END801
YÖNEYLEM ARAŞTIRMASININ TEMELLERİ
DERS NOTLARI

Dr. Y. İlker Topcu

Teşekkür:
Prof. W.L. Winston'ın "Operations Research: Applications and Algorithms" kitabı ile Prof. J.E. Beasley's YA ders notlarının bu ders notlarının oluşturulmasına olan katkıları yüzünden her iki profesöre de teşekkür ederiz....
Rastlayabileceğiniz tüm hataların sorumluluğu bize aittir. Lütfen bizi bu hatalardan haberdar ediniz!
İstanbul Teknik Universitesi OR/MS takımı

www.isl.itu.edu.tr/ya

İÇİNDEKİLER

1. YÖNEYLEM ARAŞTIRMASINA GİRİŞ

1.1 TERMİNOLOJİ

"Yöneylem Araştırması" (YA), İngiliz ve Avrupalılar tarafından "Operational Research" ve Amerikalılar tarafından "Operations Research" olarak isimlendirilir ve "OR" olarak kısaltılır.

Bu alanda kullanılan bir diğer terim de "Yönetim Bilimi"dir (Management Science) ve uluslararası literatürde MS olarak kısaltılır. İki terim birleştirilerek "OR/MS" veya "ORMS" de denilir.

YA genelde bir "Sorun Çözme" (problem solving) ve "Karar Verme Bilimi" (decision science) olarak da değerlendirilir.

Bazı kaynaklarda YA yerine Endüstri Mühendisliği (Industrial Engineering - IE) kavramı da kullanılır.

Son yıllarda bu alan için tek bir terim kullanılmaya çalışılmaktadır: OR.

Biz de derste bu alan için Yöneylem Araştırmasının Türkçe kısaltması olan YA'yı kullanacağız.

1.2 YA YÖNTEMBİLİMİ

Bir sorunun çözümü için YA kullanıldığı zaman aşağıdaki yedi adımlık süreç takip edilmelidir.

Adım 1. Sorunun Formülasyonu

YA analisti (sorunu olan karar vericiye YA teknikleri ile yardımcı olan kişi) ilk olarak sorunu tanımlar. Sorunun tanımlanması; amaçların ve sorunu oluşturan sistemin bileşenlerinin belirlenmesi ile olur.

Adım 2. Sistemin İncelenmesi

Daha sonra analist sorunu etkileyen parametrelerin değerlerini belirlemek için veri toplar. Söz konusu değerler sorunu temsil edecek bir matematiksel modelin geliştirilmesi (Adım 3) ve değerlendirilmesi (Adım 4) için kullanılır.

Adım 3. Sorunun Matematiksel Modelinin Kurulması

Analist tarafından sorunu ideal bir şekilde temsil edecek bir matematiksel model geliştirilir. Bu derste modelleme için çeşitli yöntemler öğreneceğiz.

Adım 4. Modelin Doğrulanması

Üçüncü adımda kurulan modelin gerçeği iyi yansıtıp yansıtmadığı sınanır. Şu anki durum için modelin ne kadar geçerli olduğu belirlenerek modelin gerçeğe ne kadar uyduğu test edilir.

Adım 5. Uygun bir Seçeneğin Seçilmesi

Eldeki model üzerinde bir çözüm yöntemi kullanılarak amaçları en iyi karşılayan bir seçenek (varsa) analist tarafından seçilir.

Bazen eldeki seçeneklerin kullanımı için sınırlandırmalar ve kısıtlamalar olabilir. Bu yüzden amacı karşılayan seçenek bulunamayabilir. Bazı durumlarda ise amaçları en iyi şekilde karşılayan birden fazla sayıda seçenek bulunabilir.

Adım 6. Sonuçların Karar Vericiye Sunumu

Bu adımda, analist modeli ve model çözümü sonucunda ortaya çıkan önerileri karar verici ya da vericilere sunar. Seçenek sayısı birden fazla ise karar verici(ler) gereksinimlerine göre birini seçerler.

Sonuçların sunumundan sonra, karar verici(ler) öneriyi onaylamayabilir. Bunun nedeni uğraşılan sorunun doğru tanımlanmaması ya da modelin kurulmasında karar vericinin yeterince sürece karışmaması olabilir. Bu durumda analist ilk üç adıma yeniden dönmelidir.

Adım 7. Önerinin Uygulanması ve İzlenmesi

Eğer karar verici sunulan öneriden memnun kalırsa, analistin son görevi karar vericinin öneriyi uygulamasına yardımcı olmaktır: Seçeneğin kullanılarak sorunun çözümüne nezaret etmeli ve özellikle çevre koşulları değiştikçe amaçları karşılamaya yönelik dinamik güncellemeler yaparak uygulamayı izlemelidir.

1.3 YA'NIN TARİHÇESİ

YA göreceli olarak yeni bir bilim dalıdır. 1930'lu yılların sonunda YA ilk olarak Birleşik Krallık'ta kullanıldı.

1936 yılının başında İngiliz Hava Bakanlığı; doğu kıyısında, Felixstowe yakınlarında, Suffolk'da Bawdsey Araştırma İstasyonu'nu kurdu. Söz konusu yer hava kuvvetleri savaş öncesi radar çalışmalarının yapıldığı merkezdi. Yine 1936 yılında Kraliyet Hava Kuvvetleri (RAF) içinde Britanya hava savunması için özel bir birlik oluşturuldu. Radarın kullanılmaya başlaması beraberinde bazı sorunlar da getirdi: Uçakların rotası ve kontrolu gibi elde edilen bilginin doğru ve etkin bir şekilde kullanılması gibi. 1936 yılının sonunda, Kent'deki Biggin Hill'de kurulan bir grup elde edilen radar bilgisi ile diğer uçak ile ilgili yer bilgilerinin bütünleştirilmesini hedefleyen çalışmalar yaptı. Söz konusu çalışmalar YA'nın başlangıcı olarak kabul edilebilir.

1937 yılında Bawdsey Araştırma İstasyonu deneysel çalışmaları pratiğe çevirdi ve Radar İstasyonu olarak çalışmaya başladı. Radardan elde edilen bilgiler bütünleştirilerek genel hava savunma ve kontrol sistemi oluşturuldu. Temmuz 1938'de kıyı boyunca dört yeni radar istasyonu daha kuruldu. Bu durumda da farklı istasyonlardan elde edilen ve genelde birbirleri ile çelişen bilginin doğrulanması ve eşgüdümü sorunu ortaya çıktı.

Sorunun çözümü için ve yapılan işlerin etkinliğinin ölçülmesi amacıyla Bawdsey Araştırma İstasyonu'nda A.P. Rowe başkanlığında bir bilimsel grup oluşturuldu. Söz konusu askeri operasyonların araştırılması (Research into Military Operations) işlemine "Operational Research" denildi. Genişleyen çalışma grubu, 1939 yazında, Stanmore Araştırma İstasyonu'nu merkez olarak kullanmaya başladı.

Savaş sırasında Stanmore Araştırma Merkezi, Fransa'daki Alman güçlerine karşı istenen ek uçak kuvvetlerinin uygun olup olmadığını YA teknikleri kullanarak değerlendirdi ve uygun olmadığını gösteren grafiklerle o zamanki başbakan Winston Churchill'e bir sunum yaptı ve sonuçta bölgeye ek kuvvet gönderilmeyerek hava kuvvetlerinin gücünün azalması engellendi. 1941 yılında Yöneylem Araştırması Bölümü (Operational Research Section - ORS) kuruldu ve savaş bitimine kadar söz konusu grup çalışmalar yaptı.

1941 yılında kurulan Blackett önderliğindeki bu gruba yedi ayrı bilim dalından onbir bilim adamı katılmıştı: üç fizyolog, bir fizikçi, iki matematikçi, bir astrofizikçi, iki fizik matematikçisi, bir subay, bir mühendis. Savaştan sonra YA çalışmaları özellikle ABD'de askeriye dışındaki alanlarda da hızlandı

Türkiye'de ise ilk YA çalışmaları, 1 Haziran 1956'da, Alb. Fuat Uluğ'un çabaları ile Genel Kurmay'da oluşturulan yedek subaylardan oluşan Harekat Araştırması grubu ile başladı. Seferberlik ve hava savunma konularında yurtdışından alınan destek ile araştırmalar yapıldı. Ülkemizde ilk YA dersi de İTÜ Makine Fakültesinde 1960-61 ders yılında Prof. Dr. İlhami Karayalçın tarafından verildi. 1966 yılında Harekat Araştırması ismi Yöneylem Araştırması olarak değiştirildi.

2. TEMEL YA KAVRAMLARI

Örnek

Two Mines Şirketi özel bir cevher çıkardığı iki adet maden ocağına sahiptir. Ocaklarda üretilen cevher üç sınıfa ayrılır: yüksek, orta, düşük kaliteli. Şirket bir fabrikaya haftalık olarak 12 ton yüksek, 8 ton orta ve 24 ton düşük kaliteli cevher sağlamak üzere anlaşmıştır. Söz konusu iki maden ocağı (X ve Y) ayrıntıları aşağıda verilen farklı işletim özelliklerine sahiptir.

Maden	Maliyet (£'000 / gün)	Üretim (ton/gün) Yüksek	Orta	Düşük
X	180	6	3	4
Y	160	1	1	6

Anlaşmayı gerçekleştirmek için haftasonu üretim yapılmayan maden ocakları haftada kaç gün işletilmelidir?

Tahmin

Two Mines örneğini incelemek için çok basit bir şekilde yargımızı kullanarak madenlerin haftada kaç gün çalışacağına yönelik olarak fikir yürütüp sonuçları inceleyebiliriz.

- haftada bir gün X madenini, bir gün Y madenini işletme

Bu çözüm önerisi iyi bir sonuç vermeyecek gibi gözükmektedir. Sadece 7 ton yüksek kaliteli cevher üretilecek bu durumda da 12 tonluk müşteri gereksinimi karşılanamayacaktır. Böyle bir çözüme "olurlu (uygun) olmayan" (infeasible) çözüm denilir.

- haftada 4 gün X madenini, 3 gün Y madenini işletme

Bu durumda tüm müşteri gereksinimleri karşılanabilmektedir. Böyle bir çözüme de "olurlu" (feasible) çözüm denilir. Fakat söz konusu çözüm önerisinin diğer olası uygun çözüm önerilerinden daha pahalıya mal olup olmadığının araştırılması için yeni kombinasyonlar denenmelidir.

Çözüm

Yapmamız gereken Two Mines örneğini sözel olarak ifade edip, söz konusu ifadeyi matematiksel bir tanıma çevirmektir.

Bu tipte sorunları çözmeye uğraşırken öncelikle aşağıdaki kavramları belirlemeliyiz:

- değişkenler (variables)
- kısıtlar (constraints)
- amaç.(objective)

Bu belirleme sürecine "formülasyon" ya da daha resmi bir şekilde sorunun matematiksel modelinin formülasyonu denilir.

Değişkenler

Bunlar verilmesi gereken kararları veya bilinmeyenleri temsil eder. İncelenen sorunda iki adet karar değişkeni (decision variable) vardır:

x = Bir haftada X maden ocağının işletileceği gün sayısı

y = Bir haftada Y maden ocağının işletileceği gün sayısı

Doğal olarak x >= 0 ve y >= 0 olacaktır

Kısıtlar

Kısıt, soruna özgü durumların getirdiği sınırlamalardır. Kısıt belirlemenin en iyi yolu önce sınırlayıcı durumları sözel olarak ifade edip daha sonra değişkenleri kullanıp matematiksel biçimde yazmaktır:

cevher üretim kısıdı – üretilen cevher ile müşteri gereksiniminin dengelenmesi

Cevher çeşitleri

Yüksek	6x + 1y >= 12
Orta	3x + 1y >= 8
Düşük	4x + 6y >= 24

Eşitsizlik kullanma "en iyileme" (optimization) sorunlarındaki kısıtlarda esneklik sağlar.

haftasonu kısıdı - haftada 5 günden fazla çalışılamaz

x <= 5

y <= 5

Amaç

Şirketin amacı toplam maliyeti (180x + 160y) en az seviyede tutarak müşteri gereksinimlerini karşılamaktır.

Ele alınan sorunda tüm olası olurlu çözümlerden amaç fonskiyonu değerini enküçükleyen karar değişkeni değerlerini barındıran çözüm en iyi çözümdür.

Sorunun amacının kar enbüyüklemesi olması durumunda en iyi çözüm amaç fonksiyonu değerini en büyük yapan değer olacaktır.

Genel olarak, tüm olası olurlu çözümlerden amaç fonskiyonu değerini en iyi hale getiren karar değişkeni değerlerini barındıran çözüme "en iyi" (optimum) çözüm denilir.

Sonuç olarak tüm kavramları birarada yazarak matematiksel modeli aşağıdaki gibi yazabiliriz:

enküçükle (minimize)

180x + 160y

öyle ki (subject to)

6x + y >= 12

3x + y >= 8

4x + 6y >= 24

x <= 5

y <= 5

x,y >= 0

Matematiksel model aşağıdaki biçimde olmalıdır:

- tüm değişkenler süreklidir (continuous)
- tek bir amaç vardır (enbüyükleme (maximize) veya enküçükleme (minimize))
- amaç ve kısıt fonksiyonları doğrusaldır. Fonksiyondaki her terim ya sabit sayıdır ya da bir sabitle çarpılmış değişkendir (örneğin 24, 0, 4x, 6y doğrusal terimlerdir fakat xy, x^2 doğrusal değildir).

Yukarıdaki üç koşulu sağlayan herhangi bir formülasyon bir "Doğrusal Program"dır (DP; linear program - LP).

Bir sorunu DP ile incelediğimizde yukarıdaki koşullara uymak için bazı varsayımlar yaparız. Ele aldığımız örnekte haftalık çalışma gün sayısının kesirli olabileceği (tam sayı olmak zorunda olmaması) gibi. Aslında bu tip sorunları çözmek için "Tam sayılı programlama" (*integer programming* - IP) teknikleri de kullanılabilir.

Bir diğer dikkat edilmesi gereken özellik de işaret sınırlamasıdır (sign restriction). Haftadaki gün sayısı negatif olamayacağından bir haftada bir maden ocağının işletileceği gün sayısını temsil eden karar değişkenleri büyük eşit sıfır olarak modele eklenmelidir.

Matematiksel model (formülasyon) kurulduktan sonra algoritma adı verilen sayısal bir çözüm tekniği kullanılarak amaç fonksiyonunun "en iyi" (optimum) değerini verecek (enbüyükleme sorunlarında en büyük, enküçüklemede en küçük) ve tüm kısıtları sağlayacak şekilde karar değişkeni değerleri bulunur.

"YA, gerçek hayat sistemlerinin matematiksel modellerle temsil edilmesi ve en iyi çözümü bulmak için kurulan modellere sayısal yöntemler (algoritmalar) uygulanmasıdır."

3. DOĞRUSAL PROGRAMLAMA

Two Mines örneği incelenirse, bir matematiksel modelin bir "Doğrusal Program" (DP; linear program - LP) olması için aşağıdaki koşulları sağlaması gerektiği görülür:

- Tüm değişkenler süreklidir (continuous)
- Tek bir amaç vardır (enbüyükleme (maximize) veya enküçükleme (minimize))
- Amaç ve kısıt fonksiyonları doğrusaldır. Fonksiyondaki her terim ya sabit sayıdır ya da bir sabitle çarpılmış değişkendir

DP'ler önemlidir çünkü:

- çok sayıda sorun DP olarak formüle edilebilir
- "Simpleks algoritması" kullanılarak DP'ler çözülebilir ve en iyi çözüm bulunabilir

DP'lerin temel uygulama alanlarına aşağıda çeşitli örnekler verilmiştir:

- Üretim planlama
- Rafineri yönetimi
- Karışım
- Dağıtım
- Finansal ve ekonomik planlama
- İşgücü planlaması
- Tarımsal planlama
- Gıda planlama

DP'ler için dört temel varsayım söz konusudur:

- Oransallık
    - Her karar değişkeninin amaç fonksiyonuna katkısı karar değişkeninin değeri ile orantılıdır (Dört asker üretmenin amaç fonksiyonuna (kâra) katkısı ($4\times\$3=\$12$) bir askerin amaç fonkisyonuna katkısının ($3) tam olarak dört katıdır.)
    - Her karar değişkeninin kısıtların sol tarafına katkısı karar değişkeninin değeri ile orantılıdır. (Üç asker üretmek gerekli montaj zamanı (2 saat $\times$ 3 =

6 saat) tam olarak bir asker üretmek için gerekli montaj zamanının (2 saat) üç katıdır.)

- Toplanabilirlik
    - Herhangi bir karar değişkeninin amaç fonksiyonuna katkısı diğer karar değişkenlerinin değerlerinden bağımsızdır. (Trenin (x_2) değeri ne olursa olsun, asker (x_1) üretmek her zaman amaç fonksiyonuna $3x_1$ dolar katkı yapacaktır.)
    - Herhangi bir karar değişkeninin kısıt sol tarafına katkısı diğer karar değişkenlerinin değerlerinden bağımsızdır. (x_1'in değeri ne olursa olsun, x_2 üretimi x_2 saat montaj ve x_2 saat marangozluk gerektirir.)

      Sonuç 1: Amaç fonksiyonu değeri her bir karar değişkeninin katkısının toplamına eşittir.

      Sonuç 2: Her bir kısıdın sol taraf değeri her bir karar değişkeninin katkısının toplamına eşittir.

- Bölünebilirlik

Karar değişkenleri tam sayı olmayan değerler alabilir. Eğer tam sayı değerler kullanmak şartsa TP kullanılmalıdır. (1.69 tren üretmek kabul edilebilir.)

- Kesinlik

Her parametre kesin olarak bilinmektedir.

3.1 DP'NİN FORMÜLASYONU

3.1.1 Giapetto Örneği

(Winston 3.1., s. 49)

Giapetto tahtadan oyuncak asker ve tren yapmaktadır. Satış fiyatları, bir oyuncak asker için $27, bir oyuncak tren için $21'dır. Bir asker için $10'lık hammadde ve $14'lık işçilik kullanılmaktadır. Bir tren için ise söz konusu rakamlar sırasıyla $9 ve $10'dır. Her bir asker için 2 saat montaj ve 1 saat marangozluk gerekirken, her bir tren için 1 saat montaj ve 1 saat marangozluk gerekmektedir. Eldeki hammadde miktarı sınırsızdır, fakat haftada en çok 100 saat montaj ve 80 saat marangozluk kullanabilen Giapetto'nun haftada en fazla 40 oyuncak asker satabileceğini göz önünde bulundurarak karını enbüyüklemek için hangi oyuncakdan haftada kaç adet üretmesi gerektiğini bulunuz.

Yanıt

Karar değişkenleri tam olarak verilmesi gereken (bu sorunda Giapetto tarafından) kararları tanımlamalıdır. Giapetto bir haftada kaç oyuncak asker ve tren yapacağına karar vermelidir. Bu karara göre aşağıdaki karar değişkenleri tanımlanabilir:

x_1 = bir haftada üretilen asker sayısı,

x_2 = bir haftada üretilen tren sayısı,

Amaç fonksiyonu karar değişkenlerinin bir fonksiyonudur. Gelir veya karını enbüyüklemek ya da maliyetini enküçüklemek isteyen karar vericinin amacını yansıtır. Giapetto haftalık karını (z) enbüyüklemek isteyecektir.

Bu sorunda kar

(haftalık gelir) – (hammadde satınalma maliyeti) – (diğer değişken maliyetler)

olarak formüle edilebilir. Bu durumda Giapetto'nun amaç fonksiyonu:

Enbüyükle $z = 3x_1 + 2x_2$

Kısıtlar karar değişkenlerinin alabilecği değerler üzerindeki, sınırlamaları gösterir. Herhangi bir sınırlama olmazsa Giapetto çok fazla sayıda oyuncak üreterek çok büyük kar elde edebilir. Fakat gerçek hayatta olduğu gibi burada da kısıtlar vardır

Haftalık kullanılabilen montaj işçiliği zamanı

Haftalık kullanılabilen marangozluk zamanı

Askerler için haftalık talep

İşaret sınırlamaları da eğer karar değişkenleri salt negatif olmayan değerler alıyorsa kullanılmalıdır (Giapetto negatif sayıda asker veya tren üretemez!).

Yukarıdaki tüm bu özellikler aşağıdaki **Doğrusal Programlama** (DP; Linear Programming - LP) sorununu verir:

Maks $z = 3x_1 + 2x_2$	(Amaç fonksiyonu)
s.t. $2x_1 + x_2 \leq 100$	(Montaj kısıdı)
$x_1 + x_2 \leq 80$	(Marangozluk kısıdı)
$x_1 \leq 40$	(Talep kısıdı)
$x_1, x_2 \geq 0$	(İşaret sınırlamaları)

Eğer (x_1,x_2)'nin bir değeri (bir çözüm) tüm bu kısıtları ve işaret sınırlamalarını sağlarsa, söz konusu çözüm **olurlu bölge**dedir (feasible region).

Grafik olarak ya da hesaplayarak sorun çözüldüğünde olurlu bölgedeki çözümlerden amaç fonksiyon değeri en yüksek olan çözümün $(x_1,x_2) = (20,60)$ olduğunu ve $z = 180$ değerini verdiğini buluruz. Bu çözüm **en iyi çözüm**dür (optimal solution).

<u>Rapor</u>

Haftada 20 asker ve 60 tren üretilmesi durumunda kar $180 olacaktır. Kar miktarları, eldeki işçilik ve talebe göre elde edilebilecek en büyük kar budur. Daha fazla işçilik bulunursa kar çoğalabilir.

3.1.2 Reklam Örneği

(Winston 3.2, s. 61)

Dorian şirketi, yüksek gelirli müşterileri için otomobil ve jeep üretmektedir. Televizyondaki tiyatro oyunlarına ve futbol maçlarına bir dakikalık spot reklamlar vererek satışlarını arttırmayı hedeflemektedir. Tiyatro oyununa verilen reklamın maliyeti $50bin'dir ve hedef kitledeki 7 milyon kadın ve 2 milyon erkek tarafından seyredilebilir. Futbol maçına verilen reklamın maliyeti ise $100bin'dir ve hedef kitledeki 2 milyon kadın ve 12 milyon erkek tarafından seyredilebilir. Dorian yüksek gelirli 28 milyon kadın ve 24 milyon erkeğe en az maliyetle nasıl ulaşır?

Yanıt: Karar değişkenleri aşağıdaki gibi belirlenebilir:

x_1: tiyatro oyununa verilen reklam sayısı

x_2: futbol maçına verilen reklam sayısı

Sorunun modeli:

$$\begin{array}{llcccl}
\text{min} & z = 50x_1 & + & 100x_2 & & \\
\text{öyle ki} & 7x_1 & + & 2x_2 & \geq & 28 \\
 & 2x_1 & + & 12x_2 & \geq & 24 \\
 & & & x_1, \; x_2 & \geq & 0
\end{array}$$

Grafik çözüm yapılırsa $(x_1,x_2) = (3.6,1.4)$ değerleri için amaç fonksiyonunun en iyi değeri z = 320 olarak bulunur. Grafiğe bakılarak en iyi tamsayılı çözüm $(x_1,x_2) = (4, 2)$ olarak bulunabilir.

Rapor: Hedeflenen kitleye ulaşmak için en az maliyetli çözüm 4 adet reklamı tiyatro oyununda ve 2 adet reklamı futbol maçında kullanmak gerekir. Bu durumda Dorian $400bin reklam masrafı yapacaktır.

