahyaspor	26 Ispartaspor	27 Antalyaspor
28 Alanyaspor	29 Mersin İ.Y.	30 Tarsus İ.Y.
50 K.Maraşspor	51 A.Demirspor	52 Adanaspor
53 İskenderunspor		

5. Grup

34 Erzurumspor	35 Çaykur Rizespor	36 Akçaabat Sebatspor
37 Orduspor	44 Vanspor	45 Siirt K.H.YSE.spor
46 Muşspor	47 Batman Bld.spor	48 Diyarbakırspor
49 Malatyaspor		



Şekil 5. 1992-1993 Futbol Sezonunda Optimum Plana Göre İkinci Ligdeki Alt Grupların Harita Üzerinde Görünümü

S_{109}={38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,65}

S_{162}={53,54,55,56,57,58,59,60,61,62,63,64,66,67,68,69}

S_{191}={70,71,72,73,74,75,76,77,78,79,80,81,8285,86,87}

S_{221}={83,84,88,89,90,91,92,93,94,95,96,97,98,99,100,101}

S_{265}={102,103,104,105,106,107,108,109,111,112,113,114,115,116,117,118}

S_{282}={110,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133}

S_{330}={134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149}

Optimum çözümde yeralan alt kümelerin kesişimleri boş kümeye, bileşimleri ise M evrensel kümesine eşit olması gerektiği şartını bu sonuçlar da sağlamaktadır:

$$S_4 \cap S_{16} \cap S_{52} \cap S_{109} \cap S_{162} \cap S_{191} \cap S_{221} \cap S_{265} \cap S_{282} \cap S_{330} = \emptyset$$

$$S_4 \cup S_{16} \cup S_{52} \cup S_{109} \cup S_{162} \cup S_{191} \cup S_{221} \cup S_{265} \cup S_{282} \cup S_{330} = M$$

$$M = \{1,2,\ldots\ldots\ldots,160\}$$

Üçüncü ligle ilgili problemin optimum çözüm sonuçlarına göre, takımların yeralacağı alt gruplar tablo 15. 'de verilmiştir. Bu sonuca göre, alt grubların harita üzerindeki görünümleri şekil 6. 'daki gibi olmaktadır.

1992-1993 Futbol sezonunda Türkiye Futbol Federasyonu tarafından belirlenip uygulanan alt grupların oluşum planı; ikinci lige ilişkin olanı şekil 7. 'de ve üçüncü lige ilişkin olanı ise şekil 8. 'de görülmektedir(1).

---

(1) Türkiye Futbol Federasyonu, "1992-1993 Türkiye Profesyonel Birinci-İkinci Ligleri Kupa Müsabakaları Statüsü ve Üçüncü Lig Fikstürü", 1992, Ankara, s.26-37

Tablo 15. 1992-1993 Futbol Sezonunda Üçüncü Ligdeki Takımların Optimum Plana Göre Yerleştirilecekleri Alt Gruplar

1. Grup

1 Edirnespor	2 Kırklarelispor	3 Babaeskispor
4 Lüleburgazspor	5 Uzunköprüspor	6 Keşanspor
7 Malkaraspor	8 Tekirdağspor	9 Çorluspor
10 Silivrispor	11 Bayrampaşaspor	12 Çengelköy
13 F.Karagümrük	14 Kapalıçarşıspor	15 Küçükköyspor
16 Nişantaşıspor		

2. Grup

17 Yücespor	18 Anadolu Hisarı I.Y.	19 Beylerbeyi
20 Büyükşehir Bld.spor	21 Dikilitaş	22 Feriköy
23 Galata	24 Kasımpaşa	25 Maltepespor
26 Pendikspor	27 Sümerbank Beykoz	28 Umraniyespor
29 Vefaspor	30 Darıca G.Birliği	31 Gebzespor
32 Gölcükspor		

3. Grup

33 Bilecikspor	34 Bozüyükspor	35 Esk.Demirspor
36 Eskişehirspor	37 Esk.Şekerspor	150 DMY.Ank.Demirspor
151 DSI spor	152 Şekerspor	153 Y.Sincanspor
154 Bartınspor	155 Kilimlispor	156 Erdemir Ereğlispor
157 Düzcespor	158 Akyazıspor	159 Beypazarı Bld.spor
160 Polatlıspor		

4. Grup

38 Sönmez Filanentspor	39 S.Bank Merinosspor	40 Mudanyaspor
41 Karacabeyspor	42 M.Kemalpaşaspor	43 Gönenspor
44 Bigaspor	45 Çanspor	46 Ç.Dardanelspor
47 Burhaniyespor	48 Bergamaspor	49 Soma Linyitspor
50 Soma Sotesspor	51 Tavşanlı Linyitspor	52 Akhisarspor
65 Uşakspor		

5. Grup

53 Sümerbankspor	54 Alpetspor	55 Altınordu
56 Tarişspor	57 Torbalıspor	58 Yeşilovaspor
59 Y.Turgutluspor	60 Selçuk Efesspor	61 Kuşadasıspor
62 Tirespor	63 Ödemişspor	64 Alaşehirspor
66 Buldanspor	67 Y.Milasspor	68 Marmarisspor
69 Fethiyespor		

Tablo 15.'in devamı

6. Grup

70 Y.Dinarspor	71 Y.Afyonspor	72 Burdurgücü
73 Kemer Bld.spor	74 Ant.Köy Hiz.spor	75 Ormanspor
76 Y.Akşehirspor	77 Etibank Sas	78 Konya Yolspor
79 Karamanspor	80 Kırşehirspor	81 Aksarayspor
82 Ereglispor	85 Niğdespor	86 Nevşehirspor
87 B.Ş.Bld.Erciyesspor		

7. Grup

83 Silifkespor	84 Içel Polisgücü	88 Adana Polisgücü
89 Adana G.Bir.spor	90 Kozan Bld.spor	91 Kadirli I.Y.B.Spor
92 Ceyhan Bld.spor	93 Osmaniyespor	94 Sahilspor
95 Hatayspor	96 Kırıkhan Bld.spor	97 Islahiyespor
98 Kilis Bld.spor	99 Elbistanspor	100 Nizip Bld.spor
101 Besnispor		

8. Grup

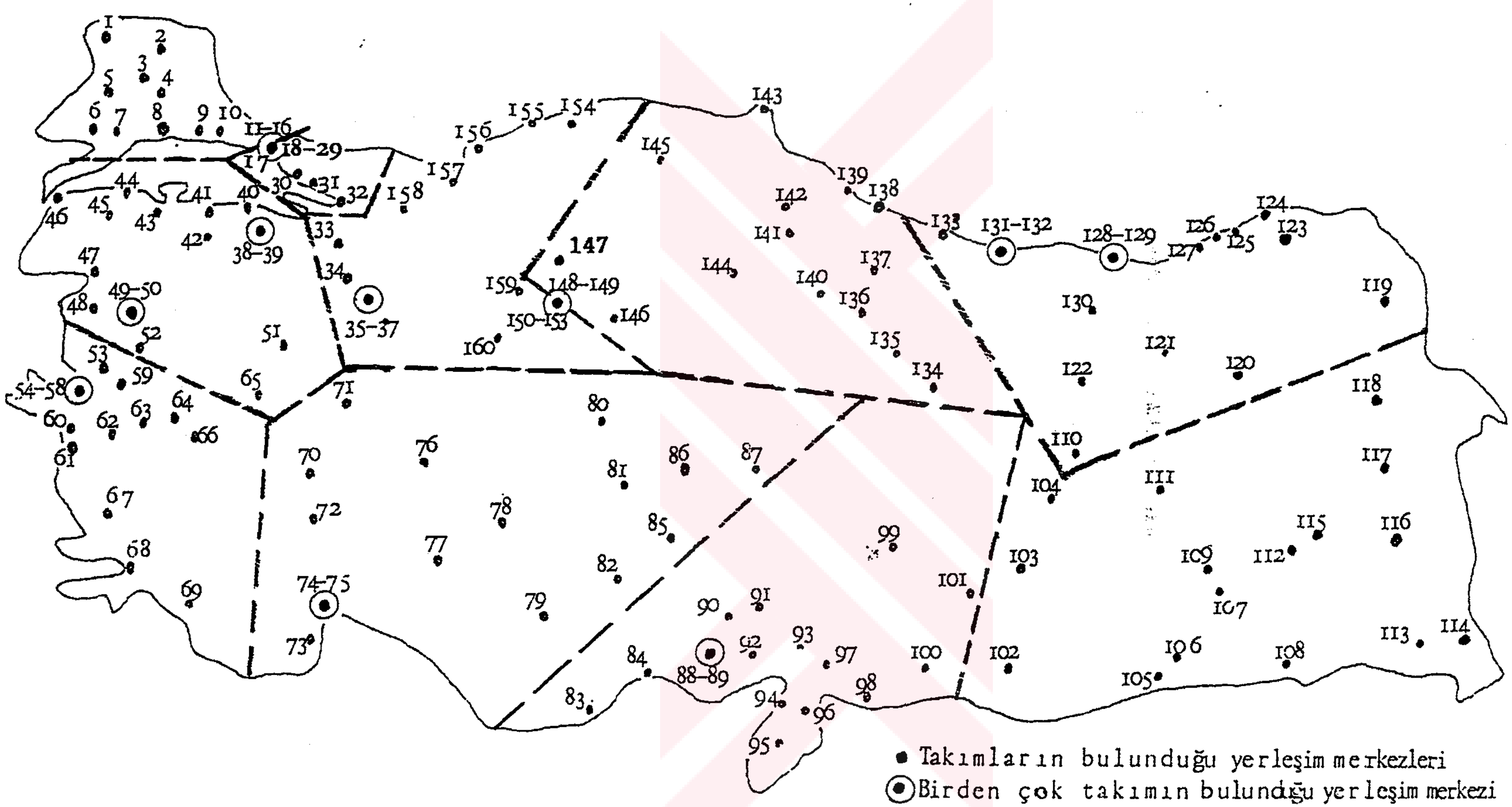
102 Ş.Urfaspor	103 Adıyamanspor	104 Elazığspor
105 Kızıltepespor	106 Mardinspor	107 Batmanspor
108 Cizrespor	109 Silvanspor	111 Bingölspor
112 Bitlisspor	113 Hakkari Ptt spor	114 Yüksekova Bld.Sp.
115 Tatvanspor	116 Van DSI spor	117 Ercişspor
118 Ağrıspor		

9. Grup

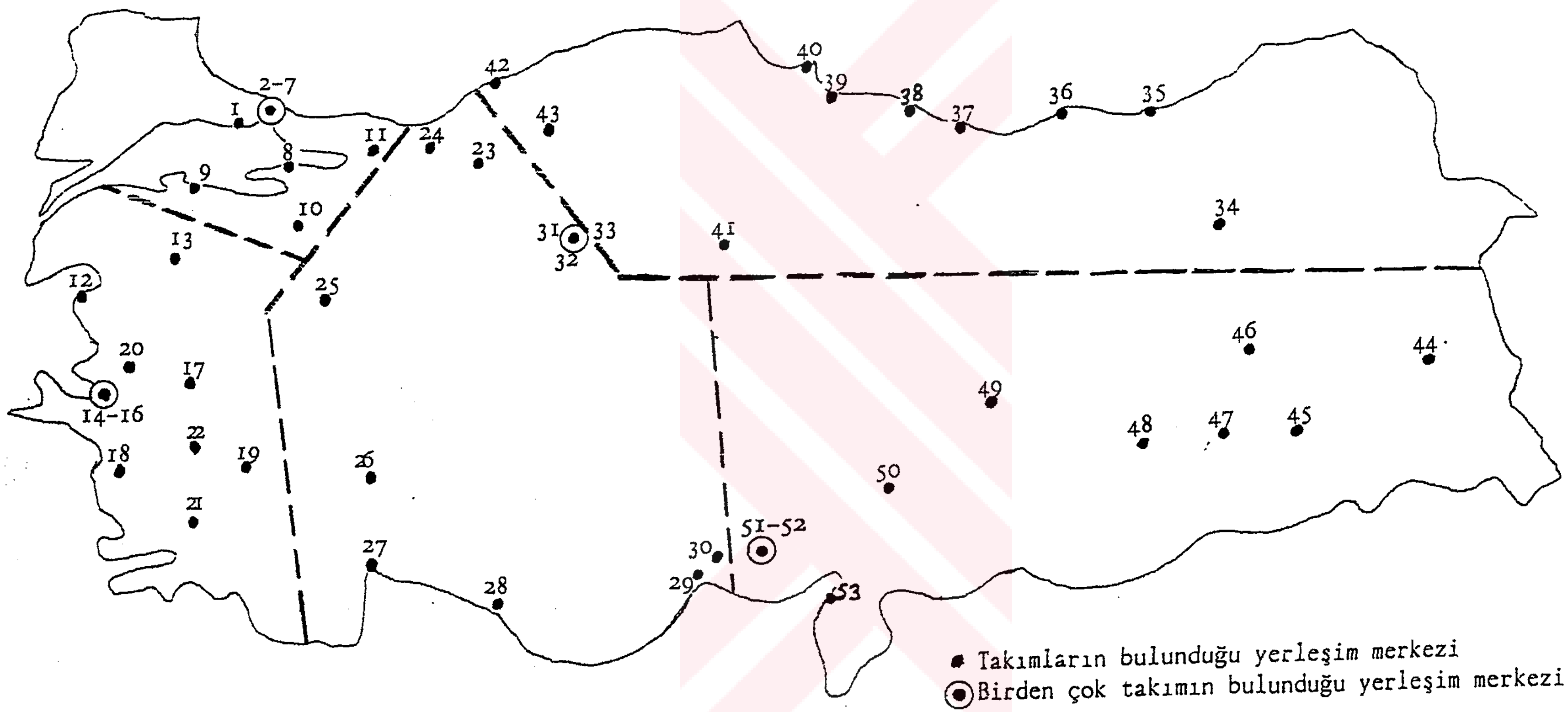
110 Tuncelispor	119 Kars Köy Hiz.spor	120 TEK 12 Martspor
121 Bayburtspor	122 Erzincanspor	123 Artvinspor
124 Hopaspor	125 Ardeşen Bld. Spor	126 Pazarspor
127 Çayelispor	128 Trabzon PTT spor	129 Yalıspor
130 Gümüşhane K.H.Spor	131 Giresunspor	132 Bulancakspor
133 Fatsaspor		

10. Grup

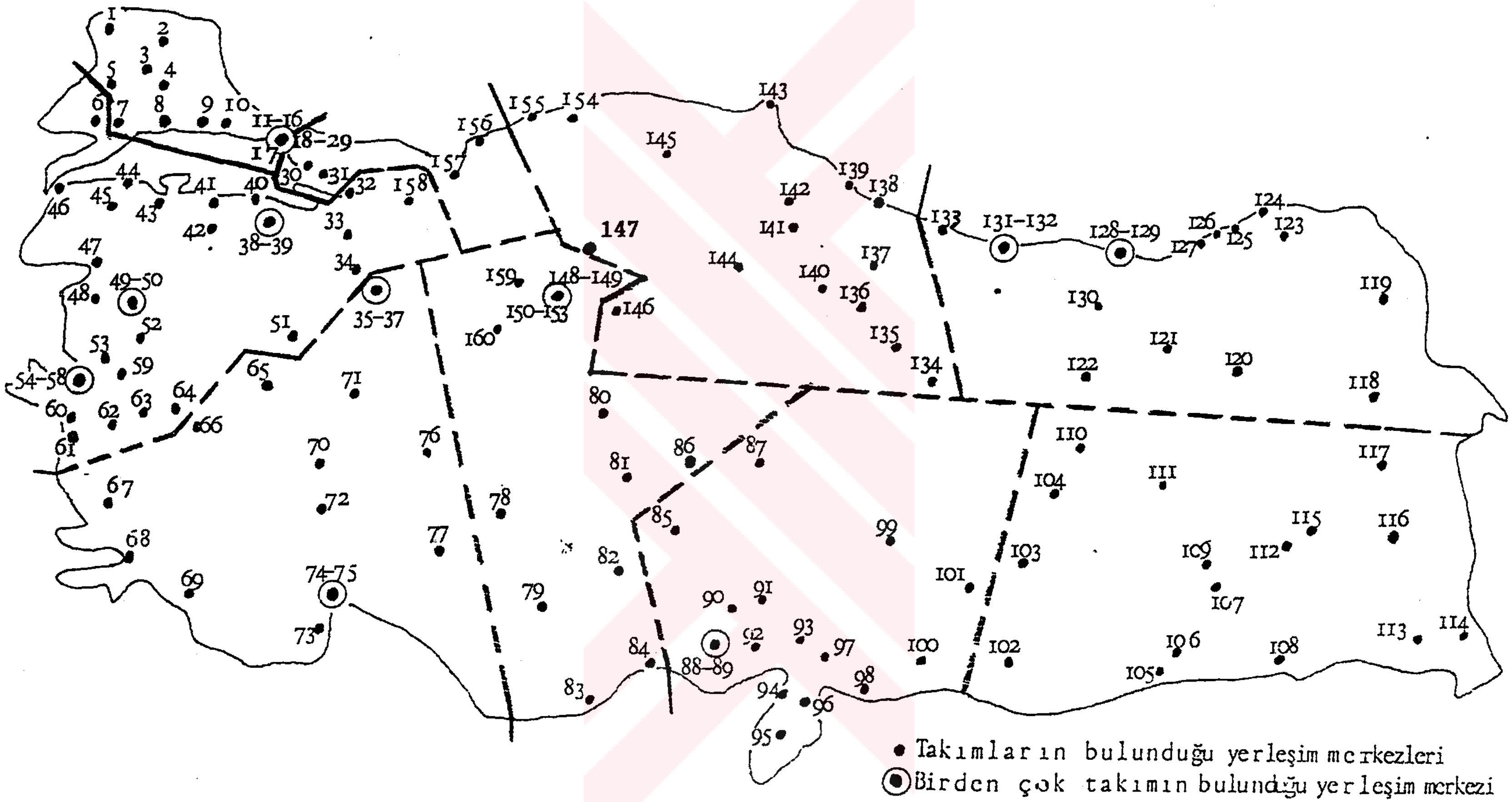
134 Sivasspor	135 Tokatspor	136 Turhalspor
137 Erbaaspor	138 Çarşambaspor	139 Kadıköyspor
140 Amasyaspor	141 Merzifonspor	142 Vezirköprüspor
143 Köy Hiz.Sinopspor	144 Çorumspor	145 Kastamonuspor
146 Kırıkkalespor	147 Çubukspor	148 Ankara Emniyetspor
149 Ankara Köy Hiz.spor		



Şekil 6. 1992-1993 Futbol Sezonunda Optimum Plana Göre
Üçüncü Ligdeki Alt Grupların Harita Üzerinde Görünümü



Şekil 7. 1992-1993 Futbol Sezonunda Uygulanan Plana Göre İkinci Ligdeki Alt Grupların Harita Üzerinde Görünümü



Şekil 8. 1992-1993 Futbol Sezonunda Uygulanan Plana Göre Üçüncü Ligdeki Alt Grupların Harita Üzerinde Görünümü

İkinci ligde uygulanan plan, Ek-3. 'de verilen küme bölme tablosuna göre aşağıdaki çözüme karşılık gelmektedir:

$X_4 = X_{54} = X_{103} = X_{151} = X_{168} = 1$

$X_j = 0$ (X_j=1 olmayan j 'ler için)

Z = 330752 Km.

