T.C.
ANKARA ÜNiVERSiTESi
SOSYAL BiLiMLER ENSTiTÜSÜ
iKTiSAT
ANABiLiM DALI

Doğrusal Programlama'ya sütun uzayında bakış

Doktora Tezi

Mehmet Murat AYGEN

Tez Danışmanı
Doç. Dr. Türkmen GÖKSEL

Ankara-2018

T.C.
ANKARA ÜNiVERSiTESi
SOSYAL BiLiMLER ENSTiTÜSÜ
iKTiSAT
ANABiLiM DALI

Doğrusal Programlama'ya sütun uzayında bakış

Doktora Tezi

Tezi Danışmanı: Doç. Dr. Türkmen GÖKSEL

Tez Jürisi Üyeleri

Adı ve Soyadı	**imzası**
.....	.....
.....	.....
.....	.....
.....	.....
.....	.....

Tez Sınavı Tarihi: 31 Mayıs 2018 Perşembe

TÜRKiYE CUMHURiYETi
ANKARA ÜNiVERSiTESi
SOSYAL BiLiMLER ENSTiTÜSÜ MÜDÜRLÜĞÜ'NE

Bu belge ile, bu tezdeki bütün bilgilerin akademik kurallara ve etik davranış ilkelerine uygun olarak toplanıp sunulduğunu beyan ederim. Bu kural ve ilkelerin gereği olarak, çalışmada bana ait olmayan tüm veri, düşünce ve sonuçları andığım ve kaynağını gösterdiğimi ayrıca beyan ederim. (31/05/2018)

Tezi hazırlayan

Mehmet Murat AYGEN

imzası

İçindekiler

Bölüm 1. Giriş 1
1.1 Tezin Konusu 5
1.2 Tezin Önemi 5

Bölüm 2. Kuramsal Çerçeve (I) 7
2.1 Simpleks Algoritması 13
2.2 Dantzig Kuralı ve eleştirisi 16
2.3 Anahtarlamanın Geometrisi ve alt-taban 18

Bölüm 3. Örnek Uygulamalar 22

Bölüm 4. Kuramsal Çerçeve (II) 25
4.1 Yoz çözüm 25
4.3 Temel Çözüm sayısını veren formül 27
4.4 Kaynak Kullanımını Destekleme (KKD) 27
4.5 Revize Simpleks Tablosu 27
4.6 Döngü (ing. cycling) 30
4.7 Leksikografik Döngü Önleme Kuralı 32

Bölüm 5. İktisadi Uygulama 34

Bölüm 6. Kuramsal Çerçeve (III) 37
6.1 Fazlalık Yoklaması 37
6.2 Mutlak Fazlalık 38

Bölüm 7. Demokrasi Prensibi 41

Bölüm 8. EN-KÜÇÜK-AÇILAR ANAHTARLAMA KURALI 43

Bölüm 9. Değerlendirme ve Sonuç 46

Kaynakça 46

Çizimler

Şekil 3.1 Teknolojik Olanaklılık Konisi 21

Şekil 7.1 Sağlıklı (küb kesiti) 39

Şekil 7.2 Fazlalık var (küb kesiti) 40

Şekil 8.1 Shively Fig.33 41

Şekil 8.2 Pan Problemi 1 42

Bölüm 1

GiRiŞ

Hukuk biliminin vekalet teorisine göre firma bir açık ve zımni mukaveleler manzumesidir (ing. nexus of contracts) [bkz: Yılmaz, s.14]. Ticari mübadelelerin yanı sıra üretime de bir çeşit mübadele nazarı ile bakabilirsek, onu taraflarından biri hep kendisi olan bir "mübadeleleler manzumesi" (ing. nexus of exchanges) olarak da görebiliriz. Piyasa fiyatlarını veri almak durumunda olan (küçük ölçekli) bir firma, hangi mübadeleye ne ölçüde gireceği kararlarını, geleneksel olarak, maliyet muhasebesi yaparak alır. Firma kuramı bağlamında Doğrusal Programlama (DP), firmanın, bu muhasebeyi, hesaba (1) kendi bünyesi dahilindeki dışsallıkları ve (2) gölge fiyatları[1] da katarak yapmasına imkan veren bir ileri hesaplama tekniği olmaktadır. Böyle dışsallıkların sözkonusu olmadığı bir firma, kararlarını, klasik maliyet muhasebesi yaparak almaya devam edebilir. Ticaret unvanlarına "entegre" sözcüğünü ekleyen imalatçı firmalar "böyle dışsallıklardan yararlanma olanağımız vardır" demiş olmaktadırlar. Doğrusal Programlama'ya, bir ulusal veya küresel ekonominin bütünü için kurulan Leontief girdi-çıktı modelinin, firmaya, matematikteki anlamı ile "kısıtlaması" (ing. restriction) olarak da bakılabilir. Bazı firmaların karar alma süreçlerinde bilfiil kullanıyor olmaları, modelin gerçekçiliğinin ve kullanışlılığının bir kanıtı sayılmalıdır.

Bu tezde bir karar alma yöntemi olarak Doğrusal Programlama'nın pratikte karşılaşılan (1) kısıtların fazlalık veya gereksizlikleri (ing. redundancy), (2) yozluk (ing. degeneracy) ve (3) döngü (ing. cycling) gibi

[1] Gölge fiyatlar kuşkusuz firmanın muhtelif kapasiteleri ile ilgilidirler. Bu kapasiteleri bizzat kullanmakla, fason üretim yapmak isteyenlere kiralamakla elde edeceği gelirin üzerine ne ölçüde çıkabildiğini hesaplamasına yarar sayısal veriler olmaktadırlar. Eğer gölge fiyatlar kira rayiçlerinin üzerinde değillerse, dar anlamda "kapitalizm", o firmada bitmiş demektir - ancak kooperatifleşerek idame ettirilebilir (ing. fall under voluntary administration).

sorunlarını daha iyi anlamağa ve çözüm yolları bulmağa çalışacağız.

"Kapasite" ve sanayi istatistikleri arasında sıkça anılan "kapasite kullanım oranı (%)" basit kavramlar değillerdir. Bir dökme kuru yük gemisinin kapasitesini tek bir birimle ifade etmek zordur. Kabul edilecek muhtelif yükün (ki bu demir cevheri de olabilir, tahıl da) özgül ağırlıklarına bağlı olarak kapasite "ton" olabileceği gibi ambarların hacmi (m^3) de olabilir. Sanayi bölgelerimizde konvansiyonel bir kamyona bir seferde sadece dört veya beş adet çelik ingot veya tek bir sac rulosu yüklenerek taşıma yapıldığı sık görülen bir durumdur. Nakliye firmaları nerede ise boş kalan kasaya yüklenerek aynı istikamete sevkedilecek mesela yaş meyve-sebze gibi özgül ağırlığı demire kıyasla çok daha az başkaca yük bulamıyor olmalıdırlar. Trafik kurallarının izin verdiği yüksekliğe kadar kuru ot balyaları yüklenmiş bir kamyonun da "ölü yük istiap haddi" %100 kullanılmıyor olabilir[2].

Mesela talaşlı imalat işkolunun kapasitesi bir torna veya freze tezgahına bir seferde azami kaç kilogramlık bir parçanın takılabileceği esas alınarak hesaplanmaktadır. Bu durumda kapasitenin ölçüleceği birim kg veya ton olmaktadır. Bu tercih demir-çelik işkolu ile koordineli çalışmak durumunda olan işkolunun ulusal ölçekte planlanmasında hesaplama kolaylığı sağlıyor olsa da, bir talaşlı imalat işyeri üretim planlamasını ton olarak değil de makinalarını işler vaziyette tutabileceği mesai saati cinsinden yapacaktır. Buna göre günlük kapasite makina sayısı çarpı (bakım/onarım, vardiya değişimleri ve yemek molaları düşüldüğünde yaklaşık) 20-21 saat kadar olmalıdır. Tornanın yapacağı işi freze, frezenin yapacağı işi de torna yapamayacağından; her biri için ayrı kapasite hesaplamak gerekebilecektir. Kapasitesini birden fazla çeşit kıt kaynağın belirlediği bir başka işkolu dökümcülüktür. Burada (1) metal ergitme ve (2) kalıp imalatı kapasiteleri dökülecek parçaların umumiyetle hafif veya ağır olmalarına bağlı olarak kısıtlayıcı olabilmektedir. Gerek mayi maden gerekse döküm kalıbı piyasaya arzedilmeleri imkansız ve birbirlerini tamamlayan ara-ürünlerdir (ing. complementary goods, work-in-progress). Kapasitenin, böyle piyasa-değeri-olmayan ara-mallar cinsinden ifadesi, firmayı yönetenlerden başka hiçkimse için bir anlam ifade etmeyecektir.

Kuru tarımdan sulu tarıma geçen bir çiftçinin kapasitesini artık işleyebildiği toprak yüzölçümü değil de, hektar başına yılda kaç m^3

[2]Bir taşıt aracının hareket etmesine (yürümesine, yüzmesine, uçmasına) şöyle veya böyle katkı yapan bütün bileşen ve aksamının ağırlıkları toplamına o aracı "darası" (ing. tara weight) denir. Ambar, antrepo v.b. tesislere teslimat yapmak için gelen kamyonun, kantara alınmakla ölçülen ağırlığından darasının düşülmesi ile bulunan sayı, getirdiği "ölü yük"ün (ing. deadweight) ölçümü olmaktadır. "Ölü yük istiap haddi" de, bir taşıt aracının taşıyabileceği maksimum ölü yük olarak tanımlanmıştır.

su bulabileceği belirleyecektir. Her iki kıt kaynaktan (toprak ve sulama) da azami randıman almak isteyen rasyonel çiftçi ise, toprakları üzerinde optimum oranlarda kuru ve sulu tarımı eşzamanlı olarak yapacaktır. İşte bu optimum oranları hesaplamak için kullanması gereken yöneylem araştırması tekniği Doğrusal Programlama'dır. Tıpkı gemisini/kamyonunu "tam" yüklemek isteyen nakliyeci gibi.. Hangi siparişleri eşzamanlı olarak dökmesi gerektiğini dökümcüye bildirecek hesaplama tekniği de budur.

Eğer kısıt sayısı m ise, en çok m tane karar-değişkenine pozitif değerler atanacağı, Doğrusal Programlama kuramının, ilerideki sayfalarda göreceğimiz gibi, bir çıkarsamasıdır. Gerek nakliyecilik, gerekse dökümcülükte $m = 2$ olduğundan; bir nakliyecinin gemisine/kamyonuna 2'den fazla çeşit mal yüklemesi ve bir dokümcünün 2'den fazla sayıda siparişi eşzamanlı olarak dökmesi verimli çalışma ortamı bulamıyor olmalarının göstergeleri sayılmalıdır. Nakliyecinin sorunu yerel yük-arzı yetersizliği, dökümcünün sorunu da "hatırlı müşterileri" ile iyi geçinme zorunluluğu olabilir ki buna da "talep yetersizliği" demek mümkündür. Bunlar bir ekonominin prodüktivitesi gibi oldukça soyut bir kavram hakkında fikir edinmek için çıplak-gözle görülebilecek kanıtlar olmaktadır.

Piyasa ekonomisi geliştikçe ve derinleştikçe birbirlerinden ayrı düşünülmesi olanaksız gibi görünen kapasitelerin ekonomik anlamda ayrıştıklarına ve ayrı ayrı fiyatlandırıldıklarına tanık oluyoruz. Karayolu nakliyeciliğinde "Tam Yükleme" (ing. Full Truck Load, FTL) eski normal olmaktadır. Piyasa ekonomisinin derinleşmesi yanı-sıra bu iş koluna özgü trafik sıkışıklığı sorununun da başgöstermesi ile, kamyon kasası hacmini fiyatlandırma gereksinimi duyumsanmış ve lojistik el-kitabı ve sözlüklerine Türkçeye "Eksik Yükleme" ve "Navlun Sınıfları" diye kazandırılabilecek yeni sözdizimleri eklenmiştir. Bunlar, sırası ile, "Less than Truck Load (LTL)" ve "Freight Classes" teknik deyimlerinin karşılıkları olmaktadır. Eski normalden bir sapma ifade eden olumsuz "LTL" deyimi, olumlu "(kamyon kasasını) tam doldurma" kavramını da beraberinde getirmektedir ki, bu da İngilizce "Stacking Efficiency"[3] ve "Stowage"[4] kavramlarının karşılığıdır.

Şimdi, bir navlun proforma faturasını, gerçekte nasıl yapılıyor olduğuna bakmaksızın, şöyle keselim:

$$F = \pi_W W + \pi_V V.$$

Burada W ve V kabul edilecek yükün ağırlığı ve hacmi, π_W ve π_V

[3] LODER, s.61

[4] Alkan & Erdal, s.367

çarpanları bir kilogram ve bir litre yükü taşımak için talep edeceğimiz fiyat rayiçleri, F de fatura yekunudur. Bu fiyatlandırmanın gerçek yaşama uygunluğunu göstermek için, denklemin her iki yanını da W ile bölelim:

$$F/W = \pi_W \cdot 1 + \pi_V(V/W) = \pi_W + \pi_V D^{-1}$$

buluruz. Burada D yükün özgül ağırlığı olmaktadır. Bulduğumuz bu formülün sözel anlatımı "kilometre başına navlun rayicinin birimi TL/kg veya TL/ton'dur" eski normaline göre ifade edecek olursak "reel navlun rayici eşittir kuramsal (veya saf) rayiç artı özgül-ağırlık ile ters orantılı bir işgaliye zammı" olmaktadır ki, fiyatlandırma konusunda bütünü ile uygulamacılara dönük kitaplarda yazılı olanlarla uyumludur[5]. Yukarıda söz ettiğimiz "Navlun Sınıfları" deyimi de muhtelif yükün özgül ağırlıklarına göre tasnif edilerek, ait olduğu sınıfın rayici ile fiyatlandırılması uygulaması ile ilgili bir deyimdir.

Bir yerel piyasada π_W ve π_V rayiçlerinin henüz oluşmamış olması böyle "akıllı fiyatlandırma" yapmak isteyen bir girişimci için engel değil, fırsattır: kanıksanmış TL/ton rayicin altına böyle "hesaplı" bir biçimde inerek pazar payını arttırması imkanı dahilindedir. Aynı "akıllı fiyatlandırma"yı bir dökümcü de yapabilir. π_W rayicini hesaplamakta akaryakıt ikmali ve piston gömlekleri, dış-iç lastik, rulman gibi aksamın yenilenmesi için yapacağı harcamaları; π_V rayicini hesaplamakta ise ödeyeceği vergileri esas almalıdır. "Karayolu yüzeyi tayınlanması" problemi her siyasi rejimin kendine özgü çözümler bulduğu bir "işgaliye vergilendirmesi" problemidir. Bu açık-uçlu bir problemdir. "Paralı yollar", "şerit tahsisi veya kısıtlaması", "tek-çift sayılı plaka tahdidi", trafiğin yoğun olduğu saatlerde yol üzerindeki sifon-otoparklarda bekleme zorunlulukları bu çözümlerden en çok bilinenleridir.

Şimdi bir tane kamyonu olan bir nakliyeci düşünelim. Kendisinden değişik 10 çeşit yükün taşınması talep edilmiş olsun. Çözmesi gereken Doğrusal Programlama problemini şöyle formüle etmek mümkündür:

$max.$	$0 \cdot x_1$	$+0 \cdot x_2+$	$\ldots$	$+0 \cdot x_{10}$	$-\pi_V s_1$	$-\pi_W s_2$	
$ST:$	$a_{1,1}x_1$	$+a_{1,2}x_2+$	$\ldots$	$+a_{1,10}x_{10}$	$+s_1$		$= V$
	$a_{2,1}x_1$	$+a_{2,2}x_2+$	$\ldots$	$+a_{2,10}x_{10}$		$+s_2$	$= W$
	x_1						$\leq d_1$
		x_2					$\leq d_2$
			$\ddots$				$\vdots$
				x_{10}			$\leq d_{10}$
ve	eksiolmazlık	kısıtları.					