3.1.3 Beslenme Örneği

(Winston 3.4., s. 70)

Bayan Fidan dört "temel gıda grubu" ile beslenmektedir: kek, çikolatalı dondurma, kola, ananaslı pasta. Bir adet kek \$0.5'a, bir kaşık dondurma \$0.2'a, bir şişe kola \$0.3'a ve bir dilim pasta \$0.8'a satılmaktadır. Her gün en az 500 kalori, 6 oz. çikolata, 10 oz. şeker ve 8 oz. yağ alması gereken Bayan Fidan en az maliyetle bu gereksinimlerini nasıl karşılar? Aşağıdaki tabloyu kullanarak bir DP modeli kurup sorunu çözünüz.

	Kalori	Çikolata (ounce)	Şeker (ounce)	Yağ (ounce)
Kek (1 adet)	400	3	2	2
Çikolatalı dondurma (1 kaşık)	200	2	2	4
Kola (1 şişe)	150	0	4	1
Ananaslı pasta (1 dilim)	500	0	4	5

Yanıt

Karar değişkenleri:

x_1: günlük yenilecek kek sayısı

x_2: günlük yenilecek kaşık dondurma sayısı

x_3: günlük içilecek şişe kola sayısı

x_4: günlük yenilecek dilim pasta sayısı

şeklinde belirlenebilir.

Bu durumda amaç fonksiyonu (cent cinsinden toplam günlük maliyet):

$$\min w = 50 x_1 + 20 x_2 + 30 x_3 + 80 x_4$$

Kısıtlar:

$$400 x_1 + 200 x_2 + 150 x_3 + 500 x_4 \geq 500 \qquad \text{(günlük kalori)}$$

$$3 x_1 + 2 x_2 \geq 6 \qquad \text{(günlük çikolata)}$$

$$2 x_1 + 2 x_2 + 4 x_3 + 4 x_4 \geq 10 \qquad \text{(günlük şeker)}$$

$$2 x_1 + 4 x_2 + x_3 + 5 x_4 \geq 8 \qquad \text{(günlük yağ)}$$

$$x_i \geq 0,\ i = 1, 2, 3, 4 \qquad \text{(işaret sınırlamaları!)}$$

Rapor:

Bayan Fidan günde 3 kaşık dondurma yiyip 1 şişe kola içerek tüm besin gereksinimlerini karşılayabilir ve sadece 90 cent harcar ($w=90$, $x_2=3$, $x_3=1$).

3.1.4 Postane Örneği

(Winston 3.5., s. 74)

Bir postanede haftanın her günü farklı sayıda elemana gereksinim duymaktadır. Sendika kurallarına göre bir eleman 5 gün peş peşe çalışmakta diğer iki gün izin yapmaktadır. Çalıştırılması gereken toplam en az eleman sayısını aşağıdaki iş yüküne göre hesaplayınız.

	Pzt	Sal	Çar	Per	Cum	Cmt	Paz
Gerekli eleman	17	13	15	19	14	16	11

Yanıt: Karar değişkenleri x_i (i. gün çalışmaya başlayan eleman sayısı) olsun

Matematiksel olarak DP modeli aşağıdaki gibi oluşturulabilir:

$$
\begin{array}{llllllllll}
\min\ z = & x_1 & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & & \\
& x_1 & & & +x_4 & +x_5 & +x_6 & +x_7 & \geq & 17 \\
& x_1 & +x_2 & & & +x_5 & +x_6 & +x_7 & \geq & 13 \\
& x_1 & +x_2 & +x_3 & & & +x_6 & +x_7 & \geq & 15 \\
& x_1 & +x_2 & +x_3 & +x_4 & & & +x_7 & \geq & 19 \\
& x_1 & +x_2 & +x_3 & +x_4 & +x_5 & & & \geq & 14 \\
& & +x_2 & +x_3 & +x_4 & +x_5 & +x_6 & & \geq & 16 \\
& & & +x_3 & +x_4 & +x_5 & +x_6 & +x_7 & \geq & 11
\end{array}
$$

Rapor: (x_i) = (4/3,10/3,2,22/3,0,10/3,5), z = 67/3 şeklindedir. Karar değişkeni değerleri yakın tamsayılara yuvarlanırsa (x_i) = (2,4,2,8,0,4,5), z = 25 çözümü bulunur: yanlış olabilir! Elde edilen Tamsayılı Lindo çözümüne göre ise amaç fonksiyonun en iyi değeri z = 23'dür ve (x_i) = (4,4,2,6,0,4,3) şeklindedir.

3.1.5 Sailco Örneği

(Winston 3.10., s. 99)

Sailco şirketi gelecek dört mevsimde kaç adet yelkenli üreteceğine karar verecektir. Talep sırasıyla 40, 60, 75 ve 25 yelkenlidir. Sailco tüm talepleri zamanında karşılamalıdır. Başlangıçta Sailco'nun envanterinde 10 yelkenli vardır. Normal mesai ile bir mevsimde 40 yelkenli üretebilen şirket yelkenli başına $400 işçilik maliyetine maruz kalmaktadır. Fazla mesai ile yapılan her ek yelkenli için ise işçilik maliyeti $450'dır. Herhangi bir mevsimde yapılan yelkenli ya talebi karşılamak için kullanılıp satılır ya da envantere konulur. Bir yelkenlinin bir mevsim envanterde tutulması durumunda ise $20 envanter taşıma maliyeti oluşmaktadır

Yanıt:

t = 1,2,3,4 için karar değişkenleri

x_t = t. mevsimde normal mesai ile üretilen yelkenli sayısı

y_t = t. mevsimde fazla mesai ile üretilen yelkenli sayısı

Envanter hesaplarının yapılabilmesi için:

i_t = t. mevsimin sonunda envanterdeki yelkenli sayısı

d_t = t. dönem için yelkenli talebi

Veri $\quad x_t \leq 40, \forall t$

Mantıksal olarak $\quad i_t = i_{t-1} + x_t + y_t - d_t, \forall t.$

Talep karşılanmalı $\quad i_t \geq 0, \forall t$

(İşaret sınırlamaları $\quad x_t$ and $y_t \geq 0, \forall t$)

Bu kısıt kümelerini kullanarak toplam maliyet z'yi enküçüklemeliyiz:

$$z = 400(x_1+x_2+x_3+x_4) + 450(y_1+y_2+y_3+y_4) + 20(i_1+i_2+i_3+i_4)$$

Rapor:

Lindo en iyi .çözümü $(x_1, x_2, x_3, x_4) = (40, 40, 40, 25)$, $(y_1, y_2, y_3, y_4) = (0, 10, 35, 0)$ ve toplam maliyet = $78450.00 olarak verir. Üretim çizelgesi:

		M1	M2	M3	M4
Normal mesai (x_t)		40	40	40	25
Fazla mesai (y_t)		0	10	35	0
Envanter (i_t)	10	10	0	0	0
Talep (d_t)		40	60	75	25

3.1.6 Müşteri Hizmet Düzeyi Örneği

(Winston 3.12, s. 108)

Bir bilgisayar şirketinde müşteri hizmetleri için deneyimli uzmana olan talep (adamsaat/ay) aşağıdaki gibidir:

t	Ocak	Şub	Mart	Nis	May
d_t	6000	7000	8000	9500	11000

Ocak ayı başında şirkette 50 deneyimli uzman vardır. Her uzman ayda 160 saat çalışabilir. Yeni bir uzmanı yetiştirmek için deneyimli uzmanlar 50 saat ayırmaktadır ve söz konusu uzmanın eğitimi bir ayda tamamlanmaktadır. Her deneyimli uzmana ayda \$2000, her yeni uzmana ise ayda \$1000 ödenmektedir. Her ay deneyimli uzmanların %5'i işten ayrılmaktadır. Şirket hem hizmet talebini karşılamak istemekte hem de maliyetleri enazlamak istemektedir. Sorunu çözmek için DP modeli kurunuz.

Yanıt: Karar değişkenleri:

x_t = t ayında eğitilecek uzman sayısı

İşlem yapabilmek için kullanılan diğer değişkenler ise

y_t = t. ayın başında şirketteki deneyimli uzman sayısı

d_t = t. ayın hizmet talebi

Bu durumda $z = 2000\ (y_1+...+y_5) + 1000\ (x_1+...+x_5)$ amaç fonksiyonu enküçüklenmelidir.

Kısıtlar:

$$160y_t - 50x_t \geq d_t, \quad t = 1,...5$$

$$y_1 = 50$$

$$y_t = .95y_{t-1} + x_{t-1}, \quad t = 2,3,4,5$$

$$x_t,\ y_t \geq 0$$

3.2 DP'NİN ÇÖZÜMÜ

3.2.1 Çözüm Türleri

Bir DP çözüldüğü zaman aşağıdaki dört durumdan biri ile karşılaşılır:

1. DP'nin **bir tek en iyi çözüm**ü vardır.
2. DP'nin **alternatif (çok sayıda) en iyi çözüm**leri vardır. Birden fazla (aslında sonsuz sayıda) en iyi çözüm bulunur.
3. DP **olurlu değildir** (infeasible). Hiç olurlu çözümü yoktur (Olurlu bölgede nokta yoktur).
4. DP **sınırlı değildir** (unbounded). Olurlu bölgedeki noktalar sonsuz büyüklükte amaç fonksiyon değeri vermektedir.

Grafik olarak,

- Bir tek en iyi çözüm varsa, eş kar doğrusu olurlu bölgeyi terkederken bir köşe (vertex - corner) ile kesişir.
- Alternatif çözümler söz konusu ise, eş kar doğrusu olurlu bölgeyi terkederken bir doğru (bir kısıdı temsil eden) ile kesişir.
- Olurlu çözüm vermeme durumunda, kısıtların sağladığı bölgelerin kesişimi dolayısıyla olurlu bölge yoktur.
- Sınırlı olmama durumunda ise eş kar doğrusu olurlu bölgeyi terkedememektedir.

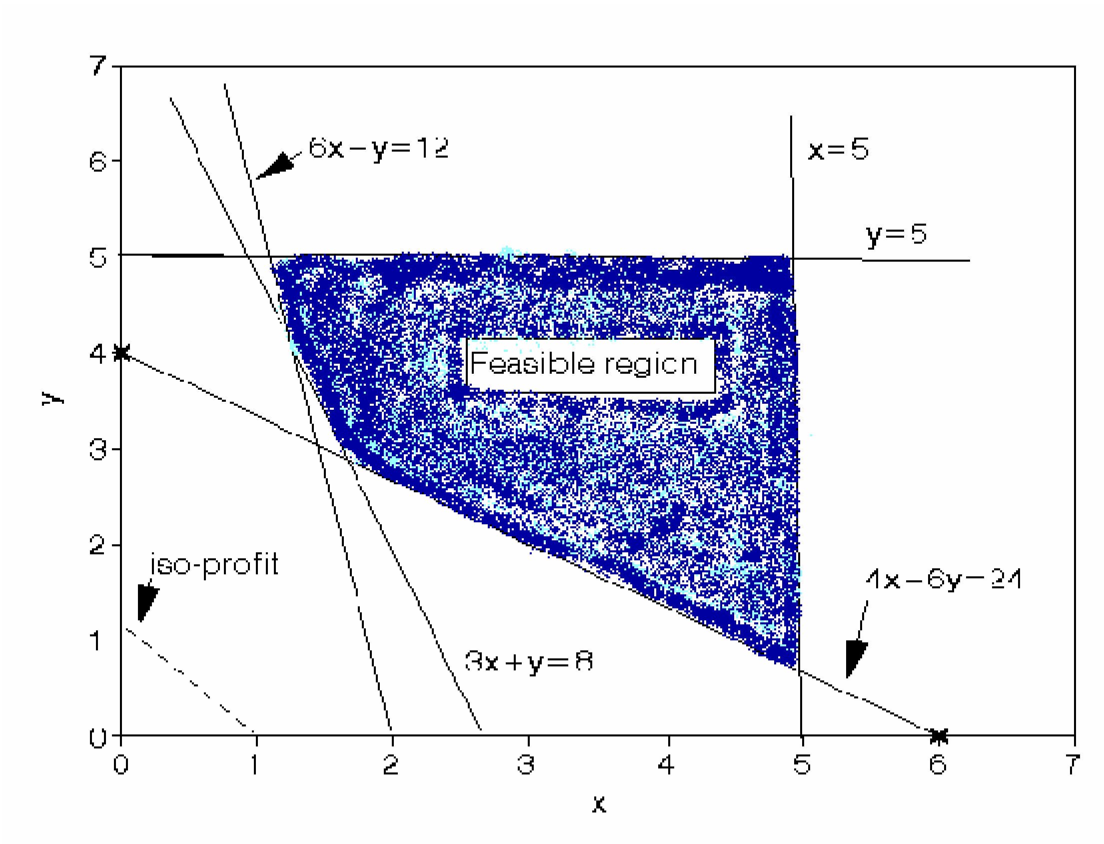
3.2.2 Grafik Çözüm (Enküçükleme)

min 180x + 160y

öyle ki

6x + y >= 12
3x + y >= 8
4x + 6y >= 24
x <= 5
y <= 5
x,y >= 0

Sadece iki karar değişkeni olduğundan iki eksenli bir grafik üzerinde çözüme gidilebilir. Öncelikle kısıtları temsil eden doğrular çizilip kısıtları sağlayan bölgelerin kesişimini veren **olurlu bölge** (feasible region) belirlenir.



Daha sonra amaç fonksiyonunu 180x + 160y = K (K sabit) şeklinde yazarak herhangi bir sabit için (grafikte K = 180) elde ettiğimiz doğruyu grafiğe çizeriz. Bu doğrulara **eş kar** doğrusu (iso-profit line) denilir. Enküçükleme sorunlarında K'nın olurlu bölgede alabileceği en küçük değere göre doğru çizildiği düşünülerek söz konusu doğrunun olurlu bölgeyle kesiştiği nokta aranır. Bu noktadaki x, y koordinatları bize **en iyi** (optimum) çözümün karar değişken değerlerini verir: 3x + y = 8 ve 4x + 6y = 24 doğrularının kesiştiği nokta (1.71, 2.86).

En iyi çözüm 765.71'dir (1.71 gün x, 2.86 gün y madeni çalıştırılmalıdır)

3.2.3 Grafik Çözüm (Enbüyükleme)

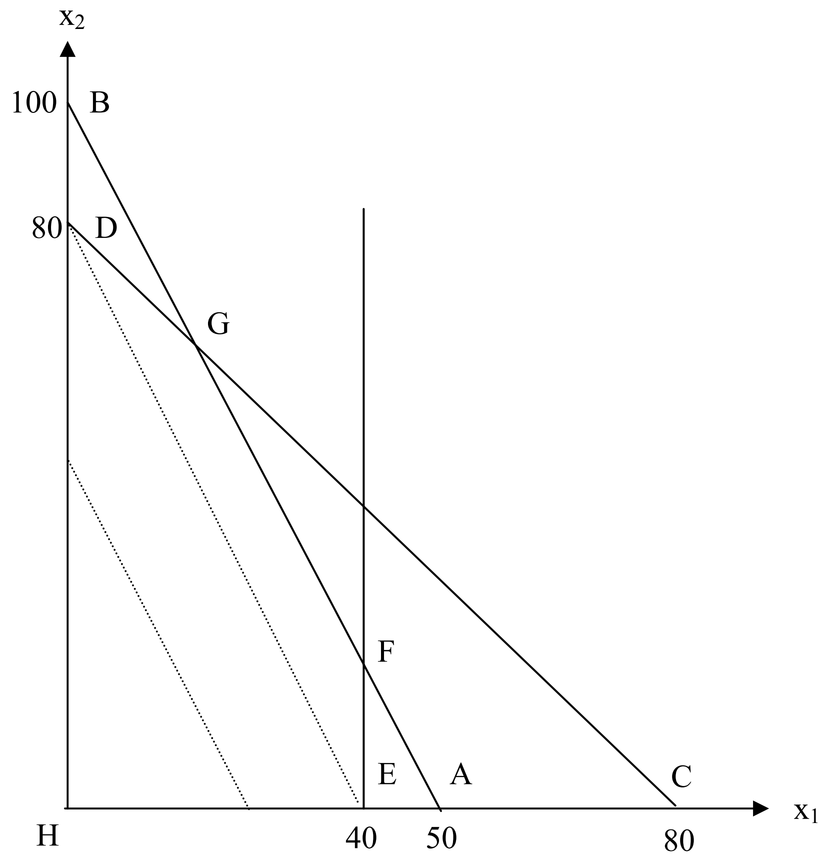
Maks $z = 4x_1 + 2x_2$ (Amaç fonksiyonu)

. $2x_1 + x_2 \leq 100$ (Montaj işçiliği kısıdı)

$x_1 + x_2 \leq 80$ (Marangozluk kısıdı)

$x_1 \leq 40$ (Talep kısıdı)

$x_1 + x_2 \geq 0$ (İşaret sınırlamaları)



G (20, 60) ve F (40, 20) noktaları arasındaki noktalar (doğru) **alternatif en iyi çözüm**leri, verir. $0 \leq c \leq 1$ için:

$c\ [20\ 60] + (1-c)\ [40\ 20] = [40-20c\ \ 20+40c]$

en iyi çözümdür.

****

Kısıt ekleyelim: $x_2 \geq 90$ (Tren talebi kısıdı)

Olurlu bölge yoktur: **olurlu olmayan DP**

****

Sadece tek kısıt olsun: $x_2 \geq 90$

Eş kar doğrusu olurlu bölgeyi terkedemez: **sınırlı olmayan DP**

3.2.4 Simpleks Algoritması

Tüm DP sorunlarının (ikiden fazla sayıda karar değişkeni olanların da) en iyi çözümü olurlu bölgenin bir köşesindedir. Simpleks algoritması bu gerçeği kullanarak çözüme gider.

Başlangıçta olurlu bölgenin bir köşesi ile işleme başlanır ve eğer söz konusu köşe en iyi çözümü vermezse yeni bir adım (iterasyon) işletilerek amaç fonksiyonunu iyileştiren (veya aynı bırakan) başka bir komşu köşeye geçilir. Bu adımlar en iyi DP çözümü bulununcaya kadar sürer.

DP'leri çözmek için kullanılan simpleks algoritması Dantzig tarafından 1940'lı yılların sonunda geliştirilmiştir. Daha sonra algoritma geliştirilip yeni versiyonları geliştirilmiştir. Bunlardan biri olan "revised simpleks algoritması" DP çözümü için kullanılan bilgisayar paketlerinde kullanılmaktadır.

Adımlar

1. DP'yi standart biçime çeviriniz
2. Bir temel olurlu çözüm (basic feasible solution - bfs) bulunuz
3. Mevcut bfs'nin en iyi çözüm olup olmadığını araştırınız. En iyi ise sorun çözülmüştür, durunuz.
4. Mevcut bfs en iyi çözüm değilse, amaç fonksiyon değerini en çok iyileştirmek için hangi temel dışı değişkenin temel değişken olacağını (çözüme gireceğini) saptayınız.
5. Hangi değişkenin çözümden çıkıp temel dışı değişken olacağını oran testi ile bulunuz. Elde edilen pivot denklemi satır işlemlerinde kullanarak yeni bir bfs bulunuz.
6. Adım 3'e dönünüz.

İlgili kavramlar:

- Standart biçim: tüm kısıtlar eşitliktir ve tüm değişkenler negatif olmayan değerler alır
- bfs: tüm değişkenlerin negatif olmayan değerler aldığı bir olurlu çözüm
- Temel dışı değişken: bfs'de değerleri 0'a eşit olan değişkenler
- Temel değişken: bfs'deki diğer değişkenler, standart biçimdeki eşitliklerin çözülmesi ile 0'dan büyük değerler alırlar

3.2.5 Simpleks için Örnek (Dakota Mobilya)

(Winston 4.3, s. 134)

Dakota mobilya şirketi sıra, masa ve sandalye yapmaktadır. Her ürün için, aşağıdaki tabloda görüldüğü gibi, sınırlı miktarda kullanılabilen tahta, marangozluk ve montaj işçiliği gerekmektedir. Aynı tabloda ürünlerin satış fiyatları da verilmiştir. Haftada en fazla 5 masa satılabilmektedir. Haftalık karı enbüyükleyecek bir üretim planı oluşturunuz.

Kaynak	Sıra	Masa	Sandalye	Kullanılabilen.
Tahta (m^2)	8	6	1	48
Montaj işçiliği	4	2	1.5	20
Marangozluk	2	1.5	.5	8
Talep (maks)	-	5	-	
Fiyat ($)	60	30	20	

DP Modeli:

x_1, x_2, x_3 bir haftada üretilen sıra, masa ve sandalye sayısı olsun. z ise Dakota'nın haftalık kar miktarını göstersin. Aşağıdaki DP'yi formüle edebiliriz

$$\text{maks } z = 60x_1+30x_2+20x_3$$

$$8x_1+6x_2+x_3 \leq 48$$

$$4x_1+2x_2+1.5x_3 \leq 20$$

$$2x_1+1.5x_2+.5x_3 \leq 8$$

$$x_2 \leq 5$$

$$x_1,x_2,x_3 \geq 0.$$

Simpleks algoritması ile çözüm

1. Öncelikle gevşek (slack) değişkenler kullanarak DP modelini standart biçime getiriniz ve modeli kanonik bir şekilde yazınız.

$$
\begin{array}{llllllllllll}
\text{maks } z = & 60x_1 & + & 30x_2 & + & 20x_3 & & & & & & \\
\text{s.t.} & 8x_1 & + & 6x_2 & + & x_3 & +s_1 & & & & = & 48 \\
 & 4x_1 & + & 2x_2 & + & 1.5x_3 & & +s_2 & & & = & 20 \\
 & 2x_1 & + & 1.5x_2 & + & .5x_3 & & & +s_3 & & = & 8 \\
 & & & x_2 & & & & & & +s_4 & = & 5
\end{array}
$$

$$x_1,x_2,x_3,s_1,s_2,s_3,s_4 \geq 0.$$

$$
\begin{array}{lllllllll}
R_0 & z & -60x_1 & -30x_2 & -20x_3 & & & & = 0 \\
R_1 & & 8x_1 & + 6x_2 & + x_3 & +s_1 & & & = 48 \\
R_2 & & 4x_1 & + 2x_2 & +1.5x_3 & +s_2 & & & = 20 \\
R_3 & & 2x_1 & +1.5x_2 & + .5x_3 & & +s_3 & & = 8 \\
R_4 & & & x_2 & & & & +s_4 & = 5
\end{array}
$$

2. Bir başlangıç temel olurlu çözümü bulunuz

Sorun için (x_1, x_2, x_3) = 0 çözümü olurlu olduğundan, aşağıda verilen nokta bir başlangıç temel olurlu çözümdür (basic feasible solution – bfs):

$x_1 = x_2 = x_3 = 0$, $s_1 = 48$, $s_2 = 20$, $s_3 = 8$, $s_4 = 5$.

Bu bfs'de üç karar değişkeni ***temel dışı değişken*** (non-basic variables) ve üç gevşek değişken de ***temel değişkendir*** (basic variables) ve değerleri kanonik modeldeki eşitliklerden bulunur.

.

3. Mevcut bfs'nin en iyi çözüm olup olmadığını kontrol ediniz

Temel dışı herhangi bir değişkenin değerinin çoğaltılması (temele girmesi) ile z'nin değerinin iyileşmesinin mümkün olup olmadığı araştırılır.

Eğer tüm temel dışı değişkenlerin amaç fonksiyon satırındaki (***0. satır; row 0 – R_0***) katsayıları 0 ya da 0'dan büyükse (nonnegative), mevcut bfs en iyi (optimal) çözümdür (z'nin değeri daha çok iyileştirilemez) .

Fakat örnekte tüm temel dışı değişkenlerin 0. satırdaki katsayıları negatiftir.