Bu uygulanan plana göre oluşan alt kümeler ise:

$S_4 = \{1,2,3,4,5,6,7,8,9,10,11\}$

$S_{54} = \{12,13,14,15,16,17,18,19,20,21,22\}$

$S_{103} = \{23,24,25,26,27,28,29,30,31,32,33\}$

$S_{151} = \{33,34,35,36,37,38,39,40,41,42,43\}$

$S_{168} = \{ 44,45,46,47,48,49,50,51,52,53\}$

olmaktadır.

Üçüncü ligde uygulanan plan ise, Ek-4 'de verilen küme bölme tablosuna göre aşağıdaki çözüme karşılık gelmektedir:

$X_j = 1$ ($j \in \{3,13,23,88,128,207,231,264,297,331\}$)

$X_j = 0$ (X_j=1 olmayan j 'ler için)

Federasyon tarafından uygulanan bu plana göre oluşan alt kümeler ise:

S_{3} ={1,2,3,4,5,7,8,9,10,11,12,13,14,15,16,17}

S_{13} ={6,32,33,34,38,39,40,41,42,43,44,45,46,47,51,158}

S_{23} ={18,19,20,21,22,23,24,25,26,27,28,29,30,31,156,157}

S_{88} ={35,36,37,65,66,67,68,69,70,71,72,73,74,75,76,77}

S_{128}={48,49,50,52,53,54,55,56,57,58,59,60,61,62,63,64}

S_{207}={78,79,80,81,82,83,84,86,148,149,150,151,152,153,159,160}

S_{231}={85,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101}

S_{264}={102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117}

S_{297}={118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133}

S_{331}={134,135,136,137,138,139,140,141,142,143,144,145,146,147,154,155}

olmaktadır.

İkinci ve üçüncü ligle ilgili küme bölme problemlerinin optimizasyonuyla elde edilen optimum planlar Türkiye Futbol Federasyonu tarafından ortaya konup uygulanan planlarla karşılaştırıldığında; uygulanan planların optimum olmadığı ve bu planların maliyetinin optimum planların maliyetine göre önemli ölçüde yüksek olduğu görülmektedir.

Uygulanan planın optimum plandan maliyet(1) farkı:

İkinci lig için; 330752 - 300072 = 30680 Km.

Üçüncü lig için; 1141220 - 991364 = 149856 Km.

Toplam; 30680 + 149856 = 180536 Km.

olmaktadır. Yani, optimum plana nazaran Futbol

---

(1) Maliyet olarak, gidilecek yolun uzunluğu dikkate alınmıştır.

Federasyonunun uyguladığı mevcut plan, takımların ve taraftarlarının 180536 Km. daha fazla yolculuk yapmalarını gerektirmektedir. Bu sonuç, ikinci ligde %10.2 , üçüncü ligde %15.1 daha fazla maliyet ve daha fazla seyahat yorgunluğu anlamına gelmektedir.

1992-1993 Futbol sezonundaki ikinci ve üçüncü ligle ilgili küme bölme problemlerinin optimizasyon çalışmaları sonucunda elde edilen bulgular tablo 16. 'da özet olarak verilmiştir.

Tablo 16. İkinci ve Üçüncü Ligle İlgili Problemlerin Optimum Çözüm Sonuçları

Açıklama	II. Lig Problemi	III. Lig Problemi
1	53	160
2	168	347
3	22	193
4	19	178
5	5	175
6	18	50
7	300072	991364
8	330752	1141220
9	30680	149856
10	10.2	15.1
11	19.7	10
12	73171	1028868
13	77170	1091416

Tablo 16. 'daki satırlar:

1. Satır sayısı (Evrensel kümedeki eleman sayısı)
2. Sütun sayısı (Küme bölme tablosunda yeralan alt küme sayısı)
3. Bloklar içindeki alt kümelerin oluşum maliyetleri küçükten-büyüğe doğru sıralandığında küme bölme tablosunun optimizasyonu için geçen süre (run tıme)(saniye)
4. Bloklar içindeki alt kümelerin oluşum maliyetleri büyükten-küçüğe doğru sıralandığında küme bölme tablosunun optimizasyonu için geçen süre (run tıme)(saniye)
5. Bloklar içindeki alt kümelerde maliyetlerin küçükten-büyüğe doğru sıralanması durumunda, optimum çözümün ortaya çıktığı süre (saniye)
6. Bloklar içindeki alt kümelerde maliyetlerin büyükten-küçüğe doğru sıralanması durumunda, optimum çözümün ortaya çıktığı süre (saniye)
7. Optimum planın maliyeti (Km)
8. Uygulanan mevcut planın maliyeti (Km)
9. Uygulanan mevcut planın optimum plandan maliyet fazlalığı (Km)
10. Fazla maliyetin optimum plan maliyetine oranı (%)
11. Katsayı matrisinin yoğunluğu (%) yani, katsayı matrisindeki 1 'lerin (mxn) 'e oranı
12. Bloklar içindeki alt kümelerin oluşum maliyetleri küçükten-büyüğe doğru sıralandığında küme bölme tablosunun optimizasyonunda, her bir ana daldan bulunan ilk uygun çözümlerin en küçük maliyetli olanı
13. Bloklar içindeki alt kümelerin oluşum maliyetleri büyükten-küçüğe doğru sıralandığında küme bölme tablosunun optimizasyonunda, her bir ana daldan bulunan ilk uygun çözümlerin en küçük maliyetli olanı

ifade ederler.

Bu problemlerin optimum çözümleri Bölüm III.C 'de tanımlanan algoritmanın uygulanmasıyla uygun bir sürede sonuçlandı. Çözüm süresi; ikinci lig için 19 saniye, üçüncü lig için 178 saniye olarak belirlendi.

Uygulama problemleri iki farklı küme bölme tablosu dikkate alınarak optimize edildi. Birinci durumda; bloklar içindeki alt kümelerin oluşum maliyetleri küçükten-büyüğe, ikincisinde ise büyükten-küçüğe doğru sıralanacak şekilde düzenlendi. Buna göre yapılan optimizasyon işlemleri sonucunda; her bir ana daldan bulunan ilk uygun çözümlerde ortaya çıkan en küçük maliyetin birinci durumda daha küçük olduğu gözlendi. Devam edecek çözüm işlemlerinde bu maliyet üst sınır olarak dikkate alınacağından ve bu sınırdan daha büyük maliyetli uygun çözümler için daha az işlem gerekeceğinden, maliyetlerin küçükten-büyüğe doğru sıralandığı birinci durumda problemlerin çözüm süreleri daha kısa olmuştur. Bu sonuçlar, tablo 16. 'nın 3,4,10 ve 11. inci satırlarında yeralmaktadır.

Uygulama problemlerinin çözümünden elde edilen bu bulgular da dikkate alınarak; merdiven formunda düzenlenmiş küme bölme tablosunda bloklar içindeki alt kümeler, maliyetleri küçükten-büyüğe olacak şekilde sıralanırsa problemin çözüm süresinin daha kısa olacağı söylenebilir.

V. SONUÇ VE ÖNERİLER

A. Sonuç

Küme bölme (set partitioning)veya eşitlik kısıtlı küme örtüleme (equality-constrained set covering) problemi, bir M evrensel kümesinin istenilen kısıtlara uygun bir şekilde alt kümelere ayrılması sorunlarında optimum ayrışım planını araştıran yöneylem problemi olarak tanımlanabilir. 0-1 tamsayılı doğrusal programlama problemlerinin özel bir halidir. Bu özelliği; katsayı matrisinin sadece 0 veya 1 değerlerinden oluşması, kısıtların eşitlik şeklinde olması ve bütün eşitliklerin sağ tarafının 1 'e eşit olmasıdır. Yani, amaç fonksiyonundaki değişkenlerin katsayılarından başka diğer bütün değerlerin 0 veya 1 sayılarından oluştuğu ve bütün kısıtların eşitlik şeklinde olduğu 0-1 tamsayılı doğrusal programlama problemi olarak düşünülebilir.

Küme bölme problemlerinin çözümü iki aşamada gerçekleşmektedir. Birinci aşamada, küme bölme tablosu oluşturulmakta, ikinci aşamada ise bu tablodaki verilerden optimum çözüm araştırılmaktadır.

Küme bölme tablosunun oluşturulmasında, olası bütün alt kümelerin tabloya dahil edilmesi halinde problemin boyutları çok büyük olmakta (alt kümelerdeki eleman sayılarının eşit olmaması halinde daha da büyük olmakta) ve bu yüzden çözümü zorlaşmaktadır. Bu olumsuzluğu önlemek için, ele alınan sorunun kendine has yapısı dikkate alınarak optimum çözümde yeralmayacağı önceden belirlenen alt kümelerin tabloya dahil edilmemesi uygun bir yol olarak görülmektedir.

Alt kümelerdeki eleman sayılarının eşit olduğu problemlerde bu yol daha da uygun olabilir. Ancak, bu işlemlerde hata yapılması halinde yani, optimum çözümde yeralacak bir alt kümenin küme bölme tablosuna konulmaması durumunda bulunacak optimum çözüm gerçek optimum çözüm olmayacağından alt kümelerin oluşturumasında bu duruma gerekli dikkatin gösterilmesi gerekmektedir.

Küme bölme tablosunun hazırlanmasından sonra, genel 0-1 tamsayılı doğrusal programlama problemlerinin çözümü için geliştirilen tamsayımlama algoritmalarından veya kesme düzlemi metoduna göre geliştirilen algoritmalardan birinin kullanılmasıyla problemin optimum çözümünün bulunması mümkün olabilmektedir. Ancak, uygulamada genellikle büyük boyutlu olarak ortaya çıkan küme bölme problemlerinin çözümünde daha kısa sürede sonuca ulaştıran algoritmaların kullanılması daha uygun olmaktadır.

Küme bölme probleminin özel yapısını dikkate alan araştırmacılar, bu problemin çözümü için çeşitli algoritmalar ortaya koydular. Bu algoritmalar, ele alınan problemin özelliğine göre farklı performanslar göstermektedir. Bu algoritmalar içinde en çok kullanılanın Marsten 'in tamsayımlama algoritması olduğu literatürden anlaşılmaktadır.

Marsten 'in algoritmasının en önemli özelliği, merdiven formunda düzenlenmiş olan küme bölme tablosu üzerinde işlemler yapması ve birkaç değişkene 0 veya 1 değeri verilerek oluşturulan kısmi çözümden uygun çözümlerin bulunması için, diğer değişken değerlerinin araştırılmasında simpleks yönteminin kullanılmasıdır. Ancak, ele alınan sorunun yapısına uygun olarak Marsten 'in algoritmasında bazı değişiklikler yapıldığında, bu algoritmanın performansı daha da yükseltilebilmektedir.

Bu tez çalışmasında ele alınan uygulama problemlerinin çözümünde kullanılan algoritma böyle bir düzenlemeyle elde edilmiştir. Bu algoritmada, Marsten 'in algoritmasında yeralan simpleks çözümler kullanılmamaktadır. Çünkü, simpleks çözümlerin uzun sürede sonuçlandığı ve çok kere tamsayı sonuçlar vermediği gözlenmiştir.

Ayrıca bu algoritmada, merdiven formunda düzenlenen küme bölme tablosunda yeralan bloklar içindeki sütunlar (alt kümeler) C_j maliyetleri küçükten-büyüğe doğru sıralanacak şekilde düzenlendiğinden, tamsayımlama yapılırken her bir ana daldan bulunan ilk uygun çözümler daha küçük maliyetli olarak ortaya çıkmaktadır. Böylece daha ilk işlemlerde üst sınır oldukça küçük (optimum çözüme çok yakın) olacağından daha az işlemle optimum çözüm bulunabilmektedir. Bu algoritma Bölüm III.D 'de tanımlanmıştır. Özellikle alt kümelerdeki eleman sayılarının eşit olduğu küme bölme problemlerinin çözümü için bu algoritmanın kullanılmasının uygun olacağı söylenebilir. Çünkü, uygun çözümde 1 değeri alacak değişken sayısı sabit olup önceden bu sayı bilinmektedir. Bu durumda, sayımlama işlemleri devam ederken, bu sayı kadar değişkenin 1 değerini aldığı bir kısmi çözüm (başka kontrollere gerek kalmadan) uygun çözüm olarak dikkate alınabilecektir.

Bu araştırmada uygulama problemi olarak seçilen Türkiye ikinci ve üçüncü futbol liglerinde alt grupların belirlenmesi sorunlarının çözümünde kullanılabilecek en uygun modelin, küme bölme modeli olduğu bu konuda yapılan ön çalışmalar sonucunda belirlenmiştir. Bu problemlerin Dinamik Programlama, Ulaştırma, Atama ve benzeri modellerden biriyle çözülebilmesi için herhangi bir yol bulunamamıştır.

Küme bölme modelinin bu tür problemlerin çözümünde en uygun model olmasının esas nedeni, olası alt kümeleri ve bunların C_j maliyetlerini içeren bir tablonun hazırlanmasıyla 0-1 tamsayılı doğrusal modelin kolayca kurulabilmesidir. Bu nedenle, bu sorunların küme bölme problemi olarak kurulması ve çözülmesi işlemleri oldukca basit ve pratik olmakta ve optimum çözüme uygun bir sürede ulaşılabilmektedir.

1992-1993 Sezonunda Türkiye futbol liglerindeki deplasmanlı maçlar; birinci, ikinci, üçüncü ligler ve deplasmanlı amatör liglerinde yapılmaktadır. Bunlardan sadece birinci futbol ligindeki takımların tamamı aynı grup içinde yer aldığından bu ligde alt gruplar bulunmamaktadır. Diğer liglerde bulunan takımların ise belli sayıdaki alt gruplara yerleştirimesi gerekmektedir. Üçüncü ligde alt gruplardaki takım sayıları eşit olmasına rağmen ikinci ligdeki alt gruplarda takım sayıları eşit değildir. Bu yüzden, bu araştırmada iki farklı durumu içeren ikinci ve üçüncü futbol liglerinde alt grupların belirlenmesi sorunları uygulama problemi olarak ele alınmıştır.

Bu tür problemlerin çözümü için tanımlanan Bölüm III.D 'deki algoritmaya uygun bir paket program bulunamadığından bu algoritmanın bilgisayar programı yapıldı.

1992-1993 Futbol sezonunda ikinci ve üçüncü liglerle ilgili alt grupların belirlenmesi sorunu, küme bölme problemi olarak düzenlendi ve Bölüm III.D 'deki algoritmanın Ek-5 ve Ek-6 'daki bilgisayar programları kullanılarak optimum çözümleri araştırıldı. Problemlerin çözümünde, hızı 33 Mhz olan "Intel" marka 80486 Dx-33 model bilgisayardan yararlanıldı.