[5]Rushton, Croucher & Baker, s.96-97

Burada, $i = 1, 2, \ldots, 10$ olmak üzere, d_i taşınması talep edilen yükün kendine has bir birimle (kutu, çuval, şişe, vb.) "miktarı", $a_{1,i}$ bir biriminin hacmi, $a_{2,i}$ de ağırlığı olmaktadır. Yani $a_{2,i}/a_{1,i}$ yükün özgül ağırlığıdır. Amaç da s_1 ve s_2 (gevşeklik) değişkenleri ile gösterilen "atıl kapasiteler"in fırsat maliyetleri toplamını en aza indirmektir. Bunun kâr maksimizasyonu ile eş-anlamlı olduğunu şöyle gösteririz:

$$\sum_{j=1}^{10}(\pi_V a_{1,j} + \pi_W a_{2,j})x_j = \pi_V(\sum_{j=1}^{10} a_{1,j}x_j) + \pi_W(\sum_{j=1}^{10} a_{2,j}x_j) =$$

$$\pi_V(V - s_1) + \pi_W(W - s_2) = (\pi_V V + \pi_W W) - \pi_V s_1 - \pi_W s_2$$

olacaktır. Bu son ifadede parantez içindeki nicelik bir sabittir. Hem hacmi hem de istiap haddini "tam" kullanabilsek elde edeceğimiz kuramsal kârı vermektedir. Sabit olması itibarı ile maksimizasyon dışı tutulabilir.

1.1 TEZiN KONUSU

Tezin konusu Doğrusal Programlama'ya, yarım asırdır terkedilmiş bir yaklaşımı tekrar benimseyerek sütun uzayında bakmak ve böyle bakmakla yapılabilecek ilerlemeleri saptamaktır. Alışıldığı üzere satır uzayında bakmakla kaydedilebilecek bir ilerlemenin artık bulunamadığı kanaati yaygındır. Yarım asır önce, bu doktora tezinin konusu olan sütun uzayına "Gereksinim(ler) Uzayı" (ing. requirement(s) space) diye bir ad da takılmıştır. Ne var ki bu terminolojiye daha sonraki yıllarda yazılan ders kitaplarının fihristlerinde rastlamak mümkün olmamaktadır. Öte yanda, satır uzayına, sanki çözüm(ler) ancak bu uzayda görülebilir ve bulunabilirmiş gibi "Çözüm Uzayı" adı takan yazarlar da vardır [bkz: Hadley, s. 162-166]. Amaç iki boyutlu uzayda bir çokgen, üç boyutluda da politop veya polihedral-küme olarak "görünen" olanaklılık bölgelerini görmekse, sütun uzayında (iki boyutlusu ile sınırlı olmak üzere) bunun dahi mümkün olduğunu ileride göreceğiz [bkz: Şekil 7.1, 7.2; s.40, 41].

1.2 TEZiN ÖNEMi

Fazla kısıtların erken teşhisi Doğrusal Programlama problemlerinin çözümünde kullanılan Simpleks algoritmasının etkinlik ve hızını arttırabilecektir.

Bu tezde yaptığımız asıl katkı "döngü" olgusunu, zannedildiği gibi Simpleks algoritmasının yaptığı bir arıza değil, fakat problemin sıra-dışı çözümlerinin mevcut olduklarını gösterir bir işaret olduğunu anlamış ve göstermiş olmamızdır. Bu sıra-dışı çözümleri Simpleks algoritması bulamaz, ancak döngüye girmekle bize böyle çözümlerin bulunabileceği iyi haberini verir.

Kuramsal Çevçeve I bölümünde Simpleks algoritmasına bir adım adım taban değiştirme algoritması nazarı ile bakacağız. Örnek Uygulamalar'da bu yaklaşımın pratik uygulanabilirliğine örnekler vereceğiz. Kuramsal Çerçeve II bölümünde taban değiştirme sırasında özen gösterilmesi gereken incelikleri, bazı tekrarlar yapmaktan kaçınmayarak, bir kez daha vurgulayacağız. Yoz çözüm ve döngü kavramlarını araştırmaya yine bu bölümde başlayacağız. İktisadi Uygulama bölümünde döngü olgusunun iktisadi anlamını açıklamaya çalışacağız. Kuramsal Çerçeve III bölümünde de fazlalık olgusuna hem satır, hem de sütun uzaylarında bakacak ve pratikte uygulaması olabilecek sonuçlara varacağız.

Bilgisayar (arka) belleklerinin ekonomik kullanımı bakımından anlamlı saptamalar gerek kukla-değişken (ing. dummy variable) kullanımı pratiği gerekse Revize Simpleks Tablosu bağlamlarında yapacağız. Bunlardan birincisinin mantıksal hiçbir dayanağı olmadığını göstermiş olduğumuza inanıyoruz.

Bölüm 2

KURAMSAL ÇERÇEVE (I)

Yukarıda verdiğimiz kamyon yükleme problemi örneğinde, amaç fonksiyonunda, sıfırdan farklı katsayılarla sadece gevşeklik değişkenleri[1] vardı ve yapısal karar değişkenleri katsayıları hep sıfırdı. Bu itibarla, verdiğimiz çok sıra-dışı bir örnek olmaktadır. Çoğu ders kitabı imalat sanayi üretim programlama problemini (ing. Product Mix Problem) esas alır ve eğer gevşeklik değişkenlerinin ölçtükleri atıl kapasiteleri fason üretim yapmak isteyen girişimcilere kiralama olanağı yoksa, bu değişkenler amaç fonksiyonunda sıfır çarpanlarla yer alırlar − bir başka deyişle yer almazlar. Bu varsayımın, bu tezin "Dantzig Kuralı ve eleştirisi" başlığı altında göreceğimiz gibi, kuramsal çıkarsamaları vardır ve bu itibarla özel bir adla anılmayı haketmektedir. "Kiralama Yok Varsayımı" adı ile anılmasını öneriyoruz. Dünya ekonomisindeki son gelişmeler, pek de gerekli olmadığını göstereceğimiz bu varsayımı, gerçekçi olmaktan giderek uzaklaştırmaktadır[2].

Bütün ders kitaplarının bu varsayım altında esas aldıkları imalat sanayi üretim programlama problemi şöyle formüle edilmektedir:

[1]Gevşeklik değişkenleri (ing. slack variables), matematikçilerin, kısıtları arasında doğallıkla $\cdots \leq \ldots$ şeklinde *eşitsizliklerin* de bulunduğu bir D.P. problemini, bir doğrusal denklem (yani *eşitlik*) sistemine dönüştürebilmek için, eşitsizliklerin sol taraflarına ekledikleri karar değişkenleri olmaktadır. Vermekte olduğumuz örnekte olduğu gibi, amaç fonksiyonu "fırsat maliyetleri toplamı"na dönüştüğünde ve ileride göreceğimiz gibi kullanıl(a)mayan kapasiteleri fason üretim yapacak girişimcilere kiralama imkanı mevcut olduğunda, bu değişkenler, atıl/münhal kapasitelerin ölçüsü olmakla, ekonomik anlam da kazanmaktadırlar.

[2]Bkz: "*Sharing economy* gains ground across sectors" (title over full page) End of Ownership news, © The Financial Times Limited 2017 No.39,664, Wednesday 27 December 2017, p.14

$max.$	c_1x_1	$+c_2x_2$	$\dots$	$+c_{n-m}x_{n-m}$	$+0s_1$	$+0s_2$	$\dots$	$+0s_m$	RHS
$ST:$	$a_{1,1}x_1$	$+a_{1,2}x_2$	$\dots$	$+a_{1,n-m}x_{n-m}$	$+1s_1$	$+0s_2$	$\dots$	$+0s_m$	$= b_1$
	$a_{2,1}x_1$	$+a_{2,2}x_2$	$\dots$	$+a_{2,n-m}x_{n-m}$	$+0s_1$	$+1s_2$	$\dots$	$+0s_m$	$= b_2$
	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\ddots$	$\dots$	
	$a_{m,1}x_1$	$+a_{m,2}x_2$	$\dots$	$+a_{m,n-m}x_{n-m}$	$+0s_1$	$+0s_2$	$\dots$	$+1s_m$	$= b_m$

ve eksiolmazlık kısıtları.

Burada $\vec{x'} = (x_1, \dots, x_{n-m})^T$ ve $\vec{s} = (s_1, \dots, s_m)^T$ girdileri yapısal karar ve gevşeklik değişkenleri olan sütun vektörleri; c_j skaleri j indisli ürünün bir biriminin satışından elde edilecek katma değeri, $b_i \geq 0$ skaleri i indisli kıt kaynağın mevcudunu, $a_{i,j}$ skalerleri de j indisli üründen bir birim üretmek için kullanılması/tüketilmesi gereken i indisli kıt kaynak miktarını gösterir katsayılar olmaktadır. $a_{i,j} < 0$ olması, j indisli üründen 1 birim üretildiğinde, i indisli kıt-kaynağın bir yan-ürün (ing. by-product) olarak açığa tam da bu miktarda çıkmakta (üretilmekte) olduğunun ifadesi olmaktadır. Yukarıda sözünü ettiğimiz "firma-içi dışsallıklar", sözkonusu, bu durumda olmaktadır.

Amaç fonksiyonunun vektörel ifadesi, $\vec{c} = (c_1, \dots, c_{n-m})$ ve $\vec{0} = (0, \dots, 0)$ satır vektörleri olmak üzere, max. $z = (\vec{c}, \vec{0})\binom{\vec{x'}}{\vec{s}} = \langle \vec{c}, \vec{x'} \rangle$ olmaktadır. z harfini, amaç fonksiyonunun aldığı değeri gösterir değişken yanı-sıra, ileride göreceğimiz z_k skalerini adlandırmak için de kullanacağız. Teamül budur ve bunların farklı şeyler oldukları unutulmamalıdır. Bu notasyon örtüşmesinin mantıksal dayanağı şu olsa gerektir: Amaç fonksiyonunun, t adet anahtarlama işleminden sonra alacağı değer, ileride matris-vektör cebiri ile bulacağımız gibi

$$z_t = \vec{c_B} B_t^{-1} \vec{b},$$

k bir sütun vektörü indisi olmak üzere aynı harfle göstereceğimiz diğer skalerin değeri de

$$z_k = \vec{c_B} B_t^{-1} \vec{a_k}$$

olmaktadır[3]. Bu iki "bilineer form" bir çarpanları hariç aynıdır ve $\vec{b}$ çarpanı yerine $\vec{a_k}$ ikame etme pratiği, karşımıza, bu tezin "Leksikografik Döngü Önleme Kuralı'nın eleştirisi" (s.31) bölümünde de çıkacaktır. Müşterek çarpanı, Bazaraa $\vec{c_B} B_t^{-1} = \vec{w}$ harfli ifadesi ile göstermektedir (s.202).

[3]Bazaraa, Formül (3.1) s.103 ve Formül (3.11) s.111

"Yapısal" kısıtları gösteren doğrusal denklem sisteminin matris denklemi de şöyle yazılmaktadır:

$$(N_0, I_m)\begin{pmatrix}\vec{x'}\\ \vec{s}\end{pmatrix} = \vec{b}$$

Girdileri $a_{i,j}$ olan bir matrisi A harfi ile göstermek gelenektir. Ancak biz bu harfi biraz farklı bir matrisi adlandırmak için kullanacağımızdan, N_0 dedik. Aşağıda anlaşılacağı üzere, bu N harfi de uygundur.

Yapısal kısıtları ele eksiolmazlık kısıtlarından ayırarak alan Simpleks algoritması benzer bir ayrımı $\vec{x'}$ ile gösterdiğimiz yapısal değişkenlerle $\vec{s}$ ile gösterdiğimiz gevşeklik değişkenleri arasında yapmaz. Dolayısı ile yapısal/gevşeklik ayrımı yapmadan bütün karar değişkenlerini aynı harf ile gösteririz: $x_{n-m+k} \equiv s_k \quad (k = 1, 2, \ldots, m)$ veya $\binom{\vec{x'}}{\vec{s}} \equiv \vec{x}$. Aynı mantıkla, Kronecker deltası kullanarak, $a_{i,n-m+k} = \delta_{i,k} \quad (i, k = 1, 2, \ldots, m)$ veya $(N_0, I_m) = A$ diye de yazılmaktadır. Bu basitleştirmelerden sonra yukarıdaki doğrusal denklem sisteminin matris ifadesi

$$A\vec{x} = \vec{b} \tag{2.1}$$

olmaktadır. Simpleks algoritması $m \times n$ ebatlarında A matrisinin $\vec{a_1}, \ldots, \vec{a_n}$ vektörleri ile göstereceğimiz sütunları arasında (yapısal/gevşeklik değil de) "taban vektörü" ve "taban-dışı vektör" ayrımı yapar. Bunların çarpıldıkları karar değişkenlerine de "temel değişken" ve "taban-dışı değişken" adları verilir. Bu ayrıma uygun olarak A matrisini algoritmanın her adımında, t indisi atılmış Simpleks adımı (Anahtarlama) sayısını göstermek üzere, N_t ve B_t ile göstereceğimiz

$$A = (N_0, B_0) = (N_1, B_1) = \ldots = (N_t, B_t) = \ldots$$

iki partisyona ayırır[4]. N'ler $m \times (n-m)$ ebatlarında, B'ler de $m \times m$ ebatlarında alt-matrislerdir. A matrisinin $\vec{a_j}$ ile gösterdiğimiz sütunları bu iki alt-matristen ya birinin ya da ötekinin sütunları olmaktadır. x_j karar-değişkeni için, eğer çarpanı olduğu $\vec{a_j}$ vektörü N_t alt-matrisinin sütunu ise "taban-dışı", B_t alt-matrisinin sütunu ise de "tabanda" diyeceğiz. B_0, yukarıda gördüğümüz üzere, I_m birim matrisine eşittir. Simpleks tablosunda bu partisyonlar böyle yan-yana değil, fakat iç-içedirler. Vektör ve matris harfli ifade yazma kolaylıkları adına,

[4]Dantzig bizim B_t dediğimiz alt-matrise semboller listesinde B^t, N_t dediğimize de kısaca N demiş, bu notasyonu ilk kez kullandığı sayfada da $B = B^t$ ve $N = N^t$ yazmıştır [bkz: Dantzig & Thapa 1, s.xxxviii, xxxvii, 90].

sütunların gerektiğinde yeniden sıralanmaları[5] ile, yukarıdaki gibi yanyana yazıldıklarını varsayıyoruz. Bu kolaylık bazı düzenlilikleri görmemize engel de olabilmektedir. Mesela sözünü yukarıda ettiğimiz $z_k = \vec{c_B} B_t^{-1} \vec{a_k}$ harfli ifadesinde $\vec{c_B}$ satır vektörünün girdileri ve B_t^{-1} karematrisinin satırları böyle yeniden sıralanmışlardır. İlk bakışta bir "bilineer form"a benzeyen bu ifadeyi, bu tezin "Dantzig Kuralı ve eleştirisi" (s.16) başlığı altında, içerdiği düzenliliği açığa çıkartacak şekilde yeniden yazacağız. Vektör-matris cebiri harfli ifadeleriyle konu anlatımı yapan yazarların sıkça düştükleri bir yanılgı da, cari taban (kare) matrisinin, matris çarpımı işlemine göre, hep tersinir (ing. invertible) olacağını varsayar olmalarıdır. Elbette $B_0 = I_m$ ilk-tabanının rankı tamdır. Bu gerçekten yola çıkarak, Simpleks algoritmasının rankı koruyacağını ispata kalkan yazarlar vardır [bkz: Lee, Lemma 4.8, s.38]. Yapılan ispat yanlıştır: "cari taban" matrisi tersi ile "müteakip taban" matrisinin[6] çarpımı o denli basit bir sonuç vermez!

Girdileri karar değişkenlerimiz olan $\vec{x}$ vektörünü de, bu taban/tabandışı ayrımına uygun olarak,

$$\vec{x} = \begin{pmatrix} \vec{x_N} \\ \vec{x_B} \end{pmatrix}$$

diye iki kısma ayırırız. $t = 0$ adımında $\vec{x_N} = \vec{x'}$ ve $\vec{x_B} = \vec{s}$ olduğunu anımsıyoruz.

$A\vec{x} = \vec{b}$ denklem sistemi böylelikle

$$(N_t, B_t) \begin{pmatrix} \vec{x_N} \\ \vec{x_B} \end{pmatrix} = N_t \vec{x_N} + B_t \vec{x_B} = \vec{b} \tag{2.2}$$

olmaktadır.