4. Yeni bfs'nin bulunması

- Enbüyüklenmek istenen z en çok x_1 sıfırdan farklı yapıldığı zaman çoğalır: x_1 ***giren değişken***dir
- R_1 incelendiğinde x_1'in en fazla 6 olabileceği görülür. Aksi takdirde $s_1 < 0$ olacaktır. Benzer şekilde R2 ve R3 sırasıyla 5 ve 4 sınırlarını verir. Son satırda x1 olmadığından herhangi bir sınırlama söz konusu değildir. Bu durumda tüm sınırlamaların (aslında sağ taraf değerlerinin giren değişken katsayılarına "oran"larının – ***oran testi***) en küçüğü olan 4, x1'in alabileceği en büyük değerdir. x1 = 4 olduğunda s3 = 0 olup çözümden çıkar ve ***çıkan değişken*** olarak isimlendirilir.
- R_3 de ***pivot denklem*** olur. x_1 temel değişken olduğu için birim matrise girecek şekilde sistem yeniden düzenlenir.

Yeni *pivot denklem* (R_3/2):

$$R_3': \quad x_1+.75x_2+.25x_3+ \quad .5s_3 \quad = 4$$

R_3' kullanılarak x_1 tüm diğer satırlarda yok edilir.

R_0'	$z \quad +15x_2 \quad -5x_3 \quad +30s_3 = 240$	$z = 240$	(R_0+60R_3')
R_1'	$-x_3 \quad +s_1 \quad -4s_3 = 16$	$s_1 = 16$	(R_1-8R_3')
R_2'	$-x_2 \quad +.5x_3 \quad + \quad s_2-2s_3 = 4$	$s_2 = 4$	(R_2-4R_3')
R_3'	$x_1+.75x_2 \quad +.25x_3 \quad +.5s_3 = 4$	$x_1 = 4$	(R_3')
R_4'	$x_2 \quad + \quad s_4 = 5$	$s_4 = 5$	(R_4)

Yeni bfs $x_2=x_3=s_3=0$, $x_1=4$, $s_1=16$, $s_2=4$, $s_4=5$ şeklindedir ve z=240 olur

5. Adım 3'e geri dönünüz. En iyi çözümü bulunana kadar adımları tekrar ediniz

- x_3 girer.
- Oran testi sonucu $x_3 = 8$ bulunur; s_2 çıkar: İkinci satır pivot denklem olur.
- Pivot denklemde (R_2') giren değişkenin katsayısı 1 yapılır:

$$R_2'' \quad -2x_2+x_3+2s_2-4s_3 = 8 \quad (R_2' \times 2).$$

R_2'' satır işlemleri ile diğer satırlarda giren değişken yok edilir:

R_0''	z	$+5x_2 \qquad + \ 10s_2+10s_3$	$= 280$	$z = 280$	$(R_0'+5R_2'')$
R_1''		$-2x_2 \qquad + \ s_1 \quad 2s_2- \ 8s_3$	$= 24$	$s_1 = 24$	$(R_1'+R_2'')$
R_2''		$-2x_2 \quad +x_3 \ + \quad 2s_2- \ 4s_3$	$= 8$	$x_3 = 8$	(R_2'')
R_3''	$x_1+1.25x_2$	$- \ .5s_2+1.5s_3$	$= 2$	$x_1 = 2$	$(R_3'-.5R_2'')$
R_4''		$x_2 \qquad + \qquad s_4$	$= 5$	$s_4 = 5$	(R_4')

Yeni bfs: $x_2=s_2=s_3=0$, $x_1=2$, $x_3=8$, $s_1=24$, $s_4=5$; $z = 280$.
Sıfırıncı satırdaki tüm temel dışı değişkenlerin katsayısı pozitiftir.
MEVCUT ÇÖZÜM EN İYİ ÇÖZÜMDÜR (OPTIMAL SONUÇ)

Rapor: Dakota mobilya şirketi haftalık karını enbüyüklemek için 2 sıra ve 8 sandalye üretmelidir. Bu durumda 280$ kar eder.

Simpleks algoritması tablolarla gösterilirse

(***Siz de tüm ödev ve sınavlarda her işlem için tablo kullanın!!!***)

Başlangıç tablosu:

z	x_1	x_2	x_3	s_1	s_2	s_3	s_4	ST	TD
1	-60	-30	-20	0	0	0	0	0	z = 0
	8	6	1	1	0	0	0	48	s_1 = 48
	4	2	1.5	0	1	0	0	20	s_2 = 20
	2	1.5	.5	0	0	1	0	8	s_3 = 8
	0	1	0	0	0	0	1	5	s_4 = 5

İlk adım:

z	x_1	x_2	x_3	s_1	s_2	s_3	s_4	ST	TD
1		15	-5			30		240	z = 240
			-1	1		-4		16	s_1 = 16
		-1	.5		1	-2		4	s_2 = 4
	1	.75	.25			.5		4	x_1 = 4
		1					1	5	s_4 = 5

İkinci adım ve en iyi tablo:

z	x_1	x_2	x_3	s_1	s_2	s_3	s_4	ST	TD
1	0	5	0	0	10	10	0	280	z = 280
	0	-2	0	1	2	-8	0	24	s_1 = 24
	0	-2	1	0	2	-4	0	8	x_3 = 8
	1	1.25	0	0	-.5	1.5	0	2	x_1 = 2
	0	1	0	0	0	0	1	5	s_4 = 5

3.2.6 Simpleks ile Diğer Çözüm Türleri

Alternatif En İyi Çözüm

Dakota örneğini $35/masa olarak değiştirelim

Yeni $z = 60 x_1 + 35 x_2 + 20 x_3$

Yeni sorun için ikinci ve en iyi (optimal) tablo:

z	x_1	x_2 ⇓	x_3	s_1	s_2	s_3	s_4	ST	TD	Oran	
1	0	**0**	0	0	10	10	0	280	z=280		
0	0	-2	0	1	2	-8	0	24	s_1=24	-	
0	0	-2	1	0	2	-4	0	8	x_3=8	-	
0	1	1.25	0	0	-.5	1.5	0	2	x_1=2	2/1.25	⇒
0	0	1	0	0	0	0	1	5	s_4=5	5/1	

Bir diğer en iyi tablo:

z	x_1	x_2	x_3	s_1	s_2	s_3	s_4	ST	TD
1	0	0	0	0	10	10	0	280	z=280
0	1.6	0	0	1	1.2	-5.6	0	27.2	s_1=27.2
0	1.6	0	1	0	1.2	-1.6	0	11.2	x_3=11.2
0	0.8	1	0	0	-0.4	1.2	0	1.6	x_2=1.6
0	-0.8	0	0	0	0.4	-1.2	1	3.4	s_4=3.4

Bu yüzden en iyi çözüm aşağıdaki gibidir:

$z = 280$ ve $0 \le c \le 1$ için

$$\begin{vmatrix} x_1 \\ x_2 \\ x_3 \end{vmatrix} = c \begin{vmatrix} 2 \\ 0 \\ 8 \end{vmatrix} + (1-c) \begin{vmatrix} 0 \\ 1.6 \\ 11.2 \end{vmatrix} = \begin{vmatrix} \mathbf{2c} \\ \mathbf{1.6 - 1.6c} \\ \mathbf{11.2 - 3.2c} \end{vmatrix}$$

Sınırlı Olmayan DP'ler

z	x_1	x_2	x_3 ⇓	x_4	s_1	s_2	ST	TD	Oran
1	0	2	**-9**	0	12	4	100	z=100	
0	0	1	-6	1	6	-1	20	x_4=20	Yok
0	1	1	-1	0	1	0	5	x_1=5	Yok

Oran testi yapılamadığı için çözülmek istenen DP sınırlı olmayan DP'dir.

3.2.7 Büyük M Yöntemi

Eğer bir DP'de $\geq$ veya = kısıtlar varsa, Simpleks yöntemi kullanılarak bir başlangıç temel olurlu çözümü (bfs) oluşturulamaz..

Bu durumda Büyük M (Big M) yöntemi veya İki Evreli (Two Phase) Simpleks yöntemi kullanılmalıdır.

Büyük M yöntemi Simpleks Algoritmasının bir türüdür: Soruna yapay (artificial) değişkenler de eklenerek bir bfs bulunur. DP'nin amaç fonksiyonu da sonuçta yapay değişkenlerin katsayıları 0 olacak şekilde yeniden düzenlenir.

Adımlar

1. Öncelikle tüm kısıtlar sağ taraf (ST; Right Hand Side - RHS) değerleri negatif olmayacak şekilde düzenlenir (ST değeri negatif olan kısıtlar -1 ile çarpılır. Bu çarpım sonucu eşitsizliğin yönünün değişeceğini unutmayınız!). Düzenlemelerden sonra her kısıt $\leq$, $\geq$ veya = kısıt olarak sınıflandırılır
2. Tüm kısıtlar standart biçime çevrilir. Eğer kısıt $\leq$ kısıtsa, sol tarafa simpleks yönteminde olduğu gibi gevşek değişken s_i eklenir. Eğer kısıt $\geq$ kısıtsa, sol tarafdan bir fazlalık (excess) değişken e_i çıkarılır.
3. Tüm $\geq$ or = kısıtların sol tarafına bir yapay değişken a_i eklenir. Aynı zamanda yapay değişkenler için işaret sınırlaması ($a_i \geq 0$) da eklenir.
4. M çok büyük bir sayı olsun. Eğer DP enküçükleme sorunu ise, amaç fonksiyonuna (her yapay değişken için) Ma_i eklenir. Eğer DP enbüyükleme sorunu ise, amaç fonksiyonuna (her yapay değişken için) $-Ma_i$ eklenir.
5. Her yapay değişken başlangıç temel çözümünde olacağı için amaç fonksiyonundan (0. satır) elenmelidir (katsayıları sıfır olacak şekilde düzenleme yapılmalıdır). Daha sonra simpleks algoritmasının adımları kullanılarak (M'nin büyük bir sayı olduğu unutulmadan!) çözüme gidilir.

Yukarıdaki 5 adımla düzenlenen yeni DP'nin en iyi çözümünde tüm yapay değişkenler 0'a eşit çıkarsa, esas sorunun **en iyi çözüm**ü bulunmuştur.

Eğer yeni DP'nin en iyi çözümünde en az bir yapay değişken pozitif bir değer alırsa, esas sorun **çözümsüzdür** (infeasible)!!!

3.2.8 Büyük M için örnek (Oranj Meyve Suyu)

(Winston 4.10., s. 164)

Bevco şirketi, portakal gazozu ile portakal suyunu karıştırarak Oranj ismiyle portakallı meyve suları üretmektedir. Portakal gazozunun bir ozunda 0.5 oz. şeker ve 1 mg C vitamini vardır. Portakal suyunun bir ozunda ise 0.25 oz. şeker ve 3 mg C vitamini vardır. Bevco bir oz. portakal sodası üretmek için 2¢, bir oz. portakal suyu üretmek için ise 3¢ harcamaktadır. Şirketin pazarlama bölümü Oranj'ı 10 oz.luk şişelerde satmak istemektedir. Bevco'nun her bir şişede en az 20 mg C vitamini bulunmasını ve en çok 4 oz. şeker olması şartını en az maliyetle karşılamasını sağlayınız.

DP Modeli

x_1 ve x_2 bir şişe Oranj'da bulunması gereken portakal gazozu ve portakal suyu miktarı olsun. DP modeli aşağıdaki gibi kurulur.

$$\min z = 2x_1 + 3x_2$$

$$\begin{aligned}
0.5x_1 + 0.25x_2 &\leq 4 && \text{(şeker kısıdı)} \\
x_1 + 3x_2 &\geq 20 && \text{(C vit. kısıdı)} \\
x_1 + x_2 &= 10 && \text{(10 oz'luk şişe kısıdı)} \\
x_1, x_2 &\geq 0
\end{aligned}$$

Büyük M yöntemi ile çözüm

<u>Adım 1. Tüm kısıtların ST değerleri negatif olmayacak şekilde kısıtları yeniden düzenleyiniz</u>

Tüm kısıtların ST değeri pozitiftir

<u>Adım 2. Tüm kısıtları standart biçime çeviriniz</u>

$$\begin{aligned}
z - 2x_1 - 3x_2 &= 0 \\
0.5x_1 + 0.25x_2 + s_1 &= 4 \\
x_1 + 3x_2 - e_2 &= 20 \\
x_1 + x_2 &= 10
\end{aligned}$$

tüm değişkenler ≥ 0

Adım 3. $\geq$ or = kısıtlara a_i yapay değişkenini ekleyiniz

$$
\begin{array}{lll}
z - 2x_1 - 3x_2 & = 0 & R0 \\
0.5x_1 + 0.25x_2 + s_1 & = 4 & R1 \\
x_1 + 3x_2 - e_2 + a_2 & = 20 & R2 \\
x_1 + x_2 + a_3 & = 10 & R3
\end{array}
$$

tüm değişkenler ≥ 0

Adım 4. Amaç fonksiyonuna Ma_i ekleyiniz (min. sorunu için)

$\min z = 2x_1 + 3x_2 + M a_2 + M a_3$

Sıfırıncı satır (R0) aşağıdaki gibi olacaktır:

$z - 2x_1 - 3x_2 - M a_2 - M a_3 = 0$

Adım 5. Yapay değişkenleri R0'dan eleyecek şekilde yeni R0 oluşturunuz

Yeni R0 = R0 + M * R2 + M * R3 $\Rightarrow$

$z + (2M-2)x_1 + (4M-3)x_2 - M e_2 = 30M$ Yeni R0

Başlangıç tablosu:

z	x_1	x_2 ⇓	s_1	e_2	a_2	a_3	ST	TD	Oran
1	2M-2	**4M-3**	0	-M	0	0	30M	z=30M	
0	0.5	0.25	1	0	0	0	4	s_1=4	16
0	1	$\boxed{3}$	0	-1	1	0	20	a_2=20	**20/3***
0	1	1	0	0	0	1	10	a_3=10	10

Enküçükleme sorununda, sıfırıncı satır katsayısı "en pozitif" olan değişken giren değişkendir!

İlk tablo:

z	x_1 ⇓	x_2	s_1	e_2	a_2	a_3	ST	TD	Oran
1	**(2M-3)/3**	0	0	(M-3)/3	(3-4M)/3	0	20+3.3M	z	
0	5/12	0	1	1/12	-1/12	0	7/3	s_1	28/5
0	1/3	1	0	-1/3	1/3	0	20/3	x_2	20
0	$\boxed{2/3}$	0	0	1/3	-1/3	1	10/3	a_3	**5***

En iyi tablo:

z	x_1	x_2	s_1	e_2	a_2	a_3	ST	TD
1	0	0	0	-1/2	(1-2M)/2	(3-2M)/2	25	z=25
0	0	0	1	-1/8	1/8	-5/8	1/4	s_1=1/4
0	0	1	0	-1/2	1/2	-1/2	5	x_2=5
0	1	0	0	1/2	-1/2	3/2	5	x_1=5

Rapor:

Bir şişe Oranj'da, 5 oz. portakal gazozu ve 5 oz. portakal suyu olmalıdır.

Bu durumda toplam maliyet 25¢ olacaktır.

3.3 DUYARLILIK ANALİZİ

3.3.1 İndirgenmiş Maliyet

Herhangi bir temel dışı değişkenin indirgenmiş maliyeti (reduced cost), değişkenin temel değişken olması (DP'nin en iyi çözümüne girmesi) için amaç fonksiyon katsayısında yapılacak iyileştirme miktarıdır.

Eğer bir x_k temel dışı değişkeninin amaç fonksiyon katsayısı indirgenmiş maliyet kadar iyileştirilirse, DP'nin bir tek en iyi çözümü olmaz: alternatif çözümler vardır. x_k, söz konusu çözümlerden en az birinde temel değişken; en az birinde ise temel dışı değişken konumundadır.

Eğer x_k temel dışı değişkeninin amaç fonksiyon katsayısı indirgenmiş maliyetten daha fazla iyileştirilirse, yeni DP'nin tek bir en iyi çözümüne ulaşılır ve bu çözümde x_k temel değişken olur (x_k>0).

Temel değişkenin indirgenmiş maliyeti sıfırdır (tanıma bakınız)!

3.3.2 Gölge Fiyat

DP modelinin *i*. kısıdının gölge fiyatı (shadow price), söz konusu kısıdın sağ taraf (ST; Right Hand Side - RHS) değerinin 1 birim çoğaltılması durumunda, en iyi amaç fonksiyon değerinin ne kadar iyileştiğini (enbüyükleme sorununda ne kadar arttığını, enküçükleme sorununda ne kadar azaldığını) gösterir.

Bu tanım sadece değişimden önceki çözümün değişimden sonra da aynı kalması durumunda geçerlidir!

Bir $\geq$ kısıdın gölge fiyatı her zaman 0 ya da 0'dan küçük (nonpositive); bir $\leq$ kısıdın gölge fiyatı ise her zaman 0 ya da 0'dan büyük (nonnegative) olacaktır.

3.3.3 Basit Örnek

maks z = 5 x1 + x2 + 10 x3

x1 + x3 $\leq$ 100

x2 $\leq$ 1

Tüm değişkenler $\geq$ 0

Bu çok kolay bir DP modelidir ve simpleks kullanılmadan elle de çözülebilir:

x2 = 1 (Bu değişken ilk kısıtta yoktur, bu durumda sorun enbüyükleme olduğundan ikinci kısıdın sol taraf değeri 1'e eşit olur)

x1 = 0, x3 = 100 (Bu iki değişken ise salt ilk kısıtta kullanılmışlardır ve x3'ün amaç fonksiyon değeri x1'inkinden büyük olduğu için x3'ün en iyi değeri birinci kısıt ST değerine eşit olur)

Bu durumda en iyi çözüm aşağıdaki gibidir:

z = 1001, [x1, x2, x3] = [0, 1, 100]

Aynı zamanda duyarlık analizi de elle hesaplanabilir:

İndirgenmiş Maliyet

x2 ve x3 temel değişken (en iyi çözümde) olduklarından, indirgenmiş maliyetleri 0'dır. x1'i temel değişken yapabilmek için amaç fonksiyon katsayısını en az x3'ün amaç fonksiyon katsayısı kadar yapmak diğer bir deyişle 5 (10-5) birim çoğaltmak gerekir. Yeni amaç fonksiyonu (maks z = 10 x1 + x2 + 10 x3) olacak ve [x1, x2, x3] için en az iki en iyi çözüm bulunacaktır: [0, 1, 100] ve [100, 1, 0].

Bu durumda x1'in indirgenmiş maliyeti 5'dir

Eğer x1'in amaç fonksiyon katsayısını indirgenmiş maliyet değerinden daha fazla çoğaltırsak en iyi çözüm bir tane olacaktır: [100, 1, 0].

Gölge Fiyat

Eğer birinci kısıdın ST değeri 1 birim arttırılırsa, x3'ün yeni en iyi çözüm değeri 100 yerine 101 olacaktır. Bu durumda da z'nin yeni değeri 1011 olacaktır.

Tanımdan faydalanıp tersten gidersek: 1011 - 1001 = 10, birinci kısıdın gölge fiyat değeridir.

Benzer şekilde ikinci kısıdın gölge fiyatı 1 olarak hesaplanır (hesaplayınız).

3.3.4 Duyarlılık için Lindo Çıktısının Kullanılması

DİKKAT: Simpleks'de sıfırıncı satır olan amaç fonksiyonu Lindo'da birinci satır (Row 1) olarak kabul edilir!

Bu yüzden ilk kısıt, Lindo'da her zaman ikinci satırdır!!!

```
MAX     5 X1 + X2 + 10 X3
SUBJECT TO
     2)  X1 + X3 <=   100
     3)  X2 <=   1
END


LP OPTIMUM FOUND AT STEP      1

     OBJECTIVE FUNCTION VALUE
     1)      1001.000

 VARIABLE        VALUE          REDUCED COST
      X1           0.000000           5.000000
      X2           1.000000           0.000000
      X3         100.000000           0.000000

     ROW   SLACK OR SURPLUS     DUAL PRICES
      2)           0.000000            10.000000
      3)           0.000000             1.000000

 RANGES IN WHICH THE BASIS IS UNCHANGED:

                        OBJ COEFFICIENT RANGES
 VARIABLE          CURRENT        ALLOWABLE        ALLOWABLE
                     COEF          INCREASE         DECREASE
     X1            5.000000         5.000000         INFINITY
     X2            1.000000        INFINITY          1.000000
     X3           10.000000        INFINITY          5.000000

                        RIGHTHAND SIDE RANGES
     ROW        CURRENT       ALLOWABLE        ALLOWABLE
                  RHS          INCREASE         DECREASE
      2         100.000000      INFINITY        100.000000
      3           1.000000      INFINITY          1.000000
```

Lindo çıktısı x1, x2 ve x3 değişkenlerinin indirgenmiş maliyetlerini (reduced costs) 5, 0 ve 0 olarak vermektedir.

Enbüyükleme sorunlarında temel dışı bir değişkenin indirgenmiş maliyeti aynı zamanda Lindo çıktısındaki amaç fonksiyon katsayıları aralığındaki (obj. coefficient

ranges) o değişken için izin verilen çoğalış (allowable increase) değeri ile de bulunabilir. Burada x1 için söz konusu değer 5'dir.
Enküçükleme sorunlarında ise temel dışı değişkenin indirgenmiş maliyeti izin verilen azalış (allowable decrease) değerine eşittir.

Aynı Lindo çıktısından, gölge fiyatlar (shadow proces) da kısıtların "dual price" değerleri okunarak bulunabilir:
Örneğimizde birinci kısıdın (satır 2) gölge fiyatı 10'dur.
İkinci kısıdın (satır 3) gölge fiyatı ise 1'dir.

Bazı önemli denklemler

Eğer bir kısıdın ST değerindeki bir değişim en iyi çözümün değişmeyeceği izin verilen ST aralıklarında (allowable RHS range) ise aşağıdaki denklemler kullanılarak yeni amaç fonksiyon değeri hesaplanabilir:
enbüyükleme sorunu için

- yeni amaç fn. değeri = eski amaç fn. değeri + (yeni ST – eski ST) × gölge fiyat

enküçükleme sorunu için

- yeni amaç fn. değeri = eski amaç fn. değeri – (yeni ST – eski ST) × gölge fiyat

Yukarıda verilen örnekte, izin verilen ST aralığı çoğalışı (allowable increase in RHS ranges) sonsuz (infinity) olduğu için her iki kısıdın da ST değerini istediğimiz kadar çoğaltabiliriz. Fakat izin verilen ST aralığı azalışına (allowable decrease) göre birinci kısıdı en fazla 100, ikinci kısıdı ise 1 birim azaltabiliriz.

Birinci kısıdın yeni ST değerinin 60 olduğunu düşünelim:
Öncelikle izin verilen aralıklar kontrol edilir. Çoğalış sonsuz olduğundan birinci denklemi kullanabiliriz (maks sorunu).
z_{yeni} = 1001 + (60 - 100) 10 = 601.

3.3.5 Ek Örnek (Simpleks Kullanarak Duyarlılık)

Dakota mobilya örneğinde x_1, x_2, x_3 sırasıyla üretilen sıra, masa ve sandalye miktarı idi.