Bu problemlerin optimum çözümleri dikkate değer sonuçlar vermiştir. 1992-1993 sezonunda, Türkiye Futbol Federasyonu tarafından belirlenip uygulanan planın maliyetinin (burada maliyet olarak, gidilecek yolun uzunluğu dikkate alınmaktadır), optimum planın maliyetinden önemli ölçüde fazla olduğu gözlenmiştir. Uygulanan planın optimum plandan maliyet farkı:

İkinci lig için; 330752 - 300072 = 30680 Km.

Üçüncü lig için; 1141220 - 991364 = 149856 Km.

Toplam; 30680 + 149856 = 180536 Km.

olmaktadır.Bu fark, ikinci ve üçüncü lig için toplam 180536 Km.'dir. Yani, bu sezonda takımlar ve taraftarları 180536 Km. (Dünyanın ekvator çevresinin 4.5 katı) daha fazla yolculuk yapmışlardır. Bu durum optimum planın maliyetine göre yaklaşık %14 fazlalık demektir.

Katedilen yol uzunluğu (Km.) olarak belirlenen bu fazlalığın parasal karşılığı yaklaşık olarak (Temmuz 1993 fiyatlarına göre) 45,134,000,000,- Tl. olmaktadır(1).

---

(1) Burada, takımların ortalama 1000 kişilik taraftarlarıyla deplasman maçına gittikleri ve bir kişinin 1 Km. 'lik yol için 250 Tl. ödediği varsayılmıştır. 250 Tl/Km. 'nin belirlenmesinde, kısa ve uzun mesafelere yolcu taşıyan otobüs firmalarının uyguladığı fiyatların ortalaması dikkate alınmıştır.

Bu rakam sadece yol masraflarının karşılığıdır. Bunun dışında, yolun uzunluğuyla doğru orantılı olarak değişen diğer masrafların (konaklama, yeme, içme vs.) da bu rakama ilave edilmesi gerekir.

Örneğin, 100-200 Km. uzaklıktaki bir yere aynı gün içinde (konaklama yapmadan) gidip-gelinebilir fakat 300 Km. 'den daha uzak bir yere gidip-gelmek için konaklama gerekebilir. Ayrıca, 50 Km. 'lik bir mesafeye yolculuk esnasında yemek yemeden gidip-gelinebildiği halde, 100 Km. 'den daha uzak mesafe için yemek yeme ihtiyacı ortaya çıkabilir. Bu masrafları tam olarak hesaplamak kolay olmamakla birlikte, yukarıda hesaplanan 45,134,000,000,-Tl. 'lik yol masrafından daha fazla olacağı tahmin edilmektedir. Buna göre, 1992-1993 futbol sezonunda ikinci ve üçüncü ligdeki alt grupların Futbol Fedarasyonunun belirlediği plana göre yapılarak uygulanması halinde toplam maliyeti (optimum planın maliyetine göre) yaklaşık olarak 100 Milyar Tl. artıracağı söylenebilir.

Bu sonuca göre, Türkiye Futbol Fedorasyonu tarafından ortaya konulup uygulanan planın kulüplere, taraftarlarına ve ülke ekonomisine önemli ölçüde ek maliyet yüklediği görülmektedir. Ayrıca, futbolcuların gereksiz yere yaptıkları fazla yolculuktan dolayı ortaya çıkan yorgunluğun oyuncu performansını düşürdüğü de söylenebilir.

Bu problemlerin optimum çözümleri, Bölüm III.D 'deki algoritmanın kullanılmasıyla uygun bir sürede elde edildi. Çözüm süresi; ikinci lig için 19 saniye, üçüncü lig için 178 saniye olarak belirlendi.

Problemler iki farklı küme bölme tablosu dikkate alınarak optimize edildi. Birinci durumda; bloklar içindeki alt kümelerin oluşum maliyetleri küçükten-büyüğe, ikincisinde ise büyükten-küçüğe doğru sıralanacak şekilde düzenlendi.

Buna göre yapılan optimizasyon işlemleri sonucunda; her bir ana daldan bulunan ilk uygun çözümlerde ortaya çıkan en küçük maliyetin birinci durumda daha küçük olduğu gözlendi. Devam edecek çözüm işlemlerinde bu maliyet üst sınır olarak dikkate alınacağından ve bu sınırdan daha büyük maliyetli uygun çözümler için daha az işlem gerekeceğinden, maliyetlerin küçükten-büyüğe doğru sıralandığı birinci durumda problemlerin çözüm süreleri daha kısa olmuştur.

1992-1993 Futbol sezonunda Türkiye ikinci ve üçüncü liglerinde alt grupların belirlenmesi sorunlarıyla ilgili olarak düzenlenen küme bölme problemlerinin küme bölme tablosunda, yukarıda belirtilen iki farklı durum dikkate alınarak bulunan optimum çözüm sonuçları Tablo 16. 'da verilmiştir.

Uygulama problemlerinin çözümünden elde edilen bu bulgular da dikkate alınarak Merdiven formunda düzenlenmiş küme bölme tablosunda bloklar içindeki alt kümeler maliyetleri küçükten-büyüğe olacak şekilde sıralanırsa problemin çözüm süresinin daha kısa olacağı söylenebilir.

Yukarıda belirtilen problemlerin çözümünde, BASIC diliyle hazırlanan bilgisayar programları (EK-5 ve EK-6) kullanıldı. Bu programlar, küme (set) işlemlerine (bileşim, kesişim vs.) daha uygun bir programlama diliyle yapılırsa, problemlerin çözüm sürelerinin daha kısa olabileceği ve yukarıda belirtilen iki ayrı durumun çözüm sürelerindeki farkının daha da yüksek olabileceği düşünülmektedir.

B. Öneriler

Alt grupların optimum şekilde planlanmasının amaçlandığı diğer spor dalları ile ilgili sorunlara da küme bölme modelinin uygulanması mümkündür. Bu alanda yapılacak çalışmaların hem daha uygun algoritmaların geliştirilmesi hem de ele alınan sorunların optimum çözüme ulaştırılması yönünde yararlı sonuçlar verebileceği düşünülmektedir.

Küme bölme probleminin çözümü için daha iyi algoritmaların ortaya konulması mümkün olabilir. Özellikle, merdiven formunda düzenlenmiş olan küme bölme tablosunda satırlar ve/veya sütunlar C_j maliyetleriyle ilişkilendirilerek yeniden düzenlemelere gidilmesi (burada uygun çözüme çıkmayacak dalların bir kısmının veya tamamının önceden bilinebilmesi ve/veya ilk işlemlerde optimum çözümün veya yaklaşık optimum çözümün bulunabilmesi amaçlanmalı) yolunda yapılacak araştırmaların yararlı sonuçlar verebileceği söylenebilir.

Örneğin, küme bölme tablosunda;

-merdiven basamaklarının genişliği küçükten-büyüğe veya büyükten-küçüğe veya ilk basamak en az genişlikte ve devam eden basamaklar büyükten-küçüğe olacak şekilde, satırlar yeniden düzenlenirse,

-Boş bloklar en sona veya en başa veya tablo içinde uygun şekilde dağıtılmış olarak düzenlenirse,

-Birinci blok, en yoğun (içindeki 1 sayısı en fazla) olacak şekilde ve bu blokun üst tarafının yoğunluğu mümkün olduğunca yüksek olacak şekilde satırlar ve sütunlar yeniden düzenlenebilirse,

-Yukarıdaki düzenlemelerden birkaçı birlikte yapılabilirse,

-Bu düzenlemeler, ana dallarla başlanan her kısmi çözüm için bu kısmi çözümün özelliğine göre yeniden yapılabilirse,

Küme bölme problemlerinin çözüm işlemlerinin ve çözüm süresinin azaltılması ile ilgili yeni bulguların elde edilmesi mümkün olabilir.

Küme bölme probleminin uygulama alanlarının hızla arttığı literatürden anlaşılmaktadır. Gelecekte, özellikle biyoloji, kimya, inşaat mühendisliği, metoroloji alanlarında ve uzay çalışmalarında bu modelle ilgili uygulama çalışmalarının yapılmasının modelin teorisi ve uygulaması açısından yararlı olacağı düşünülmektedir.

Bu çalışmada ele alınan problemlerde alt kümelerin oluşturulması elle yapılarak, küme bölme tablosunda yeralacak alt küme sayısının olabildiğince az olmasına çalışıldı. Tabloda yeralacak alt kümelerin oluşturulması, bu amaçla hazırlanacak bir bilgisayar programı kullanılarak bilgisayar yardımıyla yapılabilir. Ancak, problemin özelliğini bozmadan ve bütün kısıtları sağlayan ve gerçek optimum çözüme ulaşmaya engel olmayan fakat alt kümelerin sayısını olabildiğince azaltan bir bilgisayar programının yapılması gerekmektedir. Bu çalışmada yapılan pogramlarla elde edilen alt küme sayıları, elle yapılana göre 5-10 kat daha fazla olarak ortaya çıktığından, bu yolla elde edilen küme bölme tablosu kullanılmadı.

Ancak, benzer diğer çalışmalarda alt küme oluşturulması işlemleri için bilgisayar kullanılması daha olumlu sonuçlar verebilir.

Türkiye Futbol Fedarasyonunun ve diğer sporlarla ilgili federasyonların ilgi alanları içinde yeralan spor dallarında, alt kümelerin oluşturulmasıyla ilgili sorunların çözümünde optimizasyon hesapları yaptırarak bulacakları en uygun planı uygulamaları, birçok yönden yararlı olacaktır. En iyi (optimum) planın uygulanması halinde öncelikle, ilgili takımların kulüplerine, taraftarlarına ve ülke ekonomisine çok önemli katkılar sağlanabilir. Ayrıca bu durumda, bulundukları alt kümenin yeri ile ilgili olarak kulüp yöneticilerinin yapacakları itirazların önüne geçilerek olası suistimaller de önlenebilir.

ÖZET

Bu çalışmanın esas amacı, küme bölme (set partitioning) problemlerinin yapısal özelliklerini ortaya koyarak bu problemlerin çözümü için kullanılabilecek algoritmaları araştırmak ve Türkiye ikinci ve üçüncü futbol liglerinde alt grupların belirlenmesi sorununu çözecek uygun bir algoritma önermektir.

Küme bölme problemi, 0-1 tamsayılı doğrusal modelin özel bir hali olarak ortaya çıkar:

$$\text{Min.}\quad Z = \sum_{j\in N} C_j X_j$$

$$\text{Kısıtlar:}\quad \sum_{j\in N} a_{ij} X_j = 1 \qquad i\in M\,,\quad M = \{1,2,\ldots,m\},$$

$$X_j \in \{0,1\},\quad a_{ij} \in \{0,1\},\quad C_j \geq 0\,,\quad j\in N\,,\quad N = \{1,2,\ldots,n\}$$

Bu problemlerin çözümünde; önce, olası alt kümelerden oluşan küme bölme tablosu hazırlanarak 0-1 tamsayılı doğrusal modeli kurulur. Problemin optimizasyonunda tamsayımlama ve/veya kesme düzlemi tekniklerine dayanan fakat bu problemlerin yapısına uygun olarak hazırlanan özel algoritmalar kullanılır.

Bu tez çalışmasında, Türkiye futbol liglerinde alt grupların belirlenmesi (ve benzeri) sorunların küme bölme problemleri olarak çözümünde kullanılabilecek uygun bir algoritma önerildi. Uygulama çalışmaları sonucunda, bu tür sorunların çözümünde en uygun modelin küme bölme modeli olduğu gözlendi.

1992-1993 Sezonunda Türkiye ikinci ve üçüncü futbol liglerinde alt grupların belirlenmesi sorunu, küme bölme problemi olarak düzenlenip çözüldü. Futbol Federasyonunun bu sezonda uyguladığı plana göre takımların ve taraftarlarının deplasman maçları için gidecekleri yolun uzunluğu 1471972 Km. iken, optimum plana göre bu rakam 1291436 Km. olarak ortaya çıktı.

Bu sonuçlara göre optimum planın maliyeti Futbol Federasyonu tarafından belirlenen planın maliyetine nazaran 180536 Km. (oran olarak %12.3) daha düşük olmaktadır. Km olarak belirlenen bu rakamın parasal karışılığının 1993 yılı fiyatlarıyla yaklaşık 100 milyar lira civarında olduğu tahmin edilmektedir.

ABSTRACT

The main purpose of this study is to research algorithms which can be used for the solutions of the set partitioning problems by bringing up the structural characteristics of these problems and is suggesting a convenient algorthm which will solve the problem of determining subgroups in the second and the third football leauges in Turkey.

The set partitioning problem appears as a special case of linear model which has integers of 0-1:

$$\text{Minimize} \quad \sum_{j\in N} C_j X_j$$

$$\text{Subject to} \quad \sum_{j\in N} a_{ij} X_j = 1 \qquad i\in M \qquad M = \{1,2,\ldots,m\},$$

$$X_j \in \{0,1\}, \quad a_{ij} \in \{0,1\}, \quad C_j \geq 0, \qquad j\in N \qquad N = \{1,2,\ldots,n\}$$

In the solutions of these problems; first, the linear model is set which has integers of 0-1 by prepearing the set partitioning table composed of possible subsets. The special algorithms based on implicit enumeration and/or cutting plane techniques but prepared in accordance with the structure of the problem are used in the optimization of the problem.

In this thesis, a convenient algorithm which will be able to used in the solution of the questions of determining subgroups in the football leagues in Turkey (and similar questions) was suggested. As a result of practicing studies, it was observed that the most convenient model was set partitioning model in the solutions of such questions.

The question of determining the subgroups in the second and third football leauges in Turkey in the seasons of 1992-1993, was organized and solved as a set partitioning problem. While the length of the way which has to be covered by supporters and teams for displacement matches is 1471972 kilometres according to the existing schedules programmed by Turkish Football Federation, this figure has appeared as 1291436 kilometres according to the optimum plan.

In the light of these results, the cost of optimum plan is 180536 kilometres that is, per cent 12.3 lower with regard to the cost of the schedules which was determined by Turkish Football Federation. The monetary equivalent of this figure determined as kilometer has been guessed about hundred billions liras with the prices of 1993.

YARARLANILAN KAYNAKLAR

KİTAPLAR

BAYRAKTAR Mustafa, "Soyut Cebir ve Sayılar Teorisi", Atatürk Üniversitesi Yayınları, Erzurum, 1988

ÇAKICI Metin, "Temel İstatistik", Dokuz Eylül Üniversitesi-Manisa İ.İ.B.F., Manisa, 1991

FILEUREN Hein, "A Computational Study Of The Set Partitioning Approcach For Vehicle Routing and Scheduling Problems", Twente Üniversitesi, 1988

ÖZTÜRK Ahmet, "Yöneylem Araştırması", Uludağ Üniversitesi Yayınları No:3-040-0113, Bursa, 1987

KARA İmdat, "Doğrusal Programlama", Bilim Teknik Yayınevi, Eskişehir, 1991

KARA İmdat, "Yöneylem Araştırması, Doğrusal Olmayan Modeller", Anadolu Üniversitesi Basımevi, Eskişehir, 1986

ÖZDEN Kenan, "Yöneylem Araştırması", Hava Harp Okulu Yayınları, 1989

ÖZGÜVEN Cemal, "Doğrusal Programlama", Erciyes Üniversitesi İ.İ.B.F. Yayınları No:1, Kayseri, 1986

SERPER Özer-GÜRSAKAL Necmi, "Araştırma Yöntemleri", Filiz Kitabevi, İstanbul, 1989

SEZGİN Atilla-ADA Erhan, "İşletmeciler İçin Yöneylem Araştırması", Gazi Üniversitesi İ.İ.B.F., İşletme Bölümü, Ankara, 1990

TAHA Hamdy A., "Operations Research An Introduction", Mac Millan Publishing Company, Inc., 1987

TURİZM BAKANLIĞI ve KARAYOLLARI TEŞKİLATI (TCK), "Türkiye Karayolları Haritası", Ankara, 1993

TÜRKİYE FUTBOL FEDERASYONU, "1992-1993 Türkiye Profesyonel Birinci-İkinci Ligleri Kupa Müsabakaları Statüsü ve Üçüncü Lig Fikstürü", Ankara, 1992

TÜTEK Hülya H.-GÜMÜŞOĞLU Şevkinaz, "Sayısal Yöntemler", Dokuz Eylül Üniversitesi, İ.İ.B.F., İzmir, 1991

WAGNER Harvey M., "Principles of Management Science",Prentice-Hall, Inc., 1975

MAKALELER

ANILY S.ve FEDERGRUEN A., "Structured Partitioning Problems", Operations Research, 39/1, 1991

BALAS Egun ve PADBERG Manfred W., "Set Partitioning: A Survey", Sıam Review, 18/4, 1976

CAMPELLO Ruy Eduardo, "Updating a Hyprid Algorithm for Set Partitioning Problems" Mat.Aplic.Comp., Vol.4,No:1, Rio de Jenerio

CAMPELLO Ruy Eduardo ve MACULEN Nelson F., "Lagrangean Relaxation for a Lower Bound to a Set Partitioning Problem With Side Constrains: Properties and Algorithms", Discrete Aplied Mathematics, 18, 1987

COTTLE Richard W. ve KELMANSON Milton L. ve KORTE Bernard, "On Deep Disjunctive Cutting Planes for Set Partitioning: A Computationally Oriented Research", Proceedings of the International Congress on Mathematical Programming; Rio de Janeiro , Brazil, 6-8 April, 1981