Şimdi, belli bir $t > 0$ adımında, `B`$= B_t$ ve `b`$= \vec{b}$ atamalarını yaptıktan sonra, karar değişkenlerinin alacakları değerleri `x=B^(-1)*b`, $\vec{x_B} =$ `x`, $\vec{x_N} = \vec{0}$ diye hesaplayabiliriz[7]. Bunun bir çözüm olduğunun sağlaması da

$$A\vec{x} = N_t \vec{x_N} + B_t \vec{x_B} = N_t \vec{0} + B_t (B_t^{-1} \vec{b}) = \vec{0} + (B_t B_t^{-1}) \vec{b} = I_m \vec{b} = \vec{b}$$

[5] Yeniden sıralamanın sözel ifadesi "Temel değişkenlerin sıralanmış bir kümesi" (ing. an ordered set of basic variables) olmaktadır [bkz: Dantzig & Thapa 1, s.67]. Amaç, Tablo'nun temel değişkenlere karşı gelen sütunlarını yeni bir sıraya, I_m birim matrisi oluşturacak şekilde sokmaktır.

[6] Terminoloji için bkz: s.15

[7] Göksel, s.116

diye yapılır.

Ne var ki, bu yöntem bilgisayarın mikro-işlemci, ön-bellek, arka-bellek gibi kendine özgü kıt kaynaklarını Simpleks algoritması kadar ekonomik kullanmaz ve dev A matrislerinin sözkonusu olduğu gerçek yaşam uygulamaları için uygun olmayabilir.

Simpleks algoritmasının orta-öğretim düzeyinde cebir ve doğrusal cebir eğitimine kıyasla bir soyutlama boyutu vardır. Katsayılar matrisini (ing. "Tableau of Detached Coefficients for ..."', bkz: Dantzig & Thapa 1, s.8) değişkenler vektöründen ayırır ve bunlardan birincisi üzerine odaklanır. Bu soyutlama matris çarpımı işleminin birleşme özelliği ile mümkün olmaktadır. Aşağıda da sıkça yazacağımız

$$B_t^{-1}(A\vec{x}) = (B_t^{-1}A)\vec{x}$$

eşitliğinin sağ tarafında, parantez içinde, girdileri sadece (çarpıldıkları $\vec{x}$ değişkenlerden ayrışmış) skalerler olan bir matris vardır. $t > 0$ yani $B_t \neq B_0 = I_m$ olduğunda, bu skalerlere "güncellenmiş", B_t^{-1} kare matrislerine de "güncelleme çarpanı" diyeceğiz. $A\vec{x} = \vec{b}$ temel eşitliğinin "RHS" dediğimiz sağ-tarafının güncellenmesi de böylelikle $B_t^{-1}\vec{b}$ olmaktadır. Bir $\vec{a_j} = (a_{1,j}, \ldots, a_{m,j})^T$ sütun vektörünün güncellenmişine teamüllere uygun olarak $\vec{a'_j}$, $a_{i,j}$ girdisinin güncellenmişine $\bar{a}_{i,j}$, $(\vec{a_1}, \ldots, \vec{a_n}) = A$ matrisinin güncellenmişine de A' diyeceğiz. Bu mutabakata göre İlk Tablo'da (Sıfırıncı Tablo) veya bir Simpleks adımı (Anahtarlama) atılmadan önceki cari Tablo'da c_j, $a_{i,j}$ ve b_i harfleri ile gösterdiğimiz skalerlerin yerlerini müteakip tabloda alacak olanları $\bar{c}_j$, $\bar{a}_{i,j}$ ve $\bar{b}_i$ harfli ifadeleri ile gösteririz. Karar değişkenleri ise, adı-üzerinde "değişken" olduklarından, her adımda başka değer (ki bunlar, $t = 0$ olduğunda doğallıkla b_i, $t > 0$ olduğunda da, aşağıda göreceğimiz gibi, o adımdaki $\bar{b}_i$ skalerleri olmaktadır) alıyor olsalar da, hep aynı harf ile gösterilirler. "RHS" dediğimiz $\vec{b}$ vektörünün girdilerinin güncellemelerini, bizim de yaptığımız gibi, $\bar{b}_i$ ile gösteren Bazaraa, A matrisinin $\vec{a_j}$ sütun vektörlerinin güncellemelerine kimi yerde $\vec{y_j}$ (s.89, 103), kimi yerde $\vec{a'_j}$ (s.298), kimi yerde de $\vec{y'_j}$ (s.299) demiştir. Bu harfli-ifade enflasyonu hakkında, bu tezin "Simpleks Algoritması" başlığı altında, söyleyeceklerimiz olacaktır. Bunun nereden icap ettiğini, daha açık anlatımla hangi notasyonun nerede kullanılmasının daha uygun olduğunu anlamak mümkün değildir. Vektörleri (ve matrisleri) geniş-punto (ing. boldface) harflerle gösterme yolunu seçtiğin göre, hepsine birden, a'sı geniş olmak üzere, $\bar{a}_j$, girdilerine de $\bar{a}_{i,j}$ diyebilirdi. Matris-vektör cebiri ile yapmış olduğu konu anlatımında, "güncelleme çarpanı" dediğimiz B_t^{-1}

matrisi B_{t-1}^{-1}'den türetilmediğine göre, Cari Tablo da bir öncekinden değil fakat hep Sıfırıncı Tablo'ya dönerek, deyim yerinde ise "sil baştan" türetilmektedir. Bu nedenle gündemimizde sadece Sıfırıncı ve Cari tablolar vardır. Tefrik etmemiz gereken sadece bu ikisi ve birbirine karşı gelen girdileridir.

Yukarıda yaptığımız $\vec{x_N} = \vec{0}$ varsayımını görmezden gelerek (2) eşitliklerini "güncelleme çarpanı" B_t^{-1} ile (soldan) çarpar ve

$$B_t^{-1}A\vec{x} = B_t^{-1}N_t\vec{x_N} + I_m\vec{x_B} = B_t^{-1}\vec{b}$$

genellemesine gideriz. İşte, sıfırıncı satırı hariç tutulacak olursa, "Simpleks Tablosu" diye buna denmektedir. Değişkenlerden soyutlandığında $B_t^{-1}A$, "sağ tarafı" ile birlikte de $(B_t^{-1}A, B_t^{-1}\vec{b})$ matrisine.. Dikkat edilecek olursa birim matris I_m "Temel değişkenlerin yeniden sıralanmaları" varsayımı uyarınca her tablonun bir alt-matrisidir:

$$(B_t^{-1}A, B_t^{-1}\vec{b}) = (B_t^{-1}(N_t, B_t), B_t^{-1}\vec{b}) =$$
$$= ((B_t^{-1}N_t, B_t^{-1}B_t), B_t^{-1}\vec{b}) = ((B_t^{-1}N_t, I), B_t^{-1}\vec{b}).$$

Bu son ifadede A matrisi yerine, (N_t, B_t) değil de, $(N_0, B_0) = (N_0, I)$ yazmaya hakkımız vardır. Böyle yapacak olursak,

$$(B_t^{-1}A, B_t^{-1}\vec{b}) = (B_t^{-1}(N_0, I), B_t^{-1}\vec{b}) = ((B_t^{-1}N_0, B_t^{-1}), B_t^{-1}\vec{b})$$

buluruz. Bu cari taban B_t'nin matris çarpımına göre tersi B_t^{-1} matrisinin cari tabloda ve her adımda bir alt-matris olarak açık-seçik görülebilir olduğunun kanıtı olmaktadır ki, Simpleks'in uygulamacılara sağladığı büyük bir kolaylıktır [bkz: Bazaraa, s.132-133]. Bu kolaylığı "Temel değişkenlerin yeniden sıralanmaları" varsayımına borçlu olmadığımızı da kaydedelim.

Her tablonun I_m ve B_t^{-1} matrislerini birer alt-matris olarak içereceğini böylelikle göstermiş olduk. Sıfırıncı adımda $(t = 0)$ özdeş olan bu iki alt-matris t arttıkça ayrışacaklar, ancak hiç değilse ilk m simpleks adımı boyunca, karşımıza iç-içe geçmiş olarak çıkacaklardır. Yani bazı sütunları müşterek olacaktır. Uygulamalarda bu inceliğe dikkat etmek yarar sağlayacaktır.

(2) eşitliklerinin güncellemesi

$$B_t^{-1}(N_t, B_t)\begin{pmatrix}\vec{x_N}\\ \vec{x_B}\end{pmatrix} = B_t^{-1}N_t\vec{x_N} + I_m\vec{x_B} = B_t^{-1}\vec{b}$$

olduğuna göre,

$$\vec{x_B} = I_m \vec{x_B} = B_t^{-1}\vec{b} - B_t^{-1} N_t \vec{x_N},$$

$\vec{x_N} = \vec{0}$ alındığında da, tabandaki karar değişkenlerinin o adımda alacakları cari değerler

$$\vec{x_B} = B_t^{-1}\vec{b} = (\bar{b}_1, \ldots, \bar{b}_m)^T$$

olmaktadır. $I_m = (\vec{e_1}, \ldots, \vec{e_m})$ ve $\vec{n_i} = \vec{e_i}^T B_t^{-1}$ alarak, yukarıda sözünü ettiğimiz (değişkenlerden soyutlanmış) skalerleri güncelleyen formülleri

$$\bar{a}_{i,j} = \langle \vec{n_i}, \vec{a_j} \rangle \qquad (\vec{a_j} \in N_t) \tag{2.3a}$$

$$\bar{b}_i = \langle \vec{n_i}, \vec{b} \rangle \tag{2.3b}$$

diye bulmuş oluyoruz[8]. Bu $\vec{n_i}$ satır vektörleri, karşımıza, bu tezin "Anahtarlama Geometrisi ve Alt-Taban" bölümünde de çıkacaklardır (s.18). $\vec{a_j} \in B_t$ olduğunda, $B_t^{-1} B_t = I_m = (e_1, \ldots, e_m)$ olduğundan, $\bar{a}_{i,j} = \delta_{i,j}$ olduğu bellidir. Sıfırıncı satır skalerlerinin güncellenmeleri bundan biraz daha uzun olacaktır. Amaç fonksiyonunun "Kiralama yok" varsayımına uygun $(\vec{c}, \vec{0})$ çarpanlar satır-vektörünü $A = (N_t, B_t)$ ve $\vec{x} = \binom{\vec{x_N}}{\vec{x_B}}$ partisyonlamalarına uygun olarak $(\vec{c}, \vec{0}) = (\vec{c_N}, \vec{c_B})$ diye ayırırsak, $\vec{x_N} = \vec{0}$ olması gerekmediğinde alacağı değeri

$$z = (\vec{c_N}, \vec{c_B}) \binom{\vec{x_N}}{\vec{x_B}} = \vec{c_N}\vec{x_N} + \vec{c_B}\vec{x_B} =$$

$$= \vec{c_N}\vec{x_N} + \vec{c_B}(B_t^{-1}\vec{b} - B_t^{-1} N_t \vec{x_N}) = \vec{c_B} B_t^{-1}\vec{b} + (\vec{c_N} - \vec{c_B} B_t^{-1} N_t)\vec{x_N} =$$

$$= \langle \vec{c_B}, \vec{x_B} \rangle + (\vec{c_N} - \vec{c_B} B_t^{-1} N_t)\vec{x_N} = z_0 + (\vec{c_N} - \vec{c_B} B_t^{-1} N_t)\vec{x_N} + \vec{0} I_m \vec{x_B}$$

diye hesaplarız. Her simpleks tablosu yeni bir Doğrusal Programlama probleminin sıfırıncı tablosu olarak düşünülecek olursa, farkeden sadece amaç fonksiyonu (sıfırıncı satır) olacaktır. Amaç da, artık, amaç-fonksiyonunun z ile gösterdiğimiz cari değerinde bundan-böyle kaydedilecek artış $(z - z_0)$ niceliğini maksimize etmek olacaktır. Simpleks tablosunun sıfırıncı satırı skalerlerinin güncellenmeleri formülü böylelikle:

$$\bar{c}_j = c_j - \vec{c_B} B_t^{-1} \vec{a_j} = c_j - z_j \qquad (\vec{a_j} \in N_t) \tag{2.4}$$

olmaktadır [bkz: Gürel, s.50]. Literatürde girdileri bunlar olan satır vektörüne "indirgenmiş maliyetler vektörü" (ing. reduced costs vector)

[8] bkz: Formül (3.60), Dantzig & Thapa 1, s.90

denmektedir. Satır girdilerinin yukarıda söz ettiğimiz "Temel değişkenleri sıralanmış küme" mutabakatına göre yeniden sıralanmış oldukları unutulmamalıdır. Sözünü o bağlamda ettiğimiz $z_k = \vec{c_B} B_t^{-1} \vec{a_k}$ skalerleri de burada bulduklarımız olmaktadır. $\vec{a_j} \in B_t$ ise, $B_t^{-1}\vec{a_j} = \vec{e_j}$ olacağından, $\bar{c}_j = c_j - \vec{c_B}\vec{e_j}$ yukarıda gördüğümüz gibi hep sıfırdır. Bunun "Kiralama Yok Varsayımı" demek olduğunu anımsıyoruz.

2.1 SİMPLEKS ALGORİTMASI

Simpleks Algoritması'nı anlatmadan önce, şu önemli saptamayı yapmakta yarar vardır: algoritmanın yukarıda matris-vektör cebiri ile bulduğumuz revize skalerleri gerçekten de verir olduğu ancak "tecrübe ile sabit"tir! Bunun matematik anlamda ispata muhtaç bir husus olduğunun kimse farkında değil gibidir. "Simpleks Algoristması'nın Temel Teoremi" adı ile anılmayı hak eden böyle bir teoremin ispatını matematikçilere bırakıyoruz. Algoritmanın en önemli bileşeni "anahtarlama işlemi" şu olmaktadır: Cari tablo $(m+1) \times (n+1)$ ebatlarında T matrisinin $i_0 > 0$ satırı ve $j_0 < (m+1)$ sütununun kesiştikleri t_{i_0,j_0} girdisi "anahtar" olarak seçildiğinde, anahtarlama rutini, Matlab notasyonu ile, bu yazılım sıfır indis kabul etmediği için `i` $= i+1$, `i0` $= i_0+1$, `j` $= j, \ldots$ alarak yazacak olursak

```
function T = pivot(i0, j0, T)
 p = T(i0, j0)
 T(i0, :) = T(i0, :) / p
 for i=1, i0-1
    pi = T(i, j0)
    T(i, :) = -T(i0, :)*pi + T(i, :)
 end
 for i=i0+1, m+1
    pi = T(i, j0)
    T(i, :) = -T(i0, :)*pi + T(i, :)
 end
```

elemanter satır-işlemlerini[9] yapar. Dikkat edilecek olursa, bu işlemler

[9]Bir matrisin bir satır vektörünü belli bir skaler ile çarptıktan sonra ikinci bir satırı ile toplamak ve bulunan vektörü bu iki vektörden ikincisinin yerine yazmak (birincisi aynı kalır) bir elemanter satır operasyonudur [bkz: Dantzig & Thapa 1, s.346]. Bu teknik ile, bir sütunu (ki bu, Anahtar'ın üzerinde olduğu sütundur), "birinci satır"daki girdisi (ki bu da, Anahtar'ın ta kendisidir) haricinde "banko sıfır" yapmak mümkündür. Yukarıda kodladığımız algoritma tam da böyle yapmaktadır. MatLab yazılımı kullanarak bir `A` matrisinin mesela 2. ve 3. satır vektörlerini toplamak için `A(2, :`

sonucu $I_m = B_t^{-1}B_t$ alt-matrisinin 1 girdisi i_0 satırına denk gelen sütunu (ki buna "`j1` $= k(i_0)$ indisli sütun" diyelim) müteakip tablonun $B_{t+1}^{-1}N_{t+1}$ alt-matrisi sütunları arasına, buna mukabil $B_t^{-1}N_t$ alt-matrisinin j_0 indisli sütunu da $I_m = B_{t+1}^{-1}B_{t+1}$ birim matrisi sütunları arasına alınmaktadır. Okur kendini buna bir sayısal örneği inceleyerek inandırmalıdır. Sözünü ettiğimiz iki I_m matrisi soyut olarak aynı olmakla birlikte tablo üzerindeki tertipleri (ing. layout) farklıdır. `j0` indisli sütuna "giren vektör", çarpıldığı karar değişkenine de "giren değişken" (ing. entering column, entering variable) denmektedir. Benzer şekilde de `j1` indisli sütun ve çarpıldığı karar değişkeni "çıkan" (ing. departing or leaving) olmaktadır.