Karı enbüyüklemek için kurulan DP:

$$
\begin{array}{llllllll}
\text{maks } z = & 60x_1 & + & 30x_2 & + & 20x_3 & & \\
& 8x_1 & + & 6x_2 & + & x_3 & +s_1 & = 48 & \text{Tahta} \\
& 4x_1 & + & 2x_2 & + & 1.5x_3 & +s_2 & = 20 & \text{Montaj} \\
& 2x_1 & + & 1.5x_2 & + & .5x_3 & +s_3 & = 8 & \text{Marangozluk} \\
& & & x_2 & & & +s_4 & = 5 & \text{Talep}
\end{array}
$$

Bu sorunun en iyi çözümünü de bulmuştuk:

$$
\begin{array}{llllll}
z & +5x_2 & & + 10s_2 + 10s_3 & & = 280 \\
& -2x_2 & +s_1 & + 2s_2 - 8s_3 & & = 24 \\
& -2x_2 & +x_3 & + 2s_2 - 4s_3 & & = 8 \quad (1) \\
& x_1+1.25x_2 & & - .5s_2 + 1.5s_3 & & = 2 \\
& x_2 & & & +s_4 & = 5
\end{array}
$$

Analiz 1

Mevcut montaj işçiliği miktarı değişsin: $20 \rightarrow 20+\delta$, bu durumda sistem de değişecektir:

$$
\begin{array}{llllll}
z' = 60x_1' & + 30x_2' & + 20x_3' & & & \\
8x_1' & + 6x_2' & + x_3' & +s_1' & & = 48 \\
4x_1' & + 2x_2' & + 1.5x_3' & +s_2' & & = 20+\delta \\
2x_1' & + 1.5x_2' & + .5x_3' & +s_3' & & = 8 \\
& x_2' & & & +s_4' & = 5
\end{array}
$$

veya değişen sistem aşağıdaki gibi de yazılabilir:

$$\begin{array}{lllllll}
z' = 60x_1' & + 30x_2' & + 20x_3' & & & & \\
8x_1' & + 6x_2' & + x_3' & +s_1' & & & = 48 \\
4x_1' & + 2x_2' & + 1.5x_3' & +(s_2'-\delta) & & & = 20 \\
2x_1' & + 1.5x_2' & + .5x_3' & & +s_3' & & = 8 \\
 & x_2' & & & & +s_4' & = 5
\end{array}$$

$z', x_1', x_2', x_3', x_4', s_1', s_2'-\delta, s_3', s_4'$ esas (değişimden önceki) sorunu, dolayısıyla (1)'i sağlar. Gerekli değişiklikler yapılırsa yeni en iyi çözüm:

$$\begin{array}{llllllll}
z' & +5x_2' & & & + 10(s_2'-\delta) & + 10s_3' & & = 280 \\
 & -2x_2' & & +s_1' & + 2(s_2'-\delta) & - 8s_3' & & = 24 \\
 & -2x_2' & +x_3' & & - 2(s_2'-\delta) & - 4s_3' & & = 8 \\
 & x_1'+1.25x_2' & & & - .5(s_2'-\delta) & + 1.5s_3' & & = 2 \\
 & x_2' & & & & & +s_4' & = 5
\end{array}$$

ve böylece

$$\begin{array}{llllllll}
z' & +5x_2' & & & + 10s_2' & + 10s_3' & & = 280+10\delta \\
 & -2x_2' & & +s_1' & + 2s_2' & - 8s_3' & & = 24+2\delta \\
 & -2x_2' & +x_3' & & + 2s_2' & - 4s_3' & & = 8+2\delta \\
 & x_1'+1.25x_2' & & & - .5s_2' & + 1.5s_3' & & = 2-.5\delta \\
 & x_2' & & & & & +s_4' & = 5
\end{array}$$

elde edilir

$-4 \leq \delta \leq 4$ aralığı için yeni sistem en iyi çözümü verir: Bu araklıkta ST değerleri negatif olmaz.

δ çoğaldıkça, toplam kar da 10δ kadar çoğalmaktadır. Bu durumda montaj işçiliği kısıdının **gölge fiyat**ının \$10/saat olduğunu (4 saat azalma ve 4 saat çoğalmaya izin verildiğini unutmadan) söyleyebiliriz .

Analiz 2

Eğer sıraların fiyatı \$60+γ olursa ne olur?

Küçük bir γ için kar 2γ çoğalır çünkü en iyi çözüm 2 sıra yapılmasını önermektedir.

Peki söz konusu kar katsayısı ne kadar çoğaltılabilir?

Yeni gelir:

$$
\begin{aligned}
z' &= (60+\gamma)x_1+30x_2+20x_3 = z+\gamma x_1 \\
&= (280-5x_2-10s_2-10s_3)+\gamma(2-1.25x_2+.5s_2-1.5s_3) \\
&= 280+2\gamma-(5+1.25\gamma)x_2-(10-.5\gamma)s_2-(10+1.5\gamma)s_3
\end{aligned}
$$

şeklindedir.

Yeni üst satır aşağıdaki gibi olmalıdır:

$$z'+(5+1.25\gamma)x_2+(10-.5\gamma)s_2+(10+1.5\gamma)s_3 = 280+2\gamma$$

Optimalliğin (en iyi çözümün) bozulmaması için bu satırdaki tüm terimlerin 0 ya da 0'dan büyük olması gerektiğinden aranan aralık $-4 \leq \gamma \leq 20$ olarak bulunur.

Analiz 3

Eğer temel dışı değişkenlerden birinin kar katsayısı değişirse yeni gelir:

$$
\begin{aligned}
z' &= 60x_1+(30+\gamma)x_2+20x_3 = z+\gamma x_2 \\
&= 280-5x_2-10s_2-10s_3+\gamma x_2 \\
&= 280-(5-\gamma)x_2-10s_2-10s_3
\end{aligned}
$$

şeklinde hesaplanır. Bu durumda da optimalliğin bozulmama şartı $\gamma \leq 5$'dir. Fakat $\gamma > 5$ olursa masa üretmek daha iyi olacaktır. Bu durumda da masa için **indirgenmiş maliyet**in \$5.00 olduğu söylenebilir.

3.4 DUALİTE

3.4.1 Dual Teoremi

Bir DP modeli olurlu ise primal DP'nin en iyi çözüm (amaç fonksiyonu) değeri dual DP'nin en iyi çözüm değerine **eşittir**.

3.4.2 Normal Enbüyükleme Sorununun Dualini Bulma

m kısıtlı ve n değişkenli bir DP için matris gösterimi; b, m uzunluğunda tek boyutlu bir vektörü; c ve x, n uzunluğunda tek boyutlu vektörü ve A m satırlı n sütunlu iki boyutlu bir matrisi göstermek üzere; aşağıdaki gibidir:

maks cx
öyle ki Ax <= b
x >= 0

Bu şekilde gösterilebilen bir primal DP için, m uzunluğunda tek boyutlu bir y vektörü kullanılarak, bir **dual** (eşters) DP yazılabilir:

min by
öyle ki Ay >= c
y >= 0

Aşağıdaki tablo yardımıyla kolayca bir primal modelin duali bulunabilir. Eğer primal model normal enbüyükleme sorunu ise Tablo 1 soldan sağa okunarak model yazılır. Tablo yukarıdan aşağıya okunduğunda ise dual bulunmuş olur.

Table 1. Dualin bulunması

		maks z				
min w		$x_1 \geq 0$	$x_2 \geq 0$	....	$x_n \geq 0$	
.		x_1	x_2		x_n	
$y_1 \geq 0$	y_1	a_{11}	a_{12}	....	a_{1n}	$\leq b_1$
$y_2 \geq 0$	y_2	a_{21}	a_{22}	....	a_{2n}	$\leq b_2$
....	....	....	....		....	....
$y_m \geq 0$	y_m	a_{m1}	a_{m2}	....	a_{mn}	$\leq b_m$
		$\geq c_1$	$\geq c_2$		$\geq c_n$	

3.4.3 Normal Enküçükleme Sorununun Dualini Bulma

m kısıtlı ve n değişkenli bir DP için matris gösterimi; b, m uzunluğunda tek boyutlu bir vektörü; c ve x, n uzunluğunda tek boyutlu vektörü ve A m satırlı n sütunlu iki boyutlu bir matrisi göstermek üzere; aşağıdaki gibidir:

min cx

öyle ki Ax >= b

x >= 0

Bu şekilde gösterilebilen bir primal DP için, m uzunluğunda tek boyutlu bir y vektörü kullanılarak, bir **dual** DP yazılabilir.

maks by

öyle ki yA <= c

y >= 0

Yukarıda söz edilen tablo yöntemi normal enküçükleme sorununun dualini bulma için de kullanılabilir. Eğer primal model normal enküçükleme sorunu ise Tablo 1 yukarıdan aşağıya okunarak model yazılır. Tablo soldan sağa okunduğunda ise dual bulunmuş olur.

3.4.4 Normal Olmayan Enbüyükleme Sorununun Dualini Bulma

1. Tablo 1'i primal model soldan sağa okunacak şekilde doldurunuz
2. Aşağıdaki değişiklikler yapıldıktan sonra dual model yukarıdan aşağı okunabilir:
   a) Eğer *i.* primal kısıt $\geq$ kısıtsa, ilgili dual değişken $y_i \leq 0$ şeklinde olmalıdır.
   b) Eğer *i.* primal kısıt eşitlikse, ilgili dual değişken y_i "işareti sınırlandırılmamış" (unrestricted in sign - urs) değişkendir.
   c) Eğer *i.* primal değişken urs ise, *i.* dual kısıt eşitliktir.

3.4.5 Normal Olmayan Enküçükleme Sorununun Dualini Bulma

1. Tablo 1'i primal model yukarıdan aşağıya okunacak şekilde doldurunuz
2. Aşağıdaki değişiklikler yapıldıktan sonra dual model soldan sağa okunabilir:
   a) Eğer *i.* primal kısıt $\leq$ kısıtsa, ilgili dual değişken $x_i \leq 0$ şeklinde olmalıdır
   b) Eğer *i.* primal kısıt eşitlikse, ilgili dual değişken x_i "işareti sınırlandırılmamış" (urs) değişkendir.
   c) Eğer *i.* primal değişken urs ise, *i.* dual kısıt eşitliktir

3.4.6 Dual Ekonomik Yorum için Örnek (Dakota Mobilya)

x_1, x_2, x_3 üretilen sıra, masa ve sandalye sayısını göstersin. Haftalık kar $z iken DP modeli:

$$\text{maks } z = 60x_1+30x_2+20x_3$$
$$8x_1+6x_2+x_3 \leq 48 \quad \text{(Tahta kısıtı)}$$
$$4x_1+2x_2+1.5x_3 \leq 20 \quad \text{(Montaj kısıtı)}$$
$$2x_1+1.5x_2+0.5x_3 \leq 8 \quad \text{(Marangozluk kısıtı)}$$
$$x_1,x_2,x_3 \geq 0$$

Dual

Farzedelim ki bir girişimci Dakota'nın tüm kaynaklarını (hammadde) satın almak istiyor.

Dual sorunda y_1, y_2, y_3 sırasıyla bir m^2 tahta, bir ssat montaj işçiliği ve bir saat marangozluk için ödenmesi gereken ücreti gösterir.

$w de kaynak satın alma toplam maliyetini gösterir.

Kaynak ücretleri Dakota'yı satışa teşvik edecek kadar yüksek; girişimciyi vazgeçirmeyecek kadar az olmalıdır. Bu durumda da toplam satın alma maliyeti toplam kar kadar olur.

$$\text{min } w = 48y_1+20y_2+8y_3$$
$$8y_1+4y_2+2y_3 \geq 60 \quad \text{(Sıra kısıtı)}$$
$$6y_1+2y_2+1.5y_3 \geq 30 \quad \text{(Masa kısıtı)}$$
$$y_1+1.5y_2+0.5y_3 \geq 20 \quad \text{(Sandalye kısıtı)}$$
$$y_1,y_2,y_3 \geq 0$$

4. ULAŞTIRMA SORUNLARI

Genel olarak, bir ulaştırma sorunu aşağıdaki bilgileri barındırır:

- Bir ürün/hizmet gönderen m adet **arz noktası** (supply point). i arz noktası en fazla s_i birim arz edebilir.
- Ürünün/hizmetin gönderildiği n adet **talep noktası** (demand point). j talep noktası en az d_j birime gereksinim duyar.
- Bir birimin i arz noktasından j talep noktasına gönderilmesi maliyeti c_{ij}'dir.

Söz konusu bilgi aşağıdaki **ulaştırma tablosu** ile formüle edilebilir:

	Talep noktası 1	Talep noktası 2	.....	Talep noktası n	ARZ
Arz noktası 1	c_{11}	c_{12}		c_{1n}	s_1
Arz noktası 2	c_{21}	c_{22}		c_{2n}	s_2
.....					
Arz noktası m	c_{m1}	c_{m2}		c_{mn}	s_m
TALEP	d_1	d_2		d_n	

Eğer toplam talep miktarı toplam arz miktarına eşitse sorun **dengeli ulaştırma sorunu** olarak isimlendirilir.

4.1 ULAŞTIRMA SORUNLARININ FORMÜLASYONU

4.1.1 DP Gösterimi

x_{ij} = i arz noktasından j talep noktasına gönderilen miktar olsun.

Bu durumda ulaştırma sorununun genel DP gösterimi aşağıdaki gibidir:

$$\min \sum_i \sum_j c_{ij} x_{ij}$$

öyle ki $\sum_j x_{ij} \leq s_i$ (i=1,2, ..., m) Arz kısıtları

$\sum_i x_{ij} \geq d_j$ (j=1,2, ..., n) Talep kısıtları

$x_{ij} \geq 0$

Yukarıdaki sorun, bir enbüyükleme sorunu (ulaştırma sonucu kar elde edilmesi gibi) da olsa, kısıtlarının benzer özellikler taşıması durumunda yine bir ulaştırma sorunudur.

4.1.2 Dengeli Ulaştırma Sorunu için Powerco Örneği

Powerco şirketinin dört şehre hizmet veren üç adet elektrik santrali vardır. Her bir santral sırasıyla 35 milyon, 50 milyon ve 40 milyon kWh elektrik üretmektedir. Şehirlerin en yoğun saatlerde talep ettiği elektrik miktarı ise sırasıyla 45 milyon, 20 milyon, 30 milyon ve 30 milyon kWh'dir. 1 milyon kWh elektriğin bir santralden bir şehre gönderilmesinin maliyeti aşağıdaki tabloda verilmiştir. Her şehrin talebini en az maliyetle karşılamak üzere bir ulaştırma tablosunda dengeli bir ulaştırma sorunu formüle ediniz ve sorunun DP modelini gösteriniz.

	Şehir 1	Şehir 2	Şehir 3	Şehir 4
Santral 1	$8	$6	$10	$9
Santral 2	$9	$12	$13	$7
Santral 3	$14	$9	$16	$5

Yanıt:

1. Ulaştırma sorununun formülasyonu

	Şehir 1	Şehir 2	Şehir 3	Şehir 4	ARZ
Santral 1	8	6	10	9	35
Santral 2	9	12	13	7	50
Santral 3	14	9	16	5	40
TALEP	45	20	30	30	125

Toplam talep ve toplam arz eşit olduğundan (125 milyon kWh) sorun "dengeli"dir.

2. Sorunun DP modeli olarak gösterimi

x_{ij}: Santral *i*'de üretilen ve Şehir *j*'ye gönderilen elektrik miktarı (million kwh)

min z = 8 x11 + 6 x12 + 10 x13 + 9 x14 + 9 x21 + 12 x22 + 13 x23 + 7 x24 + 14 x31 + 9 x32 + 16 x33 + 5 x34

s.t. x11 + x12 + x13 + x14 $\leq$ 35 (arz kısıtları)

x21 + x22 + x23 + x24 $\leq$ 50

x31 + x32 + x33 + x34 $\leq$ 40

x11 + x21 + x31 $\geq$ 45 (talep kısıtları)

x12 + x22 + x32 $\geq$ 20

x13 + x23 + x33 $\geq$ 30

x14 + x24 + x34 $\geq$ 30

$x_{ij} \geq 0$ (i = 1, 2, 3; j = 1, 2, 3, 4)

4.1.3 Dengesiz bir Ulaştırma Sorununun Dengelenmesi

Fazla Arz

Eğer toplam arz miktarı toplam talep miktarını geçerse, sorunu dengelemek için talep miktarı aradaki fark (fazla arz miktarı) kadar olan bir **yapay talep noktası** yaratırız. Söz konusu noktaya yapılacak gönderimler aslında olmayacağı için bu noktaya arz noktalarından yapılacak ulaştırma maliyeti 0 olacaktır.

Karşılanmayan Talep

Eğer toplam arz miktarı toplam talep miktarından azsa, aslında olurlu bir çözüm yoktur (talepler karşılanamaz). Bu durumda karşılanamayan talep kadar arzı olan bir

yapay arz noktası yaratırız. Talebin olmayan bir arz noktasından karşılanamaması beraberinde bir "ceza maliyeti" getirir.

4.1.4 Fazla Arz için Değiştirilmiş Powerco Örneği

Şehir 1'in talebinin 40 milyon kwh olduğunu farz edelim. Bu durumda dengeli bir ulaştırma sorunu formüle ediniz.

Yanıt

Toplam talep 120 ve toplam arz 125 olduğundan sorun dengeli değildir.

Sorunu dengelemek için bir yapay talep noktası yaratırız. Söz konusu noktanın talebi 125 – 120 = 5 milyon kwh olacaktır.

Her santralden yapay talep noktasına 1 milyon kwh elektrik göndermenin maliyeti 0 olacaktır.

Tablo 4. Fazla Arz Örneği için Ulaştırma Tablosu

	Şehir 1	Şehir 2	Şehir 3	Şehir 4	Yapay	ARZ
Santral 1	8	6	10	9	0	35
Santral 2	9	12	13	7	0	50
Santral 3	14	9	16	5	0	40
TALEP	40	20	30	30	5	125

4.1.5 Karşılanmayan Talep için Değiştirilmiş Powerco Örneği

Şehir 1'in talebinin 50 milyon kwh olduğunu farz edelim. Karşılanamayan her 1 milyon kWh elektrik için 80$ ceza maliyeti kesilirse dengeli bir ulaştırma sorunu formüle ediniz.

Yanıt

5 milyon kWh elektrik arz eden bir yapay arz noktası yaratırız.

	Şehir 1	Şehir 2	Şehir 3	Şehir 4	ARZ
Santral 1	8	6	10	9	35
Santral 2	9	12	13	7	50
Santral 3	14	9	16	5	40
Talep	80	80	80	80	5
TALEP	50	20	30	30	130

4.2 TEMEL OLURLU ÇÖZÜMÜN BULUNMASI

Dengeli bir ulaştırma sorunu için genel DP gösterimi aşağıdaki gibi yazılabilir::

$\min \sum_i \sum_j c_{ij}\, x_{ij}$

öyle ki $\sum_j x_{ij} = s_i$ (i=1,2, ..., m)   Arz kısıtları

$\sum_i x_{ij} = d_j$ (j=1,2, ..., n)   Talep kısıtları

$x_{ij} \geq 0$

Söz konusu soruna bir temel olurlu çözüm (basic feasible solution - bfs) bulmak için aşağıdaki önemli gözlemi kullanmalıyız:

"Eğer dengeli bir ulaştırma sorununda x_{ij}'lerin değerler kümesi bir kısıt haricinde tüm kısıtları sağlarsa, bu değerler o kısıdı da sağlar."

Bu gözlem ulaştırma sorununun çözümü sırasında herhangi bir kısıtı gözardı edebileceğimizi ve m+n-1 kısıttan oluşan bir DP çözeceğimizi gösterir. Genel olarak ilk arz kısıtı değerlendirme dışı bırakılır.

Geri kalan m+n-1 kısıda bfs bulmak için herhangi bir m+n-1 değişkenin temel çözüm verebileceğini düşünebilirsiniz: fakat söz konusu m+n-1 değişkenin temel çözümde olabilmesi için bir **döngü oluşturmamaları** gerekir.

En az dört hücrenin bir döngü oluşturması için:

- Herhangi ardışık iki hücrenin aynı satır veya sütunda olması gerekir
- Aynı satır veya sütunda ardışık üç hücre olmamalıdır
- Serinin son hücresi ilk hücre ile aynı satır veya sütunda olup döngüyü kapatmalıdır

Dengeli bir ulaştırma sorununa temel olurlu çözüm bulmak için üç farklı yöntem kullanılabilir:

1. Kuzeybatı Köşe (Northwest Corner) Yöntemi
2. Enküçük Maliyet (Minimum Cost) Yöntemi
3. Vogel'in Yaklaşımı

4.2.1 Kuzeybatı Köşe Yöntemi

Ulaştırma tablosunun en sol üst köşesinden başlarız ve x_{11}'i mümkün olduğunca büyük bir değer atarız (tabii ki, x_{11} en çok s_1 ve d_1 ikilisinin en küçük değeri kadar olabilir).

- Eğer x_{11}=s_1 ise ilk satırı iptal ediniz ve d_1'i d_1-s_1 olarak güncelleyiniz
- Eğer x_{11}=d_1 ise ilk sütunu iptal ediniz ve s_1'i s_1-d_1 olarak güncelleyiniz
- Eğer x_{11}=s_1=d_1 ise ya ilk satırı ya da ilk sütunu iptal ediniz (her ikisini de değil!)
- Eğer satırı iptal ettinizse d_1'i sıfır yapınız
- Eğer sütunu iptal ettinizse s_1'i sıfır yapınız

Bu şekilde devam ederek (her seferinde geri kalan hücrelerde yeni sol-üst köşeye atama yaparak) tüm atamalar yapılır. Sonuçta, bir hücre geriye kalacaktır. Satır veya sütundaki değeri atayarak ve hem satırı hem de sütunu iptal ederek işlemi bitiriniz: bir bfs elde edilmiştir.

Aşağıdaki dengeli ulaştırma sorunu için bir bfs bulalım
(Bu yöntemde maliyetler gerekmediğinden verilmemiştir!).

				5
				1
				3
2	4	2	1	

Toplam talep toplam arza eşittir (9): sorun dengelidir.

2				3
				1
				3
X	4	2	1	

2	3			X
				1
				3
X	1	2	1	

2	3			X
	1			X
				3
X	0	2	1	

2	3			X
	1			X
	0	2	1	3
X	0	2	1	

m+*n*-1 (3+4-1 = 6) adet değişken atanmış olur. KBK yöntemi ile seçilen değişkenler bir döngü oluşturmadıklarından bir bfs bulunmuştur.

4.2.2 Enküçük Maliyet Yöntemi

KBK yöntemi maliyetleri göz önüne almadığından başlangıç bfs'si maliyeti yüksek olan bir çözüm olabilir ve en iyi çözümün bulunması için çok sayıda işlem gerekebilir. Bu durumla karşılaşmamak için kullanılabilecek olan enküçük maliyet yönteminde en düşük taşıma maliyeti olan hücreye atama yapılır. Bu hücreye yapılacak x_{ij} ataması yine min $\{s_i, d_j\}$ kadardır.

KBK yöntemindeki gibi atama yapılan hücrenin olduğu satır veya sütun iptal edilip arz ya da talep değeri güncellenir ve tüm atamalar yapılıncaya kadar devam edilir.

2	3	5	6	5
2	1	3	5	10
3	8	4	6	15
12	8	4	6	

2	3	5	6	5
2	**1** 8	3	5	2
3	8	4	6	15
12	X	4	6	

2	3	5	6	5
2 2	1 8	3	5	X
3	8	4	6	15
10	X	4	6	

2 5	3	5	6	X
2 2	1 8	3	5	X
3	8	4	6	15
5	X	4	6	

2 5	3	5	6	X
2 2	1 8	3	5	X
3 5	8	4 4	6 6	15
5	X	4	6	

4.2.3 Vogel'in Yaklaşımı

Her satır ve sütun için ceza hesaplanarak yönteme başlanır. Ceza o satır veya sütundaki en küçük iki maliyet arasındaki farktır.