DARZENTAS J. ve BAIRAKTARIS P., "On The Set Partitioning Type Formulation For The Discrete Location Problem", Comput ve Ops. Res, Vol.13, No:6, 1986

DELORMINE J. ve HEURGON E., "Problemes de Partitionnement:Explloration Arborescente Ou Methodes de Troncatures?", Rev. Francaise Autom., Inf. Reach. Oper., 9, v=2, 1975

ETCHEBERRY Javier, "The Set-Covering Problem A New Implicit Enumaration Algorithm" Operation Research, Vol.25, No:5, 1977

FISHER Marshall L. ve KEDIA Pradeep, "Optimal Solition of Set Covering/Partitioning Problems Using Dual Heuristics", Manegement Science, Vol.36, No.6, 1990

GARFİNKEL R.S.-NEMHAUSER G.L., "The Set Partitioning Problem: Set Covering With Equality Constraints", Operations Research, 17, 1969

GARFİNKEL R.S.-NEMHAUSER G.L., "Optimal Political Districting by Implicit Enumeration Techniques", Management Science, Vol.16, No.8, 1970

IGBAL Agha Ali ve Hemalatha Thiagarajan, "A Network Relaxation Based Enumeration Algorithm For Set Partitioning", European Journal Of Operational Research, 38,1989

MARSTEN Roy E., "An Algorithm for Set Partitioning Problems", Manegement Science, 20, 1974

MICNAUD P., "Exact Implicit Enumeration Method for Solving the Set Partitioning Problem", IBM Journal of Research and Development, Vol.16, No:6, 1972

NAWIJN W.M., "Optimizing the performance of a Blond Analyser:Aplication of the Set Partitioning Problem", European Journal of Operational Research, 36, 1988

RIVETT Patrick, "The Stucture of League Football", Opl. Res. Q.,Vol.26, 4,1975

ROY Tony J. ve WOLSEY Laurence A., "Solving Mixed Integer Programming Problems Using Automatic Reformulation", Operations Research, Vol.35, No. 1, 1987

ROHDE Martin, "Set Partitioning mit Linearen Randbedingungen", OR Spektrum, 1, 1979

THUVE H., "Frequency Planning as a Set Partitioning Problem", European Journal of Operational Research, Vol:6, 1981

35- WARREN P. Adams-HANIF D. Sherali, "Linearization Strategies for a Class of Zero-One Mixed Integer Programming Problems", Operations Research, Vol.38, No.2, 1990

EK-1. 1992-1993 Sezonunda İkinci Ligteki Takımlar Ve Bulundukları Yerleşim Merkezleri

SIRA NO:	TAKIM ADI	YERLEŞIM YERI	BULUNDUĞU İL
1	Eyüpspor	Merkez	İstanbul
2	Gaziosmanpaşaspor	Merkez	İstanbul
3	İstanbulspor	Merkez	İstanbul
4	Kartalspor	Merkez	İstanbul
5	Küçükçekmecespor	Merkez	İstanbul
6	Usküdar Anadolu	Merkez	İstanbul
7	Zeytinburnuspor	Merkez	İstanbul
8	Yalovaspor	Yalova	İstanbul
9	Bandırmaspor	Bandırma	Balıkesir
10	İnegölspor	İnegöl	Bursa
11	Sakaryaspor	Merkez	Sakarya
12	Ayvalıkgücü	Ayvalık	Balıkesir
13	Balıkesirspor	Merkez	Balıkesir
14	Bucaspor	Merkez	İzmir
15	Göztepe	Merkez	İzmir
16	İzmirspor	Merkez	İzmir
17	Y.Salihlispor	Salihli	Manisa
18	Sökespor	Söke	Aydın
19	Denizlispor	Merkez	Denizli
20	Manisaspor	Merkez	Manisa
21	Muğlaspor	Merkez	Muğla
22	Y.Nazillispor	Nazilli	Aydın
23	Boluspor	Merkez	Bolu
24	Düzce Doğsanspor	Düzce	Bolu
25	Kütahyaspor	Merkez	Kütahya

26	Ispartaspor	Merkez	Isparta
27	Antalyaspor	Merkez	Antalya
28	Alanyaspor	Alanya	Antalya
29	Mersin İ.Y.	Merkez	Mersin
30	Tarsus İ.Y.	Tarsus	Mersin
31	Keçiörengücü	Merkez	Ankara
32	P.T.T	Merkez	Ankara
33	Petrolofisspor	Merkez	Ankara
34	Erzurumspor	Merkez	Erzurum
35	Çaykur Rizespor	Merkez	Rize
36	Akçaabat Sebatspor	Akçaabat	Trabzon
37	Orduspor	Merkez	Ordu
38	Unyespor	Unye	Ordu
39	Samsunspor	Merkez	Samsun
40	Bafraspor	Bafra	Samsun
41	Y.Yozgatspor	Merkez	Yozgat
42	Zonguldakspor	Merkez	Zonguldak
43	D.Ç.Karabükspor	Karabük	Zonguldak
44	Vanspor	Merkez	Van
45	Siirt K.H.YSE.spor	Merkez	Siirt
46	Muşspor	Merkez	Muş
47	Batman Bld.spor	Merkez	Batman
48	Diyarbakırspor	Merkez	Diyarbakır
49	Malatyaspor	Merkez	Malatya
50	K.Maraşspor	Merkez	K.Maraş
51	A.Demirspor	Merkez	Adana
52	Adanaspor	Merkez	Adana
53	İskenderunspor	İskenderun	Hatay

EK-2. 1992-1993 Sezonunda Üçüncü Ligteki Takımlar Ve Bulundukları Yerleşim Merkezleri

SIRA NO:	TAKIM ADI	YERLEŞIM YERI	BULUNDUĞU IL
1	Edirnespor	Merkez	Edirne
2	Kırklarelispor	Merkez	Kırklareli
3	Babaeskispor	Babaeski	Kırklareli
4	Lüleburgazspor	Lüleburgaz	Kırklareli
5	Uzunköprüspor	Uzunköprü	Edirne
6	Keşanspor	Keşan	Edirne
7	Malkaraspor	Malkara	Tekirdağ
8	Tekirdağspor	Merkez	Tekirdağ
9	Çorluspor	Çorlu	Tekirdağ
10	Silivrispor	Silivri	İstanbul
11	Bayrampaşaspor	Merkez	İstanbul
12	Çengelköy	Merkez	İstanbul
13	F.Karagümrük	Merkez	İstanbul
14	Kapalıçarşıspor	Merkez	İstanbul
15	Küçükköyspor	Merkez	İstanbul
16	Nişantaşıspor	Merkez	İstanbul
17	Yücespor	Merkez	İstanbul
18	Anadolu Hisarı İ.Y.	Merkez	İstanbul
19	Beylerbeyi	Beylerbeyi	İstanbul
20	Büyükşehir Bld.spor	Merkez	İstanbul
21	Dikilitaş	Merkez	İstanbul
22	Feriköy	Merkez	İstanbul
23	Galata	Merkez	İstanbul
24	Kasımpaşa	Merkez	İstanbul

SIRA NO:	TAKIM ADI	YERLEŞIM YERI	BULUNDUGU IL
25	Maltepespor	Merkez	İstanbul
26	Pendikspor	Merkez	İstanbul
27	Sümerbank Beykoz	Merke	İstanbul
28	Umraniyespor	Merkez	İstanbul
29	Vefaspor	Merkez	İstanbul
30	Darıca G.Birligi	Darıca	Kocaeli
31	Gebzespor	Gebze	Kocaeli
32	Gölcükspor	Gölcük	Kocaeli
33	Bilecikspor	Merkez	Bilecik
34	Bozüyükspor	Bozüyük	Bilecik
35	Esk.Demirspor	Merkez	Eskişehir
36	Eskişehirspor	Merkez	Eskişehir
37	Esk.Şekerspor	Merkez	Eskişehir
38	Sönmet Filanentspor	Merkez	Bursa
39	S.Bank Merinosspor	Merkez	Bursa
40	Mudanyaspor	Mudanya	Bursa
41	Karacabeyspor	Karacabey	Bursa
42	M.Kemalpaşaspor	M.Kemalpaşa	Bursa
43	Gönenspor	Gönen	Balıkesir
44	Bigaspor	Biga	Çanakkale
45	Çanspor	Çan	Çanakkale
46	Ç.Dardanelspor	Ç.Dardanel	Çanakkale
47	Burhaniyespor	Burhaniye	Balıkesir
48	Bergamaspor	Bergama	İzmir
49	Soma Linyitspor	Soma	Manisa

SIRA NO:	TAKIM ADI	YERLEŞİM YERİ	BULUNDUĞU İL
50	Soma Sotesspor	Soma	Manisa
51	Tavşanlı Linyitspor	Tavşanlı	Kütahya
52	Akhisarspor	Akhisar	Manisa
53	Sümerbankspor	Merkez	Manisa
54	Alpetspor	Merkez	İzmir
55	Altınordu	Merkez	İzmir
56	Tarişspor	Merkez	İzmir
57	Torbalıspor	Torbalı	İzmir
58	Yeşilovaspor	Yeşilova	İzmir
59	Y.Turgutluspor	Turgutlu	Manisa
60	Selçuk Efesspor	Selçuk	İzmir
61	Kuşadasıspor	Kuşadası	Aydın
62	Tirespor	Tire	İzmir
63	Ödemişspor	Ödemiş	İzmir
64	Alaşehirspor	Alaşehir	Manisa
65	Uşakspor	Merkez	Uşak
66	Buldanspor	Buldan	Denizli
67	Y.Milasspor	Milas	Muğla
68	Marmarisspor	Marmaris	Muğla
69	Fethiyespor	Fethiye	Muğla
70	Y.Dinarspor	Dinar	Afyon
71	Y.Afyonspor	Merkez	Afyon
72	Burdurgücü	Merkez	Burdur
73	Kemer Bld.spor	Kemer	Antalya
74	Ant.Köy Hiz.spor	Merkez	Antalya

SIRA NO:	TAKIM ADI	YERLEŞIM YERI	BULUNDUGU IL
75	Ormanspor	Merkez	Antalya
76	Y.Akşehirspor	Akşehir	Konya
77	Etibank Sas	Merkez	Konya
78	Konya Yolspor	Merkez	Konya
79	Karamanspor	Merkez	Karaman
80	Kırşehirspor	Merkez	Kırşehir
81	Aksarayspor	Merkez	Aksaray
82	Ereğlispor	Ereğli	Konya
83	Silifkespor	Silifke	Içel
84	Içel Polisgücü	Merkez	Içel
85	Niğdespor	Merkez	Niğde
86	Nevşehirspor	Merkez	Nevşehir
87	B.Ş.Bld.Erciyesspor	Merkez	Kayseri
88	Adana Polisgücü	Merkez	Adana
89	Adana G.Bir.spor	Merkez	Adana
90	Kozan Bld.spor	Kozan	Adana
91	Kadirli I.Y.B.Spor	Kadirli	Adana
92	Ceyhan Bld.spor	Ceyhan	Adana
93	Osmaniyespor	Osmaniye	Adana
94	Sahilspor	Merkez	Hatay
95	Hatayspor	Merkez	Hatay
96	Kırıkhan Bld.spor	Kırıkhan	Hatay
97	Islahiyespor	Islahiye	Gaziantep
98	Kilis Bld.spor	Kilis	Gaziantep
99	Elbistanspor	Elbistan	K.Maraş

SIRA NO:	TAKIM ADI	YERLEŞIM YERI	BULUNDUĞU İL
100	Nizip Bld.spor	Nizip	Gaziantep
101	Besnispor	Besni	Adıyaman
102	Ş.Urfaspor	Merkez	Şanlıurfa
103	Adıyamanspor	Merkez	Adıyaman
104	Elazığspor	Merkez	Elazığ
105	Kızıltepespor	Kızıltepe	Mardin
106	Mardinspor	Merkez	Mardin
107	Batmanspor	Merkez	Batman
108	Cizrespor	Cizre	Şırnak
109	Silvanspor	Silvan	Diyarbakır
110	Tuncelispor	Merkez	Tunceli
111	Bingölspor	Merkez	Bingöl
112	Bitlisspor	Merkez	Bitlis
113	Hakkari Ptt spor	Merkez	Hakkari
114	Yüksekova Bld.Sp.	Yüksekova	Hakkari
115	Tatvanspor	Tatvan	Bitlis
116	Van DSİ spor	Merkez	Van
117	Ercişspor	Erciş	Van
118	Ağrıspor	Merkez	Ağrı
119	Kars Köy Hiz.spor	Merkez	Kars
120	TEK 12 Martspor	Merkez	Erzurum
121	Bayburtspor	Merkez	Bayburt
122	Erzincanspor	Merkez	Erzincan
123	Artvinspor	Merkez	Artvin
124	Hopaspor	Hopa	Artvin

SIRA NO:	TAKIM ADI	YERLEŞIM YERI	BULUNDUGU IL
125	Ardeşen Bld. Spor	Ardeşen	Rize
126	Pazarspor	Pazar	Rize
127	Çayelispor	Çayeli	Rize
128	Trabzon PTT spor	Merkez	Trabzon
129	Yalıspor	Yalı	Trabzon
130	Gümüşhane K.H.Spor	Merkez	Gümüşhane
131	Giresunspor	Merkez	Giresun
132	Bulancakspor	Bulancak	Giresun
133	Fatsaspor	Fatsa	Ordu
134	Sivasspor	Merkez	Sivas
135	Tokatspor	Merkez	Tokat
136	Turhalspor	Turhal	Tokat
137	Erbaaspor	Erbaa	Tokat
138	Çarşambaspor	Çarşamba	Samsun
139	Kadıköyspor	Merkez	Samsun
140	Amasyaspor	Merkez	Amasya
141	Merzifonspor	Merzifon	Amasya
142	Vezirköprüspor	Vezirköprü	Samsun
143	Köy Hiz.Sinopspor	Merkez	Sinop
144	Çorumspor	Merkez	Çorum
145	Kastamonuspor	Merkez	Kastamonu
146	Kırıkkalespor	Merkez	Kırıkkale
147	Çubukspor	Çubuk	Ankara
148	Ankara Emniyetspor	Merkez	Ankara
149	Ankara Köy Hiz.spor	Merkez	Ankara

SIRA NO:	TAKIM ADI	YERLEŞIM YERI	BULUNDUĞU IL
150	DMY.Ank.Demirspor	Merkez	Ankara
151	DSI spor	Merkez	Ankara
152	Şekerspor	Merkez	Ankara
153	Y.Sincanspor	Y.Sincan	Ankara
154	Bartınspor	Merkez	Bartın
155	Kilimlispor	Kilimli	Zonguldak
156	Erdemir Ereğlispor	Erdemir	Zonguldak
157	Düzcespor	Düzce	Bolu
158	Akyazıspor	Akyazı	Sakarya
159	Beypazarı Bld.spor	Beypazarı	Ankara
160	Polatlıspor	Polatlı	Ankara

EK – 3.