Matris-vektör cebiri ile yapılan konu anlatımlarında tefrik etmemiz gereken sadece sıfırıncı ve cari tablolardı. Simpleks hesabı, cari tabloyu, öyle sıfırıncı değil, fakat bir öncekinden türetir olduğundan, bu ayrım yerine "cari tablo - müteakip tablo ayrımı"nı, notasyonumuzu aynen muhafaza ederek ikame edebiliriz. Her cari tabloya "yeni" (aslında aynı) bir Doğrusal Programlama probleminin sıfırıncı tablosu nazarı ile bakılabileceğini yukarıda göstermiş olduğumuza göre, yaptığımız bu notasyon ekonomisi sorun çıkartmayacaktır. Açık anlatımla c_j, $a_{i,j}$ ve b_i harfli-ifadeleri ile, bundan böyle, cari tablonun; $\bar{c}_j$, $\bar{a}_{i,j}$ ve $\bar{b}_i$ harfli-ifadeleri ile de müteakip tablonun skalerlerini göstereceğiz. Yukarıdaki T matrisini de, bu mutabakat uyarınca

$$T = \begin{pmatrix} \vec{c} & z_0 \\ A & \vec{b} \end{pmatrix}$$

diye yazacağız. Okuyucu, MatLab kodlamasını yukarıda verdiğimiz algoritmanın, bu T simpleks tablosunun $t_{i,j}$ girdilerini tam da

$$\bar{t}_{i,j} = t_{i,j}/t_{i_0,j_0} \qquad (i = i_0) \tag{2.5a}$$

$$\bar{t}_{i,j} = (t_{i,j}t_{i_0,j_0} - t_{i,j_0}t_{i_0,j})/t_{i_0,j_0} \qquad (i \neq i_0) \tag{2.5b}$$

formülüne uygun olarak güncellediğinden emin olmalıdır. (5b) formülünü, içerisine (5a)'yı sokarak

$$\bar{t}_{i,j} = t_{i,j} - t_{i,j_0}\bar{t}_{i_0,j} \tag{2.6}$$

diye yazan yazarlar da vardır [bkz: Llewellyn, Formül (5.6), s.117]. Böyle hibrit formül yazmak zorunda biz de kalacağız. "Anahtar" t_{i_0,j_0} sıfırıncı satır ve RHS sütunu üzerinde olamayacağına göre, inceleyeceğimiz üç

`) + A(3, : )`, ve yine 2. ve 3. sütun vektörlerini toplamak için de `A( : , 2) + A( : , 3)` kodlarız. Dantzig bu notasyonu sadece `:` yerine • yazarak aynen benimsemiştir [bkz: ibid., s.xxxviii].

şıkkın üçünde de, $t_{i_0,j_0} = a_{i_0,j_0}$ olmaktadır. Bununla birlikte (i) $t_{i,j} = a_{i,j}$ alındığında $\bar{t}_{i,j} = \bar{a}_{i,j}$, (ii) $t_{i,j} = b_i$ alındığında $\bar{t}_{i,j} = \bar{b}_i$, (iii) $t_{i,j} = c_j$ alındığında da $\bar{t}_{i,j} = \bar{c}_j$ çıkmaktadır! Bu güncelleme formüllerinin yukarıda matris-vektör cebiri ile bulduğumuz (3a), (3b) ve (4) formüllerine denk oldukları, anlayabildiğimiz kadarı ile, ancak tecrübe ile sabittir. Dantzig & Thapa 1 kitabının "Appendix B: Linear Equations" bölümü, bu denklik ispatları aranarak okunduğunda, ispatın ilk iki şık için orada bulunduğu söylenebilir. Geriye c_j skalerlerini güncelleyen (4) formülünün denkliği kalmaktadır ki, bunun ispatını bulamadık.

Giren değişkenin şu veya bu olmasını gerektiren bir kural yoktur, ancak şu veya bu olmasını salık veren alternatif politikalar vardır. Bunlardan "Dantzig Kuralı" diye bilineni aşağıda açıklayacak ve eleştireceğiz. Giren değişken hernasılsa seçildikten sonra çıkacak değişkenin seçimi ise kesin kurallara tabidir. Giren değişkenin indisine yukarıda yaptığımız gibi j_0 diyecek olursak, çıkan değişken $a_{i,j_0} > 0$ tablo girdileri ile sınırlı olarak hesaplanacak $\frac{b_i}{a_{i,j_0}}$ kesiri en küçük çıkan i_0 indisli satıra karşı gelen $j_1 = k(i_0)$ indisli temel değişken olarak saptanır. Burada gözetilen eksiolmazlıktır. Anahtarlama işlemi yapıldıktan sonra bütün temel değişkenlere eksi olmayan değerler atanacağı güvence altına:

$$\frac{b_{i_0}}{a_{i_0,j_0}} \leq \frac{b_i}{a_{i,j_0}} \Rightarrow -\frac{b_{i_0}}{a_{i_0,j_0}} \geq -\frac{b_i}{a_{i,j_0}}$$

$$b_i - a_{i,j_0}\frac{b_{i_0}}{a_{i_0,j_0}} \geq b_i - a_{i,j_0}\frac{b_i}{a_{i,j_0}} = 0$$

diye alınır. Yaptığımız, ikinci eşitsizliğin her iki tarafını da pozitif a_{i,j_0} tablo-girdisi ile çarptıktan sonra yine iki tarafa b_i "sağ taraf" girdisini ilave etmekten ibarettir. İkinci satırın sol başındaki kemiyetin böylelikle b_i tablo girdisinin güncellenmişi olduğu, Llewellyn'in yukarıda zikrettiğimiz (6) formülü uyarınca açıktır[10].

2.2 Dantzig Kuralı ve eleştirisi

c_j skalerlerini güncelleyen (4) formülünü, orada B indisi ile temsil edilen "Temel değişkenlerin sıralanmış kümesi" varsayımına başvurmaksızın ve (3a) güncelleme formülünü içine Llewellyn'in yaptığı gibi sokarak

[10] $t_{i,j} = b_i$, $t_{i,j_0} = a_{i,j_0}$ ve (5a)'ya bakarak $\bar{t}_{i_0,j} = \frac{b_{i_0}}{a_{i_0,j_0}}$ alıyoruz.

$$\bar{c}_j = c_j - z_j = c_j - \sum_{i=1}^{m} c_{k(i)} \bar{a}_{i,j} \tag{2.7}$$

diye yazarız. Yukarıda da kullanmış olduğumuz $k(i)$ fonksiyonu, tabandan, anahtar i.nci satırda olsa idi, çıkacak karar değişkeninin indisini veren bir fonksiyondur. Bu (7) formülünün üstünlüğü, notasyon konusunda yaptığımız iki mutabakattan ikisine de uygun olmasıdır[11]. Açık anlatımla, c_j skalerleri sıfırıncı tablodan seçilseler de, cari tablodan seçilseler de, hep doğru çıkmaktadır! $\bar{a}_{i,j}$ skalerinin ise, birinci mutabakata göre cari tablo, ikincisine göre de müteakip tablo skalerleri olduklarını anımsıyoruz. Yani hep tek bir tablodan seçilmelidirler.

Dantzig Kuralı, tabana, bu $c_j - z_j$ sayılarından en büyük olanı ile çarpılan karar değişkeninin alınmasını salık verir. "Giren" sütun vektörü ve alt-taban normalinin vektör iç-çarpımı ile bulunan skaler "giren" vektör yeni tabana "1 ölçü" alındığında "çıkan"ın "kaç ölçü" çıkacağının sayısal ifadesi olmaktadır[12]. Anahtar'ın üzerinde olduğu tablo satırını, anahtar'a, anahtar girdisini 1 yapacak şekilde böler olmamızın nedeni bu ikinci ölçüyü 1 yapma zorunluluğudur. İlke olarak, sütun vektörleri tabandan 1 ölçüde çıkar ve oraya yine 1 ölçüde girerler. Dantzig Kuralı ise "bir taban-dışı vektör tabana 1 ölçü girerse, hangi taban vektörü oradan *kaç ölçü* çıkar" hesabı yapmaktadır [bkz: Bazaraa, s.111]. Ne var ki, bu "ölçü" 1'den büyük olduğunda eksiolmazlık kısıtları etken olacaktır ki, bunu bütünü ile gözardı etmektedir. Bu bakımdan, bir eleştirisini bu tezin bir Bölüm 8'i olarak verdiğimiz "En Küçük Açılar Anahtarlama Kuralı"ndan (s.42) daha isabetli değildir.

Dantzig Kuralı'nın, şayet varsa bir yararı, bu (4) ve onun daha anlaşılır özdeşi (7) formülüne dayanılarak savunulabilir. Ne var ki, sıfırıncı tablodaki c_j skalerleri veridirler, yani bu formülle hesaplanmış değillerdir! Kuralın, buna karşın uygulanabilir (ve uygulanmakta) olması "Kiralama Yok Varsayımı" sayesinde mümkün olmaktadır. Şöyle ki, sıfırıncı tablodan hareketle atılacak birinci Simpleks adımında, tabana isteristemez bir yapısal karar-değişkeni ile çarpılan sütun vektörü girecek, buna mukabil oradan bir gevşeklik değişkeni ile çarpılan sütun çıkacaktır. Yapısal değişken ile çarpılan vektör tabana 1 ölçü girdiğinde, gevşeklik değişkeni ile çarpılan kaç ölçü çıkarsa çıksın, amaç fonksiyonu üze-

[11] Elbette bu (4) formülünün üstünlüğü de olmaktadır. Ne var ki, bu formül, bu özelliği kolay denetlenebilir bir açıklıkta yazılmamıştır.

[12] Dikkat edilecek olursa, bu skaler, cari tabloda, karşımıza $a_{i,j}$ girdisi olarak çıkmaktadır [bkz: Bazaraa, s.131]

rinde yapacağı kısmi değişiklik, Varsayım uyarınca hep sıfır olacaktır. Kukla değişkenleri (ing. dummy variables) Varsayım'a geçerlilik kazandırma maksadı ile kullanma pratiği, bilgisayar hafızasının ekonomik kullanımını gözetmemektedir. $B_0 = I_m$ tabanını oluşturmak için, gevşeklik değişkenleri, Varsayım geçerli olmasa da, yeterli olabilirler. Yapılması gereken $\bar{c}_j = c_j - z_j$ eşitliğini unutmak ve $c_j - z_j$ skalerlerini (7) formülü ile hesaplamaktan ibarettir.

2.3 ANAHTARLAMANIN GEOMETRİSİ VE ALT-TABAN

Bu çalışmada Doğrusal Programlama'nın simpleks algoritmasını ele bir "adım-adım taban değiştirme algoritması" olarak alacağız. "Computational" doğrusal cebir kitaplarının fihristlerinde (1) "taban değiştirme" ve (2) "adım-adım yöntemler" (ing. iterative methods) başlıklarını bulmak mümkün olsa da bu ikisinin arakesiti hep boş çıkmaktadır. Yani konuya böyle yaklaşan kitap bulunamamıştır.

Sütun uzayına yeni bir öge eklendiğinde satır uzayına da yeni bir boyut eklemek ancak ve yalnız yeni öge mevcut ögelerin (sütunların) gerdikleri alt-uzayın dışında kalıyorsa gerekir. Bu "Bir matrisin satır rankı sütun rankına daima eşittir" diyen kanun gereği böyledir.

Charnes ve Cooper satır uzayının boyutundan ekonomi yapmak için bu kanuna başvurmamışlar ve boyut sayısını hesapsızca arttırmışlardır. Bu savurganlıklarının bedelini de sayısal örneklerini 2×2 ebatları ile sınırlandırmak zorunluluğu cinsinden ödemişlerdir.

Evet, bir gevşeklik değişkenine karşı gelen $(0, \ldots, 0, 1, 0, \ldots, 0)^T$ sütununun yapısal değişkenlere karşı gelen sütunların gerdikleri alt-uzayın dışında kalması olasıdır. Ancak sütun uzayını kurmağa yapısal değişkenlerle başlamak zorunluluğu yoktur. Gevşeklik değişkenlerine karşı gelen sütunlar, dikkat edilecek olursa, R^m uzayının standart tabanını oluşturmaktadırlar. Bu standart taban ögelerinden biri yerine bir yapısal sütun vektörü koymakla sütun uzayının boyutunun büyümeyeceği çok açıktır. Yukarıda zikrettiğimiz kanun uyarınca, satır uzayının boyutunun da.. Bu ikincisini, yapısal kısıt sayısı m ile sınırlı gibi görmek böylelikle mümkün olmaktadır.

Satır uzayı boyutundan yaptığımız bu ekonomi sayesinde, bu çalışmada 3×3 ebatlarında sayısal örnekler verebileceğiz.

Doğrusal cebirdeki vektör uzayı tabanı kavramı ile Doğrusal Programlama'daki taban kavramı büyük ölçüde örtüşürler. Bunlardan ikincisi, birincisinin ötesinde şunu da içerir: bir de "sağ taraf" vektörü $\vec{b}$ vardır ve bunun hangi tabanın ögelerinin bir doğrusal bileşimi olarak ifade edileceği sorusuna yanıt bulmak algoritmanın başlıca amacıdır.

Önce taban değiştirmeye bakalım, adım-adım taban değiştirmeyi oradan öğreneceğiz.

Genel-geçer notasyonla "$max.\langle \vec{c}, \vec{x} \rangle$ s.t. $A\vec{x} = \vec{b}$ ve eksiolmazlık kısıtları" Doğrusal Programlama problemini ele alacağız.

Simpleks algoritmasının başlangıç tablosunda (ing. initial tableau) taban standart $\{(1, 0, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, 0, \ldots, 1)\}$ tabanıdır. Buna şimdilik "eski taban" diyelim. $\vec{b}$ bu tabana göre yazılmış bir m boyutlu vektör, $m \times m$ ebatlarında B matrisinin sütunları da bir yeni tabanın ögeleri olsun. Bu ögelerin B'den yapısal karar değişkeni sayısı kadar fazla sütunu olan bir A matrisinin sütunları arasından seçildiğini düşünüyoruz. Değişken-skaler ayrımını unutarak

$$\vec{b} = I_m \cdot \vec{b} = (B \cdot B^{-1})\vec{b} = B \cdot (B^{-1}\vec{b}) = (\vec{a_{i_1}}, \ldots, \vec{a_{i_m}}) \cdot \vec{x_B} \qquad (2.8)$$

diye yazalım. Son ifade $m \times m$ ebatlarında bir matris ile yine m boyutlu bir sütun vektörünün çarpımıdır. Vektörün girdileri, $\vec{b}$ vektörünün yeni tabanın ögelerinin bir doğrusal bileşimi olarak ifade, hangi çarpanlar kullanılarak edilebileceği sorusunun yanıtları olmaktadır. Bu çarpanlardan her birinin, B^{-1} matrisinin belli bir satırının $\vec{b}$ sütun vektörü ile iç-çarpımı sonucu olduğu âşikardır. Sıfır değeri alması gerekmeyen karar değişkenlerini girdileri bunlar olan $(x_{i_1}, \ldots, x_{i_m})^T = \vec{x_B}$ sütun vektörü ile gösterdiğimizi anımsayacak olursak, $B^{-1}\vec{b} = \vec{x_B}$ olduğu da.. Şimdi B^{-1} matrisinin yukarıda $\vec{n_i}$ ile gösterdiğimiz (s.12) satırlarının neleri ifade ettiklerine bakalım. $B^{-1} \cdot B = I_m$ olduğuna göre, sözkonusu satırların her biri, taban ögelerinin biri hâriç tutulduğunda, geriye kalan ögelerin gerdikleri $m - 1$ boyutlu alt-uzayın bir normalinin vektörel ifadesi olmaktadır. Bu $m - 1$ adet ögeye bir "alt taban" diyeceğiz. Normale de "B^{-1} matrisinin bizi (belli bir simpleks adımında) ilgilendiren satırı" diyeceğiz. Dışarıda-bırakılan-öge ile ona tekabül eden alt-tabanın gerdiği uzayın normalinin (yani "bizi ilgilendiren satır" oluyor bu) iç-çarpımının 1 çıkması olgusu, bir taban ögesinin genellikle ancak

kendi cinsinden ifade edilebileceği gerçeğinin gereğidir. Buna, kendimizi, (8) ifadesinde, $\vec{b}$ yerine dışarıda-bırakılan-ögeyi yazarak inandırabiliriz. Dışarıda-bırakılan-öge ve alt-tabanın-gerdiği-uzay konfigürasyonu üç boyutlu çizimlerle modellemeye elverişlidir ve karşımıza doğrusal cebir ve konveks politoplar konularınının bazılarında çıkmaktadır.