Daha sonra cezası enbüyük olan satır veya sütun bulunur.

Söz konusu satır veya sütundaki en düşük maliyetli hücre ilk temel değişkeni verir.

Yine KBK yöntemindeki gibi bu değişkene atanacak değer, ilgili hücrenin arz ve talep miktarlarına bağlıdır. Gerekli iptaller ve güncellemeler yapılır

Yeniden geri kalan tablo için yeni cezalar hesaplanır ve prosedüre benzer adımlarla devam edilir.

				Arz	Satır cezası
	6	7	8	10	7-6=1
	15	80	78	15	78-15=63
Talep	15	5	5		
Sütun cezası	15-6=9	80-7=**73**	78-8=70		

				Arz	Satır cezası
	6	**7** 5	8	5	8-6=2
	15	80	78	15	78-15=63
Talep	15	X	5		
Sütun cezası	15-6=9	-	78-8=**70**		

				Arz	Satır cezası
	6	7 5	**8** 5	X	-
	15	80	78	15	-
Talep	15	X	0		
Sütun cezası	15-6=9	-	-		

				Arz
	6	7 5	8 5	X
	15 15	80	78 0	15
Talep	15	X	0	

4.3 ULAŞTIRMA SİMPLEKSİ

4.3.1 Yöntemin Adımları

1. Eğer ulaştırma sorunu dengesiz ise dengeleyiniz.
2. Bir bfs bulmak için KBK, Enküçük Maliyet veya Vogel yöntemlerinden birini kullanınız
3. $u_1 = 0$ olarak kabul edip mevcut bfs'deki tüm temel değişkenler için $u_i + v_j = c_{ij}$ denklemini kullanarak *u*'ları ve *v*'leri hesaplayınız.
4. Tüm temel dışı değişkenler için $u_i + v_j - c_{ij} \leq 0$ ise, en iyi çözüm bulunmuştur. Eğer bu koşul sağlanmazsa $u_i + v_j - c_{ij}$ değeri en pozitif olan değişken *pivot işlemleri* ile temele girer ve temeldeki değişkenlerden biri çözümden çıkar. Böylece yeni bir bfs bulunmuş olur.
5. Yeni bfs için Adım 3 ve 4'ü tekrar ediniz.

Enbüyükleme sorunu için yine yukarıdaki adınlar uygulanır. Sadece 4. adımda aşağıdaki değişiklik yapılmalıdır:

Tüm temel dışı değişkenler için $u_i + v_j - c_{ij} \geq 0$ ise, en iyi çözüm bulunmuştur. Eğer bu koşul sağlanmazsa $u_i + v_j - c_{ij}$ değeri en negatif olan değişken *pivot işlemleri* ile temele girer ve temeldeki değişkenlerden biri çözümden çıkar. Böylece yeni bir bfs bulunmuş olur

Pivot işlemleri

1. Çözüme girecek olan değişken ile temel değişkenlerin bazıları veya hepsi bir döngü oluşturur (sadece bir olası döngü vardır!).
2. Döngüdeki hücreleri çözüme giren hücreden başlayarak sayınız. Sayısı çift olanları (0, 2, 4, vb.) *çift hücreler* olarak işaretleyiniz. Döngüdeki diğer hücreleri de *tek hücreler* olarak işaretleyiniz.
3. Tek hücrelerde değeri en küçük olan değişkeni bulunuz. Bu değere Φ diyelim. Bu değişken temel dışı kalacaktır. İşlemi tamamlamak için tüm tek hücrelerdeki değerlerden Φ çıkaralım ve çift hücrelerdeki değerlere Φ ekleyelim. Döngüde olmayan değişkenlerin değeri değişmez. Eğer $\Phi = 0$ ise giren değişken 0 değeri ile çözüme girecektir.

4.3.2 Powerco Örneğinin Çözümü

Sorun dengelidir (toplam talep toplam arza eşittir).

Powerco örneğine KBK yöntemi uygulanırsa, aşağıdaki tabloda göreülen bfs elde edilir ($m+n-1=6$ temel değişken!).

	Şehir 1	Şehir 2	Şehir 3	Şehir 4	ARZ
Santral 1	[8] 35	[6]	[10]	[9]	35
Santral 2	[9] 10	[12] 20	[13] 20	[7]	50
Santral 3	[14]	[9]	[16] 10	[5] 30	40
TALEP	45	20	30	30	125

u1 = 0

u1 + v1 = 8 $\Rightarrow$ v1 = 8

u2 + v1 = 9 $\Rightarrow$ u2 = 1

u2 + v2 = 12 $\Rightarrow$ v2 = 11

u2 + v3 = 13 $\Rightarrow$ v3 = 12

u3 + v3 = 16 $\Rightarrow$ u3 = 4

u3 + v4 = 5 $\Rightarrow$ v4 = 1

Tüm temel dışı değişkenler için $\hat{c}_{ij} = u_i + v_j - c_{ij}$ hesaplanır:

$\hat{c}_{12} = 0 + 11 - 6 = 5$

$\hat{c}_{13} = 0 + 12 - 10 = 2$

$\hat{c}_{14} = 0 + 1 - 9 = -8$

$\hat{c}_{24} = 1 + 1 - 7 = -5$

$\hat{c}_{31} = 4 + 8 - 14 = -2$

$\hat{c}_{32} = 4 + 11 - 9 = 6$

$\hat{c}_{32}$ en pozitif olan değeri verdiğinden, x_{32} temel değişken olacaktır.

x_{32}'nin de olduğu döngü (3,2)-(3,3)-(2,3)-(2,2) şeklindedir: $\Phi = 10$ bulunur.

	Şehir 1	Şehir 2	Şehir 3	Şehir 4	ARZ
Santral 1	[8] 35	[6]	[10]	[9]	35
Santral 2	[9] 10	[12] **20–Φ**	[13] **20+Φ**	[7]	50
Santral 3	[14]	[9] Φ	[16] **10–Φ**	[5] 30	40
TALEP	45	20	30	30	125

x_{33} temel dışı değişken olacaktır. Yeni bfs aşağıdaki tabloda verilmiştir:

u_i/v_j	8	11	12	7	ARZ
0	8 35	6	10	9	35
1	9 10	12 10	13 30	7	50
-2	14	9 10	16	5 30	40
TALEP	45	20	30	30	125

$\hat{c}_{12}$ = 5, $\hat{c}_{13}$ = 2, $\hat{c}_{14}$ = -2, $\hat{c}_{24}$ = 1, $\hat{c}_{31}$ = -8, $\hat{c}_{33}$ = -6

$\hat{c}_{12}$ en pozitif değeri verdiğinden, x_{12} çözüme girer.

x_{12}'nin de olduğu döngü (1,2)-(2,2)-(2,1)-(1,1) şeklindedir ve Φ = 10'dur

	Şehir 1	Şehir 2	Şehir 3	Şehir 4	ARZ
Santral 1	8 **35–Φ**	6 **Φ**	10	9	35
Santral 2	9 **10+Φ**	12 **10–Φ**	13 30	7	50
Santral 3	14	9 10	16	5 30	40
TALEP	45	20	30	30	125

x_{22} çözümden çıkar. Yeni bfs aşağıdaki tabloda verilmiştir:

u_i/v_j	8	6	12	2	ARZ
0	8 25	6 10	10	9	35
1	9 20	12	13 30	7	50
3	14	9 10	16	5 30	40
TALEP	45	20	30	30	125

$\hat{c}_{13}$ = 2, $\hat{c}_{14}$ = -7, $\hat{c}_{22}$ = -5, $\hat{c}_{24}$ = -4, $\hat{c}_{31}$ = -3, $\hat{c}_{33}$ = -1

$\hat{c}_{13}$ en pozitif olan değeri verdiğinden, x_{13} temel değişken olacaktır.

x_{13}'ün de olduğu döngü (1,3)-(2,3)-(2,1)-(1,1) şeklindedir. Φ = 25

	Şehir 1	Şehir 2	Şehir 3	Şehir 4	ARZ
Santral 1	[8] **25–Φ**	[6] 10	[10] **Φ**	[9]	35
Santral 2	[9] **20+Φ**	[12]	[13] **30–Φ**	[7]	50
Santral 3	[14]	[9] 10	[16]	[5] 30	40
TALEP	45	20	30	30	125

x_{11} temel dışı değişken olur. Yeni bfs:

u_i/v_j	6	6	10	2	ARZ
0	[8]	[6] 10	[10] 25	[9]	35
3	[9] 45	[12]	[13] 5	[7]	50
3	[14]	[9] 10	[16]	[5] 30	40
TALEP	45	20	30	30	125

$\hat{c}_{11}$ = -2, $\hat{c}_{14}$ = -7, $\hat{c}_{22}$ = -3, $\hat{c}_{24}$ = -2, $\hat{c}_{31}$ = -5, $\hat{c}_{33}$ = -3

Tüm $\hat{c}_{ij}$'ler negatif olduğundan en iyi çözüm bulunmuştur.

Rapor

Santral 2'den Şehir 1'e 45 milyon kwh elektrik gönderilmelidir.

Santral 1'den Şehir 2'ye 10 milyon kwh elektrik gönderilmelidir. Benzer şekilde Santral 3'den Şehir 2'ye 10 milyon kwh elektrik gönderilmelidir.

Santral 1'den Şehir 3'e 25 milyon kwh ve Santral 2'den Şehir 3'e 5 milyon kwh elektrik gönderilmelidir.

Santral 3'den Şehir 4'e 30 milyon kwh elektrik gönderilmelidir

Toplam taşıma maliyeti:

$$z = .9\,(45) + 6\,(10) + 9\,(10) + 10\,(25) + 13\,(5) + 5\,(30) = \$\ 1020$$

4.4 GEÇİCİ KONAKLAMA SORUNLARI

Bazı durumlarda gönderim sürecindeki bir nokta hem ürün/hizmet gönderebilir, hem de söz konusu noktaya ürün/hizmet gönderilebilir. Ürün/hizmetin arz noktasından talep noktasına gönderimi sırasında geçici olarak konakladığı bu nokta **geçici konaklama noktası** olarak isimlendirilir.

Bu özelliği olan bir gönderim sorunu geçici konaklama sorunudur.

Geçici konaklama sorununa en iyi çözüm söz konusu sorunu ulaştırma sorununa dönüştürüp ulaştırma sorununu çözerek bulunabilir.

Uyarı

"Ulaştırma Sorunlarının Formülasyonu" bölümünde belirtildiği gibi, bir başka noktaya bir ürün/hizmet gönderen fakat hiç bir noktadan ürün/hizmet alamayan nokta **arz noktası** olarak isimlendirilir.

Benzer şekilde, bir **talep noktası** da diğer noktalardan ürün/hizmet alabilir fakat hiç bir noktaya ürün/hizmet gönderemez.

4.4.1 Adımlar

1. Eğer sorun dengesiz ise sorunu dengeleyiniz.
   s = dengeli sorun için toplam arz (veya talep) miktarı olsun
2. Aşağıdaki şekilde bir ulaştırma tablosu kurunuz:
   Her arz ve geçici konaklama noktası için tabloda bir satır gerekecektir
   Her talep ve geçici konaklama noktası için bir sütun gerekecektir
   Her arz noktasının arzı o noktanın arz miktarı kadar olacaktır
   Her talep noktasının talebi o noktanın talep miktarı kadar olacaktır
   Her geçici konaklama noktasının arzı "o noktanın arz miktarı + *s*" kadar olacaktır
   Her geçici konaklama noktasının talebi "o noktanın talep miktarı + *s*" kadar olacaktır
3. Ulaştırma sorununu çözünüz

4.4.2 Kuruoğlu Örneği

(Winston 7.6.'dan esinlenilmiştir)

Kuruoglu Malatya ve G.Antep'deki fabrikalarında buzdolabı üretmektedir. Malatya'daki fabrika günde en fazla 150 adet, G.Antep'teki fabrika ise günde en fazla 200 adet buzdolabı üretebilmektedir. Buzdolapları uçak ile İstanbul ve İzmir'deki müşterilere gönderilmektedir. Her iki şehirdeki müşterilerin günlük talebi 130 adet buzdolabıdır. Gönderim maliyetlerindeki değişiklikler yüzünden bazı buzdolaplarının fabrikalardan uçakla öncelikle Ankara veya Eskişehir'e gönderilmesi ve daha sonra nihai müşterilere bu şehirlerden gönderilmesi düşünülmektedir. Bir buzdolabının taşıma maliyeti aşağıdaki tabloda verilmiştir. Kuruoğlu toplam taşıma maliyetlerini enazlayacak şekilde müşteri taleplerini karşılamak istemektedir.

Tablo 1. Gönderim Maliyetleri

TL	Malatya	G.Antep	Ankara	Eskişehir	Istanbul	İzmir
Malatya	0	-	8	13	25	28
G.Antep	-	0	15	12	26	25
Ankara	-	-	0	6	16	17
Eskişehir	-	-	6	0	14	16
Istanbul	-	-	-	-	0	-
İzmir	-	-	-	-	-	0

Yanıt:

Bu sorunda Ankara ve Eskişehir *geçici konaklama* noktalarıdır.

Adım 1. Sorunu dengeleme

Toplam arz = 150 + 200 = 350

Toplam talep = 130 + 130 = 260

Yapay talep = 350 – 260 = 90

s = 350 (dengeli sorun için toplam arz veya talep miktarı)

Adım 2. Bir ulaştırma tablosu kurma

Geçici konaklama noktası talebi = O noktanın talep miktarı + *s* = 0 + 350 = 350

Geçici konaklama noktası arzı = O noktanın arz miktarı + *s* = 0 + 350 = 350

Tablo 2. Geçici Konaklama sorununun Ulaştırma Sorunu Olarak Gösterimi

	Ankara	Eskişehir	Istanbul	Izmir	Yapay	Arz
Malatya	8	13	25	28	0	150
G.Antep	15	12	26	25	0	200
Ankara	0	6	16	17	0	350
Eskişehir	6	0	14	16	0	350
Talep	350	350	130	130	90	

Adım 3. Ulaştırma sorununun çözümü

Tablo 3. Ulaştırma Sorununun En iyi Çözümü

	Ankara	Eskişehir	Istanbul	Izmir	Yapay	Arz
Malatya	8 130	13	25	28	0 20	150
G.Antep	15	12	26	25 130	0 70	200
Ankara	0 220	6	16 130	17	0	350
Eskişehir	6	0 350	14	16	0	350
Talep	350	350	130	130	90	1050

Rapor:

Kuruoglu Malatya'da 130 buzdolabı üretip bunların tamamını Ankara üzerinden İstanbul'a göndermelidir.

G.Antep'de üretilecek 130 buzdolabı ise doğrudan İzmir'e gönderilmelidir.

Bu durumda toplam taşıma maliyeti 6370 TL olacaktır.

Uyarı: Geçici konaklama sorununun çözümünü yorumlarken yapay'a gönderilen (veya yapay'ın gönderdiği) ve bir noktanın kendisine gönderdiği miktarlara önem verilmez.

4.5 ATAMA SORUNLARI

Ulaştırma sorunlarında her arz noktasının bir talep noktasına atanmasını ve her talebin karşılanmasını gerektiren özel bir durum söz konusudur. Bu tip sorunlar "atama sorunları" olarak isimlendirilir. Örneğin hangi işçinin veya makinenin hangi işi yapacağını belirlemek bir atama sorunudur.

4.5.1 DP Gösterimi

Bir atama sorununda bir arz noktasını bir talep noktasına atamanın maliyeti c_{ij}'dir.
Öte yandan, bir x_{ij} 0-1 tamsayı değişkeni aşağıdaki gibi tanımlanır:

x_{ij} = 1 eğer *i*. arz noktası *j*. talep noktasının talebini karşılamak üzere atanırsa

x_{ij} = 0 eğer *i*. arz noktası *j*. talep noktasının talebini karşılamazsa

Bu durumda, bir atama sorununun genel DP gösterimi

$\min \sum_i \sum_j c_{ij}\, x_{ij}$

öyle ki $\sum_j x_{ij} = 1$ (i=1,2, ..., m) Arz kısıtları

$\sum_i x_{ij} = 1$ (j=1,2, ..., n) Talep kısıtları

$x_{ij} = 0$ veya $x_{ij} = 1$

şeklindedir.

4.5.2 Macar Yöntemi

Tüm arz ve talep miktarları tamsayı olduğundan, en iyi çözümdeki tüm değişkenler de tamsayı olmalıdır. Her kısıtın ST değeri 1'e eşit olduğundan, her x_{ij} 1'den büyük olmayan ve negatif olmayan bir tamsayı olmalıdır. Bu durumda her x_{ij} 0 veya 1 olmalıdır.

$x_{ij} = 0$ veya $x_{ij} = 1$ kısıtlamasını DP gösteriminde ihmal edersek, her arz noktasının bir adet arz ettiği ve her talep noktasının bir adet talep ettiği dengeli bir ulaştırma sorunu ile karşılaşırız.

Fakat atama sorununun ulaştırma simpleks yöntemi ile çözülmesi yukarıda verilen kısıtlamayı kullanmayacağı için etkin olmayacaktır.

Bu yüzden simpleks'den daha basit bir algoritma olan Macar Yöntemi ile atama sorunları çözülür.

Uyarı

1. Amaç fonksiyonunun enbüyüklenmesi istenilen atama sorunlarında karlar matrisindeki elemanların –1 ile çarpılarak sorunun **enküçükleme** sorunu olarak Macar Yöntemi ile çözülmesi gerekir
2. Eğer maliyet matrisinde satır ve sütun sayıları eşit değilse atama sorunu **dengesiz**dir. Bu durumda sorunu Macar Yöntemi ile çözmeden önce bir veya daha fazla sayıda yapay nokta eklenerek dengelenmelidir..

Adımlar

1. $m \times m$'lik maliyet matrisinin her satırındaki en küçük maliyeti bulunuz.
2. Her maliyetten kendi satırındaki en küçük maliyeti çıkararak bir matris kurunuz
3. Yeni matrisde her sütunun en küçük maliyetini bulunuz
4. Bu sefer her maliyetten kendi sütunundaki en küçük maliyeti çıkararak yeni bir matris (indirgenmiş maliyet matrisi) kurunuz
5. İndirgenmiş maliyet matrisindeki tüm sıfırları örtecek şekilde en az sayıda (yatay veya düşey) çizgi çiziniz. Eğer bu işlem için m adet çizgi gerekli ise en iyi çözüm bulunmuştur. Eğer gerekli çizgi sayısı m adetten az ise bir sonraki adıma geçiniz
6. İndirgenmiş maliyet matrisinde Adım 5'de çizilen çizgiler ile örtülmemiş ve sıfır'dan farklı en küçük maliyeti (k) bulunuz
7. Her üstünden çizgi geçmeyen maliyetten k'yı çıkarınız ve çift çizgi ile örtülen her maliyete k'yı ekleyiniz. Adım 5'e dönünüz.

4.5.3 Uçuş Ekibi Örneği

(Winston 7.5.'den esinlenilmiştir)

Dört adet kaptan pilot (Selçuk, Serkan, Ümit, Volkan) uçuşlarda beraber oldukları dört adet uçuş teknisyenini (Tuncay, Önder, Servet, Kemal) yetkinlik, uyum ve moral motivasyon açısından 1-20 ölçeğinde değerlendirmişlerdir (1: çok iyi, 20: çok kötü). Değerlendirme notları Tablo 1'de verilmiştir. Havayolu şirketi her uçuş teknisyeninin uçuş atamasını mümkün olduğunca kendisine iyi not veren kaptan pilotla yapmak istemektedir. Uçuş ekipleri nasıl olmalıdır?

Tablo 1. Makaleler için Zamanlar (gerekli hafta sayısı)

	Tuncay	Önder	Servet	Kemal
Selçuk	2	4	6	10
Serkan	2	12	6	5
Ümit	7	8	3	9
Volkan	14	5	8	7

Yanıt:

Adım 1. Tablo 1'deki her satır için en küçük maliyetler sırasıyla 2, 2, 3 ve 5'dir.

Adım 2 & 3. Her maliyetten kendi satırındaki en küçük maliyeti çıkararak Tablo 2'yi elde edilir. Yeni matrisin her sütunun en küçük maliyeti.bulunur.

Tablo 2. Satır minimumları çıkarıldıktan sonra maliyet matrisi

	0	2	4	8
	0	10	4	3
	4	5	0	6
	9	0	3	2
Sütun minimumu	0	0	0	2

Adım 4. Bu sefer her maliyetten kendi sütunundaki en küçük maliyeti çıkararak indirgenmiş maliyet matrisi elde edilir (Tablo 3).

Tablo 3. İndirgenmiş maliyet matrisi

0	2	4	6
0	10	4	1
4	5	0	4
9	0	3	0

Adım 5. Tablo 4'de gösterildiği gibi 3. ve 4. satır ile 1. sütunda çizilecek çizgiler indirgenmiş maliyet matrisindeki tüm sıfırları örter. Gerekli en az çizgi sayısı 3'dür. 4'den az çizgi gerektiğinden çözüm en iyi değildir. Bir sonraki adıma geçilir.

Tablo 4. Sıfırları örten çizgilerle indirgenmiş maliyet matrisi

0	2	4	6
0	10	4	1
4	5	0	4
9	0	3	0

Adım 6 & 7. Örtülememiş en küçük maliyet 1'dir. Her örtülmemiş maliyetten 1 çıkarılır ve iki çizgi ile örtülenlere 1 eklenir. Tablo 6 elde edilir.

Table 6. Sonuç matrisi

0	1	3	5
0	9	3	0
5	5	0	4
10	0	3	0

Tüm sıfırları örtmek için şimdi dört çizgiye gerek vardır. En iyi çözüm bulunmuştur. Sütun 3'deki tek sıfır x_{33}'de ve Sütun 2'deki tek sıfır x_{42}'dedir. Satır 4 tekrar kullanılmayacağı için Sütun 4 için kalan sıfır x_{24}'dedir. Son olarak x_{11}'i seçeriz. Seçilen tüm karar değişkenleri 1'e eşittir.

Rapor:

KP Selçuk, UT Tuncay ile; KP Serkan, UT Kemal ile; KP Ümit, UT Servet ile; KP Volkan, UT Önder ile uçmalıdır.

5. TAMSAYILI PROGRAMLAMA

DP sorunlarının modeli kurulurken bazı karar değişkenlerinin kesinlikle tamsayı değerler alması gerektiğini görürüz fakat DP'ye uygun olması için büyük değerler alabilecek karar değişkeninin kesirli kısmı yok sayılabilir mantığı ile kesirli değerler almalarına izin veririz. Bazı durumlarda bu varsayım kabul edilemez olabilir. Bulunacak sayısal sonucun tamsayı değerler alması gerekebilir.

Bu tip durumlardaki sorunlara ***Tamsayılı Program*** (***TP***; Integer Programming – IP) denilir ve söz konusu programların çözümü konusu ***Tamsayılı Programlama*** (***TP***) olarak isimlendirilir.

Sonlu sayıda seçenek kümesindeki seçenekler ile ilgili birçok kararın yap/yapma, evet/hayır gibi kesikli olması yüzünden TP'ler sıklıkla kullanılır.

Tüm değişkenlerin tamsayı olduğu bir programlama sorunu ***tamamen tamsayılı programlama sorunu*** olarak isimlendirilir.

Eğer değişkenlerin bazıları tamsayı, bazıları kesirli ise söz konusu sorun ***karma tamsayılı programlama sorunu*** adını alır.

Bazı durumlarda tamsayı değişkenler sadece 0 veya 1 değerlerini alabilir. Bu tip sorunlar da ***tamamen (karma) 0-1 programlama sorunları*** veya ***tamamen (karma) ikili tamsayı programlama sorunları*** olarak isimlendirilir.