1992-1993 Sezonunda Türkiye IKINCI Futbol Ligi ile Ilgili
Küme Bölme Probleminde KUME BÖLME TABLOSU

j	C(j)	ALT KUMELER										
1	520.3637	1	2	3	4	5	6	7	8	9	11	24
2	520.6364	1	2	3	4	5	6	7	8	11	23	24
3	524.6364	1	2	3	4	5	6	7	8	10	11	24
4	551	1	2	3	4	5	6	7	8	9	10	11
5	673.9091	1	2	3	4	5	6	7	11	23	24	42
6	675.4545	1	2	3	4	5	6	7	8	9	10	25
7	704.5455	1	2	3	4	5	6	7	8	9	10	13
8	797.5	1	2	3	4	5	6	23	24	42	43	
9	878.9091	1	2	3	4	5	6	7	8	9	12	13
10	976	2	3	4	5	6	7	8	9	10	12	13
11	1095.182	3	4	5	6	7	8	9	10	12	13	25
12	1277	4	5	6	7	8	9	10	12	13	20	25
13	1419.273	4	5	6	7	23	24	31	32	33	42	43
14	1348	5	6	7	8	9	10	12	13	17	20	25
15	1415.273	5	6	7	8	9	10	12	13	15	17	20
16	1603.455	6	7	11	23	24	31	32	33	40	42	43
17	1232.091	7	9	10	12	13	14	15	16	17	18	20
18	1268	7	8	10	11	23	24	31	32	33	42	43
19	1276.091	7	8	9	10	12	13	14	15	16	17	20
20	1077.818	8	9	10	12	13	14	15	16	17	18	20
21	1230.273	8	10	11	23	24	25	31	32	33	42	43
22	1366	8	10	11	23	24	31	32	33	41	42	43
23	1462.545	8	10	11	23	24	31	32	33	40	42	43
24	1615.818	8	11	23	24	31	32	33	39	40	42	43
25	881.5455	9	12	13	14	15	16	17	18	19	20	22
26	995.4545	9	10	12	13	14	15	16	17	18	20	22
27	1085	9	10	12	13	14	15	16	17	18	20	25
28	917	10	12	13	14	15	16	17	18	19	20	22
29	1464.818	10	11	23	24	25	31	32	33	41	42	43
30	1613.9	10	11	23	24	25	26	27	28	31	32	
31	1659.4	10	19	23	24	25	26	27	28	31	32	
32	1515.1	11	23	24	31	32	38	39	40	42	43	
33	1545.8	11	23	24	31	38	39	40	41	42	43	
34	1614.9	11	23	24	25	31	38	39	40	42	43	
35	1631.3	11	23	24	26	27	31	32	33	42	43	
36	1647	11	23	24	25	31	32	33	40	41	42	43
37	1695.455	11	23	24	31	32	33	39	40	41	42	43
38	1705.455	11	23	24	25	31	32	33	39	40	42	43

j	C(j)	ALT KUMELER										
39	1765.182	11	23	24	25	26	27	31	32	33	42	43
40	1770.2	11	23	24	25	38	39	40	41	42	43	
41	1771.091	11	23	24	31	32	33	38	39	40	42	43
42	1790.545	11	23	24	31	32	37	38	39	40	42	43
43	1812.455	11	23	24	31	37	38	39	40	41	42	43
44	1847.273	11	23	24	25	32	33	38	39	40	42	43
45	1861.2	11	19	23	24	25	26	27	28	31	32	
46	1905.3	11	23	24	25	26	27	28	30	31	32	
47	1907.7	11	23	24	36	37	38	39	40	42	43	
48	1908.909	11	23	24	26	27	31	32	33	41	42	43
49	1916	11	23	24	25	31	37	38	39	40	42	43
50	1916.7	11	23	24	25	26	27	28	29	31	32	
51	1991.091	11	23	24	25	33	37	38	39	40	42	43
52	2024.273	11	23	24	26	27	28	31	32	33	42	43
53	2059.637	11	23	24	25	37	38	39	40	41	42	43
54	865	12	13	14	15	16	17	18	19	20	21	22
55	875.4	12	13	14	15	16	17	18	20	22	25	
56	976.6364	12	13	14	15	16	17	18	19	20	22	25
57	994.6364	12	14	15	16	17	18	19	20	21	22	25
58	1071	14	15	16	17	18	19	20	21	22	26	27
59	1089	14	15	16	17	18	19	20	21	22	25	26
60	1101.091	14	15	16	17	18	19	20	21	22	25	27
61	1309.091	15	16	17	18	19	20	21	22	26	27	28
62	1301.5	17	18	19	21	22	25	26	27	28	31	
63	1387	17	19	21	22	25	26	27	28	31	32	
64	1554.8	17	19	21	22	25	26	27	28	30	31	
65	1556	17	19	21	22	25	26	27	28	29	31	
66	1510.8	19	21	22	23	25	26	27	28	31	32	
67	1580.5	19	21	22	25	26	27	28	31	32	33	
68	1685.4	19	22	23	24	25	26	27	28	31	32	
69	1786.6	19	26	27	28	29	30	31	51	52	53	
70	1836.4	19	21	22	25	26	27	28	29	30	31	
71	1842.7	19	26	27	28	29	30	50	51	52	53	
72	1867	19	25	26	27	28	29	30	31	51	52	
73	1946.7	19	22	25	26	27	28	29	30	31	51	
74	1981.2	19	23	24	25	26	27	28	31	32	33	
75	1982.6	19	21	25	26	27	28	29	30	31	51	
76	1538.1	23	24	31	32	38	39	40	41	42	43	
77	1589.8	23	24	25	31	32	33	39	40	42	43	
78	1615.4	23	24	31	32	33	38	39	40	42	43	
79	1635.9	23	24	25	32	33	39	40	41	42	43	
80	1672.2	23	31	32	33	38	39	40	41	42	43	
81	1675	23	24	32	33	37	38	39	40	42	43	
82	1704	23	24	33	37	38	39	40	41	42	43	
83	1708.5	23	24	25	32	33	38	39	40	42	43	
84	1765.9	23	24	25	33	38	39	40	41	42	43	

j	C(j)	ALT KUMELER										
85	1799.091	23	24	25	31	32	33	39	40	41	42	43
86	1808.364	23	24	31	32	33	38	39	40	41	42	43
87	1832.818	23	24	31	33	37	38	39	40	41	42	43
88	1836.727	23	24	25	31	32	38	39	40	41	42	43
89	1864.7	23	24	33	36	37	38	39	40	42	43	
90	1866.455	23	24	31	32	33	37	38	39	40	42	43
91	1881.9	23	24	36	37	38	39	40	41	42	43	
92	1887.273	23	24	25	31	32	33	38	39	40	42	43
93	1919.273	23	24	25	31	32	37	38	39	40	42	43
94	1939.091	23	24	25	26	27	31	32	33	41	42	43
95	1941.182	23	24	25	31	37	38	39	40	41	42	43
96	1988.7	23	25	26	27	28	29	30	31	32	51	
97	1990	23	35	36	37	38	39	40	41	42	43	
98	2008.5	23	33	35	36	37	38	39	40	42	43	
99	2030.6	23	24	35	36	37	38	39	40	42	43	
100	2038.727	23	24	25	26	27	28	31	32	33	42	43
101	2067.182	23	31	32	33	36	37	38	39	40	41	43
102	2073.4	23	24	25	26	27	28	29	30	31	32	
103	2108.1	23	25	26	27	28	29	30	31	32	33	
104	1758.2	25	26	27	28	29	30	31	51	52	53	
105	1821.3	25	26	27	28	29	30	50	51	52	53	
106	1885.9	25	26	27	28	29	30	31	32	51	52	
107	2083.9	25	26	27	28	29	30	31	32	33	51	
108	2143.2	25	26	27	28	29	30	31	32	33	41	
109	2211.364	25	26	27	28	29	30	31	32	33	51	52
110	1650.7	26	28	29	30	31	41	50	51	52	53	
111	1785.4	26	28	29	30	41	49	50	51	52	53	
112	1788.2	26	27	28	29	30	41	50	51	52	53	
113	1878.2	26	27	28	29	30	33	41	51	52	53	
114	1959.3	26	27	28	29	30	32	33	41	51	52	
115	1970.818	26	27	28	29	30	31	41	50	51	52	53
116	2023.909	26	27	28	29	30	31	32	41	51	52	53
117	2083.273	26	27	28	29	30	31	32	33	51	52	53
118	2131.364	26	27	28	29	30	41	49	50	51	52	53
119	2131.8	26	27	28	29	30	31	32	33	41	51	
120	2213.637	26	27	28	29	30	48	49	50	51	52	53
121	2238.182	26	27	28	29	30	31	32	33	41	51	52
122	1613.6	27	28	29	30	31	41	50	51	52	53	
123	1738.1	27	28	29	30	41	49	50	51	52	53	
124	2130.637	27	28	29	30	41	48	49	50	51	52	53
125	1525.7	28	29	30	31	41	49	50	51	52	53	
126	1535.9	28	29	30	31	32	41	50	51	52	53	
127	1691.5	28	29	30	41	48	49	50	51	52	53	
128	1753.1	28	29	30	47	48	49	50	51	52	53	
129	1758.3	28	29	30	31	32	33	41	51	52	53	
130	2115.182	28	29	30	41	47	48	49	50	51	52	53
131	2148.091	28	29	30	45	47	48	49	50	51	52	53

j	C(j)	ALT KUMELER										
132	1479.6	29	30	31	32	41	49	50	51	52	53	
133	1514.3	29	30	31	41	48	49	50	51	52	53	
134	1622.3	29	30	31	32	33	41	50	51	52	53	
135	1702	29	30	45	47	48	49	50	51	52	53	
136	1706	29	30	41	47	48	49	50	51	52	53	
137	2083	29	30	41	45	47	48	49	50	51	52	53
138	1921.2	30	41	45	47	48	49	50	51	52	53	
139	1741	31	32	33	37	38	39	40	41	42	43	
140	1926.8	31	32	33	35	36	37	38	39	40	41	
141	2069.091	31	32	33	36	37	38	39	40	41	42	43
142	2185	31	32	33	35	36	37	38	39	40	41	43
143	2310.364	31	32	33	34	35	36	37	38	39	40	41
144	1937	32	33	35	36	37	38	39	40	41	43	
145	1998.6	32	33	34	35	36	37	38	39	40	41	
146	2004.6	32	33	35	36	37	38	39	40	42	43	
147	1987.1	33	35	36	37	38	39	40	41	42	43	
148	2022.6	33	34	35	36	37	38	39	40	41	46	
149	2054.2	33	34	35	36	37	38	39	40	41	43	
150	2190.6	33	34	35	36	37	38	39	40	41	44	
151	2405	33	34	35	36	37	38	39	40	41	42	43
152	2016.7	34	44	45	46	47	48	49	50	52	53	
153	2102.7	34	35	36	37	38	39	40	41	42	43	
154	2111.5	34	35	36	44	45	46	47	48	49	50	
155	2133.7	34	35	36	37	44	45	46	47	48	49	
156	2152.1	34	41	44	45	46	47	48	49	50	53	
157	2166.8	34	35	44	45	46	47	48	49	50	53	
158	2231.5	34	35	41	44	45	46	47	48	49	50	
159	2248.8	34	35	36	41	44	45	46	47	48	49	
160	2252.1	34	35	36	37	38	44	45	46	47	48	
161	2264.1	34	35	36	37	38	39	40	41	44	46	
162	2299.8	34	35	36	37	38	39	44	45	46	47	
163	2366	34	35	36	37	41	44	45	46	47	48	
164	2372.6	34	35	36	37	38	39	40	44	45	46	
165	2458.3	34	35	36	37	38	39	41	44	45	46	
166	2482.9	34	35	36	37	38	41	44	45	46	47	
167	2193.1	41	44	45	46	47	48	49	50	52	53	
168	1992.3	44	45	46	47	48	49	50	51	52	53	

EK-4

1992-1993 Sezonunda Türkiye ÜÇÜNCÜ Futbol Ligi İle İlgili
Küme Bölme Probleminde KÜME BÖLME TABLOSU

ALT KUME NO: j	Cj	ALT KUMELER															
1	11050	1	2	3	4	8	9	10	11	12	13	14	15	16	17	18	19
2	12453	1	2	3	4	5	8	9	10	11	12	13	14	15	16	17	18
3	13338	1	2	3	4	5	7	8	9	10	11	12	13	14	15	16	17
4	14094	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
5	16072	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	46
6	18541	1	2	3	4	5	6	7	8	9	10	11	12	13	14	45	46
7	19210	1	2	3	4	5	6	7	8	9	10	11	12	13	14	44	46
8	20819	1	2	3	4	5	6	7	8	9	10	11	12	13	44	45	46
9	25903	5	6	7	38	39	40	41	42	43	44	45	46	47	48	49	50
10	23177	6	38	39	40	41	42	43	44	45	46	47	48	49	50	52	53
11	24155	6	34	38	39	40	41	42	43	44	45	46	47	48	49	50	51
12	24516	6	7	38	39	40	41	42	43	44	45	46	47	48	49	50	52
13	25068	6	32	33	34	38	39	40	41	42	43	44	45	46	47	51	158
14	900	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
15	1615	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
16	3270	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
17	7484	17	18	19	20	21	22	23	24	25	26	27	28	29	30	38	39
18	8537	17	18	19	20	21	22	23	24	25	26	27	28	29	30	40	41
19	9553	17	18	19	20	21	22	23	24	25	26	27	28	29	38	39	40
20	6573	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
21	15248	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	158
22	15398	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	157
23	26091	18	19	20	21	22	23	24	25	26	27	28	29	30	31	156	157
24	9903	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34
25	17417	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	158
26	25443	19	20	21	22	23	24	25	26	27	28	29	30	31	32	157	158
27	11643	20	21	22	23	24	25	26	27	28	29	30	31	32	33	38	39
28	11721	20	21	22	23	24	25	26	27	28	29	30	31	32	33	38	40
29	19128	20	21	22	23	24	25	26	27	28	29	30	31	32	33	38	158
30	19183	20	21	22	23	24	25	26	27	28	29	30	31	32	33	40	158
31	19613	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	158
32	21476	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
33	26533	20	21	22	23	24	25	26	27	28	29	30	31	32	33	157	158
34	33940	20	21	22	23	24	25	26	27	28	29	30	31	32	156	157	158
35	29072	30	31	32	33	34	35	36	38	39	40	51	71	155	156	157	158
36	30224	30	31	32	33	34	35	36	37	151	152	154	155	156	157	158	159
37	30445	30	31	32	33	34	35	36	38	39	40	51	65	71	156	157	158

NO: j	Cj	ALT KUMELER															
38	26628	31	32	33	34	35	36	37	38	39	152	155	156	157	158	159	160
39	27249	31	32	33	34	35	36	37	38	39	154	155	156	157	158	159	160
40	28131	31	32	33	34	35	36	37	150	151	152	154	155	156	157	158	159
41	28131	31	32	33	34	35	36	37	151	152	153	154	155	156	157	158	159
42	21637	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	51
43	22093	33	38	39	40	41	42	43	44	45	46	47	48	49	50	52	53
44	22158	33	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52
45	22489	33	34	38	39	40	41	42	43	44	45	46	47	48	49	50	51
46	22619	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	49
47	22989	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
48	23059	33	34	37	38	39	40	41	42	43	44	45	46	47	48	49	50
49	23424	33	34	36	37	38	39	40	41	42	43	44	45	46	47	48	49
50	24256	33	35	36	37	147	149	151	152	153	154	155	156	157	158	159	160
51	24314	33	34	35	36	37	38	39	40	41	51	71	76	157	158	159	160
52	24779	33	34	35	36	37	150	151	152	153	154	155	156	157	158	159	160
53	25119	33	34	35	36	37	38	39	40	51	65	71	76	157	158	159	160
54	25462	33	35	36	37	146	147	149	152	153	154	155	156	157	158	159	160
55	26024	33	34	35	36	37	38	39	40	151	152	154	155	156	157	158	159
56	26113	33	34	35	36	37	38	39	151	152	153	154	155	156	157	158	159
57	26264	33	35	36	37	51	151	152	153	154	155	156	157	158	159	160	160
58	26596	33	35	36	37	38	39	40	51	154	155	156	157	158	159	160	160
59	26671	33	35	36	37	145	147	149	152	153	154	155	156	157	158	159	160
60	26955	33	35	36	37	38	39	40	41	154	155	156	157	158	159	160	160
61	27058	33	35	36	37	38	51	152	153	154	155	156	157	158	159	160	160
62	27092	33	35	36	37	38	39	51	153	154	155	156	157	158	159	160	160
63	27733	33	35	36	37	38	39	51	71	154	155	156	157	158	159	160	160
64	29809	33	35	36	37	144	145	147	152	153	154	155	156	157	158	159	160
65	31831	33	35	36	37	38	51	71	76	154	155	156	157	158	159	160	160
66	32805	33	35	36	37	143	145	147	152	153	154	155	156	157	158	159	160
67	21819	34	35	36	37	148	149	150	151	152	153	155	156	157	158	159	160
68	24121	34	35	36	37	149	150	151	152	153	154	155	156	157	158	159	160
69	25958	34	35	36	37	51	150	151	152	153	154	155	156	157	158	159	160
70	41514	34	35	36	37	51	63	64	65	66	67	68	69	70	71	72	76
71	44107	34	35	36	37	51	64	65	66	67	68	69	70	71	72	73	76
72	44134	34	35	36	37	51	62	63	64	65	66	67	68	69	70	71	72
73	46350	34	35	36	37	51	64	70	71	72	73	74	75	76	77	78	160
74	47604	34	35	36	37	51	64	66	67	68	69	70	71	72	73	74	76
75	48068	34	35	36	37	51	65	70	71	72	73	74	75	76	77	78	79
76	21742	35	36	147	148	149	150	151	152	153	154	155	156	157	158	159	160
77	21854	35	36	37	147	148	149	150	151	152	153	154	155	156	157	159	160
78	23044	35	36	37	148	149	150	151	152	153	154	155	156	157	158	159	160
79	25375	35	36	37	71	149	150	151	152	153	154	155	156	157	158	159	160
80	30232	35	36	37	80	149	150	151	152	153	154	155	156	157	158	159	160
81	40911	35	36	37	51	59	62	63	64	65	66	67	68	69	70	71	72
82	44188	35	36	37	51	65	70	71	72	73	74	75	76	77	78	79	160
83	44943	35	36	37	51	64	66	69	70	71	72	73	74	75	76	77	78
84	45798	35	36	37	51	64	66	67	68	69	70	71	72	73	74	75	76
85	45972	35	36	37	51	64	66	68	69	70	71	72	73	74	75	76	77
86	46302	35	36	37	51	64	69	70	71	72	73	74	75	76	77	78	160