Simpleks algoritmasında alt-taban olarak tabandaki yerlerini bir simpleks adımı (ing. iteration) atıldıktan sonra da koruyacak ögeler kümesini alacağız. Bunları $a_{\vec{i_1}}, \ldots, a_{\vec{i_{m-1}}}$ ile gösterelim. $a_{\vec{i_m}}$ ile göstereceğimiz dışarıda-bırakılan-ögeden de tabandan çıkartılacak ögeyi (ing. departing variable) anlayacağız. Şimdi bu öge yerine A matrisinin eski tabanda olmayan $a_{\vec{i_{m+1}}}$ sütununu ikame ettiğimizi düşünelim – buna da "giren değişken" (ing. entering variable) denmektedir. Bu sütunun B^{-1} matrisinin bizi ilgilendiren satırı ile iç-çarpımı, dışarıda-bırakılan-ögenin iç-çarpımı ile aynı (yani 1) ise, B^{-1} matrisinin yeni tabana göre revizyonunda bizi ilgilendiren satır, binaenaleyh $\vec{b}$ vektörü ile iç-çarpımı aynen korunacaktır. Bizi ilgilendirmeyen satırları ise kaçınılmaz olarak değişeceklerdir. Dolayısı ile, $\vec{b}$'nin bir doğrusal bileşke olarak ifadesinde, alt-taban ögelerinin çarpanları sabit kalmayacaklar, ancak kısmî bileşkelerinin alt-tabanın gerdiği uzay içinde kalacağını bilmek bizim için yeterli olacaktır. (8) ifadesinin, geçerliliğini, B^{-1} matrisinin böyle revizyonunu müteakip de koruyacağı düşünüldüğünde, bu önemli bir buluştur:

Önkuram (ing. lemma): Giren ve çıkan ögelerin "bizi ilgilendiren satır" ile iç-çarpımları aynı ise, giren öge tabanda (daha doğrusu $\vec{b} = \ldots$ ifadesinde) çıkan ögenin çıkmadan önce çarpıldığı skaler ile çarpılacaktır.

Okuyucu, bu Önkuram'ın yukarıda yaptığımız sözel ispatının arkaplanında, matris çarpımı işleminin (a) "birleşme" ve (b) çarpım I_m olduğunda "değişme" özelliklerine defaatle başvurulduğuna dikkat etmelidir.

Şimdi de Önkuram'ın hipotezini "(...) ile iç-çarpımları bir skaler çarpan farkı ile aynı ise" diye revize edelim. Açıktır ki, bu takdirde bizi-ilgilendiren-satır da ancak "bir skaler çarpan farkı ile" korunacaktır. λ ile göstereceğimiz bu skaleri, yukarıdaki mülahazalardan ötürü, $\langle \lambda \vec{n_{i_0}}, a_{\vec{i_{m+1}}} \rangle = 1$ denklemini çözerek bulacağız. $(1/\lambda)$'in Simpleks tablosunun adına "anahtar" (ing. pivot) dediğimiz girdisi olduğuna dikkat edilmelidir.

Tabandan çıkartılacak değişken (ing. departing variable) belirleme Simpleks kıstasının ne göstereceğine bağlı olarak, B^{-1} matrisinin her satırının "bizi ilgilendiren satır" olabileceği aşikardır.

Bölüm 3

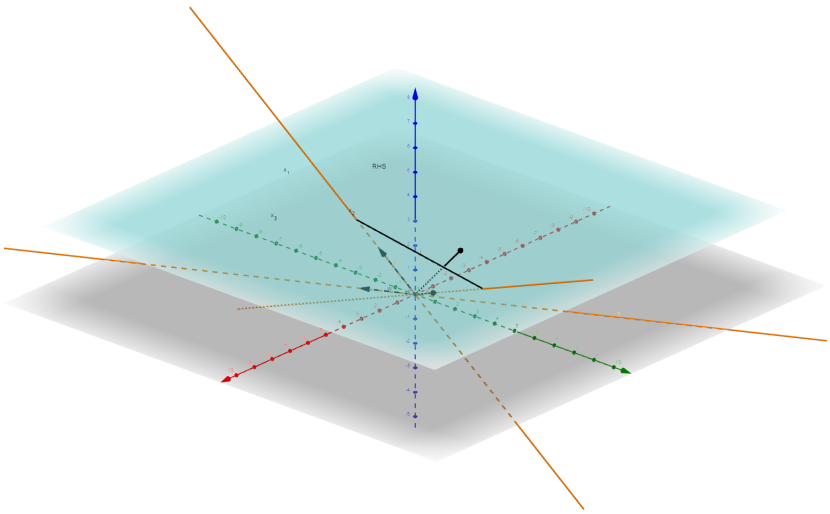
Örnek Uygulamalar

Örnek 1 (Bazaraa Example 5.2, s.210):

$max.$	x_1	$+2x_2$	$-x_3$	RHS
$ST:$	x_1	$+x_2$	$+x_3$	≤ 4
	$-x_1$	$+2x_2$	$-2x_3$	≤ 6
	$2x_1$	$+x_2$		≤ 5
ve	eksiolmazlık	kısıtları.		

Bu örnekte $(4, 6, 5)^T = \vec{b} = RHS$ vektörü, aşağıdaki ölçekli çizimde de açıkça görülebileceği gibi, sütun vektörlerinin gerdikleri teknolojik olanaklılık konisinin dışında kalmaktadır:

Şekil 3.1: Teknolojik olanaklılık konisi



Çünkü, sırası ile x_1 ve x_2 karar değişkenleri ile çarpılan $(1, -1, 2)^T$ ve $(1, 2, 1)^T$ sütunlarının gerdikleri yarıyüz x_3 ile çarpılan $(1, -2, 0)^T$ sütun

vektörü ile $\vec{b}$ vektörünü, yukarıdaki çizimde görüldüğü gibi ayırmaktadır. Bunu gösterelim: Sözkonusu yarıyüzün denklemi

$$\begin{vmatrix} x & y & z \\ 1 & -1 & 2 \\ 1 & 2 & 1 \end{vmatrix} = 0$$

dır. Determinantı açarsak $-5x + y + 3z = 0$ buluruz. $(x, y, z)^T = (1, -2, 0)^T$ ve $(4, 6, 5)^T$ değerleri verildiğinde, birbirine zıt işaretli iki sayı çıkmaktadır. Bir başka deyişle $\vec{b}$ vektörü, teknolojik olanaklılık konisinin bu yüzüne (ing. facet) göre koninin ötesine (ing. beyond) düşmektedir. Oysa koninin içinde olsa idi, her bir yüze göre berisinde (ing. beneath) kalması gerekirdi.

Bu saptama bizi üçüncü $(1, -2, 0)^T$ sütun vektötünün fazla (ing. redundant) olduğu sonucuna götürmekte ve teknolojik olanaklılığı

$$-5x + y + 3z = 0 \tag{3.1}$$

alt-uzayı ile sınırlı gibi düşünmemize izin vermektedir.

Bu alt-uzay sütun uzayında birbirlerini dik kesen ve koordinat eksenlerini ya dik kesen ya da (paralel olduğundan) hiç kesmeyen kısıt yarıyüzleri ile şu üç uç noktada kesişecektir [bkz: Demokrasi Prensibi bölümündeki çizimler, s.40, 41]:

$$\begin{array}{lcccc} A) & (& x_0, & 6, & 5 &)^T \\ B) & (& 4, & y_0, & 5 &)^T \\ C) & (& 4, & 6, & z_0 &)^T \end{array}$$

Gevşeklik değişkeni kullanmadan yapmakta olduğumuz bu çözümlemede $(4-x_0)$, $(6-y_0)$ ve $(5-z_0)$ sırası ile birinci, ikinci ve üçüncü kısıtların gevşeklik değişkenlerinin, A, B, ve C şıklarında, diğer ikisi sıfır olmak üzere aldıkları değerler olmaktadır ve eksi olmamaları gerekmektedir. Şimdi bu üç durumu ayrı ayrı çözümleyelim:

A) $-5(x_0) + (6) + 3(5) = 0$, $x_0 = \frac{21}{5} > \frac{20}{5} = 4$ ve $4 - x_0 < 0$ olduğundan, bu çözüm ekonomik bakımdan olanaksızdır.

B) $-5(4) + (y_0) + 3(5) = 0$, $y_0 = 5 < 6$ olduğundan bu çözüm olanaklılığın ekonomik koşulunu sağlamaktadır. Bir başka gerek koşul, alt-uzayı geren iki sütun vektörünün arasında olmasıdır. Şimdi de buna bakalım:

$$\mu^{-1}\begin{bmatrix}4\\5\\5\end{bmatrix} = \lambda\begin{bmatrix}1\\-1\\2\end{bmatrix} + (1-\lambda)\begin{bmatrix}1\\2\\1\end{bmatrix} = \begin{bmatrix}1\\2-3\lambda\\1+\lambda\end{bmatrix}$$

ise $\mu = 4$; $\frac{5}{4} = 2 - 3\lambda$, $\lambda = \frac{1}{4}$ olur. $0 \leq \lambda \leq 1$ çıktığından arasındadır. Sağlaması $\frac{5}{4} = 1+\frac{1}{4}$ olmaktadır. Böylelikle B çözümünün hem teknolojik hem de ekonomik olarak olanaklı olduğunu göstermiş olduk. Ne var ki C çözümü de öyledir ve amaç fonksiyonu onu (C çözümünü) yeğlememizi gerektirmektedir:

C) $-5(4) + (6) + 3(z_0) = 0$, $z_0 = \frac{20-6}{3} = \frac{14}{3} < \frac{15}{3} = 5$ olduğundan ekonomik olanaklılık vardır.

$$\mu^{-1}\begin{bmatrix}4\\6\\14/3\end{bmatrix} = \qquad \begin{bmatrix}1\\2-3\lambda\\1+\lambda\end{bmatrix}$$

ise $\mu = 4$; $\frac{6}{4} = 2 - 3\lambda$, $3\lambda = 2 - \frac{3}{2} = \frac{1}{2}$, $\lambda = \frac{1}{6}$. Sağlama: $\frac{14}{3\cdot 4} = 1 + \frac{1}{6}$ evet. Bulduğumuz $x_1 = \mu \cdot \lambda = 4 \cdot \frac{1}{6} = \frac{2}{3}$ ve $x_2 = \mu(1-\lambda) = 4(1-\frac{1}{6}) = \frac{20}{6} = \frac{10}{3}$ sonuçları M. Bazaraa'nın sayfa 212'de verdiği optimum sonuçların aynılarıdır.

Bölüm 4

KURAMSAL ÇERÇEVE (II)

Nelerin vektör, nelerin ise skaler (veya tamsayı) oldukları artık iyice anlaşılmış olması gerektiğinden, bundan böyle vektörleri üzerlerine mutlaka ok işareti koymadan da göstereceğiz.

4.1 YOZ ÇÖZÜM

'Düzlem" dediğimiz 2 boyutlu uzayda 2 doğrunun, 3 boyutlu uzayda 3 düzlemin ve $k > 3$ bir tamsayı olmak üzere k boyutlu uzayda da k tane $k-1$ boyutlu ara-yüzün (ing. hyperplane) kesişimi hep bir noktadır. Doğrusal Programlama'nın "temel çözümleri" kısıt arayüzlerinin böyle kesişimleri ile bulunan noktalardır. Normal olan, m tane yapısal kısıt ve $(n-m) \geq 1$ tane yapısal karar-değişkeni olan bir problemde, her çözümün, $(n-m)$ boyutlu uzayda yine bu $(n-m)$ sayıda $(n-m-1)$-arayüzün kesişimi olmasıdır. Eğer bir temel çözüde $(n-m)$'den fazla sayıda arayüz kesişiyorlarsa ve buna koşut olarak 1 veya daha çok sayıda gevşeklik değişkeni tabanda olmalarına karşın 0 değeri alıyorsa, bu temel çözüme "yoz" denmektedir.

Yukarıda verdiğimiz örnekte gördüğümüz ve iyice anladığımız A, B ve C şıklarının örtüşmeleri ortaya böyle yoz çözümler (ing. degenerate solutions) çıkartır.

$$A = B = C$$

eşitliklerinden ikisi sağlandığında üçüncüsünün de kendiliğinden sağlanacağı gösterilebilir.

Verdiğimiz örnekte $R\vec{H}S$ vektörünü $R\vec{H}S = (21/5, 6, 5)^T$, $RHS = (4, 5, 5)^T$ veya $RHS = (4, 6, 14/3)^T$ olarak değiştirirsek, üç yoz problem buluruz. Bunlardan birincisinin standart simpleks algoritması ile çözümü "yoz problem" bildirimi ile şu adımda (ing. iteration) stop etmektedir:

x_1	x_2	x_3	s_1	s_2	s_3	RHS
0	0	-0.2	0	0.6	0.8	7.6
0	0	1.4	1	-0.2	-0.6	0
0	1	-0.8	0	0.4	0.2	3.4
1	0	0.4	0	-0.2	0.4	0.8

Bu çözüme "optimum" demek zordur. Zira sıfırıncı satırda eksi işaretli katsayı vardır. Taban-dışı x_3 değişkeninin tabana alınması amaç fonksiyonunu ilerleteceğe benzemektedir.

Yukarıdaki (3.1) denkleminin katsayıları simpleks tablosunun birinci satırında karşımıza "bir skaler çarpan farkı ile" çıkmaktadır ki, bu skaler -5'tir. Bu katsayılardan hareketle üçüncü kısıtın 3 katını ikinci kısıta eklersek

$$5x_1 + 5x_2 - 2x_3 \leq 21$$

buluruz. Bunu da birinci kısıta

$$x_1 + x_2 + x_3 \leq 21/5 + \frac{7}{5}x_3$$

şeklinde benzetmek mümkün olmaktadır.

Bu kısıtın "cebirsel fazlalığı"ndan söz edilemez. Zira sağ tarafta x_3 değişkeninin katsayısı pozitiftir.

Birinci kısıt geometrik anlamda fazladır ki, bu doktora tezinin yapmayı umduğu katkı budur.

Bir anahtarlama (ing. pivoting) işlemi sırasında (tabana) giren değişken (ing. entering variable) ile çıkan değişkenin (ing. departing variable) çarpıldıkları sütun vektörleri arasında bir ikame ilişkisi vardır ve bu ikame her iki sütun vektörünün "alt-taban normali" üzerindeki izdüşümlerinin *yöndeş* olması ile mümkündür. Mesela C şıkkında tabandaki $s_3 = +\frac{1}{3}$ değişkeni ile taban-dışı $x_3 = 0$ değişkeni arasında bir anahtarlama,

$$\langle(-5, 1, 3), (1, -2, 0)\rangle = -7$$

ve

$$\langle(-5,1,3),(0,0,1)\rangle = +3$$

olduğundan olanaksızdır.

Yoz probleme dönecek olursak, giren değişken olarak sıfırıncı satırdaki katsayısı eksi olan x_3 seçildiğinde, bunun $(1,-0.2,-0.6)^T$ veya $(-5,1,3)^T$ vektörü üzerindeki izdüşümü

$$\langle(-5,1,3),(1,-2,0)\rangle = -7$$

dir. Tabandaki $(1,-1,2)^T$, $(1,2,1)^T$ ve $(1,0,0)^T$ sütun vektörlerinden, yöndeşlik koşulunu ancak üçüncüsü sağlamaktadır (s_2 ve s_3 tabanda olsalar dahi aday olamazlardı, çünkü yöndeşlik koşulunu sağlamamaktadırlar). Ne var ki, buna tekabül eden s_1 değişkeninin değeri sıfırdır ve daha da kısalması eksiolmazlık kısıtları yüzünden mümkün değildir. Maamafih Simpleks algoritması s_1 değişkeninin tabana sıfır skaler çarpanla alınmasına engel olmaz ki, buna aşağıda "yoz anahtarlama" diyeceğiz (s.29).