5.1 TP'NİN FORMÜLASYONU

5.1.1 Bütçeleme Sorunları

5.1.1.1 Sermaye Bütçeleme Örneği

Farzedelim ki $14,000 yatırım yapmak istiyoruz. Dört yatırım fırsatı belirledik. Yatırım 1 $5,000'lık bir yatırım gerektirmekte ve getirisinin şimdiki değeri $8,000 olmaktadır. Yatırım 2 $7,000'lık bir yatırım gerektirmekte ve getirisinin şimdiki değeri $11,000 olmaktadır. Yatırım 3 $4,000'lık bir yatırım gerektirmekte ve getirisinin şimdiki değeri $6,000 olmaktadır. Yatırım 4 ise $3,000'lık bir yatırım gerektirmekte ve getirisinin şimdiki değeri $4,000 olmaktadır.

Hangi yatırımlara para yatırarak toplam getirimizi enbüyükleyebiliriz?

Yanıt

DP'de olduğu gibi ilk adım değişkenleri belirlemektir. Tamsayı sınırlamasının getirdiği çeşitli karışıklıklar ve numaralar yüzünden TP'de bu işlem daha zordur.

Örnekte her yatırım için bir 0-1 x_j değişkeni kullanacağız..

x_j 1 ise *j*. yatırım yapılır

x_j 0 ise *j*. yatırım yapılmaz

Bu durumda 0-1 programlama sorunu söz konusudur:

$$\begin{array}{ll} \text{maks} & 8x_1 + 11x_2 + 6x_3 + 4x_4 \\ \text{öyle ki} & 5x_1 + 7x_2 + 4x_3 + 3x_4 \le 14 \\ & x_j = 0 \text{ veya } 1 \quad j = 1, \dots 4 \end{array}$$

Doğrudan yapılacak bir değerlendirmede (amaç fonksiyonu katsayılarının kısıt katsayılarına oranlanması) Yatırım 1'in en iyi seçenek olduğu önerilebilir.

Tamsayı kısıtlarına önem verilmezse en iyi DP çözümü aşağıdaki gibidir:

$x_1 = 1$, $x_2 = 1$, $x_3 = 0.5$ ve $x_4 = 0$; toplam getiri değeri $22,000.

Ne yazık ki bu çözüm tamsayılı çözüm değildir. x_3 0'a yuvarlanırsa $19,000.toplam getiri değeri olan olurlu bir çözüm elde edilir.

TP çözüm tekniği kullanılarak en iyi tamsayılı çözüm bulunursa $x_1 = 0$, $x_2 = x_3 = x_4 = 1$ ve toplam getiri değeri $21,000 olur.

Bu örnek yuvarlamanın her zaman en iyi değeri vermediğini gösterir.

5.1.1.2 Çok Dönemli Sermaye Bütçeleme Örneği

Önümüzdeki üç yıllık planlama dönemi için dört projenin değerlendirilmesi yapılacaktır. Her projeye ait beklenen getiriler ile yıllık harcamalar aşağıdaki tabloda gösterilmiştir. Üç yıl boyunca uygulamaya konulacak projeleri belirleyiniz.

		Harcamalar		
Proje	Getiri	Yıl1	Yıl2	Yıl3
1	0.2	0.5	0.3	0.2
2	0.3	1	0.5	0.2
3	0.5	1.5	1.5	0.3
4	0.1	0.1	0.4	0.1
Kullanılabilir sermaye		3.1	2.5	0.4

Yanıt

$x_j = 1$, j. proje uygulanırsa

$x_j = 0$, aksi takdirde (j. proje uygulanmaz).

Kurulacak 0-1 programlama modeli:

Maks $0.2\, x_1 + 0.3\, x_2 + 0.5\, x_3 + 0.1\, x_4$

Öyle ki $0.5\, x_1 + 1\quad x_2 + 1.5\, x_3 + 0.1\, x_4 \leq 3.1$

$0.3\, x_1 + 0.8\, x_2 + 1.5\, x_3 + 0.4\, x_4 \leq 2.5$

$0.2\, x_1 + 0.2\, x_2 + 0.3\, x_3 + 0.1\, x_4 \leq 0.4$

$x_j = 0$ veya $1 \quad j = 1, \ldots 4$

Uyarı

Amaç fonksiyonu ve kısıtlar doğrusaldır. Bu derste Tamsayılı Doğrusal Programlama ile ilgileneceğiz. Doğrusal olmayan TP'ler de vardır – fakat bu dersin kapsamı dışındadırlar.

Sorunun Simpleks Yöntemi ile çözümü sonucunda x_1=0.4975 çıktığını varsayalım. Bu durumda kesirli kısma önem vermeyip yuvarlama işlemi mantıksız olacaktır. Bu yüzden DP yerine, TP algoritmaları kullanılarak en iyi çözüm bulunur.

5.1.1.3 Değiştirilmiş Sermaye Bütçeleme Örneği

Bazı yeni kısıtlarla sorunu genişletelim.

Örneğin, aşağıdaki önermeleri ve getirdiği kısıtları düşünelim:

Sadece iki yatırım yapabiliriz

$x_1 + x_2 + x_3 + x_4 \leq 2$

Üç veya dört adet yatırım yapılması durumunda $x_1 + x_2 + x_3 + x_4 \geq 3$ olacaktır

Eğer Yatırım 2 yapılırsa,Yatırım 4 de yapılmalıdır

$x_2 \leq x_4$ veya $x_2 - x_4 \leq 0$

Eğer x_2 1 ise, x_4 de istediğimiz gibi 1 olur; eğer x_2 0 ise, x_4 için herhangi bir kısıtlama olmaz (x_4 0 veya 1 olur)

Eğer Yatırım 1 yapılırsa, Yatırım 3 yapılamaz

$x_1 + x_3 \leq 1$

Eğer x_1 1 ise, x_3 istediğimiz gibi 0 olur; eğer x_1 0 ise, x_3 için herhangi bir kısıtlama olmaz (x_3 0 veya 1 olur)

Ya Yatırım 1 ya Yatırım 2 yapılabilir

$x_1 + x_2 = 1$

Eğer x_1 1 ise, x_2 0 olur (sadece Yatırım 1 yapılır); eğer x_1 0 ise, x_2 1 olur (sadece Yatırım 2 yapılır).

5.1.2 Sırt Çantası Sorunları

Sadece bir kısıdı olan herhangi bir TP sırt çantası sorunu olarak isimlendirilir.

Bir diğer özellik de kısıdın ve amaç fonksiyonunun katsayıları negatif olmayan sayılardır.

Örneğin aşağıdaki model sırt çantası sorunudur:

Maks $8 x_1 + 11 x_2 + 6 x_3 + 4 x_4$

Öyle ki $5 x_1 + 7 x_2 + 4 x_3 + 3 x_4 \leq 14$

$x_j = 0$ veya 1 $j = 1, \ldots 4$

Geleneksel öykü, bir sırt çantasında (örnekte kapasitesi 14'dür) bazı eşyaları (dört çeşit eşya) taşımak ile ilgilidir. Her eşyanın bir büyüklüğü ve taşıma sonucu elde edilecek bir değeri vardır (eşya 2 için büyüklük 7, değer 11'dir).

Amaç, sırt çantasına sığabilecek eşyaların toplam değerini enbüyüklemektir.

Sırt çantası sorunları genellikle kolay çözülürler.

5.1.3 Sabit Maliyet Sorunları

Çok sayıda üretim ve yer seçimi sorunlarının modellenmesi için de TP kullanılabilir. Burada üretim sorununa örnek olarak Gandhi örneği ve yer seçimi sorununa örnek olarak da Çek tahsilat ofisi örneği verilmiştir.

5.1.3.1 Gandhi Örneği (Üretim)

(Winston 9.2., s. 470)

Gandhi Co sınırlı işçilik ve kumaş kullanarak gömlek, şort ve pantolon üretmektedir. Ayrıca her ürünü üretmek için makine kiralanması söz konusudur.

	Gömlek	Şort	Pant.	Toplam Mevcut
İşçilik (saat/hafta)	3	2	6	150
Kumaş (m^2/hafta)	4	3	4	160
Makine kirası ($/hafta)	200	150	100	
Değişken birim maliyet	6	4	8	
Satış fiyatı	12	8	15	

Yanıt

x_j üretilecek *j*. giysi sayısı olsun

y_j de *j*. giysinin üretim kararı olsun.

$y_j = 1$, *j*. giysi üretilecekse; $y_j = 0$, aksi takdirde.

Kar = Satış geliri – Değişken maliyet – Makine kirası

Örneğin gömlekten elde edilecek kar $z_1 = (12 – 6) x_1 – 200 y_1$.

İşçilik ve kumaş sınırlı olduğundan Gandhi iki kısıtla karşılaşır.

$x_j > 0$ iken $y_j = 1$ olmasını sağlamak için aşağıdaki kısıtlar eklenir

$$x_j \leq M_j y_j$$

Kumaş kısıtı göz önüne alınarak en fazla 40 gömlek yapılabilir ($M_1 = 40$), böylece gömlek için eklenecek kısıt x_1 üzerinde gereksiz bir sınır oluşturmaz (Eğer M_1 yeterince büyük seçilmezse (örn. $M_1 = 10$) x_1'in alabileceği değere gereksiz bir sınır getirilmiş olur)

Benzer şekilde kumaş kısıdı göz önüne alınarak en fazla 53 şort yapılabileceği ($M_2 = 53$) ve işçilik kısıdı göz önüne alınarak en fazla 25 pantolon yapılabileceği ($M_3 = 25$).bulunur.

Böylece aşağıdaki karma 0-1 TP elde edilir::

maks $6 x_1 + 4 x_2 + 7 x_3 – 200 y_1 – 150 y_2 – 100 y_3$

öyle ki $3 x_1 + 2 x_2 + 6 x_3 \leq 150$ (İşçilik kısıdı)

$4 x_1 + 3 x_2 + 4 x_3 \leq 160$ (Kumaş kısıdı)

$x_1 \leq 40 y_1$ (Gömlek üretimi ve makine kısıdı)

$x_2 \leq 53 y_2$ (Şort üretimi ve makine kısıdı)

$x_3 \leq 25 y_3$ (Pantolon üretimi ve makine kısıtı)

$x_1, x_2, x_3 \geq 0$ ve tamsayı

$y_1, y_2, y_3 = 0$ veya 1

Gandhi sorununa en iyi çözüm: $z = \$ 75$, $x_3 = 25$, $y_3 = 1$.

Bu durumda 25 pantolon yapmaları ve bir pantolon makinesi kiralamaları durumunda \$75 kar elde edeceklerdir.

5.1.3.2 Çek Tahsilat Ofisi Örneği (Yer Seçimi)

(Winston 9.2., s. 473'den uyarlanmıştır)

Bir Amerikan şirketinin Birleşik Devletlerdeki müşterilerinin ödemelerini gönderdikleri çekler ile topladığını varsayalım. Çekin tanzim edilip postalanması (müşterinin yükümlülüğünü yerine getirmesi) ile çekin şirketin eline geçmesi (paraya dönüştürülmesi) arasında bir zaman olacağı ve şirketin parayı kullanmasının gecikeceği açıktır. Bu yüzden şirket çeşitli şehirlerde çek tahsilat ofisleri açarak çekleri paraya dönüştürmeyi hızlandırmayı ve zaman kaybından dolayı uğrayacağı zararı azaltmayı hedeflemektedir.

Şirketin batı, ortabatı, doğu ve güney bölgelerindeki müşterileri günde sırasıyla $70,000, $50,000, $60,000 ve $40,000'lık ödemeyi çekle yapsınlar. Şirket L.A., Chicago, New York, ve/veya Atlanta'da çek tahsilat ofisleri açabilir. Bir çek tahsilat ofisini işletmek için yıllık masraf $50,000'dır. Bölgelerden ofislere çeklerin tanzim edilip gönderimi ve paraya çevrilmesi süreleri tabloda verilmiştir.

	LA	Chicago	NY	Atlanta
Batı	2	6	8	8
Ortabatı	6	2	5	5
Doğu	8	5	2	5
Güney	8	5	5	2

Hangi ofisler açılmalıdır? Gönderim zamanı yüzünden oluşacak kayıpları ve ofis masraflarını enazlayacak şekilde bir TP modeli kurunuz. Faiz oranının % 20 olduğunu ve her bölgenin sadece bir şehirdeki ofise çeklerini gönderebileceğini varsayınız.

Yanıt

Öncelikle her atamanın getireceği gecikmeler yüzünden oluşacak kaybı hesaplamalıyız. Örneğin Batı bölgesi çeklerini New York'a gönderirse ortalama olarak günlük işlem gören miktar $560,000 (= 8 * $70.000) olacaktır. Bu durumda % 20 faiz oranı ile yıllık kayıp $112,000 olur. Bu şekilde her bölgeden her şehre gönderimler sonucu oluşacak kayıpları hesaplayıp aşağıdaki tabloyu oluşturabiliriz.

	LA	Chicago	NY	Atlanta
Batı	28	84	112	112
Ortabatı	60	20	50	50
Doğu	96	60	24	60
Güney	64	40	40	16

y_j 0-1 değişken olsun ve j. ofis açılırsa 1 değerini, açılmazsa 0 değerini alsın
x_{ij} = 1, i. bölge j. ofise gönderirse; x_{ij} = 0, aksi takdirde.

Amacımız yıllık kayıpları ve masrafları enazlamaktır:

$$28 x_{11} + 84 x_{12} + 112 x_{13} + \ldots + 50 y_1 + 50 y_2 + 50 y_3 + 50 y_4$$

Bir kısıt kümesi her bölgenin sadece bir şehirdeki ofise çek göndermesini sağlamalıdır:

TOP [j=1'den to n'ye] x_{ij} = 1 her i için

(TOP [j=1'den n'ye], "j'nin 1'den (1 dahil) n'ye (n dahil) alabileceği tüm tamsayı değerleri için toplam" olarak okunmalıdır)

Bir kısıt kümesi de bir bölgenin açık bir ofise atanması koşulunu sağlamalıdır

$$x_{1j} + x_{2j} + x_{3j} + x_{4j} \leq M y_j$$

Dört bölge olduğundan M en az dört olacak şekilde herhangi bir sayıdır

(L.A.'da ofis açılmadığını (y_1 = 0) varsayalım. Bu durumda x_{11}, x_{21}, x_{31} ve x_{41} de 0 olacaktır. Eğer y_1 1 ise x değerleri üzerinde bir kısıtlama olmayacaktır.)

Bu sorun için yirmi değişken (dört y değişkeni, onaltı x değişkeni) ve sekiz kısıt kullanılır ve aşağıdaki 0-1 TP kurulur:

$$\begin{aligned}
\text{Min}\quad & 28 x_{11} + 84 x_{12} + 112 x_{13} + 112 x_{14} \\
& + 60 x_{21} + 20 x_{22} + 50 x_{23} + 50 x_{24} \\
& + 96 x_{31} + 60 x_{32} + 24 x_{33} + 60 x_{34} \\
& + 64 x_{41} + 40 x_{42} + 40 x_{43} + 16 x_{44} \\
& + 50 y_1 + 50 y_2 + 50 y_3 + 50 y_4 \\
\text{st}\quad & x_{11} + x_{12} + x_{13} + x_{14} = 1 \\
& x_{21} + x_{22} + x_{23} + x_{24} = 1 \\
& x_{31} + x_{32} + x_{33} + x_{34} = 1 \\
& x_{41} + x_{42} + x_{43} + x_{44} = 1 \\
& x_{11} + x_{21} + x_{31} + x_{41} \leq 4y_1 \\
& x_{12} + x_{22} + x_{32} + x_{42} \leq 4y_2 \\
& x_{13} + x_{23} + x_{33} + x_{43} \leq 4y_3 \\
& x_{14} + x_{24} + x_{34} + x_{44} \leq 4y_4 \\
& \text{Tüm } x_{ij} \text{ ve } y_j = 0 \text{ veya } 1
\end{aligned}$$

Aynı sorun için başka formülasyonlar da yapılabilir.

Örneğin aşağıdaki şekilde oluşturulan dört kısıt yerine

$$x_{1j} + x_{2j} + x_{3j} + x_{4j} \leq M\ y_j$$

onlatı kısıtlı başka bir küme kullanılabilir:

$$x_{ij} \leq y_j \qquad i = 1, 2, 3, 4;\ j = 1, 2, 3, 4$$

Bu kısıtlar da bir bölgenin açık bir ofisi kullanmasını sağlar.

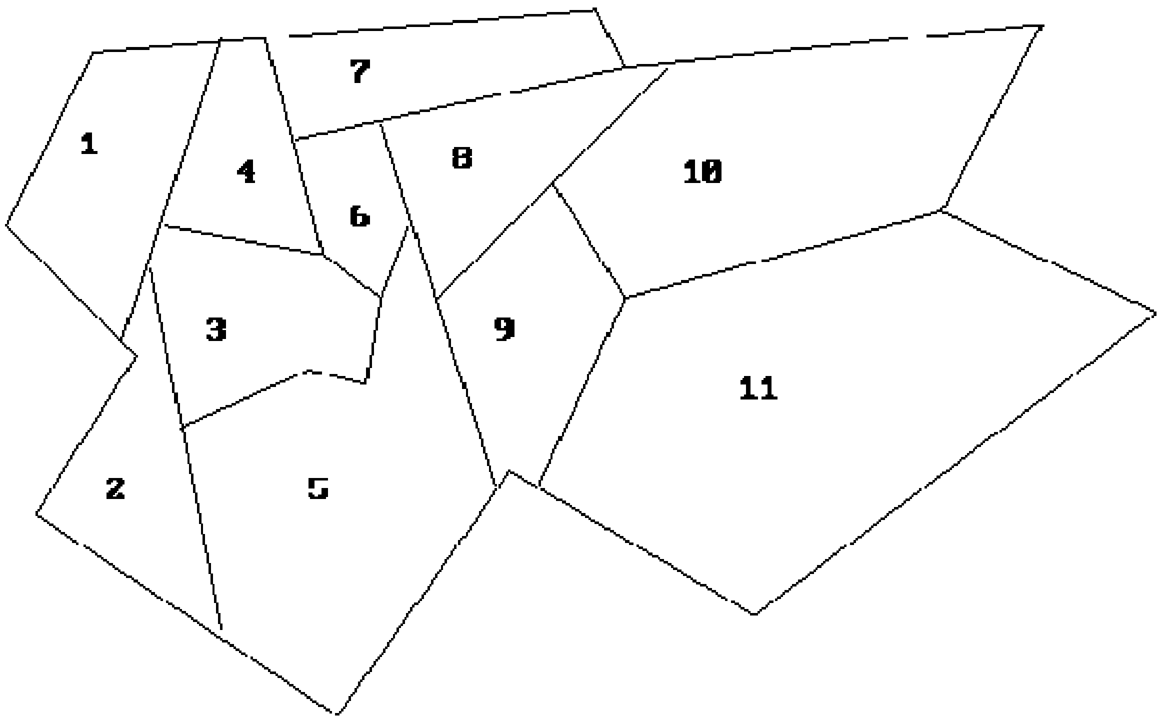
(y_j 0 ise ilgili dört kısıt kullanılarak x_{1j}, x_{2j}, x_{3j} ve x_{4j}'nin de 0 olduğu bulunur. Eğer y_j 1 ise x değerleri üzerinde kısıtlama olmaz.

Diğer bir bakış açısı da $x_{ij} = 1$ iken istediğimiz şekilde $y_j = 1$ olur. Aynı zamanda $x_{1j} = x_{2j} = x_{3j} = x_{4j} = 0$ iken y değerinde bir kısıtlama yapmaz. Ofis maliyetlerinin enazlanmak istenmesi y_j'yi 0 yapar)

Bu şekilde daha büyük bir modelin (yirmi değişken ve yirmi kısıt) formülasyonu daha az etkin gözükse de söz konusu modeli DP olarak varsayarak çözersek tamsayılı bir çözüm buluruz. Çözüm bu yüzden TP için de en iyidir!)

5.1.4 Kapsama Sorunu

Bu modeli anlatabilmek için, aşağıdaki yerleşme sorununu düşünelim. Bir şehirde itfaiye merkezlerinin yeri gözden geçirilmektedir. Aşağıdaki şekildeki gibi şehrin onbir adet ilçesi vardır.



Bir itfaiye merkezi herhangi bir ilçede kurulabilir. Merkez kendi ilçesindeki ve komşu ilçelerdeki yangınlara müdahale edebilir. Amaç merkez sayısını enazlayarak tüm ili yangına karşı koruyacak bir kapsama alanı kurmaktır.

Yanıt

Her j şehri için bir x_j değişkeni kullanabiliriz..

Şehirde itfaiye merkezi varsa x değişkeni 1 değerini, yoksa 0 değerini alacaktır. Bu durumda aşağıdaki model kurulabilir:

$$
\begin{array}{ll}
\text{Min} & x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} + x_{11} \\
\text{st} & x_1 + x_2 + x_3 + x_4 \geq 1 \text{ (şehir 1)} \\
& x_1 + x_2 + x_3 + x_5 \geq 1 \text{ (şehir 2)} \\
& x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 1 \text{ (şehir 3)} \\
& x_1 + x_3 + x_4 + x_6 + x_7 \geq 1 \text{ (şehir 4)} \\
& x_2 + x_3 + x_5 + x_6 + x_8 + x_9 \geq 1 \text{ (şehir 5)} \\
& x_3 + x_4 + x_5 + x_6 + x_7 + x_8 \geq 1 \text{ (şehir 6)} \\
& x_4 + x_6 + x_7 + x_8 \geq 1 \text{ (şehir 7)} \\
& x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} \geq 1 \text{ (şehir 8)} \\
& x_5 + x_8 + x_9 + x_{10} + x_{11} \geq 1 \text{ (şehir 9)} \\
& x_8 + x_9 + x_{10} + x_{11} \geq 1 \text{ (şehir 10)} \\
& x_9 + x_{10} + x_{11} \geq 1 \text{ (şehir 11)} \\
& \text{Her } x_j = 0 \text{ veya } 1
\end{array}
$$

İlk kısıt, Şehir 1'deki yangınların söndürülmesi için o şehirde veya komşu şehirlerde toplam en az bir merkezin kurulmasını sağlar.
a_{ij} kısıt katsayının şehir i'nin şehir j'ye komşu olması durumunda 1; i=j veya komşu olmaması durumunda 0 olduğuna dikkat ediniz.

Kısıt matrisinin j. sütunu, j. şehrindeki bir merkez tarafından hizmet görebilecek şehirler kümesini temsil eder. Çözüm olarak aranan, en az sayıda kümenin birleşiminin tüm şehirleri kapsamasıdır.

5.1.5 Eğer-Öyleyse Kısıtı

Aşağıdaki durumla matematik programlama sorunlarında karşılaşılabilir.

$f(x_1, x_2,\ldots, x_n) > 0$ kısıtının sağlanması durumunda $g(x_1, x_2,\ldots, x_n) \geq 0$:kısıtının da sağlanması istenir.

Eğer $f(x_1, x_2,\ldots, x_n) > 0$ sağlanmazsa $g(x_1, x_2,\ldots, x_n) \geq 0$'nin sağlanıp sağlanmaması önemli değildir.

Söz konusu koşulu sağlamak için aşağıdaki kısıtları formülasyona eklemeliyiz:

$$-g(x_1, x_2,\ldots, x_n) \leq M\,y \qquad (1)$$

$$f(x_1, x_2,\ldots, x_n) \leq M\,(1 - y) \qquad (2)$$

Burada y bir 0-1 değişken ve M de yeterince büyük bir sayıdır.