NO: j	Cj	ALT KUMELER															
87	46534	35	36	37	51	64	70	71	72	73	74	75	76	77	78	79	160
88	47626	35	36	37	65	66	67	68	69	70	71	72	73	74	75	76	77
89	47730	35	36	37	65	69	70	71	72	73	74	75	76	77	78	79	160
90	48636	35	36	37	51	64	69	70	71	72	73	74	75	76	77	78	79
91	49071	35	36	37	51	64	70	71	72	73	74	75	76	77	78	80	160
92	49670	35	36	37	65	69	70	71	72	73	74	75	76	77	78	80	160
93	49958	35	36	37	64	69	70	71	72	73	74	75	76	77	78	79	160
94	52287	35	36	37	65	70	71	72	73	74	75	76	77	78	79	80	81
95	52454	35	36	37	68	69	70	71	72	73	74	75	76	77	78	79	160
96	54555	35	36	37	68	69	70	71	72	73	74	75	76	77	78	80	160
97	54832	35	36	37	70	71	72	73	74	75	76	77	78	79	80	81	82
98	21182	36	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
99	21742	36	37	147	148	149	150	151	152	153	154	155	156	157	158	159	160
100	38790	36	37	51	59	61	62	63	64	65	66	67	68	69	70	71	72
101	43170	36	37	51	65	69	70	71	72	73	74	75	76	77	78	79	160
102	45332	36	37	51	64	69	70	71	72	73	74	75	76	77	78	79	160
103	45619	36	37	51	65	69	70	71	72	73	74	75	76	77	78	80	160
104	34560	37	51	59	60	61	62	63	64	65	66	67	68	69	70	71	72
105	45655	37	70	71	72	73	74	75	76	77	78	79	80	81	82	85	86
106	46531	37	70	71	72	73	74	75	76	77	78	79	80	81	82	86	87
107	21964	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53
108	22783	38	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
109	23434	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	65
110	22783	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54
111	21529	40	41	42	43	44	45	46	47	48	49	50	52	53	54	55	58
112	21887	40	41	42	43	44	45	46	47	48	49	50	52	53	54	58	59
113	22958	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
114	20198	41	42	43	44	45	46	47	48	49	50	52	53	54	55	57	58
115	20316	41	42	43	44	45	46	47	48	49	50	53	54	55	56	57	58
116	20576	41	42	43	44	45	46	47	48	49	50	52	53	54	55	56	59
117	22196	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56
118	19493	42	43	44	45	46	47	48	49	50	52	53	54	55	56	57	58
119	19891	42	43	44	45	46	47	48	49	50	52	53	54	55	56	57	59
120	21943	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57
121	18929	43	44	45	46	47	48	49	50	52	53	54	55	56	57	58	59
122	21391	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58
123	20286	44	45	46	47	48	49	50	52	53	54	55	56	57	58	59	60
124	16666	45	47	48	49	50	52	53	54	55	56	57	58	59	60	61	62
125	19431	45	46	47	48	49	50	52	53	54	55	56	57	58	59	60	61
126	19455	45	46	47	48	49	50	52	53	54	55	56	57	58	59	60	62
127	12598	47	48	49	50	52	53	54	55	56	57	58	59	60	61	62	63

NO: j	Cj	ALT KUMELER															
128	11843	48	49	50	52	53	54	55	56	57	58	59	60	61	62	63	64
129	12810	48	50	52	53	54	55	56	57	58	59	60	61	62	63	64	66
130	12948	48	49	50	52	53	54	55	56	57	58	59	60	61	62	63	67
131	13236	48	50	52	53	54	55	56	57	58	59	60	61	62	63	64	67
132	14910	48	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
133	15259	48	50	52	53	54	55	56	57	58	59	60	61	62	63	67	68
134	17935	48	52	53	54	55	56	57	58	59	60	61	62	63	67	68	69
135	12886	49	50	52	53	54	55	56	57	58	59	60	61	62	63	64	66
136	13357	49	50	52	53	54	55	56	57	58	59	60	61	62	63	64	67
137	14961	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
138	13943	50	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
139	14011	50	52	53	54	55	56	57	58	59	60	61	62	63	64	66	67
140	16240	50	53	54	55	56	57	58	59	60	61	62	63	64	66	67	68
141	19424	50	54	55	56	57	58	59	60	61	62	63	64	66	67	68	69
142	20688	50	54	55	56	57	58	60	61	62	63	64	65	66	67	68	70
143	21133	50	54	55	56	57	58	60	61	62	63	64	65	66	67	68	69
144	21946	50	54	55	56	57	58	60	61	62	63	64	66	67	68	69	70
145	16407	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66
146	18993	51	53	54	55	56	57	58	59	60	61	62	63	64	65	66	71
147	20448	51	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68
148	21161	51	54	55	56	57	58	59	60	61	62	63	64	65	66	70	71
149	27136	51	56	57	59	60	61	62	63	64	65	66	67	68	69	70	71
150	27158	51	55	56	57	60	61	62	63	64	65	66	67	68	69	70	71
151	28498	51	56	59	60	61	62	63	64	65	66	67	68	69	70	71	72
152	28946	51	52	59	60	61	62	63	64	65	66	67	68	69	70	71	72
153	30288	51	59	60	61	62	63	64	65	66	67	68	69	70	71	72	76
154	36518	51	65	66	68	69	70	71	72	73	74	75	76	77	78	79	160
155	38074	51	65	67	68	69	70	71	72	73	74	75	76	77	78	79	160
156	38987	51	65	66	70	71	72	73	74	75	76	77	78	79	80	81	82
157	39401	51	65	66	68	69	70	71	72	73	74	75	76	77	78	80	160
158	41043	51	65	67	68	69	70	71	72	73	74	75	76	77	78	80	160
159	14861	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67
160	15705	52	53	54	55	56	57	58	59	60	61	62	63	64	66	67	68
161	17870	52	53	54	55	56	57	58	59	60	61	62	63	64	67	68	69
162	18203	53	54	55	56	57	58	59	60	61	62	63	64	66	67	68	69
163	18296	53	54	55	56	57	58	59	60	61	62	63	64	65	66	70	71
164	19891	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69
165	21052	54	55	56	57	58	59	60	61	62	63	64	65	66	70	71	72
166	21793	54	55	56	57	58	59	60	61	62	63	64	65	66	70	71	76
167	24132	54	55	56	57	59	60	61	62	63	64	65	66	70	71	72	76
168	22287	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70
169	24132	55	56	57	58	59	60	61	62	63	64	65	66	70	71	72	76
170	24618	55	56	57	59	60	61	62	63	64	65	66	67	68	69	70	71
171	24626	55	56	57	59	60	61	62	63	64	65	66	67	68	69	70	72
172	26415	55	56	59	60	61	62	63	64	65	66	67	68	69	70	71	72

NO: j	Cj	ALT KUMELER															
173	29105	56	59	60	61	62	63	64	66	67	68	69	70	72	73	74	75
174	30005	59	60	61	62	63	64	65	66	67	68	69	70	72	73	74	75
175	37889	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	160
176	38027	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	83
177	39727	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
178	40513	65	66	67	68	69	70	71	72	73	74	75	76	77	78	80	160
179	43832	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82
180	44553	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	83
181	41465	68	69	70	71	72	73	74	75	76	77	78	79	81	82	83	84
182	41543	68	69	70	71	72	73	74	75	76	77	78	79	81	82	83	85
183	44921	68	69	70	72	73	74	75	76	77	78	79	80	81	82	83	85
184	44942	68	69	70	72	73	74	75	76	77	78	79	80	81	82	83	84
185	40171	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84
186	35917	70	71	72	73	74	75	76	77	78	79	80	81	149	150	151	160
187	37133	70	71	72	73	74	75	76	77	78	79	80	81	86	150	151	160
188	37169	70	71	72	73	74	75	76	77	78	79	80	81	82	86	151	160
189	37830	70	71	72	73	74	75	76	77	78	79	80	81	82	85	86	160
190	38490	70	71	72	73	74	75	76	77	78	79	80	81	82	83	85	86
191	39257	70	71	72	73	74	75	76	77	78	79	80	81	82	85	86	87
192	40043	70	72	73	74	75	76	77	78	79	80	81	82	83	85	86	87
193	39767	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	160
194	40204	72	73	74	75	76	77	78	79	80	81	82	83	85	86	87	160
195	40227	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87
196	38792	73	74	75	77	78	79	81	82	83	84	85	86	87	88	89	90
197	39175	73	74	75	77	78	79	80	81	82	83	84	85	86	88	89	90
198	39344	73	74	75	77	78	79	80	81	82	83	84	85	86	87	88	89
199	39667	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	160
200	39888	73	74	75	76	77	78	79	80	81	82	83	84	85	86	88	160
201	40154	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	90
202	40167	73	74	75	77	78	79	80	81	82	83	84	85	86	87	88	90
203	37178	74	75	77	78	79	80	81	82	83	84	85	86	87	88	89	90
204	34736	75	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91
205	30480	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92
206	28682	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93
207	29091	78	79	80	81	82	83	84	86	148	149	150	151	152	153	159	160
208	27549	79	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
209	28404	79	81	82	83	84	85	86	87	88	89	90	91	92	94	95	96
210	28661	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94

NO: j	Cj	ALT KUMELER															
211	24686	80	145	146	147	148	149	150	151	152	153	154	155	156	157	159	160
212	25666	80	144	145	146	147	148	149	150	151	152	153	154	155	156	159	160
213	26164	80	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
214	28127	80	81	82	84	85	86	87	88	89	90	91	92	93	94	95	96
215	28403	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
216	29258	80	81	82	83	84	85	86	87	88	89	90	91	92	94	95	96
217	26409	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	97
218	26653	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
219	25313	82	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98
220	25344	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97
221	22821	83	84	88	89	90	91	92	93	94	95	96	97	98	99	100	101
222	23571	83	84	85	88	89	90	91	92	93	94	95	96	97	98	99	100
223	24430	83	84	87	88	89	90	91	92	93	94	95	96	97	98	99	100
224	25195	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98
225	25226	83	84	85	87	88	89	90	91	92	93	94	95	96	97	98	99
226	22747	84	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102
227	24076	84	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101
228	24724	84	85	87	88	89	90	91	92	93	94	95	96	97	98	99	100
229	24788	84	87	88	89	90	91	92	93	94	95	96	97	98	99	100	102
230	25646	84	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
231	25567	85	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101
232	26799	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
233	25027	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102
234	26185	87	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103
235	33035	87	91	92	93	94	95	96	97	98	99	100	101	102	103	104	110
236	23063	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103
237	26292	88	89	91	92	93	94	95	96	97	98	99	100	101	102	103	104
238	30745	88	89	92	93	94	95	96	97	98	99	100	101	102	103	104	110
239	34966	89	92	93	94	95	96	97	98	99	100	101	102	103	104	110	111
240	30592	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	110
241	38603	92	93	94	95	96	97	98	99	100	101	102	103	104	108	110	111
242	38059	93	94	95	96	97	98	99	100	101	102	103	104	105	109	110	111
243	38201	93	94	95	96	97	98	99	100	101	102	103	104	105	106	108	110
244	39321	93	94	95	96	97	98	99	100	101	102	103	104	105	108	110	111
245	39919	93	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
246	42291	93	97	98	99	100	101	102	103	104	105	106	107	109	110	112	122
247	37074	94	95	96	97	98	99	100	101	102	103	104	105	106	109	110	111
248	37535	95	96	97	98	99	100	101	102	103	104	105	106	107	109	110	111

NO: j	Cj	ALT KUMELER															
249	37737	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
250	37842	96	97	98	99	100	101	102	103	104	105	106	107	109	110	111	112
251	38923	96	97	98	99	100	101	102	103	104	105	106	107	109	110	111	134
252	39377	96	97	98	99	100	101	102	103	104	105	106	107	109	110	111	122
253	40551	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	122
254	38885	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	122
255	38351	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	122
256	37562	99	100	101	102	103	104	105	106	107	108	109	110	111	112	115	122
257	42788	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114
258	40446	100	101	102	103	104	105	106	107	108	109	110	111	112	113	115	122
259	39809	101	102	103	104	105	106	107	108	109	111	112	113	114	115	116	117
260	40358	101	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117
261	40509	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116
262	41065	101	103	104	105	106	107	108	109	111	112	113	114	115	116	117	118
263	42312	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	122
264	39757	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117
265	40006	102	103	104	105	106	107	108	109	111	112	113	114	115	116	117	118
266	40116	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118
267	40252	103	104	105	106	107	108	109	111	112	113	114	115	116	117	118	120
268	39116	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	120
269	40906	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119
270	40844	105	106	107	108	109	110	111	112	113	114	115	116	117	118	120	121
271	42098	105	106	107	108	109	111	112	113	114	115	116	117	118	119	120	121
272	44102	105	106	107	108	109	111	112	113	114	115	116	117	118	119	120	123
273	45844	106	107	108	109	111	112	113	114	115	116	117	118	119	120	121	123
274	48336	107	108	109	111	112	113	114	115	116	117	118	119	120	121	123	124
275	49751	107	109	111	112	113	114	115	116	117	118	119	120	121	123	124	125
276	52500	107	108	109	112	113	114	115	116	117	118	119	120	121	123	124	125
277	54171	107	108	112	113	114	115	116	117	118	119	120	121	123	124	125	129
278	55597	107	108	112	113	114	115	116	117	118	119	120	123	124	125	127	129
279	53431	108	112	113	114	115	116	117	118	119	120	121	123	124	125	127	129
280	53891	108	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
281	54172	108	113	114	115	116	117	118	119	120	121	123	124	125	126	127	128
282	32065	110	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133
283	33306	110	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132
284	34267	110	119	120	121	122	123	124	125	126	127	128	129	130	131	132	134
285	34936	110	119	121	122	123	124	125	126	127	128	129	130	131	132	133	134
286	36988	110	119	121	122	123	124	125	126	127	128	129	130	131	132	134	135

NO: j	Cj	ALT KUMELER															
287	48955	111	112	113	114	115	116	117	118	119	120	121	123	124	125	126	127
288	49795	112	113	114	115	116	117	118	119	120	121	123	124	125	126	127	130
289	50088	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
290	50217	112	113	114	115	116	117	118	119	120	121	123	124	125	126	127	128
291	50630	112	113	114	115	116	117	118	119	120	121	122	123	124	125	127	129
292	49092	113	114	116	117	118	119	120	121	123	124	125	126	127	128	129	130
293	50476	113	114	115	116	117	118	119	120	121	122	123	124	125	127	129	130
294	40870	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130
295	42468	115	116	117	118	119	120	121	122	123	124	125	127	129	129	129	130
296	43074	117	118	119	120	121	122	123	124	125	127	129	129	129	130	134	135
297	32810	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133
298	35058	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	134
299	36232	118	119	120	121	122	123	124	125	127	129	129	129	130	131	132	134
300	33413	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134
301	35512	119	120	121	122	123	124	125	126	127	128	129	130	131	132	134	135
302	35617	119	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
303	33896	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
304	34578	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	136
305	36295	121	122	124	125	126	127	128	129	130	131	132	133	134	135	136	137
306	36456	121	122	123	124	125	126	127	128	129	130	131	132	133	134	136	136
307	30894	122	129	130	131	132	133	134	135	136	137	138	139	140	141	142	144
308	31811	122	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
309	31834	122	128	129	130	131	132	133	134	136	136	137	138	139	140	141	142
310	34258	122	127	128	129	130	131	132	133	134	136	136	137	138	139	140	141
311	35286	122	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140
312	35720	122	126	127	128	129	130	131	132	133	134	136	136	137	138	139	140
313	36187	122	125	126	127	128	129	130	131	132	133	134	136	136	137	138	139
314	37598	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137
315	37690	122	124	125	126	127	128	129	130	131	132	133	134	136	136	137	138
316	38266	122	123	124	125	126	127	128	129	130	131	132	133	134	136	136	137
317	29830	128	129	131	132	133	134	135	136	137	138	139	140	141	142	143	144
318	30797	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	144
319	31564	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
320	30278	129	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145
321	30457	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
322	29066	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146
323	30284	131	132	133	135	136	137	138	139	140	141	142	143	144	145	146	147
324	30584	132	133	135	136	137	138	139	140	141	142	143	144	145	146	147	148

NO: j	Cj	ALT KUMELER															
325	29632	133	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149
326	29760	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148
327	30184	133	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
328	30184	133	136	137	138	139	140	141	142	143	144	145	146	147	148	149	153
329	31777	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	154
330	29912	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149
331	34069	134	135	136	137	138	139	140	141	142	143	144	145	146	147	154	155
332	29069	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
333	29069	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	153
334	28846	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	153
335	28247	137	138	139	140	141	142	143	144	145	146	147	148	149	150	152	153
336	26801	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153
337	29399	138	139	140	141	142	143	144	145	146	147	148	149	150	151	153	154
338	27070	139	141	142	143	144	145	146	147	148	149	150	151	152	153	154	159
339	27358	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154
340	26615	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	159
341	28167	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156
342	27222	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	159
343	25104	143	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
344	25887	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	159
345	22397	144	145	146	147	148	149	150	151	152	153	154	155	156	157	159	160
346	23303	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
347	21366	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160

EK-5.