4.2 TERMiNOLOJi

v bir vektör, λ da bir skaler ise $v\lambda$ çarpımına "v vektörünün λ şiddeti" diyeceğiz.

4.3 Temel Çözüm sayısını veren formül

Yaygın olarak bilinen $\binom{n}{m} = \frac{n!}{m!(n-m)!}$ formülünü n ile gösterilen tamsayının ne olduğunun takdirini okura bırakarak veren yazarlara diyecek yoktur. Ama n'yi m yapısal kısıt artı $(n-m)$ yapısal değişken eksiolmazlık kısıtı olarak aldıklarını açık edenler, gevşeklik değişkeni eksiolmazlık kısıtlarını düpedüz unutmaktadırlar. Şimdi formülün hiçbir kitapta bulamadığımız ispatını da verelim. $n > m$ veya $n+m$ boyutlu uzayda her m arayüzün birbirleri ile kesişimlerinin bir temel-çözüm değil fakat $n-m$ veya $(m+n)-m = n$ boyutlu bir "yüz" vereceği çok açıktır[1]. Öyleyse kombinezon hesabı neden yapmaktayız? Şundan: $k = 1, 2, \ldots$

[1] Grünbaum, Teorem 3.1.7, s.35; m yerine $k-h+1$, d yerine n veya $n+m$, k yerine de $d-1 = n-1$ veya $n+m-1$ yazdığımızda; $h = k-m+1$ de $(n-1)-m+1 = n-m$ veya $(n+m-1)-m+1 = n$ olmaktadır.

tamsayıları için, n-boyutlu uzayda $n+m$ adet $(n-1)$-yüzü olan bir politopu $n+k$ boyutlu uzayda yine aynı sayıda $(n+k-1)$-yüzü olan bir politopun bir n-kesiti olarak elde etmek mümkündür[2]. Bu süreçte dikkat edilecek olursa uzayın $n+k$ boyutunun birer birer artmasına karşın n-kesiti alınacak politopun $(n+k-1)$-yüz sayısı $n+m$ olarak sabit kalmaktadır ve belli bir boyutta en az sayıda yüzü olan politop bir simplekstir. Öyleyse k dizimiz $k=m-1$ olduğunda sonlanacak ve bir $(n+m-1)$-simpleksin n-kesitini alıyor olacağız. Bu n-kesitin $(n+m-1)$-simpleksin $(n+m-1)-n=(m-1)$-yüzleri ile kesişimi birer temel çözümdür ve $\binom{n}{m}$ veya $\binom{n+m}{m}$ ifadelerinin altlarında gördüğümüz m'nin açıklaması bu olmaktadır. Gerçekten de bir $(n+m-1)$-simpleksin bu kombinezon hesabı ile bulunacak sayıda $(m-1)$-yüzü vardır[3].

4.4 Kaynak kullanımını destekleme (KKD)

Türkiye'de tahsil olunan vergilerden biri de Kaynak Kullanımını Destekleme Fonu'na yapılan kesintilerdir. Bu fon "kıt kaynaklar ne kadar çok kullanılırlarsa, gerçekçi bir amaç fonksiyonu da o kadar maksimum değer alır" düşüncesine dayanır. Bu varsayım fantastik bir amaç fonksiyonu için doğru olmayabilir. Yeryüzünün kıt kaynaklarının çılgınca tüketilmekte olduklarına bakılırsa, çoğu firmanın amaç fonksiyonu kaynak kullanımını destekler nitelikte olmalıdır.

4.5 REVİZE SİMPLEKS TABLOSU

Her Simpleks Tablosu'nun I_m birim matrisini bir alt-matris olarak içerdiğini yukarıda göstermiştik. Onu, bilgisayarın (arka) belleğine bu sabit ile birlikte depolamak hem ekonomik değildir, hem de profesyonel bilgisayar programcısına yakışmaz. Dışarıda bırakıldığında ise, geriye (kuşkusuz B_t^{-1}'yi de kısmen içeren) $B_t^{-1}N_t$ kalacaktır (s.12). Anahtarlama rutinini (s.14) çalıştırmak için bir de $e_{k(i_0)}^{\rightarrow}$ vektörüne gereksinim olacaktır. Kısacası Revize Tablo, sıfırıncı satırı hariç tutulduğunda

$$(B_t^{-1}N_t, e_{k(i_0)}^{\rightarrow}, B_t^{-1}\vec{b}) = (B_t^{-1}N_t, B_t^{-1}a_{k(i_0)}^{\rightarrow}, B_t^{-1}\vec{b}) = B_t^{-1}(N_t, a_{k(i_0)}^{\rightarrow}, \vec{b})$$

diye yazılmalıdır.

[2]Grünbaum, Teorem 5.1.1 ispatı, s.71

[3]Grünbaum, s.133; d yerine $m+n-1$, k yerine de $d-m=n-1$ yazmaktayız.

KKD önyargısı Mokhtar S. Bazaraa'yı vahim yanılgılara sürüklemiştir. Klasik "Simpleks Tablosu" konusuna girdiği 126. sayfada verdiği harfli ifade

	z	x_B	x_N	RHS	
z	1	0	$c_B B^{-1} N - c_N$	$c_B B^{-1} b$	Sıfırıncı satır
x_B	0	I	$B^{-1} N$	$B^{-1} b$	1. ilâ m. satırlar

tablosundan, "Revize Simpleks Tablosu" konusuna girdiği 202. sayfada verilen harfli ifade

BASIS INVERSE	RHS
w	$c_B \bar{b}$
B^{-1}	$\bar{b}$

tablosuna geçerken, açıkça belirtmeksizin,

(1) $n = 2m$,

(2) $c_N = \vec{0}^T$ ve

(3) $N = I_m$

varsayımları yapmakta, $\vec{c_B} B^{-1}$ yerine w, $B^{-1} b$ yerine de $\bar{b}$ yazmaktadır. Bu üç varsayım gerekli değildir ve okurun "Revize Tablo"nun klasik tablonun bir (alt-matris gibi) alt-tablosu olduğunu bir bakışta görmesine olanak vermemektedir.

Revize Tablo'nun harfli ifadesini $m \times (n - m)$ ebatlarında olması gereken N matrisini kullanmaksızın yazmak, klasik tablonun genel-geçerliğine kıyasla, büyük bir genelleme kaybına neden olmaktadır. N yerine R diyen Padberg, Tablo konusuna hiç girmemekle birlikte, böyle yapmamaktadır [bkz: Padberg, s.34].

Bazaraa'yı Revize Tablo'nun ebatlarını $(m+1) \times (m+1)$ olarak vermeye sevkeden yaptığı (1), (2), (3) varsayımlardır (s.205). Revize Tablo, $n = 2m$ eşitliğinin hangi yönde bozulduğuna bağlı olarak bundan daha büyük olabileceği gibi, daha küçük de olabilir.

Önce bu N matrisinin gerçekte ne olduğuna bakalım: N, sütunları arasında, tabana hiç girmemiş veya girdi ise bile sonradan çıkartılmış yapısal karar değişkeni vektörlerinin de olduğu bir matristir. KKD'den mülhem $N = I_m$ varsayımı geçerli olduğunda ise, tamamı gevşeklik değişkeni vektörü oldu demektir. (1), (2), (3) varsayımlarıı işte ancak bu çok çok özel durumda yapılabilir. "Revize" tabloyu $N = I_m$ varsayarak N'siz yazmaktansa; N matrisinin, I_m birim matrisinin bir alt-matrisini (ki başlangıç tablosunda bu I_m'nın bütünüdür) bir alt-matris olarak içereceğini öğretmek daha uygun olabilir. KKD uyarınca bütün gevşeklik değişkenleri tabandan çıkartılacak olsa bile, $n > 2m$ olduğunda, orada bazı yapısal değişken vektörlerinin kalmaları kaçınılmazdır.

Sayfa 182'de Leksikografik Kural ispatında sunulan "$m \times (m+1)$ matrix $(\bar{b}, B^{-1})$"in de aslında $m \times (n+1)$ ebatlarında Revize Simpleks Tablosu'ndan başka bir şey olmaması ve okurun alışkanlıklarını bozmamak adına bunun $(B^{-1}N, \bar{b})$ veya $(B^{-1}N, B^{-1}b)$ şeklinde yazılması gerekir. $\bar{b}$ ile gösterdiğimiz RHS "Sağ Taraf" demektir ve sağa yazılması daha uygundur. Okurun ilk kez burada karşılaştığı leksikografik pozitiflik ve eksiolmazlık kavramları da, buna uygun olarak, satır vektörlerini (Arapça gibi) sağdan sola okuyarak tanımlanmalıdır. Bazaraa'nın $(\bar{b}, B^{-1})$ dediği matrisin aslında Revize Tablo olduğunu ve çıkan-değişken-saptama kuralının eksiolmazlık gözettiğini bilirsek, sayfa 183'de ispat gayreti ile yazdıklarını okumamıza gerek kalmaz. Zira leksikografik yöntemin fazladan yaptığı, Revize tablonun $(n+1)$.inci sütunu $\bar{b}$ kullanılarak yapılan minimalite arayışının, 1'den fazla sayıda bulgu çıktığı müddetçe, sağdan-sola sırası ile n.inci, $(n-1)$.inci, ... sütunlara teşmilidir ki, sağdan-sola tanımlanacak leksikografik pozitiflik ve eksiolmazlık güvence altına tam da böyle yapmakla alınmış olacaktır.

Kaynak kullanımını destekleyen amaç fonksiyonlarını tanımlamak ve sınıflandırmak bir başka çalışmanın konusu olabilir.

4.6 DÖNGÜ (ing. cycling)

Yozluğu bir sütun vektörünün tabanda yer sıfır şiddeti ile alması olarak tanımlamak mümkündür. Bu durumda taban olanca sütun uzayını ne denli geriyor olsa da, şiddeti sıfırdan farklı ögeleri ancak bir alt-uzay gerebileceklerdir ki, bunlara gene "alt-taban" diyeceğiz. Verdiğimiz yoz örnekte simpleks algoritmasının optimalite (veya sonlanma) koşulu, tabanda sıfır şiddetinde bulunan s_1 gevşeklik değişkeni yerine oraya x_3

yapısal değişkenini yine sıfır şiddeti ile almakla sağlanabilir ki, buna bir "yoz anahtarlama" işlemi denmektedir [bkz: Bazaraa, s.106, 175].

x_1	x_2	x_3	s_1	s_2	s_3	RHS
0	0	0	$+\frac{1}{7}$	$+\frac{4}{7}$	$+\frac{5}{7}$	7.6
0	0	1	$\frac{5}{7}$	$-\frac{1}{7}$	$-\frac{3}{7}$	0
		0				3.4
		0				0.8

Yoz anahtarlama işlemlerinin daima sonuca götüreceğinin bir garantisi yoktur. "Döngü" denen olgu ortaya bu durumda çıkmaktadır. Şimdi E. M. L. Beale tarafından verilen klasik döngü örneğine bakalım:

$max.$	$\frac{3}{4}x_4$	$-20x_5$	$+\frac{1}{2}x_6$	$-6x_7$	RHS
$ST:$	$+\frac{1}{4}x_4$	$-8x_5$	$-x_6$	$+9x_7$	≤ 0
	$+\frac{1}{2}x_4$	$-12x_5$	$-\frac{1}{2}x_6$	$+3x_7$	≤ 0
			$+x_6$		≤ 1
ve	$eksiolmazlık$	$kısıtları.$			

Bu örnekte döngü olgusunun ortaya nasıl çıktığını görmek zor değildir. Dört yapısal değişkenden üçüne tekabül eden sütun vektörleri RHS vektörüne dik bir alt-uzay germektedirler. Kıt üçüncü kaynağı tüketen yegâne aktivite x_6 ise, tabana girdiğinde, ancak sıfır şiddetinde girmektedir ki döngü bundan ileri gelmektedir.

Teorem 1. *Döngü olgusu varsa, sütun uzayında, RHS vektörü, bir alt-tabanın gerdiği uzaya diktir.*

Kanıt. Eğer dik olmasaydı, KKD amaç fonksiyonu döngüye mahal vermezdi. Alt-taban ögelerinden birinin RHS üzerinde izdüşümü pozitif olsa idi, RHS ile temsil edilen kıt kaynaklar bir ölçüde tüketilmiş olacağından, KKD amaç fonksiyonu daha yüksek bir değer alacaktı. Döngü durumunda ise, amaç foksiyonu, bildiğimiz gibi hep aynı değeri alır. □

Döngü olgusu, ortaya, bu alt-uzaydaki ögelerin, tabana, mutlaka sıfır olması gerekmeyen şiddetlerle, münavebeli olarak girmeleri ile çıkmaktadır. Beale örneğinde hep sıfır şiddetinde giriyor olmalarının nedeni şudur: bu alt-taban ögelerinin RHS vektörü üzerinde izdüşümleri sıfır olduğundan, belli bir vektörel bileşkeleri de sıfır olmalıdır. Ve hiçbiri bir diğerinin eksisi değildir. Dolayısı ile, biri tabana pozitif bir şiddet ile girdiğinde, böyle pozitif şiddetle diğer ikisinin de girmeleri gerekmektedir. Ne var ki, tabanda sadece 3 vektöre yer vardır ve kıt üçüncü kaynağın gevşeklik değişkenini oradan ihraç etmek mümkün olmamaktadır.

Döngü olgusunun önüne kaynak kullanımını gözeten bir anahtarlama kuralı ile geçmek mümkün gözükmektedir.

4.7 Leksikografik Döngü Önleme Kuralı'nın eleştirisi

Döngü vaki olduğunda, "çıkan değişken" (ing. "departing variable") adayının tek olmadığı gerçeğinden hareketle, böyle bir döngü-önleme veya erken-teşhisi kuralı önerilmiştir. Oysa birden çok sayıda temel değişkenin tabandan çıkmağa aday olduğu durumlarda uygulanabilecek akıllı politikalar (ing. "heuristics") vardır. Mesela sabit RHS vektörünün, olası alt-tabanlardan hangisi üzerindeki izdüşümünün en büyük olacağını araştırmak gibi.. Bu politikanın müteakip adımda aday sayısını azaltacağının, eğer azaltmıyorsa da döngünün sabit olduğunun ispat edilebilir olduğuna inanıyoruz. Adayın tek olduğu durumda gözetilen eksiolmazlık kriteri de, bu genel politikanın bir özel hali, şöyle olmaktadır:

k taban-dışı bir değişkenin indisi olduğunda, a_k vektörünü tabana $\frac{\langle n_i, RHS\rangle}{\langle n_i, a_k\rangle}$ skaler-çarpanı [bkz: (2.3b) ve (2.3a), $RHS = \vec{b}$, s.12] $i \in Taban$ taraması üzerinden en küçük çıkan kesirle çarparak, yani RHS vektörünün $(\frac{\langle n_i, RHS\rangle}{\langle n_i, a_k\rangle})a_k$ bileşeni olarak almaktayız ki, burada n_i i indisli taban vektörü ("çıkan değişken"in sütun vektörü) hariç tutulduğunda kalan alt-tabanın gerdiği alt-uzayın "bizi ilgilendiren satır" diye de andığımız normali olmaktadır. Giren değişken sütun vektörünün (konuşlandırılması olmasa da) boyunun, bulduğumuz bu son vektörel kemiyetin payını da paydasını da aşağı-yukarı eş-etkilediği düşünülecek olursa, minimalite arayışının maksadının aslında RHS vektörünün normali üzerindeki izdüşümünün en küçük, buna mukabil gerdiği alt-uzay üzerindeki izdüşümünün en büyük olduğu alt-tabanı bulmak olduğu anlaşılır.

Çıkan vektör, o alt-tabanın dışarıda-tutulmuş-ögesi olacaktır.