M, sorunun diğer kısıtlarını sağlayan her x_j değişkeni için $-g \leq M$ ve $f \leq M$ kısıtlarını sağlamalıdır.

$f > 0$ ise (2)'nin sağlanması için $y = 0$ olmalıdır. $y = 0$ iken, (1) $-g \leq 0$ veya $g \geq 0$ olmasını sağlar. Bu durumda istenilen eğer-öyleyse koşulu gerçekleşir.

Örnek

Çek ofisi örneğine takip eden koşulu ekleyelim. Eğer 1. bölgedeki müşteri ödemelerini Şehir 1'e göndermek istiyorsa başka hiç bir bölge bu ofisi kullanamaz .Matematiksel olarak,

Eğer $x_{11} = 1$ ise $x_{21} = x_{31} = x_{41} = 0$

Tüm değişkenler 0-1 olduğundan aşağıdaki çevrimi yapabiliriz:

Eğer $x_{11} > 0$ ise $x_{21} + x_{31} + x_{41} \leq 0$ (veya $-x_{21} - x_{31} - x_{41} \geq 0$)

Eğer $f = x_{11}$ ve $g = -x_{21} - x_{31} - x_{41}$ şeklinde tanımlarsak, aşağıdaki kısıtları kullanarak koşulu gerçekleştirebiliriz:

$$x_{21} + x_{31} + x_{41} \leq My$$

$$x_{11} \leq M\,(1 - y)$$

$$y = 0 \text{ veya } 1$$

$-g$ ve f hiç bir zaman 3'ü geçemeyeceklerinden, M 3 olarak alınabilir.

5.1.6 Ya – Ya da Kısıtları

Aşağıdaki durumla matematik programlama sorunlarında karşılaşılabilir.

İki farklı biçimde kısıtlar verilsin:

$f(x_1, x_2, \dots, x_n) \leq 0$ (1)

$g(x_1, x_2, \dots, x_n) \leq 0$ (2)

eğer (1) veya (2)'den ***en az*** birinin sağlanmasını istersek bu kısıtlar ***ya – ya da*** kısıtları olarak isimlendirilir.

Söz konusu koşulu sağlamak için (1') ve (2') kısıtlarını formülasyona eklemeliyiz:

$f(x_1, x_2, \dots, x_n) \leq M y$ (1')

$g(x_1, x_2, \dots, x_n) \leq M(1 - y)$ (2')

Burada y bir 0-1 değişkenidir

M ise sorunun diğer kısıtlarını sağlayan her x_j değişkeni için $f(x_1, x_2, \dots, x_n) \leq M$ ve $g(x_1, x_2, \dots, x_n) \leq M$ koşullarını sağlayacak şekilde yeterince büyük bir sayıdır.

y = 0 iken (1) mutlaka ve (2) bazı durumlarda sağlanır.

y = 1 iken (2) mutlaka ve (1) bazı durumlarda sağlanır.

Örnek

Bir otomobil üretimi için 1.5 ton çelik ve 30 saat işçilik gerekmektedir. Mevcut miktarlar 6,000 ton çelik ve 60,000 saat işçiliktir.

Ekonomik olarak uygun bir üretim için en az 1000 otomobil üretilmelidir.

Kısıt: $x_1 \leq 0$ veya $x_1 \geq 1000$

$[f(x_1, x_2, \dots, x_n) = x_1;\ g(x_1, x_2, \dots, x_n) = 1000 - x_1]$

İşaret sınırlaması $x_1 \geq 0$ ve tamsayı

Kısıtı aşağıdaki kısıt çifti ile değiştirerek modeli doğrusallaştırabiliriz:

$$x_1 \leq M_1 y_1$$

$$1000 - x_1 \leq M_1 (1 - y_1)$$

$$y_1 = 0 \text{ veya } 1$$

$$M_1 = \min(6.000/1.5,\ 60.000/30) = 2000$$

5.1.7 Gezgin Satıcı Sorunu

Bir satıcı on adet şehiri birer kez ziyaret edip evine dönecektir. "Satıcının yapacağı bu gezide katedeceği toplam uzaklığı enazlayacak rota (şehir sırası) nasıl olmalıdır?" sorunu ***Gezgin Satıcı Sorunu*** (***GSS***; Traveling Salesperson Problem – TSP) olarak isimlendirilir.

GSS'nin TP Formülasyonu

N adet şehir olduğunu varsayalım

$i \neq j$ için c_{ij} = şehir i'den şehir j'ye uzaklık ve

c_{ii} = M (gerçek uzaklıklara göre çok büyük bir uzaklık) olsun

Aynı zamanda x_{ij} aşağıdaki şekilde bir 0-1 değişken olarak tanımlansın:

x_{ij} = 1, GSS çözümü şehir i'den şehir j'ye gitmeyi önersin

x_{ij} = 0, aksi takdirde

GSS formülasyonu:

min TOP[i] TOP[j] c_{ij} x_{ij}

öyle ki TOP[i] x_{ij} = 1 her j için

TOP[j] x_{ij} = 1 her i için

$u_i - u_j + N x_{ij} \leq N - 1$ her $i \neq j$ için; i=2,3,…,N; j=2,3,…,N

Her x_{ij} = 0 veya 1, Her $u_i \geq 0$

(TOP[j], "j'nin 1'den (1 dahil) n'ye (n dahil) alabileceği tüm tamsayı değerleri için toplam" olarak okunmalıdır)

İlk kısıt kümesi satıcının her şehire salt bir kez uğramasını sağlar.

İkinci kısıt kümesi satıcının her şehirden salt bir kez ayrılmasını sağlar.

Üçüncü kısıt kümesi aşağıdakileri sağlar:

Tam tur yapmayan herhangi bir x_{ij}'ler kümesi olurlu olmayan çözümdür Tam tur yapan herhangi bir x_{ij}'ler kümesi olurlu çözümdür

UYARI: Büyük GSS için kurulacak bir TP modelinin çözümü etkin değildir karmaşık ve zordur. Çok sayıda şehirin söz konusu olması durumunda GSS'yi dal sınır yöntemi ile çözmek çok fazla bilgisayar zamanı gerektirir. Bu yüzden genellikle sezgisel yöntemler kullanılarak (her zaman en iyi çözüm bulunmasa da) GSS çözülür.

5.2 TP'NİN ÇÖZÜMÜ

"TP Sorunlarının Formülasyonu" bölümünde belirli bir sayıda tamsayılı programlama örneği inceledik. Bu tip sorunların çözümü için başlıca iki yaklaşım vardır.

İlk geliştirilen yöntem ***kesme düzlemleri*** (cutting planes) olarak isimlendirilir. Tamsayılığı sağlamak için kısıtlar eklemeye dayanır.

1980'lerden beri kullanılan ve daha etkin olarak nitelendirilen yöntem ise ***dal sınır*** (branch and bound) olarak isimlendirilir. Sorunu alt sorunlara bölen bir ağaç arama yaklaşımıdır.

Aslında, tüm yaklaşımlar tekrarlı bir şekilde DP'ler çözmeyi içerir. DP çözümü için g*enel amaçlı* (çözülen DP'den bağımsız herhangi bir DP'yi çözebilecek yapıda) ve *etkin işlem yapabilen* (büyük DP'leri çözebilen) algoritmalar (örn. simpleks) vardır. TP çözümü için ise hem genel amaçlı hem de etkin işlem yapabilen benzer bir algoritma yoktur.

TP için çözüm yöntemleri iki sınıfa ayrılabilir:

- *Genel amaçlı* (herhangi bir TP'yi çözebilecek yapıda) fakat etkin işlem yapamayan (göreli olarak daha küçük sorunları çözen); veya
- *Özel amaçlı* (Belirli bir tipteki TP sorunu için geliştirilen) ve daha etkin işlem yapabilen.

TP çözüm yöntemleri aynı zamanda çözüme gitmeleri açısından aşağıdaki gibi de sınıflandırılabilir:

- *En iyi (Optimal)*
- *Sezgisel (Heuristic)*

"En iyi" algoritma matematiksel olarak en iyi çözümü bulmayı *garantiler*

Bazı durumlarda en iyi çözümle çok fazla ilgilenmeyebiliriz çünkü:

- çözmek istediğimiz sorunun büyüklüğü mevcut en iyi algoritmalarının etkin işlem yapabilme sınırlarının üstündedir (kullanabileceğimiz bilgisayar zamanı gibi)
- harcanacak zamana, paraya en iyi çözümü bulmak değmez, yaklaşık bir çözüm yeterli görülebilir

Bu gibi durumlarda sezgisel algoritmalar kullanılabilir. Bulunan çözüm olurlu bir çözümdür ve büyük bir olasılıkla iyi tasarlanmış bir yöntem kullanılıyorsa en iyi çözüme yakın kaliteli bir çözümdür.

Bu durumda çözüm algoritmaları için dört farklı kategori vardır:

- Genel Amaçlı, En iyi

  Sayma, Dal sınır, kesme düzlemi

- Genel Amaçlı, Sezgisel (*ders kapsamında değil*)

  Genel amaçlı bir en iyi algoritmasını çalıştırmak ve belirli bir süre sonunda durmak

- Özel Amaçlı, En iyi (*ders kapsamında değil*)

  Sınır üretmeye dayalı ağaç arama yaklaşımları

- Özel amaçlı, Sezgisel (*ders kapsamında değil*)

  Sınır tabanlı sezgiseller, tabu arama, genetik algoritmalar....

5.2.1 DP Gevşetmesi (DP ile İlişki)

Herhangi bir TP için aynı amaç fonksiyonu ve kısıtları kullanarak fakat değişkenlerin tamsayı olma gereksinimini aşağıdaki gibi değiştirerek bir DP elde edebiliriz:

"x_i = 0 veya 1" değişkeni 0 <= x_i <= 1 sürekli aralığında değerler alır

"$x_i \geq 0$ ve tamsayı" değişkeni salt x_i >= 0 işaret sınırlamasını sağlar

Değişkenler üzerindeki tüm tamsayı ve 0-1 şartlarını yoksayarak elde edilen DP, ***TP'nin DP Gevşetmesi*** (LP Relaxation of the IP) olarak isimlendirilir. Söz konusu doğrusal gevşetmeyi (DG) DP çözüm algoritmalarını kullanarak çözebiliriz.

Eğer DG'nin en iyi çözümündeki tüm değişkenler tamsayılı değerler alıyorlarsa bu durumda bulunan en iyi çözüm orijinal TP sorununun da en iyi çözümüdür (*doğal tamsayılı DP*)

DG, TP sorununa göre daha az kısıtlı (gevşek) olduğundan aşağıdaki durumlarla karşılaşılabilir:

- Eğer TP enbüyükleme sorunu ise, DG'nin en iyi amaç değeri TP'ninkine eşit veya daha büyüktür.
- Eğer TP enküçükleme sorunu ise, DG'nin en iyi amaç değeri TP'ninkine eşit veya daha küçüktür.
- Eğer DG olurlu değilse (olurlu çözümü yoksa), TP de olurlu değildir.

Bu durumda DG'nin çözülmesinin bir bilgi vereceği açıktır: en iyi amaç değeri sınırı belirlenir ve şanslı isek en iyi amaç değerini buluruz. Fakat daha önce de gördüğümüz gibi en iyi çözümdeki değişken değerlerini tamsayı değerlere yuvarlamak genel olarak en iyi TP çözümünü vermeyebilir; hatta yeni çözüm olurlu bile olmayabilir.

Genel olarak DP tabanlı TP çözüm paketleri aşağıdaki çözüm sürecini kullanır

- TP'nin belirlenmesi: amaç, kısıtlar, tamsayılı değişkenler (x_j = tamsayı ve 0 ile n arasında).
- DP Gevşetmesinin yapılması: TP'deki amaç ve kısıtlara ek olarak $0<=x_j<=n$ fakat tamsayı olma şartının kaldırılması.
- Dal sınır algoritması ile TP'nin en iyi çözümünün bulunması

5.2.2 Sayma

DP'den (değişkenlerin sürekli aralıkta değerler alabildiği) farklı olarak TP'de her değişken sadece sonlu sayıda kesikli (tamsayı) değerler alabilir.

Tüm olası çözümleri ***sayma*** (enumerate) – her biri için amaç fonksiyon değerini hesaplama ve olurlu çözümlerden en iyisini seçme – bir çözüm yaklaşımı olabilir.

Örneğin aşağıdaki çok dönemli sermaye bütçeleme sorununu inceleyelim,

$$\begin{array}{ll} \text{maks} & 0.2\,x_1 + 0.3\,x_2 + 0.5\,x_3 + 0.1\,x_4 \\ \text{Öyle ki} & 0.5\,x_1 + 1\;\;x_2 + 1.5\,x_3 + 0.1\,x_4 \le 3.1 \\ & 0.3\,x_1 + 0.8\,x_2 + 1.5\,x_3 + 0.4\,x_4 \le 2.5 \\ & 0.2\,x_1 + 0.2\,x_2 + 0.3\,x_3 + 0.1\,x_4 \le 0.4 \\ & x_j = 0 \text{ or } 1 \qquad j = 1, \ldots 4 \end{array}$$

Bu sorun için 2^4=16 olası çözüm vardır. Bunlar:

x_1	x_2	x_3	x_4	
0	0	0	0	hiç proje uygulama
0	0	0	1	bir proje uygula
0	0	1	0	
0	1	0	0	
1	0	0	0	
0	0	1	1	iki proje uygula
0	1	0	1	
1	0	0	1	
0	1	1	0	
1	0	1	0	
1	1	0	0	
1	1	1	0	üç proje uygula
1	1	0	1	
1	0	1	1	
0	1	1	1	
1	1	1	1	dört proje uygula

En iyi çözümü kesin olarak bulabilmek için, örneğimizde, 16 olası çözümü hesaplamamız gerekmektedir. Bu örnek TP için genel bir doğruyu gösterir. Küçük bir sorunu kolay bir şekilde çözmek için kullanılacak algoritma, sorun büyüdükçe artan bir hızla zorlaşmaktadır.

Örneğin ikili değerler (0-1) alabilen 100 değişkenli bir TP modeli için tek tek sayılacak olası çözüm sayısı 2x2x2x ... x2 = 2^{100} (yaklaşık olarak 10^{30}) adettir Tüm olası çözümleri saymak ve olurlular içinden en iyisini seçmek ***kavramsal*** olarak bir sorun yaratmasa da ***işlemsel*** (sayısal) olarak imkansızdır.

5.2.3 Dal Sınır Algoritması

Sistematik bir şekilde olurlu çözümlerin sayılarak en iyi tamsayılı çözümün bulunması için kullanılan genel amaçlı bir DP tabanlı ağaç arama (LP-based tree search) yöntemi olan ***Dal Sınır Algoritması*** 1960'ların başında Land ve Doig tarafından önerilmiştir.

Sayma yöntemi gibi tüm olurlu çözümleri saymak yerine sadece belirli sayıda olurlu çözümü inceleterek (küçük bir kısmının inceleneceği ümidi ile) en iyi çözümü garanti bir şekilde bulur.

5.2.3.1 Örnek

Çok dönemli sermaye bütçeleme sorunu için TP modeli:

maks $\quad 0.2\,x_1 + 0.3\,x_2 + 0.5\,x_3 + 0.1\,x_4$

Öyle ki $\quad 0.5\,x_1 + 1\quad x_2 + 1.5\,x_3 + 0.1\,x_4 \leq 3.1$

$0.3\,x_1 + 0.8\,x_2 + 1.5\,x_3 + 0.4\,x_4 \leq 2.5$

$0.2\,x_1 + 0.2\,x_2 + 0.3\,x_3 + 0.1\,x_4 \leq 0.4$

$x_j = 0$ veya $1 \quad j = 1, \dots 4$

Bu sorunun çözümünü güçleştiren değişkenlerin tamsayı değerler (sıfır veya bir) almaya zorlanmasıydı.

Eğer değişkenler kesirli değerler alabilseydi (sıfır ve bir arasındaki tüm değerler) sorunu DP olarak çözebilirdik.

Sorunun bu şekilde bir DP gevşetmesini [$x_j = 0$ veya 1 (j=1,...,4) yerine 0 <= x_j <= 1 (j=1,...,4) kısıtlarını kullanalım] çözeceğimizi varsayalım.

Herhangi bir DP çözüm paketi ile x_2=0.5, x_3=1, x_1=x_4=0 ve en iyi amaç fonksiyon değeri 0.65 olarak hesaplanır.

Bu durumda esas tamsayılı sorunumuzun en iyi amaç fonksiyon değeri hakkında fikir sahibi oluruz (<= 0.65). 0.65 en iyi tamsayılı çözüm için bir ***üst sınır***dır (upper bound).

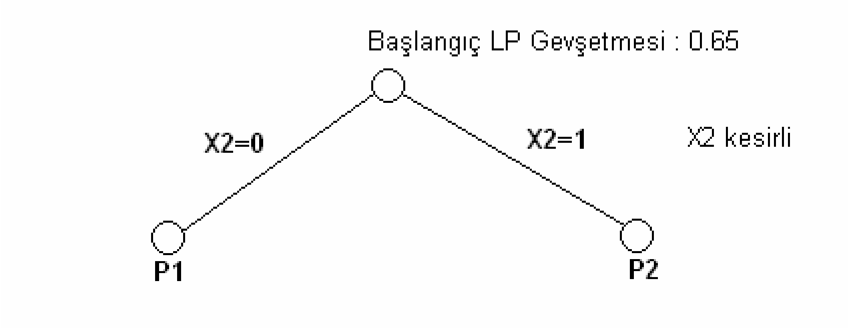
Söz konusu değerin üst sınır olmasının nedeni tamsayı zorunluluğunu kaldırınca elde edilen DP gevşetmesinin çözümünün değerinin en az tamsayılı çözümünki kadar (belki de daha iyi) olacağıdır.

Elde edilen DP gevşetmesi çözümünde x_2 değişkeninin tamsayılı yerine kesirli bir değer aldığını görürüz. Tamsayılı değer alması için iki yeni sorunun çözümü ile ilgilenebiliriz:

P1: orijinal DP gevşetmesine ek olarak x_2=0

P2: orijinal DP gevşetmesine ek olarak x_2=1

Bu şekilde sadece bir değişkenin kesirli değer yerine tamsayılı değer almaya zorlanarak yeni çözüm aranması işlemine ***dal işlemi*** (branching) denilir. Dal işlemi, aşağıdaki gibi bir ağaç üzerinde (ağaç arama isminin nasıl ortaya çıktığını gösterecek şekilde) şematize edilebilir.

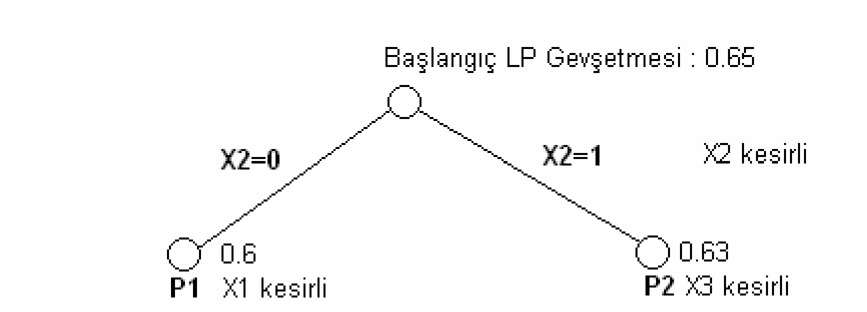


P1 ve P2 DP gevşetmelerini çözersek, aşağıdaki çözümleri elde ederiz:

P1'in çözümü x_1=0.5, x_3=1, x_2=x_4=0, amaç fn. değeri 0.6

P2'in çözümü x_2=1, x_3=0.67, x_1=x_4=0, amaç fn. değeri 0.63

Aşağıdaki şekilde de görüldüğü gibi yeni çözümlerdeki değişken değerleri de tamsayılı değildir:



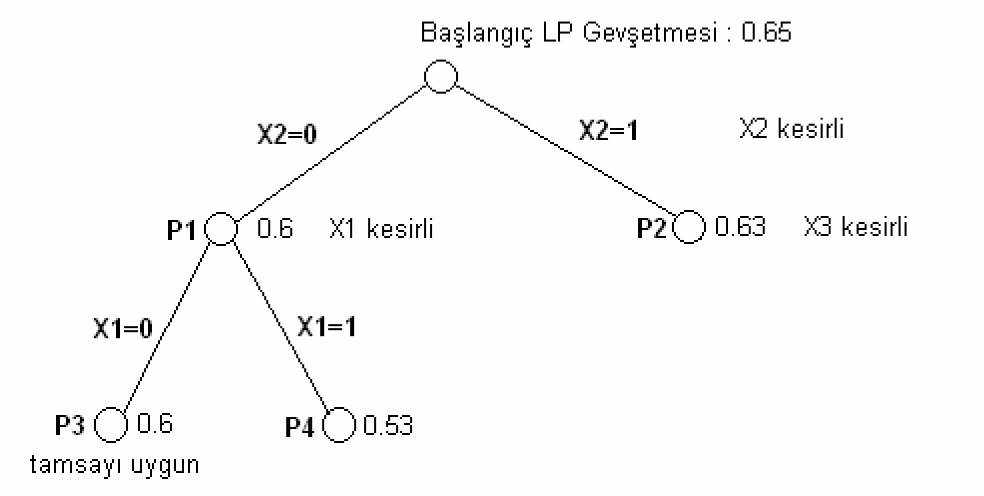
En iyi tamsayılı çözümü bulmak için süreci tekrar ederiz, kesirli değer alan bir değişkeni tamsayılı olmaya zorlayarak iki yeni sorun üretiriz..

P1 sorunu üzerinde dal işlemi yapalım. Bu durumda elde edilecek DP gevşetmeleri ve çözümlerinin listesi:

P3 (P1'e ek olarak x_1=0) çözümü x_3=x_4=1, x_1=x_2=0, amaç fn.: 0.6

P4 (P1'e ek olarak x_1=1) çözümü x_1=1, x_3=0.67, x_2=x_4=0, amaç fn. 0.53

P2 çözümü x_2=1, x_3=0.67, x_1=x_4=0, amaç fn: 0.63



Görüldüğü gibi bu adımda P3 sorununun çözümü 0.6'dır ve tüm değişkenlerin değerleri tamsayılıdır. P3 üzerinde yeni bir dal işlemi yapmaya gerek kalmaz ve DP gevşetmeleri listesinden çıkarılır.

En iyi tamsayılı çözüm değeri hakkında yeni bir bilgi elde etmiş oluruz: amaç fonksiyonunun en iyi çözüm değeri 0.6 ile 0.65 arasında (her iki değer dahil) olacaktır.

P4'ü düşünelim, şu anki değeri 0.53'dür ve x_3 değişkeni kesirlidir. Yeni bir dal işlemi yapmamız durumunda amaç fonksiyon değeri daha iyi duruma gelemez. Zaten 0.6'lık bir tamsayılı çözümümüz olduğundan P4 de DP gevşetmeleri listesinden çıkarılır. Bu şekilde ***alt sınır***dan (lower bound) daha iyi bir olurlu çözüm bulunamayacağı için eleme yapılmasına ***sınır işlemi*** (branching) denilir.