İKİNCİ Futbol Ligi İle İlgili Bilgisayar Programı

```
10 REM II.LIG ICIN****KUME BOLME ALGORITMASININ BILGISAYAR PROGRAMI**********************************
***BLOKLARI MALIYET SIRALI*******************
20 DIM X(53,53),Y(176,12),T(177),N(176),BB(176),M(55),C(100),ZOR(176),NA(60)
30 DIM D(30),S(53,53),SS(10,50),E(10),A(10),MINS(10),KON(10,53),OY(168)
40 INPUT "UZAKLIKLAR MATRISINI GORMEK ISTERMISIN (E/H)";M$
50 FOR I=1 TO 52
60 IF M$="E" THEN GOTO 70 ELSE GOTO 80
70 PRINT "X";I;"J";"=";
80 A=I+1
90 FOR J=A TO 53
100 READ X(I,J)
110 IF M$="E" THEN GOTO 120 ELSE GOTO 130
120 PRINT X(I,J);
130 NEXT J
140 IF M$="E" THEN GOTO 150 ELSE GOTO 160
150 PRINT "      "
160 NEXT I
170 INPUT "ALT SETLERI GORMEK ISTIYORMUSUN (E/H)";D$
180 CLS
190 FOR K=1 TO 176
200 IF D$="E" THEN GOTO 210 ELSE GOTO 220
210 PRINT "S";K;"=";
220 FOR L=1 TO 11
230 READ Y(K,L)
240 IF D$="E" THEN GOTO 250 ELSE 260
250 PRINT "_"Y(K,L);
260 NEXT L
270 IF D$="E" THEN GOTO 280 ELSE GOTO 290
280 PRINT "---"
290 NEXT K
300 IF E$="E" THEN GOTO 310 ELSE GOTO 320
310 PRINT "TAMAMLANDI"
320 GOTO 330
330 INPUT "ALT SET MALIYELERINI GORMEK ISTERMISIN (E/H)";E$
340 IF E$="E" THEN GOTO 350 ELSE GOTO 370
350 CLS
360 PRINT "ALT SET MALIYETLERI"
370 FOR K=1 TO 176
380 B=1
390 FOR C=1 TO 10
400 L=B
410 I=Y(K,L)
420 B=B+1
430 FOR L=B TO 11
440 J=Y(K,L)
450 T(K)=T(K)+X(I,J)
460 NEXT L
470 NEXT C
480 IF E$="E" THEN GOTO 490 ELSE GOTO 500
490 PRINT "T";K;"=";T(K),
500 NEXT K
```

```
510 INPUT "MERDIVEN FORMUNU GORMEK ISTERMISIN (E/H)";VF$
520 IF VF$="E" THEN GOTO 540 ELSE GOTO 2750
530 REM MERDIVEN FORMUNA GETIRIR
540 FOR R=1 TO 53
550 FOR K=1 TO 176
560 L=1
570 IF Y(K,L)=R THEN GOTO 580 ELSE GOTO 670
580 LET Z=Z+1
590 LET A=T(Z)
600 LET T(Z)=T(K)
610 LET T(K)=A
620 FOR L=1 TO 11
630 LET A=Y(Z,L)
640 LET Y(Z,L)=Y(K,L)
650 LET Y(K,L)=A
660 NEXT L
670 NEXT K
680 LET N(R)=Z
690 NEXT R
700 REM *** BLOKLAR ICINDEKI ALT KUMELERI MALIYETLERINE GORE KUCUKTEN-BUYUGE    DOGRU SIRALAR
710 FOR G=1 TO 176
720 IF Y(G,11)=0 THEN LET T(G)=T(G)/10 ELSE LET T(G)=T(G)/11
730 NEXT G
740 FOR Z=1 TO 176
750 LET R=Y(Z,1)
760 FOR K=Z+1 TO N(R)
770 IF T(K)<T(Z) THEN SWAP T(Z), T(K) ELSE 810
780 FOR L=1 TO 11
790 SWAP Y(Z,L), Y(K,L)
800 NEXT L
810 NEXT K
820 NEXT Z
830 REM **************ESIT KOLONLARI IPTAL EDER
840 FOR X=1 TO 176
850 IF T(X)=T(X+1) THEN LET T(X)=0
860 NEXT X
870 GOTO 1200
880 LET E=0
890 FOR R=1 TO 53
900 LET P=E+1 :LET L=2
910 IF N(R)<P THEN GOTO 1190
920 LET KON=55:A=0:B=0:C=0:H=0
930 FOR Z=P TO N(R)
940 GOTO 980
950 LET KON=55
960 FOR H=1 TO B
970 LET Z=BB(H)
980 IF L>11 THEN GOTO 1060
990 IF Y(Z,L)<KON THEN LET KON=Y(Z,L):LET I=1:LET M1=1:LET BB(I)=Z:GOTO 1010
1000 IF Y(Z,L)=KON THEN LET I=I+1:LET BB(I)=Z:LET M1=M1+1
```

```
1010 IF B=0 THEN GOTO 1040
1020 NEXT H
1030 IF B>0 THEN GOTO 1070
1040 NEXT Z
1050 GOTO 1070
1060 LET M1=1:LET G=BB(1):LET T(G)=0
1070 IF M1=1 THEN LET Z=N(R):LET A=T(Z):LET C=BB(1):GOTO 1090
1080 IF M1>1 THEN LET L=L+1:LET B=M1:GOTO 950
1090 LET T(Z)=T(C)
1100 LET T(C)=A
1110 FOR L=1 TO 11
1120 LET A=Y(Z,L)
1130 LET Y(Z,L)=Y(C,L)
1140 LET Y(C,L)=A
1150 NEXT L
1160 LET E=E+1
1170 LET N(R)=N(R)-1
1180 LET L=2:GOTO 910
1190 NEXT R
1200 LET A=0:LET F=0
1210 FOR Z=1 TO 176
1220 IF T(Z)=0 THEN LET A=A+1:LET C(A)=Z:LET F=F+1
1230 NEXT Z
1240 PRINT "ESIT KOLON SAYISI      =";F
1250 LET A=0:LET S=0
1260 LET A=A+1:LET B=A+1
1270 LET N=C(B)-B
1280 IF A=F THEN LET N=177-B
1290 LET P=C(A)-S
1300 FOR Z=P TO N
1310 LET K=Z+A
1320 LET T(Z)=T(K)
1330 FOR L=1 TO 11
1340 LET Y(Z,L)=Y(K,L)
1350 NEXT L
1360 NEXT Z
1370 LET S=S+1
1380 IF A=F THEN GOTO 1390 ELSE 1260
1390 FOR Z=1 TO 168
1400 IF Y(Z,11)=0 THEN ZOR(Z)=10 ELSE ZOR(Z)=11
1410 NEXT Z
1420 REM***************BLOKLARI UST SINIRI [M(R)] VE ICINDEKI KUME SAYISI [N(R)]
1430 REM***************HESAPLANIR
1440 LET J=0
1450 ERASE N:DIM N(60)
1460 FOR Z=1 TO 168
1470 LET J=J+1
1480 IF Y(Z,1)<Y(Z+1,1) OR Z=168 THEN LET D=Y(Z,1):LET M(D)=Z:LET N(D)=J:LET J=0 :GOTO 1490 ELSE 152
0
1490 IF Y(Z+1,1)-Y(Z,1)>1 THEN LET R=Y(Z+1,1)-Y(Z,1):LET W=0:GOTO 1500 ELSE 1520
1500 LET W=W+1:LET D=D+W:LET M(D)=Z
```

```
1510 IF W=R-1 THEN GOTO 1520 ELSE 1500
1520 NEXT Z
1530 REM GOTO 1580
1540 REM**************KUME BÖLME TABLOSUNUN GÖSTERIMI******************
1550 ' LPRINT TAB(21)"EK-3"
1555 ' LPRINT TAB(21)"1992-1993 Sezonunda Türkiye IKINCI Futbol Ligi Ile Ilgili"
1560 ' LPRINT TAB(21)"          Küme Bölme Probleminde KUME BÖLME TABLOSU"
1570 'LPRINT TAB(21)"=========================================================================="
1580 LPRINT TAB(21)" j       C(j)                ALT KUMELER        "
1590 LPRINT TAB(21)"---     --------     ----------------------------------------------------"
1600 FOR Z=132 TO 168
1610 LPRINT TAB(21)Z;TAB(28)T(Z);
1620 LET K0=38
1630 FOR L=1 TO ZOR(Z)
1640 LET K0=K0+5
1650 LPRINT TAB(K0)Y(Z,L);
1660 NEXT L
1670 IF Y(Z,1)<Y(Z+1,1) THEN LPRINT
1680 LPRINT " "
1690 NEXT Z
1700 OPEN "R",#1. "OT", 5
1710 FIELD #1, 5 AS BOL$
1720 REM GOTO 1970
1730 REM***********ARA VERILERININ (R BLOKTA 1 SATIRINI ÖRTEN SUTUNLAR)
1740 REM***********DOSYAYA KAYDI***********
1750 LET Z=1:LET K=1
1760 LET T=Y(Z,1)
1770 FOR Z=K TO M(T)
1780 FOR L=1 TO ZOR(Z)
1790 LET A=Y(Z,L):LET NA(A)=NA(A)+1
1800 LET NUM=2809*NA(A)+53*T+A-2862
1810 LSET BOL$=MKS$(Z)
1820 PUT #1, NUM
1830 NEXT L
1840 NEXT Z
1850 FOR R=T TO 53
1860 IF NA(R)=0 THEN  1920
1870 LET NUM=53*T+R+84947!
1880 J=NA(R)
1890 LSET BOL$=MKS$(J)
1900 PUT #1, NUM
1910 NA(R)=0
1920 NEXT R
1930 IF Z>=168 THEN 1970
1940 LET Z=Z+1:LET K=K+N(T)
1950 GOTO 1760
1960 REM***************ALGORITHANIN UYGULANMASI*************
1970 MINZ=100000!
1980 FOR SSS=1 TO M(1)
1990 LET T=1:LET H=1:LET P=SSS:LET E(H)=P:LET TT(H)=T(P):LET RR(H)=1
2000 LET SS(2,1)=0
```

```
2010 LET H=H+1:LET KOR(H)=KOR(H-1)+ZOR(P):PRINT "H=";H
2020 LET A=0:LET B=0:LET L=1:LET TAS=0
2030 FOR I=RR(H-1) TO 53
2040 IF L=ZOR(P)+1 THEN 2060
2050 IF I=Y(P,L) THEN KON(H,I)=1000:LET L=L+1:GOTO 2070
2060 LET KON(H,I)=KON(H-1,I)
2070 IF A>0 AND KON(H,I)=1000 THEN LET B=B+1:LET D(B)=I:LET TAS=TAS+1
2080 IF A=0 AND KON(H,I)<1000 THEN LET A=I:LET T=I:LET RR(H)=I:LET TAS=A-1
2090 NEXT I
2100 IF N(T)=0 THEN GOTO 2410
2110 IF B>0 THEN GOTO 2200
2120 LET Z=M(T)-N(T)+1
2130 FOR G=1 TO N(T)
2140 IF H=5 THEN GOTO 2150 ELSE 2160
2150 IF MINZ<TT(4)+T(Z) OR KOR(5)+ZOR(Z)><53 THEN 2170
2160 LET SS(H,G)=Z
2170 LET Z=Z+1
2180 NEXT G
2190 LET A(H)=N(T):GOTO 2410
2200 LET G=0:LET A=0
2210 FOR K=1 TO B
2220 I=D(K)
2230 LET NUM=53*T+I+84947!
2240 GET #1, NUM
2250 LET ABA=CVS(BOL$)
2260 FOR J=1 TO ABA
2270 LET NUM=2809*J+53*T+I-2862
2280 GET # 1, NUM
2290 LET Q=CVS(BOL$)
2300 OY(Q)=Q
2310 NEXT J
2320 NEXT K
2330 LET TA=M(T)-N(T)+1
2340 FOR K=TA TO M(T)
2350 IF OY(K)>0 THEN LET OY(K)=0:GOTO 2400
2360 IF H<5 THEN 2390
2370 LET TT(H)=TT(H-1)+T(K)
2380 IF TT(5)>MINZ OR KOR(5)+ZOR(K)><53 THEN 2400
2390 LET G=G+1:LET SS(H,G)=K :LET A(H)=G
2400 NEXT K
2410 IF VAR=1 THEN VAR=0:GOTO 2490
2420 IF VAR=2 THEN VAR=0:GOTO 2540
2430 IF VAR=3 THEN VAR=0:GOTO 2590
2440 IF SS(2,1)=0 THEN GOTO 2670
2450 LET U2=0
2460 IF U2=A(H) THEN GOTO 2670
2470 LET U2=U2+1:LET P=SS(H,U2):LET E(H)=P:LET TT(H)=TT(H-1)+T(P):LET SS(3,1)=0
2480 VAR=1:GOTO 2010
2490 IF SS(H,1)=0 THEN LET H=H-1:GOTO 2460
2500 LET U3=0
```

```
2510 IF U3=A(H) THEN LET H=H-1:GOTO 2460
2520 LET U3=U3+1:LET P=SS(H,U3):LET E(H)=P:LET TT(H)=TT(H-1)+T(P):LET SS(4,1)=0
2530 VAR=2:GOTO 2010
2540 IF SS(H,1)=0 THEN LET H=H-1:GOTO 2510
2550 LET U4=0
2560 IF U4=A(H) THEN LET H=H-1:GOTO 2510
2570 LET U4=U4+1:LET P=SS(H,U4):LET E(H)=P:LET TT(H)=TT(H-1)+T(P):LET SS(5,1)=0
2580 VAR=3:GOTO 2010
2590 IF SS(H,1)=0 THEN LET H=H-1:GOTO 2560
2600 LET P=SS(H,1):LET E(H)=P:LET TT(H)=TT(H-1)+T(P)
2610 IF TT(5)<MINZ THEN LET MINZ=TT(5):GOTO 2620 ELSE 2660
2620 PRINT :PRINT MINZ
2630 FOR I=1 TO 5
2640 LET WX(I)=E(I):PRINT WX(I)
2650 NEXT I
2660 H=H-1:GOTO 2560
2670 NEXT SSS
2680 CLOSE #1
2690 REM****************OPTIMUM COZUMUN GSTERIMI****************
2700 PRINT "MINZ=";MINZ
2710 PRINT
2720 FOR I=1 TO 5
2730 PRINT "S";"(";I;")=";" ";WX(I)
2740 NEXT I
2750 END
2755 '---UZAKLIK MATRISI VE ALT KUME VERILERINI DATAYA GIRINIZ---
2760 DATA .....
```