Bölüm 5

İktisadî uygulama

Çevresel, toplumsal ve kamusal (ESG) duyarlılık

Çevrecilik, sosyal yükümlülükler ve kamu denetimleri (ing. ESG = environmental, social, governmental) hesaplara katılmak istendiğinde, ESG-optimum çözümler üreten Doğrusal Programlama kuramı görüşümüze göre ancak sütun uzayında yapılabilir. Böyle ESG-optimum çözümler bulan yazılımlar batıda günün konusudurlar [bkz: Grene, S., "Quants are the new ethical investors" (title over full page) Viewpoint, FTfm annex of Financial Times Europe, © The Financial Times Limited 2016 No.39,069, Monday 25 January 2016, p.9]. Kapitalizmin bir çelişkisi de küresel ekonomi (veya kapitalist üst-akıl) için kıt olan kaynakların firma için; firma için kıt kaynakların da küresel ekonomi için kıt olmamalarıdır. Bunlara kısaca "g-kıt kaynaklar" ve "f-kıt kaynaklar" diyelim. Petrol, doğal gaz ve kömür gibi yakıtlar g-kıt kaynaklara verilebilecek başlıca örneklerdir. Ancak firma küresel ekonominin çok küçük bir cüzzü olduğundan, g-kıt kaynakları piyasadan dilediği kadar temin edebilir. Yani verdiğimiz üç örnekten hiçbiri bir f-kıt kaynak değildir. Tersine, firma için, arttırılması, teknik nedenlerden ötürü, hiç değilse kısa erimde olanaksız kaynaklar; küresel ekonominin yenilenebilir kaynaklar kullanarak ürettiği sermaye mallarıdır. Bunlara örnek olarak da ocak, fırın, kazan, silo gibi donanımın hacimleri ve makinaların kapasiteleri gösterilebilir. Kâr maksimize eden firma elindeki f-kıt kaynaklardan "tam randıman almak" adına, g-kıt kaynakları olmasa bile, bunları tüketerek ürettiği mesela atık ısı gibi yan ürünleri düpedüz israf ediyor olabilir.

Gevşeklik değişkenlerinin amaç fonksiyonunda genellikle sıfır katsayı ile yer almaları, daha açık anlatımla hiç yer almamaları, f-kıt kaynakların piyasaya arzedilmeleri ile bir gelir elde etmenin mümkün olmadığı

anlamına gelir. Kurulu kapasiteler fason imalat yapacak girişimcilere kiralanabilir olmasalar bile, ki buna "Kiralama Yok Varsayımı" demiştik, eğer f-kıt kaynağımız bir yan ürünse, onu, en iyi, yine firma değerlendirebilecektir. Dökümcülükte en kıymetli hammadde, aynı siparişin bir önceki dökümünden arta kalan ve "yolluk" tabir edilen döküm döküntüsüdür. Bu yan ürün piyasaya arzedilecek olsa ancak hurda fiyatına satılabilecektir ki, bir dökümhane bu yola ancak iflas ettiğinde gider.

Bir gevşeklik değişkeni, karşı geldiği kıt kaynağın niteliğine bağlı olarak, âtıl (ing. idle) kalan bir kurulu kapasiteyi ölçüyor olabileceği gibi, bir yan ürünün "çöpe atılacak" kesirini ölçüyor da olabilir. Eğer bu yan ürün, atık ısı örneğinde olduğu gibi, depo edilemez nitelikte ise; ilgili kısıtın RHS girdisi de sıfır olacaktır. d bir doğal sayı olsun. Eğer $d+1$ veya daha fazla sayıda sütun m boyutlu sütun uzayının d boyutlu bir alt uzayını eksiolmaz katsayılarla gerebiliyorsa ve bu alt uzay RHS vektörüne dikse; çöpe dökülecek yan ürün miktarlarını sıfırlayan ESG-optimum bir üretim programı yapmak imkan dahiline girecektir. Dikkat: bu yukarıda zikrettiğimiz döngü gerek koşuludur da! Böylelikle döngü Simpleks algoritmasının yaptığı bir ârıza olmaktan çıkmakta ve bir ESG-optimum çözümün ta kendisi (Beale'nin döngü örneği de verilebilecek en uygun sayısal örneklerden biri) olmaktadır.

Üretimi yan ürün üreten ya da tüketen ürünleri temsil eden sütunların gerdikleri alt uzayın RHS vektörüne dik olma koşulunun iktisadi açıklaması şöyle olmaktadır: Eğer salt yan ürünleri değerlendirmek adına üretilen ürünler firmaya kar değil, fakat maliyet getiriyorlarsa; böyle üretim(!)i f-kıt kaynakların RHS girdileri pozitif olanlarından (hepsi sıfır olsa idi, diklik sözkonusu olamazdı) hiç kullanmaksızın yapmak, ESG kaygılarla yapılacak fedakârlığı da minimize edecektir. Elbette burada amaç fonksiyonunun "(f-kıt) kaynak kullanımını destekler (KKD)" nitelikte olduğunu varsayıyoruz.

Şimdi sayısal örneğe geçelim. Beale örneğinde üretilecek miktarı x_4 değişkeni ile ölçülen ürün (buna bundan böyle "dördüncü ürün" diyeceğiz) iki çeşit yan ürün tüketilerek üretilen ve firmaya bir kâr bırakan sorunsuz bir üründür ve böyle olduğundan da hem optimum hem de ESG-optimum bir üretim programına dahil olması çok olasıdır.

Beşinci ürün her iki yan üründen de üretmesine karşın, katkı marjı (yani satış fiyatı eksi birim değişken maliyet) negatif olduğundan firmaya maliyet getiren sorunlu bir üründür. Bu beşinci ürünün -20 olan katkı marjını -15 olarak değiştirelim. Beale örneği üzerinde başka hiçbir

değişiklik yapmayacağız. Amaç fonksiyonu ile sınırlı bu değişikliğin olanaklılık polihedral kümesini hiçbir surette değiştirmeyeceği kesindir.

Altıncı ürün her iki yan ürünü de üretiyor olmasına karşın firmanın f-kıt kaynağını tüketen yegane üründür ve eğer amaç fonksiyonumuz KKD ise her üretim programında yer alması kaçınılmazdır.

Yedinci ürün tıpkı dördüncü ürün gibi yan ürün tüketilerek üretilen bir ürün olmasına karşın, ondan farklı olarak firmaya kâr değil maliyet getirmektedir. Buna karşın elde kalması arzu edilmeyen iki çeşit yan ürünü çok daha büyük bir iştahla yutmaktadır.

Şimdi dikkat: bu sayısal örneğin üç değil fakat dört karar değişkenine pozitif reel sayılar atayan şu sıra-dışı çözümü de vardır: η pozitif bir reel sayı olduğunda, $x_4 = \frac{2}{5} + \frac{112}{5}\eta$, $x_5 = 0 + \eta$, $x_6 = 1$ ve $x_7 = \frac{1}{10} + \frac{4}{15}\eta$. Karar değişkenleri bu pozitif değerleri aldıklarında, amaç fonksiyonu da $(1+\eta)/5$ gibi hiçbir kısıtın sınırlandırmadığı bir değer almaktadır! Firma f-kıt kaynağı kullanmayacak olsa, karar değişkeni değerlerindeki sabitler sıfırlanacağından, kâr $(0 + \eta)/5$ gibi yine sınırsız bir sayı olacaktır. Dikkat edilecek olursa standart simpleks algoritması bu çözümü bulamaz ancak mevcudiyeti hakkında bize bir işaret (döngü uyarısı) verebilir. Simpleks ile bulunan maksimum kâr sonlu bir sayı çıkarken, ESG-optimum kârın ∞ çıkması matematiğin bir cilvesi olsa gerektir.

Hal-i hazırdaki Doğrusal Programlama eğitimi, bu tekniğin uygulayıcılarını, $m \times n$ ebatlarında bir problemin bir temel çözümünün (ing. basic solution) en çok m tane karar değişkenine pozitif değerler atayacağı saplantısına sürüklemiş olabilir. Bunun her zaman böyle olmayacağını, bu tezin yazarı da, tezi yazmağa başladıktan sonra anlamıştır. Bunun kurama ciddi bir katkı olduğunu düşünmektedir.

Bölüm 6

KURAMSAL ÇERÇEVE (III)

6.1 FAZLALIK YOKLAMASI

Bir Doğrusal Programlama probleminin A matrisinin her bir $a^{(i)}$ satırını RHS vektörünün ilgili girdisine bölerek "sağ taraf(ı)" komple 1 yapalım. Böyle yapmakla, sadece, karar-değişkenlerinin ölçüsü oldukları niceliklerin birimlerini yeniden-tanımlamış oluruz. Bulduğumuz yeni matrisi, yine A ile göstermeye devam, böyle olduğundan edeceğiz. $A = (N_0, B_0) = (N_1, B_1) = \ldots = (N_t, B_t)$ ve $B_0 = I_m$ olduğunu anımsıyoruz (s.9). Gevşeklik değişkenlerini bu bölümde gözardı edeceğimizden, A değil fakat onun N_0 alt-matrisinin satırları ile ilgileneceğiz ve n harfini daha önce başka bir vektörü göstermek için kullanmış olduğumuzdan, bu satırları gene a harfi ile göstereceğiz (bunu daha önce de yapmıştık, bkz. s.9). Bu mutabakata göre, gevşeklik değişkenleri, $\vec{x} = (x_1, x_2, \ldots, x_{n-m})$ karar değişkenlerimiz arasında olmayacak ve

$$N_0 = \begin{bmatrix} \vec{a}^{(0)} \\ \vdots \\ \vec{a}^{(m)} \end{bmatrix} = \begin{bmatrix} a_{1,1} & \ldots & a_{1,(n-m)} \\ \vdots & & \vdots \\ a_{m,1} & \ldots & a_{m,(n-m)} \end{bmatrix}$$

olacaktır.

Teorem 2.

$$\begin{aligned} &max.\ \langle a^{(0)}, x\rangle \\ &S.T.\ \langle a^{(i)}, x\rangle \leq 1, i = 1, \ldots, m \end{aligned} \tag{6.1}$$

Doğrusal Programlama probleminde m.inci kısıtın "fazla" olması için bir gerek koşul

$$a^{(m)} \in conv\{a^{(1)}, \ldots, a^{(m-1)}\} \tag{6.2}$$

dir.

Kanıt. Hipotez olarak $\{x \in R^{(n-m)} : \langle a^{(i)}, x\rangle \leq 1, i = 1, \ldots, m-1\} \subset \{x \in R^{(n-m)} : \langle a^{(m)}, x\rangle \leq 1\}$ alıyoruz. İki tarafın da polarını alacak olursak, Problem 3 ve 3° [Barvinok, s.144] yardımı ile $conv\{0, a^{(m)}\} = \{y \in R^{(n-m)} : y = \lambda a^{(m)}, 0 < \lambda \leq 1\} \subset conv\{0, a^{(1)}, \ldots, a^{(m-1)}\}$ yazarız. Bu da $\alpha_i \geq 0, i = 0, 1, \ldots, m-1$ için $\lambda a^{(m)} = \alpha_0 0 + \sum_{i=1}^{m-1} \alpha_i a^{(i)}$ ve $1 = \sum_{i=0}^{m-1} \alpha_i$ demektir. İspat $\alpha_0 = 1 - \lambda \geq 0$ alınarak ve vektör eşitliğinin her iki tarafı λ'ya bölünerek tamamlanır. □

(2) koşulu sağlandığında, içerdiği $conv\{\ldots\}$ ifadesi olanaklılık polihedral kümesinin polarıdır (ing. polar body) ve $\vec{0} \in R^{(n-m)}$ noktası (1) Doğrusal Programlama probleminin olanaklı çözümlerinden biri olduğundan bunun tam tersi de Exercise 3.4.5(i) uyarınca doğrudur [Grünbaum, s.49]. Bu, Exercise 3.4.5(vii) uyarınca, olanaklılık bölgesi polihedral kümesinin polarının bir politop olmasını gerektirir. Tam aksinin varit olduğu minimizasyon problemleri, dolayısı ile, bu doktora tezinin ilgi alanı dışında kalmaktadırlar. Konveks Politoplar mevcut literatürü, eğer orijin olanaklı bir çözüm değilse, olanaklılık polihedral kümesinin onu (içerisinde değil, fakat) sınırında içerecek şekilde ötelenmesini (ing. translation, shifting) gerektirmektedir. Bu da hesapların daha da karışması demek olacaktır.

6.2 Mutlak fazlalık

Yukarıdaki ispatta $\alpha_0 = 1 - \lambda$ kabulünü hiçbir zorunluluk olmadan yaptık ve bulunabilecek gerek koşullardan sadece birini bulmuş olduk.

Bu koşul sağlandığında fazlalığa yozluk eşlik eder. Bulduğumuz içerme ilişkisine göre $a^{(m)} \in conv\{0, a^{(1)}, \ldots, a^{(m-1)}\}$ ifadesi esastır ve bu (gerek) koşul $a^{(m)} = \ldots$ ifadesinde $\vec{0}$ vektörünün katsayısının sıfır olmasını gerektirmeyebilir. Eğer değilse, m.inci kısıtın mutlak anlamda fazla olduğu ve yozluğun sözkonusu olmadığı açıktır. Okuyucu bütün bu iddialarımızı (1) eşitsizliklerini ilgili α_i katsayıları ile genişlettikten sonra taraf-tarafa toplayarak doğrulamalıdır.

Nasıl ki homojen doğrusal denklemler lineer cebir ile projektif geometri arasında bir köprü ise "Gale transforms" da yine bu cebir ile konveks politoplar kuramı arasında köprüdür. Bu kuramının bulgularından yararlanabilmek için, olası yeter koşullar arasında, yukarıdaki (2)'i esas alacak ve çıkartacağımız sonuçların geçerliliklerini diğer olası gerek koşullardan biri sağlandığında da koruduklarının ispatını matematikçilere bırakacağız. $a^{(i)}$ satırının Gale-transformunu $\hat{a}^{(i)}$ ile göstereceğiz. Cari Tablo'nun bize bunları dahi verir olduğunu az ileride göreceğiz.

Fazlalık yoklaması yapmak için bir gerek koşul bulmak yeterlidir. Önce pratikte nasıl sınanması gerektiğine bakalım. N_0 matrisinin $a^{(1)}, \ldots, a^{(m)}$ satırlarının ilgin bağımlılığı (ing. affine dependence) sütunlarının da doğrusal bağımlılıklarını gerektirir. O halde fazlalık (ve yozluk) için bir gerek koşul da rank düşüklüğü (ing. being short of full rank) olsa gerektir. Düşüklüğün 1'den fazla olmadığını varsayacağız. Bu varsayım işimizi kolaylaştıracak, uyarınca $\hat{a}^{(i)}$ vektörleri tek boyutlu, açıkçası R reel sayılar kümesinin elemanları olacaklardır. N_0 matrisimizin rankı $m-1$ ise:

$$B_t^{-1}(N_0, I_m) = \left[\begin{array}{ccc|c} \bar{a}_{1,1} & \ldots & \bar{a}_{1,n-m} & B_t^{-1} \text{ kare} \\ \vdots & & \vdots & \text{matrisinin ilk} \\ \bar{a}_{m-1,1} & \ldots & \bar{a}_{m-1,n-m} & (m-1) \text{ satırı} \\ \hline 0 & \ldots & 0 & \hat{a}_1^{(m)} \quad \ldots \quad \hat{a}_m^{(m)} \end{array}\right]$$

olur. Bu da B_t^{-1} matrisinin son (en alt) satırının ispatta geçen $(\alpha_1, \ldots, \alpha_{m-1}, -\lambda)$ "ilgin bağlayıcılar" (ing. affine dependences) satırının bir skaler çarpan farkı ile aynısı olduğunu gösterir. Biz bu satırı "bizi ilgilendiren satır" adı ile anmıştık. (2) koşulunda geçen $conv\{\ldots\}$ ifadesi bütün katsayıların eksiolmazlığını gerektirir. Bu da $a^{(1)}, \ldots, a^{(m-1)}, a^{(m)}$ vektörleri arasında bir ilgin bağımlılık ilişkisinde bir ve yalnız bir katsayının işaretinin diğerlerininkilerden farklı olması demektir ki, bilgisayarın fazla devre açıp-kapamadan yapabileceği bir testtir. Katsayıları Simpleks algoritmasının birer yan ürün olarak verdiği ve girdileri bunlar olan vektörü "bizi

ilgilendiren satır" adı ile andığımız unutulmamalıdır. Bu satırı Gale-transform notasyonuna uygun olarak $(\hat{a}_1^{(m)}, \ldots, \hat{a}_m^{(m)})$ diye yazdık.

Teorem 3. *$m > 3$ olduğunda, fazlalık yoklaması yapmağa lüzum, B_t^{-1} matrisinin bizi ilgilendiren satırında en az 2 + işaretli ve en az 2 − işaretli katsayı varsa yoktur. Fazlalık yoklaması, bizi ilgilendirenler arasında, bu koşulu sağlamayan satıra karşı gelen kısıt için yapılmalıdır.*

Kanıt. Teorem 5.4.2'den yararlanacağız [Grünbaum, s.88]. Teoremin hipotezindeki "(...) some n-polytope P with m vertices" ayrıntısı hiçbir uç-noktanın diğerlerinin gerdikleri konveks kabuğun içerisinde olmadığı anlamına gelir ki, bunun (2) koşulunu hiçbir satır vektörünün sağlamadığı demek olduğu açıktır. Teorem 5.4.2, bu durumda, ilgin bağlayıcılardan ikisinin pozitif, ikisinin de negatif olacaklarını söylemektedir ki, ispat etmek istediğimiz de budur. □

$3 < 2 + 2$ olduğundan, bu teorem $m = 3$ durumunda uygulanabilir değildir. Bu durumu ele özel olarak alacağız [bkz: Demokrasi Prensibi bölümü, s.40].

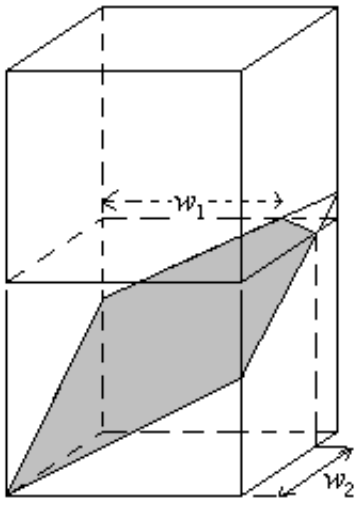
Bölüm 7

DEMOKRASi PRENSiBi

$3 \times n$ ebatlarında bir Doğrusal Programlama probleminde fazlalık olmadığı garantisini Teorem 3'e (s.39) başvurarak vermenin mümkün olmadığı açıktır. Bu durumun ele özel olarak alınması gerekmektedir.

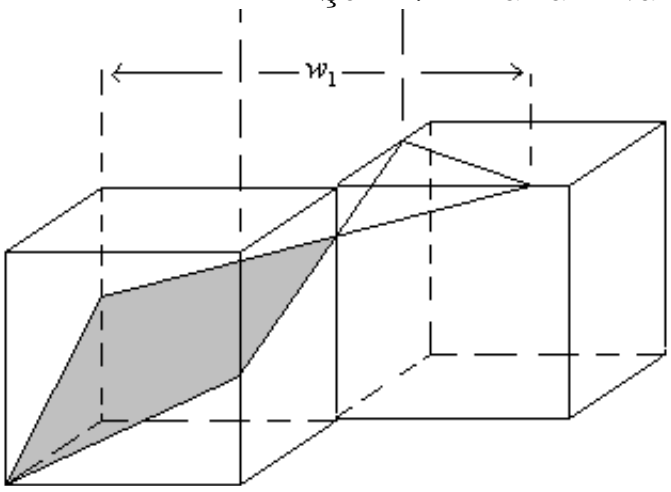
Sütun uzayında teknolojik olanaklılık alt-uzayı ekonomik olanaklılık sınırı düzlemleri aşağıdaki iki şekilde görüldüğü gibi kesecektir. Gölgelendirdiğimiz olanaklılık bölgeleri satır uzayında âşina olduğumuz olanaklılık bölgeleri ile Exercise 3.4.13 gereği eş-biçimlidir [Grünbaum, s.51].

Şekil 7.1: Sağlıklı



"Sağlıklı" çizimde her üç kısıt da yerine göre kısıtlayıcı (ing. binding) olabilmekte iken, "Fazlalık var" çiziminde üçüncü kısıt (dikey

Şekil 7.2: Fazlalık var



boyut) kısıtlayıcı ancak olanaklılık bölgesi dışında olmaktadır ki, "fazlalık" dediğimiz tam da budur.

Gösterecek olduğumuz gibi, teknolojik olanaklılık alt-uzayı

$$(1 - w_2)x + (1 - w_1)y + (w_2w_1 - 1)z = 0 \qquad (7.1)$$

denklemini sağlayan $(x, y, z) \in R^3$ noktaların kümesi olmaktadır. Bunu (x, y, z) yerine sırası ile $(0, 0, 0)$, $(w_1, 1, 1)$ ve $(1, w_2, 1)$ koyarak doğrulayabiliriz. Bu doğrusal denklemin katsayıları "bizi ilgilendiren satır(ın)" girdileri olmaktadır. Zira, yukarıdaki denkleme bir iç-çarpım nazarı ile bakılacak olursa anlaşılan odur ki, girdileri bunlar olan sütun vektörü teknolojik olanaklılık alt-uzayını geren ve geriyor olabilecek her sütun vektörüne diktir!

Katsayıların toplamı ilk iki katsayının çarpımına eşit çıkmaktadır:

$$(1 - w_2) + (1 - w_1) + (w_2w_1 - 1) = (1 - w_2)(1 - w_1) \qquad (7.2)$$

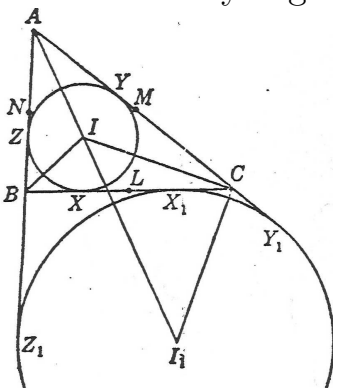
Çizimlerden ve (2)'den anlaşılan odur ki, + işaretli girdiler çoğunluğu sağlıyorlarsa problemimiz sağlıklıdır. Aksi takdirde azınlıkta kalan girdiye karşı gelen kısıt tıpkı Teorem 2'de (s.37) olduğu gibi fazladır.

Bölüm 8

En-Küçük-Açılar Anahtarlama Kuralı'nın eleştirisi

Bir Doğrusal Programlama probleminin olanaklılık bölgesinin Shively'nin klasik ders kitabından aldığımız "Şekil 33"de görülen çeşitkenar ABC üçgeninin içi, kenarları ve köşeleri olduğunu varsayalım [Shively, s.46]. İki-boyutlu vektör uzayımızın orijini de orada görülen dış-teğet-çemberin I_1 merkezi olsun. Açıktır ki, $-I_1\vec{X}_1$, $I_1\vec{Y}_1$ ve $I_1\vec{Z}_1$ vektörleri karşı geldikleri kısıtların dışa-yönelik gradyanlarıdır. Amaç fonksiyonu gradyanı bunlardan biri ile çakışırsa, karşı geldiği kısıtın sınırındaki olanaklı noktaların hepsi birer optimum çözüm olacaktır. Dahası iki komşu kenarın ikisi de ikinci-kısıtlayıcı-kısıt olabilecektir. Demek ki, kısıt gradyanları ikinci-kısıtlayıcı-kısıt belirlemede birer eşik olmaktadırlar. Eşikten saat yönünde veya tersi küçük bir sapma ikinci-kısıtlayıcı-kısıtı değiştirecektir.

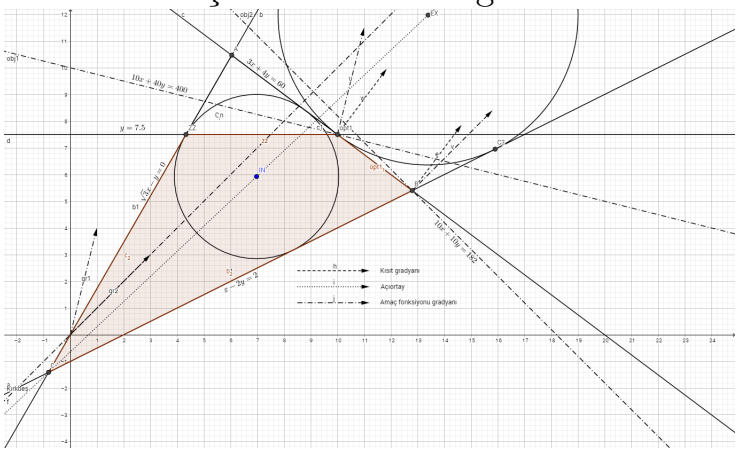
Şekil 8.1: Shively Fig. 33



Dikkat edilecek olursa, eğer amaç fonksiyonu gradyanı en küçük açıyı mesela $-I_1\vec{X}_1$ vektörü ile yapmakta ise, ikinci-en-küçük-açı eşiği de $\angle Z_1 I_1 Y_1$ açısının $-I_1\vec{A}$ ortayı olmaktadır. Çeşitkenar bir üçgende bu iki eşiğin üstüste gelmesi gerekmeyebilir. Eğer aralarında bir açı olacaksa, buna "tehlikeli bölge" demek uygun olacaktır. Amaç fonksiyonu gradyanı bu bölge içine düştüğünde, en-küçük-açılar kuralının bizi yanlış köşeye yönelteceği çok açıktır. Bir örnek vermek gerekirse, amaç fonksiyonu gradyanının $-\vec{I}_1 L$ olduğunu varsayalım. İkinci en küçük açıyı $I_1\vec{Z}_1$ kısıt gradyanı ile yapmasına karşın, optimaliteye bu kısıtın kısıtlayıcı olmadığı C köşesinde ulaşılacaktır.

Bu kriter elimizde oldukça dilediğimiz kadar çok karşı-örnek üretebilir, literatürde verilenlerin de neden böyle olduklarını açıklayabiliriz. Pan'ın Problem 1'i ile başlayalım [Pan, s.221]. Oradaki "Fig. 1"in olanaklılık alanı, iki karar değişkeni üzerine konmuş üst ve alt limitler gözardı edilecek olursa, bir çeşitkenar üçgenin içi, kenarları ve köşeleridir. Bu üçgenin iç-teğet çemberinin koordinatları yuvarlak-hesap $x = 6.9728463129$ ve $y = 5.9254339837$, yarı-çapı da 3.07594505'tir. Açı ortayı ve $3x+4y = 60$ kısıtının gradyanının eğimleri de sırası ile 0.941776 ve 4/3'tür. "Tehlikeli bölge" bu ikisinin arası olmaktadır. Eğimi 4 olan amaç-fonksiyonu gradyanı bu iki sınır arasında kalmadığından, metod tam da umulan sonucu vermektedir. Şimdi gradyan eğimi 1, yani tehlikeli bölge dahilinde olan alternatif $z = 10x + 10y$ amaç-fonksiyonunu benimseyelim. İkinci-en-küçük-açı eşiği 0.941776 aşılmış olmadığından optimum çözümü yine aynı noktada bulmayı umuyor olmalıyız. Yeni amaç-fonksiyonumuz, şekilde görülebilen koordinatlarıyla ile orada $175 = 100 + 75$ değeri almaktadır. Buna karşın olanaklı $(x, y) = (12.8, 5.4)$ noktasında daha büyük değer alabilmektedir.

Şekil 8.2: Pan Fig. 1



Şimdi de, metodun, Li'nin karşı-örneğinde başarısızlığa neden uğradığını açıklamaya çalışalım [Li, s.263]. Örneği üç-boyutludur, fakat olanaklılık bölgesi polihedronunu $x_1 = 0$ düzlemi üzerine izdüşürmek sureti ile olanaklılık bölgesi köşe noktaları $(50., 100.)$, $(100., 0.)$ ve $(100., 99.5)$ olan üçgen, amaç-fonksiyonu gradyanı eğimi de -100 olan bir 2-boyutlu Doğrusal Programlama problemi elde ederiz. İç-teğet çemberi merkezi $(80.96101150945, 80.6504494693)$ noktasında, yarı-çapı da 19.038988491'dir. En dik kısıt gradyanı eğimi $+100$, açı-ortayı eğimi de 4.236067977 olmaktadır. Amaç fonksiyonu gradyanının tam da bu ikisi arasında kaldığını şöyle gözlemliyoruz: $\arctan(-4.236) > \arctan(-100) > \arctan(+100)$ veya $\pi - 1,339 > \pi - 1.56 > 1.56$.

Özet olarak, en-küçük-açılar anahtarlama kuralı ikinci, üçüncü, ilah. galibinin saptanması çoğu kez olanaksız bir müsabaka gibidir.

Bölüm 9

DEĞERLENDiRME ve SONUÇ

Bu çalışmada, Doğrusal Programlama'ya sütun uzayında baktık ve göze ilk çarpanları göstermeğe çalıştık. Bütünü ile satır uzayına özgü bir olgu olduğu düşünülen fazlalığın sütun uzayında içeriği-daha-zengin bir kuramını geliştirdik ve bunun konveks politoplar kuramının ispatlanmış bazı teoremlerinin Doğrusal Programlama gözönünde tutularak yeniden okunmalarından fazla bir gayret gerektirmediğini anladık.

Kaynakça

[1] Alkan, M. & Erdal, M., Lojistik ve Dış Ticaret Sözlüğü, ISBN 975-92469-4-5, Eylül 2004 İstanbul

[2] Barvinok, A., A Course in Convexity, Amer. Math. Soc. Graduate Studies in Mathematics Vol.54, RI 2002

[3] Bazara, M. S., Jarvis, J. J. & Sherali, H. D., Linear Programming and Network Flows, Copyright ©2010 by John Wiley & Sons Inc.

[4] Beale, E. M. L., Cycling in the Dual Simplex Algorithm, Naval Research Logistics Quarterly 2(4), pp.269-276 December 1955

[5] Charnes, A. & Cooper, W. W., Management Models and Industrial Applications of Linear Programming, Copyright ©1961 by John Wiley & Sons Inc.

[6] Gass, S. I. & Vinjamuri, S., Cycling in linear programming problems, Computers & Operations Research 31 (2004) 303-311

[7] Gürel, O., Lineer Programlama, İstanbul Teknik Üniversitesi Kütüphanesi Elektronik Hesap Merkezi Yayınları Sayı 4, Ankara Üniversitesi Basımevi 1966

[8] Hadley, G., Linear Programming, Copyright ©1962 by Addison-Wesley Publishing Company, LoCCC No.61-5306

[9] Lee, J., https://github.com/jon77lee/JLee_LinearOptimizationBook/blob/master/JLee.3.0.pdf

[10] Li, W., A note on two direct mothods in linear programming, European Journal of Operational Research 158 (2004) 262-265.

[11] Llewellyn, R. W., Linear Programming, Copyright ©1964 by Holt Rinehart and Winston Inc. LoCCC No.64-10124

[12] LODER, Lojistik Terimleri Sözlüğü, Dördüncü Baskı, Haziran 2010 İstanbul

[13] Pan, P.-Q., Practical finite pivoting rules for the simplex method, OR Spectrum 12 (1990) 219-225.

[14] Padberg, M., Linear Optimization and Extensions, Springer-Verlag, 1999

[15] Rushton, A., Croucher, P. & Baker, P., The Handbook of Logistics and Distribution Management, 5th Edition, Kogan Page, ISBN 978 0 7494 66 27 5, LoCCC No.658.7-dc23

[16] Shively, L. S., An Introduction to Modern Geometry, John Wiley & Sons Inc. New York, 1939.

[17] Stojkovic, N. V. & Stanimirovic, P. S., Two direct methods in linear programming, European Journal of Operational Research 131 (2001) 417-439.

[18] Trefethen, L. N., & Bau III, D., Numerical Linear Algebra, SiAM OT50 1997

[19] Vanderbei, R. J., Linear Programming, Springer International Series in OR & Management Science Vol.196, 2008

[20] Yılmaz, A. Ö., "Halka Açık Anonim Ortaklıklarda Yöneticinin Mali Hakları" Yüksek Lisans Tezi, A.Ü.SosBilEns Özel Hukuk (Ticaret Hukuku) Anabilim Dalı, Ankara 2009