EK-6:
Uçüncü Futbol Ligiyle ilgili Bilgisayar Programı
==

```
10 WIDTH "LPT1:", 100
20 LPRINT CHR$(27); CHR$(15);
30 REM******III.LiG iCiN********MALiYET SIRALI***************************************
40 DIM Y(415, 16), T(415), N(415), M(165), C(165), NA(165)
50 DIM D(80), SS(10, 40), E(10), A(10), MINS(10), KON(10, 165), OY(415)
60 OPEN "R", #3, "DOSYA", 10
70 FIELD #3, 10 AS AL$
80 GOTO 1560
90 INPUT "UZAKLIKLAR MATRISINI GORMEK ISTERMISIN (E/H)"; M$
100 IF M$ = "E" THEN GOTO 110 ELSE GOTO 330
110 FOR I = 1 TO 159
120 IF M$ = "E" THEN GOTO 130 ELSE GOTO 140
130 PRINT "X"; I; "J"; "========================="; I; I; I
140 A = I + 1
150 FOR J = A TO 160
160 IF EK < 0 THEN LET EK = EK + 1: LET MA = 999: GOTO 240
170 IF EH < 0 THEN LET EH = EH + 1: LET MA = AD: GOTO 240
180 IF KE < 0 THEN LET KE = KE + 1: LET MA = 0: GOTO 240
190 LET AD = MA
200 READ MA
210 IF MA < -2000 THEN LET KE = MA + 2001: LET MA = 0
220 IF MA < 0 AND MA > -1000 THEN LET EK = MA + 1: LET MA = 999
230 IF MA < -1000 THEN LET EH = MA + 1002: MA = AD
240 LET NUM = 160 * I + J - 160
250 LSET AL$ = STR$(MA)
260 PUT #3, NUM
270 IF M$ = "E" THEN GOTO 280 ELSE GOTO 290
280 PRINT J; MA
290 NEXT J
300 IF M$ = "E" THEN GOTO 310 ELSE GOTO 320
310 PRINT "      "
320 NEXT I
330 INPUT "ALT KUMELERI GORMEK ISTIYORMUSUN (E/H)"; D$
340 FOR K = 1 TO 411
350 IF D$ = "E" THEN GOTO 360 ELSE GOTO 370
360 REM PRINT "S"; K; "=";
370 FOR L = 1 TO 16
380 READ Y(K, L)
390 IF D$ = "E" THEN GOTO 400 ELSE 410
400 REM PRINT Y(K, L);
410 NEXT L
420 IF D$ = "E" THEN GOTO 430 ELSE GOTO 440
430 REM PRINT "---"
440 NEXT K
450 FOR I = 1 TO 411
460 FOR L = 1 TO 15
470 FOR K = L + 1 TO 16
480 IF Y(I, K) < Y(I, L) THEN SWAP Y(I, L), Y(I, K)
```

```
490 IF D$ = "E" THEN GOTO 500 ELSE GOTO 590
500 NEXT K
510 NEXT L
520 NEXT I
530 FOR I = 1 TO 411: PRINT I;
540 FOR L = 1 TO 16
550 PRINT Y(I, L);
560 NEXT L
570 PRINT
580 NEXT I
590 INPUT "ALT KUME MALIYETLERINI GORMEK ISTERMISIN (E/H)"; E$
600 IF E$ = "E" THEN GOTO 610 ELSE GOTO 830
610 CLS
620 PRINT "ALT KUME MALIYETLERI"
630 FOR K = 1 TO 411
640 B = 1
650 FOR C = 1 TO 15
660 LET L = B
670 LET I = Y(K, L)
680 LET B = B + 1
690 FOR L = B TO 16
700 LET J = Y(K, L)
710 LET NUM = 160 * I + J - 160
720 GET #3, NUM
730 LET AY = VAL(AL$)
740 LET T(K) = T(K) + AY
750 NEXT L
760 NEXT C
770 PRINT "T"; K; "="; T(K),
780 NUM = 30000 + K
790 LET KB = T(K)
800 LSET AL$ = STR$(KB)
810 PUT #3, NUM
820 NEXT K
830 INPUT "MERDIVEN FORMUNU GORMEK ISTERMISIN (E/H)"; VF$
840 REM DOSYADAN ALT KUME MALIYETLERININ OKUNMASI
850 FOR I = 1 TO 411
860 LET NUM = 30000 + I
870 GET #3, NUM
880 LET T(I) = VAL(AL$)
890 NEXT I
900 IF VF$ = "E" THEN GOTO 920 ELSE GOTO 3420
910 REM MERDIVEN FORMUNA GETIRIR
920 FOR R = 1 TO 160
930 FOR K = 1 TO 411
940 L = 1
950 IF Y(K, L) = R THEN GOTO 960 ELSE GOTO 1050
960 LET Z = Z + 1
970 LET A = T(Z)
980 LET T(Z) = T(K)
990 LET T(K) = A
1000 FOR L = 1 TO 16
```

```
1010 LET A = Y(Z, L)
1020 LET Y(Z, L) = Y(K, L)
1030 LET Y(K, L) = A
1040 NEXT L
1050 NEXT K
1060 LET N(R) = Z
1070 NEXT R
1080 REM ALT KUME MALIYETLERINI SIRALAR
1090 FOR Z = 1 TO 411
1100 LET R = Y(Z, 1)
1110 FOR K = Z + 1 TO N(R)
1120 IF T(K) < T(Z) THEN SWAP T(Z), T(K) ELSE 1160
1130 FOR L = 1 TO 16
1140 SWAP Y(Z, L), Y(K, L)
1150 NEXT L
1160 NEXT K
1170 NEXT Z
1180 REM ESIT KOLONLARI AYIKLAR
1190 FOR X = 1 TO 411
1200 IF T(X) = T(X + 1) AND Y(X, 4) = Y(X + 1, 4) AND Y(X, 7) = Y(X + 1, 7)
     AND Y(X, 10) = Y(X + 1, 10) AND Y(X, 13) = Y(X + 1, 13) AND Y(X, 16) = Y(X + 1, 16) THEN LET T(X) = 0
1210 NEXT X
1220 LET A = 0: LET F = 0
1230 FOR Z = 1 TO 411
1240 IF T(Z) = 0 THEN LET A = A + 1: LET C(A) = Z: LET F = F + 1
1250 NEXT Z
1260 PRINT "ESIT KOLON SAYISI      ="; F
1270 LET A = 0: LET S = 0
1280 LET A = A + 1: LET B = A + 1
1290 LET N = C(B) - B
1300 IF A = F THEN LET N = 412 - B
1310 LET P = C(A) - S
1320 FOR Z = P TO N
1330 LET K = Z + A
1340 LET T(Z) = T(K)
1350 FOR L = 1 TO 16
1360 LET Y(Z, L) = Y(K, L)
1370 NEXT L
1380 NEXT Z
1390 LET S = S + 1
1400 IF A <> F THEN GOTO 1280
1410 REM SON DURUMDAKI Y(Z,L),T(Z)' LERI DOSYAYA KAYDEDER
1420 FOR Z = 1 TO 411 - F
1430 FOR L = 1 TO 16
1440 LET NUM = 16 * Z + L + 31000
1450 LET V = Y(Z, L)
1460 RSET AL$ = STR$(V)
1470 PUT #3, NUM
1480 NEXT L
1490 LET NUM = 30000 + Z
1500 LET KB = T(Z)
1510 LSET AL$ = STR$(KB)
1520 PUT #3, NUM
```

```
1530 NEXT Z
1540 PRINT "F="; F
1550 REM DOSYADAN Y(Z,L), T(Z)' LERIN OKUNMASI
1560 FOR Z = 1 TO 347
1570 FOR L = 1 TO 16
1580 LET NUM = 16 * Z + L + 31000
1590 GET #3, NUM
1600 LET Y(Z, L) = VAL(AL$)
1610 NEXT L
1620 LET NUM = 30000 + Z
1630 GET #3, NUM
1640 LET T(Z) = VAL(AL$)
1650 NEXT Z
1660 REM M(R), N(R)' LERIN HESAPLANMASI
1670 LET J = 0
1680 FOR KO = 1 TO 165: LET N(KO) = 0: NEXT KO
1690 FOR Z = 1 TO 347
1700 LET J = J + 1
1710 IF Y(Z, 1) < Y(Z + 1, 1) OR Z = 347 THEN LET D = Y(Z, 1): LET M(D) = Z: LET
 N(D) = J: LET J = 0: GOTO 1720 ELSE 1750
1720 IF Y(Z + 1, 1) - Y(Z, 1) > 1 THEN LET R = Y(Z + 1, 1) - Y(Z, 1): LET W = 0:
 GOTO 1730 ELSE 1750
1730 LET W = W + 1: LET D = D + W: LET M(D) = Z
1740 IF W = R - 1 THEN GOTO 1750 ELSE 1730
1750 NEXT Z: REM GOTO 1760
1760 'LPRINT TAB(11); "EK-4"
1770 'LPRINT TAB(11); "1992-1993 Sezonunda Türkiye UÇUNCU Futbol Ligi Ile Ilgili
"
1780 'LPRINT TAB(11); "          Küme Bölme Probleminde KUME BOLME TABLOSU"
1790 'LPRINT TAB(11); "======================================================
================================"
1800 'LPRINT TAB(11); "ALT"
1810 'LPRINT TAB(11); "KUME"
1820 LPRINT TAB(11); "NO:"; TAB(17); "Cj"; TAB(32); "ALT KUMELER"
1830 LPRINT TAB(11); "__"; TAB(17); "___"; TAB(23); "_______________________
________________________________________________"
1840 WW = 0: FOR Z = 38 TO 347
1850 LET T = Y(Z, 1): LET SU = 20
1860 LPRINT TAB(11); Z; TAB(16); T(Z); : WW = WW + 1
1870 FOR L = 1 TO 16
1880 LET SU = SU + 5
1890 LPRINT TAB(SU); RIGHT$(STR$(Y(Z, L)), LEN(STR$(Y(Z, L))) - 1);
1900 NEXT L
1910 IF Y(Z, 1) < Y(Z + 1, 1) THEN LPRINT : WW = WW + 1
1920 LPRINT " "
1930 IF WW >= 52 THEN X$ = INPUT$(2): WW = 0: GOSUB 1970
1940 NEXT Z
1950 CLOSE #3
1960 END: ''''''
1970 LPRINT TAB(11); "NO:"; TAB(17); "Cj"; TAB(32); "ALT KUMELER"
1980 LPRINT TAB(11); "__"; TAB(17); "___"; TAB(24); "______________________
________________________________________________"
1990 RETURN
2000 END
```

```
2010 REM VERILERIN GOSTERIMI
2020 FOR Z = 1 TO 347
2030 LET T = Y(Z, 1)
2040 PRINT Z; T(Z); " "; M(T); N(T); "  ";
2050 FOR L = 1 TO 16
2060 PRINT Y(Z, L);
2070 NEXT L
2080 IF Y(Z, 1) < Y(Z + 1, 1) THEN PRINT
2090 PRINT " "
2100 NEXT Z
2110 CLOSE #3
2120 REM S(T,I,J) VERILERININ DOSYAYA KAYDI
2130 OPEN "R", #2, "SAMAN", 2
2140 FIELD #2, 2 AS AHM$
2150 REM GOTO 2040
2160 LET Z = 1: LET K = 1
2170 LET T = Y(Z, 1): REM PRINT "T=";T;"Z";Z
2180 FOR Z = K TO M(T)
2190 FOR L = 1 TO 16
2200 LET A = Y(Z, L): LET NA(A) = NA(A) + 1
2210 REM PRINT "=============================NA(A)"; NA(A); "T"; T; "I"; A; "
   "; "Z"; Z
2220 LET NUM = 25600 * NA(A) + 160 * T + A - 25760
2230 LSET AHM$ = MKI$(Z)
2240 PUT #2, NUM
2250 NEXT L
2260 NEXT Z
2270 FOR R = T TO 160
2280 REM IF NA(R) = 0 THEN 1950
2290 LET NUM = 160 * T + R + 1000000!
2300 LET J = NA(R): REM PRINT "J="; J; "T="; T; "I="; R
2310 LSET AHM$ = MKI$(J)
2320 PUT #2, NUM
2330 NA(R) = 0
2340 NEXT R
2350 IF Z >= 347 THEN 2400
2360 LET K = K + N(T)
2370 GOTO 2170
2380 REM ALGORITHANIN UYGULANMASI
2390 AAA = TIMER
2400 LET MINZ = 1E+09
2410 FOR SSS = 1 TO M(1)
2420 LET T = 1: LET H = 1: LET P = SSS: LET E(H) = P: LET TT(H) = T(P): LET RR(H
) = 1
2430 LET SS(2, 1) = 0
2440 LET H = H + 1: LET KOR(H) = KOR(H - 1) + 16: PRINT "H="; H
2450 LET A = 0: LET B = 0: LET L = 1: LET TAS = 0
2460 FOR I = RR(H - 1) TO 160
2470 IF L = 17 THEN 2490
2480 IF I = Y(P, L) THEN KON(H, I) = 1000: LET L = L + 1: GOTO 2500
2490 LET KON(H, I) = KON(H - 1, I)
2500 IF A > 0 AND KON(H, I) = 1000 THEN LET B = B + 1: LET D(B) = I
2510 IF A = 0 AND KON(H, I) < 1000 THEN LET A = I: LET T = I: LET RR(H) = I
```

```
2530 REM PRINT "KON(H,I)"; H, I, KON(H, I)
2540 NEXT I
2550 IF N(T) = 0 THEN GOTO 2860
2560 IF B > 0 THEN GOTO 2650
2570 LET Z = M(T) - N(T) + 1
2580 FOR G = 1 TO N(T)
2590 IF H = 10 THEN GOTO 2600 ELSE 2610
2600 IF MINZ < TT(9) + T(Z) THEN 2620
2610 LET SS(H, G) = Z
2620 LET Z = Z + 1
2630 NEXT G
2640 LET A(H) = N(T): GOTO 2860
2650 LET G = 0: LET A = 0
2660 FOR K = 1 TO B
2670 LET I = D(K)
2680 LET NUM = 160 * T + I + 1000000!
2690 GET #2, NUM
2700 LET AGA = CVI(AHM$): REM PRINT "AGA=";AGA
2710 FOR SA = 1 TO AGA
2720 LET NUM = 25600 * SA + 160 * T + I - 25760
2730 GET #2, NUM
2740 LET Q = CVI(AHM$): REM PRINT "Q=";Q;"T=";T;"I=";I
2750 LET OY(Q) = Q
2760 NEXT SA
2770 NEXT K
2780 LET TA = M(T) - N(T) + 1: REM PRINT "M(T)"; M(T); "TA"; TA
2790 FOR K = TA TO M(T)
2800 IF OY(K) > 0 THEN LET OY(K) = 0: GOTO 2850
2810 IF H < 10 THEN 2840
2820 LET TT(H) = TT(H - 1) + T(K)
2830 REM IF TT(10) > MINZ THEN 2570
2840 LET G = G + 1: LET SS(H, G) = K: LET A(H) = G: REM PRINT "SS(H,G)"; H; G; S
S(H, G); "A(H)"; A(H)
2850 NEXT K
2860 IF VAR = 1 THEN LET VAR = 0: GOTO 2990
2870 IF VAR = 2 THEN LET VAR = 0: GOTO 3040
2880 IF VAR = 3 THEN LET VAR = 0: GOTO 3090
2890 IF VAR = 4 THEN LET VAR = 0: GOTO 3140
2900 IF VAR = 5 THEN LET VAR = 0: GOTO 3190
2910 IF VAR = 6 THEN LET VAR = 0: GOTO 3240
2920 IF VAR = 7 THEN LET VAR = 0: GOTO 3290
2930 IF VAR = 8 THEN LET VAR = 0: GOTO 3340
2940 IF SS(2, 1) = 0 THEN GOTO 3420
2950 LET U2 = 0
2960 IF U2 = A(H) THEN GOTO 3420
2970 LET U2 = U2 + 1: LET P = SS(H, U2): LET E(H) = P: LET TT(H) = TT(H - 1) + T
(P): LET SS(3, 1) = 0
2980 LET VAR = 1: GOTO 2440
2990 IF SS(H, 1) = 0 THEN LET H = H - 1: GOTO 2960
3000 LET U3 = 0
3010 IF U3 = A(H) THEN LET H = H - 1: GOTO 2960
3020 LET U3 = U3 + 1: LET P = SS(H, U3): LET E(H) = P: LET TT(H) = TT(H - 1) + T
(P): LET SS(4, 1) = 0
```

```
3030 LET VAR = 2: GOTO 2440
3040 IF SS(H, 1) = 0 THEN LET H = H - 1: GOTO 3010
3050 LET U4 = 0
3060 IF U4 = A(H) THEN LET H = H - 1: GOTO 3010
3070 LET U4 = U4 + 1: LET P = SS(H, U4): LET E(H) = P: LET TT(H) = TT(H - 1) + T
(P): LET SS(5, 1) = 0
3080 LET VAR = 3: GOTO 2440
3090 IF SS(H, 1) = 0 THEN LET H = H - 1: GOTO 3060
3100 LET U5 = 0
3110 IF U5 = A(H) THEN LET H = H - 1: GOTO 3060
3120 LET U5 = U5 + 1: LET P = SS(H, U5): LET E(H) = P: LET TT(H) = TT(H - 1) + T
(P): LET SS(6, 1) = 0
3130 LET VAR = 4: GOTO 2440
3140 IF SS(H, 1) = 0 THEN LET H = H - 1: GOTO 3110
3150 LET U6 = 0
3160 IF U6 = A(H) THEN LET H = H - 1: GOTO 3110
3170 LET U6 = U6 + 1: LET P = SS(H, U6): LET E(H) = P: LET TT(H) = TT(H - 1) + T
(P): LET SS(7, 1) = 0
3180 LET VAR = 5: GOTO 2440
3190 IF SS(H, 1) = 0 THEN LET H = H - 1: GOTO 3160
3200 LET U7 = 0
3210 IF U7 = A(H) THEN LET H = H - 1: GOTO 3160
3220 LET U7 = U7 + 1: LET P = SS(H, U7): LET E(H) = P: LET TT(H) = TT(H - 1) + T
(P): LET SS(8, 1) = 0
3230 LET VAR = 6: GOTO 2440
3240 IF SS(H, 1) = 0 THEN LET H = H - 1: GOTO 3210
3250 LET U8 = 0
3260 IF U8 = A(H) THEN LET H = H - 1: GOTO 3210
3270 LET U8 = U8 + 1: LET P = SS(H, U8): LET E(H) = P: LET TT(H) = TT(H - 1) + T
(P): LET SS(9, 1) = 0
3280 LET VAR = 7: GOTO 2440
3290 IF SS(H, 1) = 0 THEN LET H = H - 1: GOTO 3260
3300 LET U9 = 0
3310 IF U9 = A(H) THEN LET H = H - 1: GOTO 3260
3320 LET U9 = U9 + 1: LET P = SS(H, U9): LET E(H) = P: LET TT(H) = TT(H - 1) + T
(P): LET SS(10, 1) = 0
3330 LET VAR = 8: GOTO 2440
3340 IF SS(H, 1) = 0 THEN LET H = H - 1: GOTO 3310
3350 LET P = SS(H, 1): LET E(H) = P: LET TT(H) = TT(H - 1) + T(P)
3360 IF TT(10) < MINZ THEN LET MINZ = TT(10): GOTO 3370 ELSE 3410
3370 PRINT : PRINT MINZ
3380 FOR I = 1 TO 10
3390 LET WX(I) = E(I): PRINT WX(I)
3400 NEXT I
3410 LET H = H - 1: GOTO 3310
3420 NEXT SSS
3430 CLOSE #2
3440 PRINT "OPTIMUM ÇOZUM BULUNDU"
3450 PRINT "Minimum.Z="; MINZ
3460 PRINT
3470 FOR I = 1 TO 10
3480 PRINT "S"; "("; I; ")="; " "; WX(I)
3490 NEXT I
```

```
3500 PRINT "ÇOZUM SURESI"
3510 PRINT "SANIYE="; TIMER - AAA
3520 END
4000 DATA **BURAYA, UZAKLIK MATRISI VE TABLODA YERALACAK ALT KUMELER GIRILECEK**
```