Geriye sadece P2 kalır:

P2 çözümü x_2=1, x_3=0.67, x_1=x_4=0, amaç fn: 0.63

P2 için dal işlemi yapılırsa

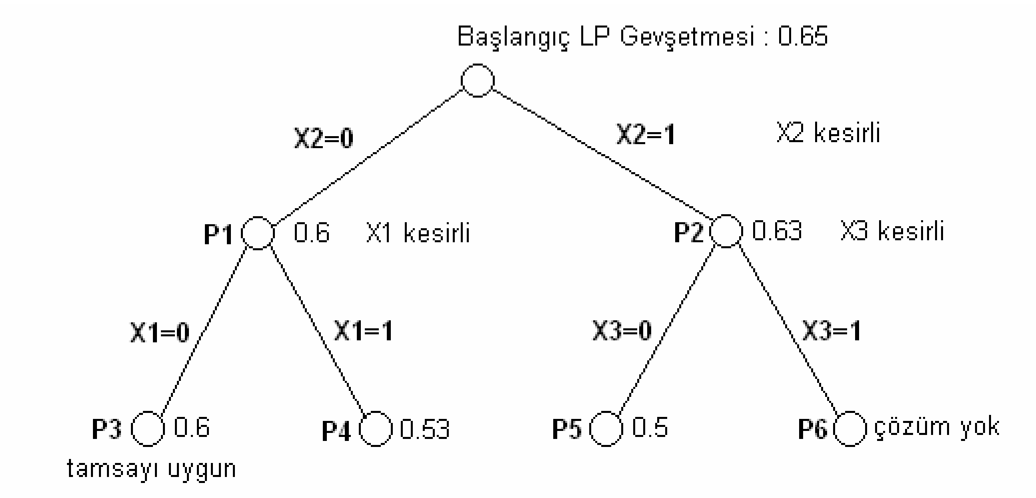
P5 (P2'ye ek olarak x_3=0) çözümü x_1=x_2=1, x_3=x_4=0, amaç fn: 0.5

P6 (P2'ye ek olarak x_3=1) çözümü olurlu değil

Bu durumda ne P5 ne de P6'dan dal işlemi yapmak gerekmez.

En iyi tamsayılı çözüm (esas sorunun çözümü) amaç fonksiyon değeri 0.6 ve x_3=x_4=1, x_1=x_2=0 şeklindedir.

En iyi çözümü elde etmek için kullanılan tüm süreç:



Dikkat edileceği gibi sayma yönteminde sayılacak tüm olası çözümler 16 (2^4) adet iken biz sadece 7 DP çözdük. Dal sınır algoritması çok büyük sorunlarla ilgilenilmemesi durumunda en iyi tamsayılı çözümü bulmak için etkin bir yoldur.

5.2.3.2 Grafik Çözümde Dal Sınır Algoritması Örneği

(Winston 9.3., s. 503)

DP Gevşetmesi ve ilk iki alt problem

(http://www.isl.itu.edu.tr/ya/branch_and_bound_graphical_f1.jpg)

Geri kalan çözüm

(http://www.isl.itu.edu.tr/ya/branch_and_bound_graphical_f2.jpg)

LINDO ÇIKTISI:

```
     MAX      8 X1 + 5 X2
     SUBJECT TO
         2)    X1 + X2 <=   6
         3)    9 X1 + 5 X2 <=   45
     END
     GIN       2

 LP OPTIMUM FOUND AT STEP 2
 OBJECTIVE VALUE = 41.2500000

 SET X1 TO <= 3 AT 1, BND= 39.00 TWIN= 41.00   9

 NEW INTEGER SOLUTION OF 39.0000000 AT BRANCH 1 PIVOT 9
 BOUND ON OPTIMUM:  41.00000
 FLIP X1 TO >= 4 AT 1 WITH BND= 41.000000
 SET X2 TO <= 1 AT 2, BND= 40.56 TWIN=-0.1000E+31   12
 SET X1 TO >= 5 AT 3, BND= 40.00 TWIN= 37.00   15

 NEW INTEGER SOLUTION OF 40.0000000 AT BRANCH 3 PIVOT 15
 BOUND ON OPTIMUM:  40.00000
 DELETE       X1 AT LEVEL     3
 DELETE       X2 AT LEVEL     2
 DELETE       X1 AT LEVEL     1
 ENUMERATION COMPLETE. BRANCHES= 3 PIVOTS= 15

 LAST INTEGER SOLUTION IS THE BEST FOUND
 RE-INSTALLING BEST SOLUTION...

       OBJECTIVE FUNCTION VALUE

       1)       40.00000

  VARIABLE        VALUE          REDUCED COST
        X1         5.000000        -8.000000
        X2         0.000000        -5.000000

       ROW   SLACK OR SURPLUS     DUAL PRICES
        2)         1.000000         0.000000
        3)         0.000000         0.000000

 NO. ITERATIONS=       15
 BRANCHES=     3 DETERM.=  1.000E    0
```

5.2.4 Kesme Düzlemi

Tamsayılı Programlama sorunlarının çözümü için Dal Sınır algoritmasına alternatif bir yaklaşım ***Kesme Düzlemi*** (Cutting Planes) yöntemidir.

Kesme düzleminde temel fikir, DP'ye kısıtlar ekleyerek tamsayılı değerler barındıran en iyi olurlu çözüme ulaşmaktır.

Tabii ki çözmek istediğimiz sorunu değiştirmemek için hangi kısıtları ekleyeceğimize dikkat etmeliyiz. Ekleyeceğimiz özel kısıtlar ***kesme*** (cut) olarak isimlendirilir.

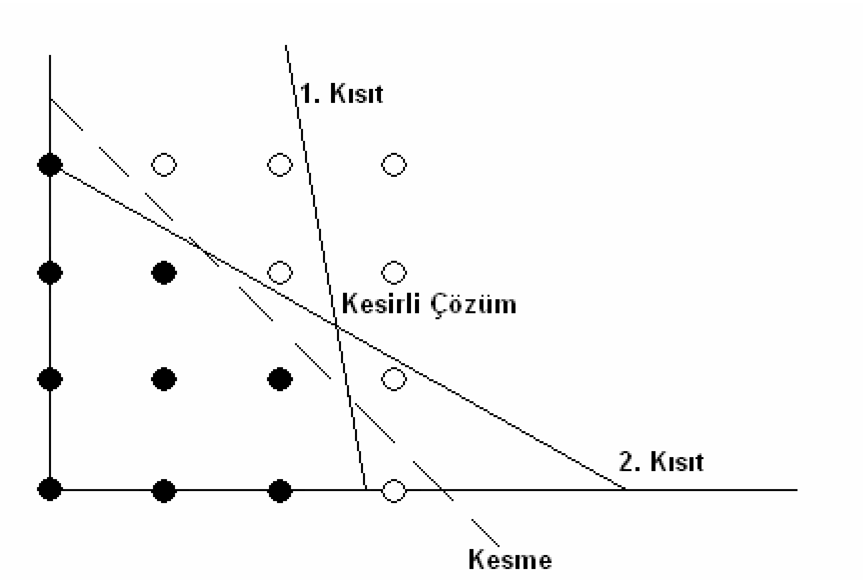
Bir kesirli çözüm için eklenecek kesme aşağıdaki kriterleri sağlamalıdır:

Tüm olurlu tamsayılı çözümler kesme için de olurlu olmalıdır

O anki kesirli çözüm kesme için olurlu olmamalıdır.

Söz konusu kriterler aşağıdaki gibi gösterilebilir

.



Kesme üretmek için iki farklı yaklaşım vardır:

Gomory kesmeleri bir DP tablosundan kesme üretir. Bu yaklaşımın üstünlüğü herhangi bir sorunu çözebilmesi, zayıf yönü ise çok yavaş olmasıdır..

İkinci yaklaşım ise sorunun yapısını kullanarak çok iyi kesmeler üretmeye dayanır. Sorundan soruna değişen bu yaklaşım daha etkin hesaplama yapar.

5.2.4.1 Algoritma Adımları

1. TP'nin DP gevşetmesi için en iyi çözüm tablosunu bulunuz. Eğer tüm değişkenler tamsayılı değerler almışsa en iyi çözüm bulunmuştur! Aksi takdirde bir sonraki adıma geçiniz
2. ST değerinin kesirli kısmı ½'ye en yakın olan kısıtı seçiniz.
3. Seçilen kısıt için tüm tamsayı değerlerini (aşağıya yuvarlayıp) sol tarafa ve artan tüm kesirli değerleri sağ tarafa toplayınız.
4. "Değiştirilmiş kısıtın yeni ST'si" $\leq$ 0 olacak şekilde bir kesme üretiniz
5. Dual simpleks yöntemi kullanarak kesme eklenen yeni DP gevşetmesinin en iyi çözümünü bulunuz. Eğer tüm değişkenler tamsayılı değerler almışsa en iyi çözüm bulunmuştur! Aksi takdirde ikinci adıma dönünüz

5.2.4.2 Dual Simpleks Yöntemi

(Enbüyükleme sorunu için)

En negatif ST'yi seçeriz

Bu pivot satırın temel değişkeni çözümden çıkar

Pivot satırdaki negatif katsayılı değişkenler için oranlar hesaplanır (sıfırıncı satırdaki katsayı / pivot satırdaki katsayı)

Mutlak değerce en küçük oranlı değişen çözüme girer.

5.2.4.3 Örnek

Aşağıdaki TP söz konusu olsun:

maks $z = 8x1 + 5x2$
st $x1 + x2 \leq 6$
$9x1 + 5x2 \leq 45$
$x1, x2 \geq 0$ ve tamsayı

Eğer tamsayı olma koşulunu yoksayarsak ve DP gevşetmesini çözersek elde edeceğimiz en iyi çözüm tablosu::

z	x1	x2	s1	s2	ST
1	0	0	1.25	0.75	41.25
0	0	1	2.25	- 0.25	2.25
0	1	0	- 1.25	0.25	3.75

ST değerinin kesirli kısmı ½'ye en yakın olan kısıtı seçelim (ikisi de eşit uzaklıkta olduğu için herhangi biri seçilebilir):

$x1 - 1.25s1 + 0.25s2 = 3.75$

Seçilen kısıt için tüm tamsayı değerlerini (aşağıya yuvarlayıp) sol tarafa ve artan tüm kesirli değerleri sağ tarafa toplayalım:

$$x1 – 2\ s1 + 0\ s2 – 3 = 0.75 – 0.75\ s1 – 0.25\ s2$$

Sol taraf sadece tamsayılı değer aldığından, sağ taraf değeri tamsayı yapılmalıdır. Şu anda bir pozitif kesirli sayıdan bazı pozitif değerler çıkarılmaktadır. Bu durumda sağ taraf pozitif olamaz ($s \geq 0$) ve aşağıdaki kısıt üretilebilir.

$$0.75 – 0.75\ s1 – 0.25\ s2 \leq 0$$

Tüm olurlu tamsayılı çözümler üretilen kısıt için de olurludur. Fakat şu anda s1 ve s2 0 olduğundan mevcut en iyi (kesirli) çözüm olurlu değildir. Böylece eklenen kısıt bir kesme olur ve *Gomory kesme düzlemi* olarak isimlendirilir.

Kesmeyi DP'ye ekleyerek yeni bir çözüm ararız.

z	x1	x2	**s1**	s2	s3	RHS
1	0	0	**1.25**	0.75	0	41.25
0	0	1	2.25	- 0.25	0	2.25
0	1	0	- 1.25	0.25	0	3.75
0	0	0	- 0.75	- 0.25	1	**- 0.75**

Dual simpleks oran testi s1'in s3 yerine çözüme girmesini önerir. En iyi çözüm aynı zamanda tamsayılı çözümdür:

z = 40, x1 = 5, x2 = 0

6. PROJE YÖNETİMİ

Proje

Belirli bir amacı olan ve bir zaman süreci içerisinde birbirleri ile ilgili faaliyetlerin gerçekleştirildiği bir sistem çalışması

Süresi, maliyeti ve kaynak gereksinimi belirli faaliyetler dizisi

Proje Özellikleri

- Tek ve belirli bir amaç
- Geçici bir organizasyon
- Sınırlı zaman
- Sınırlı kaynak
- Birbirleri ile ilişkili faaliyetler dizisi
- Çevre ve dış kısıtlar

Kriterler

Bir çalışmanın proje olarak kabul edilebilmesi için çalışmanın

- başının ve sonunun iyice tanımlanmış
- her gün karşılaşılmayan işlerle ilgili
- özel bir önemi ve ağırlığı
- karmaşık yapıda
- çeşitli organizasyon, bölüm ve personelle ilişki içinde; çok disiplinli olması gerekir

Projelerde Kullanılan Kaynaklar

- Zaman,
- Finansman,
- İşçilik,
- Malzeme,
- Teçhizat,
- Personel

Proje Çeşitleri

- Yatırım,
- Sosyal,
- Kültürel,
- Askeri,
- Ekonomik, vb...

Proje Örnekleri

- İnşaat,Hizmet,İmalat, Araştırma-Geliştirme,
- Savunma,
- Yönetim...

Proje Yönetimi

- Proje Planlama
    1. Amacin belirlenmesi
    2. Projenin tanımlanması
    3. Faaliyet gereksinimlerinin belirlenmesi
    4. Takım oluşturulması
- Proje Çizelgeleme
    5. Kaynakların faaliyetlere atanması
    6. Faliyetler arası ilişkilerin düzenlenmesi
    7. Gerekli güncellemeler ve düzenlemelerin yapılması
- Proje Kontrol
    8. Kaynakların, maliyetlerin, kalitenin ve bütçenin izlenmesi
    9. Planların revizyonu ve gerekli değişikliklerin yapılması
    10. Kaynakların değişimi ile talebin karşılanması

Proje Yönetiminde Kullanılan Gösterim Türleri

- Çubuk (Gannt) Diyagramları
    - Zaman ölçeğine bağlı olarak çizilmiş bir seri yatay çizgi
    - Her çubuk projedeki bir faaliyetin başlangıç ve bitiş süresi ile faaliyet için gerekli zamanı gösterir

- Kritik faaliyet zincirini gösteremez

- Devre Diyagramları
    - İş süreleri ve müşterek bir birimle ifade edilen iş miktarı eksenleri üzerinde faaliyet çizgileri
    - Çizginin eğimi faaliyet gerçekleşme hızını (miktar/zaman) gösterir
- Ağ Diyagramları
    - Düğüm ve oklardan oluşan grafik
    - Faaliyetler arasındaki ilişkiyi gösterir
    - İki tür ağ diyagramı vardır:
        - **Ok Diyagramları** (Activity on Arc / Activity on Link)
          Oklar faaliyetleri, düğümler de faaliyetlerin başlangıç/bitişini gösterir
        - **Blok Diyagramları** (Activity on Node)
          Düğümler faaliyetleri, oklar da faaliyetler arası öncelik ilişkilerini gösterir

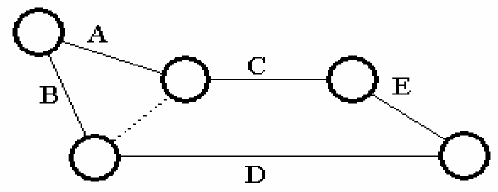
Örnek Gösterim

Beş faaliyetten oluşan bir proje olduğunu varsayalım. A ve B faaliyetleri bitmeden C faaliyeti başlayamasın. D, B bittikten sonra başlasın. E, C'den sonra olsun.

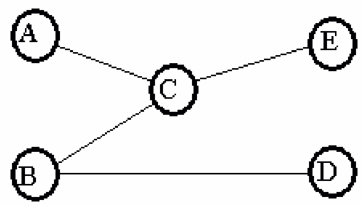
OK DİYAGRAMI:

Gerekli öncelik ilişkilerini göstermek için yapay ok (kesikli çizgi) kullanılabilir

Aynı başlangıç ve bitiş düğümünü birden fazla faaliyet kullanamaz



BLOK DİYAGRAMI



6.1 CPM / PERT

Eğer faaliyetlerin süreleri kesin olarak biliniyorsa, Kritik Yol Yöntemi (Critical Path Method – **CPM**) kullanılarak projenin tamamlanma süresi hesaplanabilir.

Eğer faaliyet sürelerinde bir belirsizlik varsa, Program Değerlendirme ve Gözden Geçirme Tekniği (Program Evaluation and Review Technique – **PERT**) kullanılarak projenin verilen bir zamanda bitirilme olasılığı hesaplanabilir.

Adımları

1. Projenin ve ilgili tüm faaliyetlerin tanımlanması
2. Faaliyetler arası ilişkilerin ve başlangıç-bitiş önceliklerin belirlenmesi
3. Ağ diyagramının çizilmesi
4. Her faaliyete gerekli zaman ve/veya maliyetin atanması
5. Ağ üzerindeki en uzun süreli yolun (*kritik yol*) hesaplanması
6. Proje planı, çizelgelemesi, izlenmesi ve kontrolu için ağın kullanımı

CPM

$\mathbf{t_{ij}}$: i. ve j. düğüm arasındaki faaliyetin süresi

ES: Bir faaliyetin en erken başlangıç zamanı,

EF: Bir faaliyetin en erken bitiş zamanı

LS: Bir faaliyetin en geç başlangıç zamanı,

LF: Bir faaliyetin en geç bitiş zamanı

$\mathbf{S_{ij}}$: Toplam Bolluk (S = 0 ise i ve j düğümleri arasındaki faaliyet kritik yoldadır)

$ES_{ij} = \text{Maks}\,(EF_i)$, $\quad ES_{1j} = 0$

$EF_{ij} = ES_{ij} + t_{ij}$,

$LS_{ij} = LF_{ij} - t_{ij}$,

$LF_{ij} = \text{Min}\,(LS_j)$ $\quad LF_{in} = EF_{in}$

$S_{ij} = LS_{ij} - ES_{ij}$ veya $S_{ij} = LF_{ij} - EF_{ij}$

PERT

Süresi kesin olarak bilinmeyen faaliyetler için *iyimser* (a), *kötümser* (b) ve *en olası* (m) süre tahminleri kullanılır.

Beta dağılımı esas alınarak tahmini ortalama ve varyans hesaplanır:

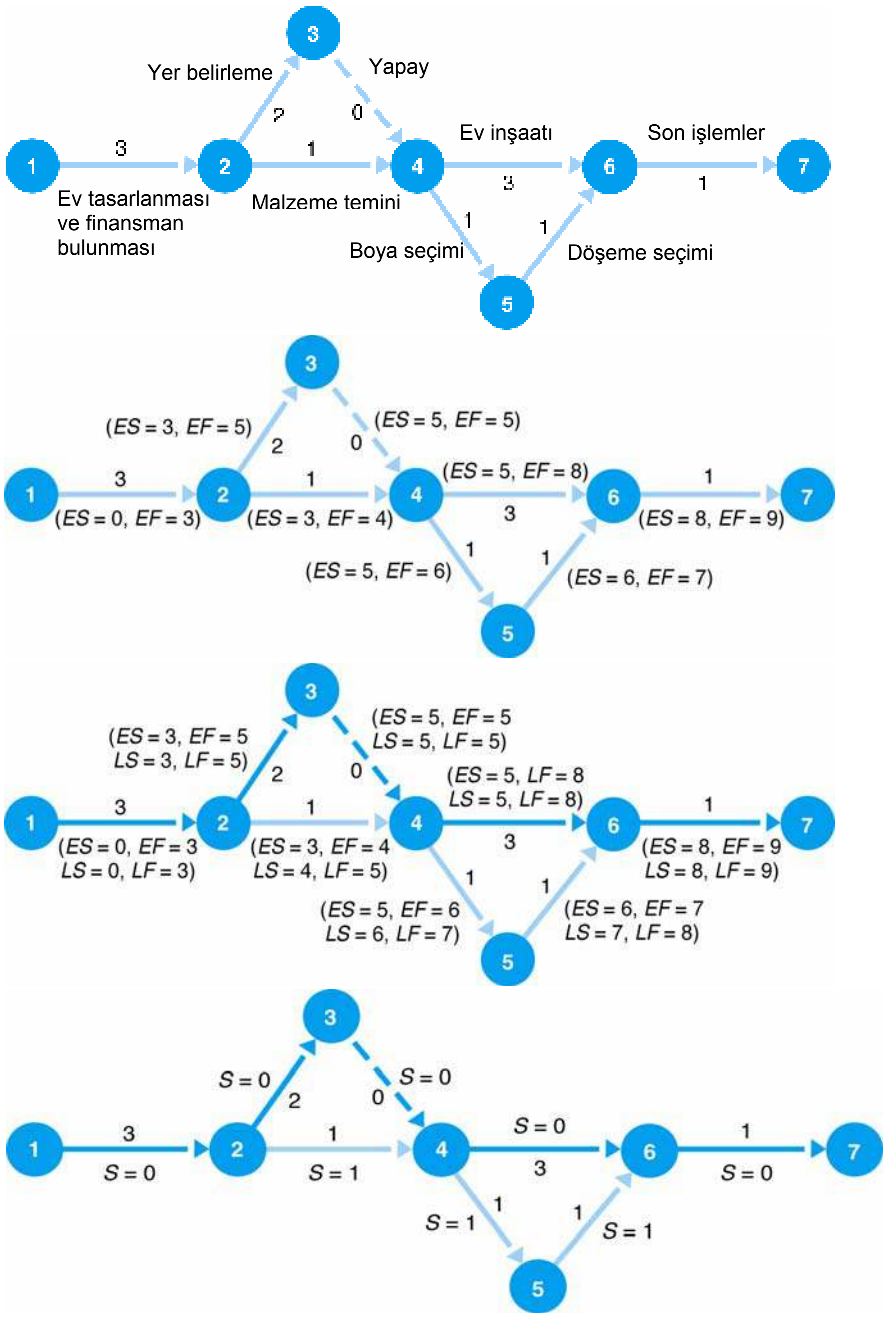
Beklenen süre: $\mathbf{t} = (a + 4m + b) / 6$

Varyans: $\boldsymbol{\sigma^2} = (b - a)^2 / 36$

6.1.1 CPM Örneği

Bir ev yapımı için öncelikle evin tasarlanması ve gerekli finansmanın bulunması gerekir (3 ay). Daha sonra ev yapılacak yer belirlenir (2 ay) ve aynı zamanda inşaat için gerekli malzeme temin (1 ay) edilir. Yerin belirlenmesi ve malzemenin temini gerçekleşir gerçekleşmez ev inşaatına (3 ay) başlanır. Bu sırada gerekli boyanın seçimi (1 ay) ve gerekli döşemenin seçimi (1 ay) peşpeşe gerçekleşir. Tüm bu işlemler bittiğinde son işlemler (1 ay) yapılır ve ev yapım projesi biter.

Projenin tamamlanma süresini ve kritik yolu bulunuz.



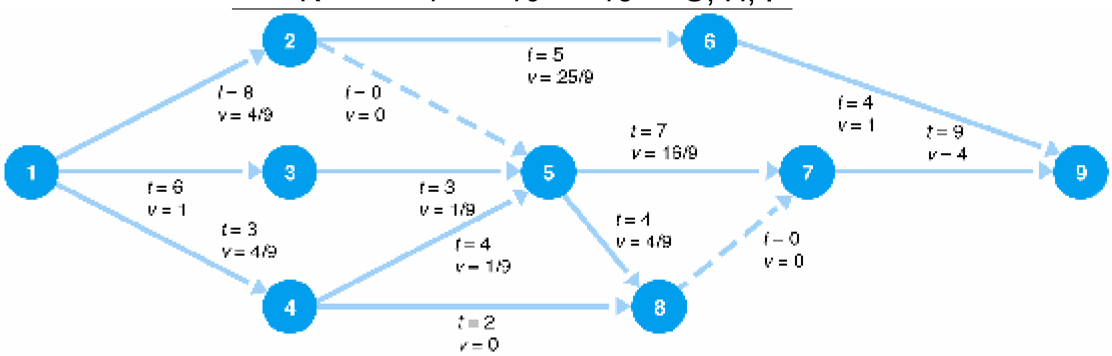
Kritik yol: 1–2–3–4–6–7; Projenin tamamlanma süresi: 9 ay

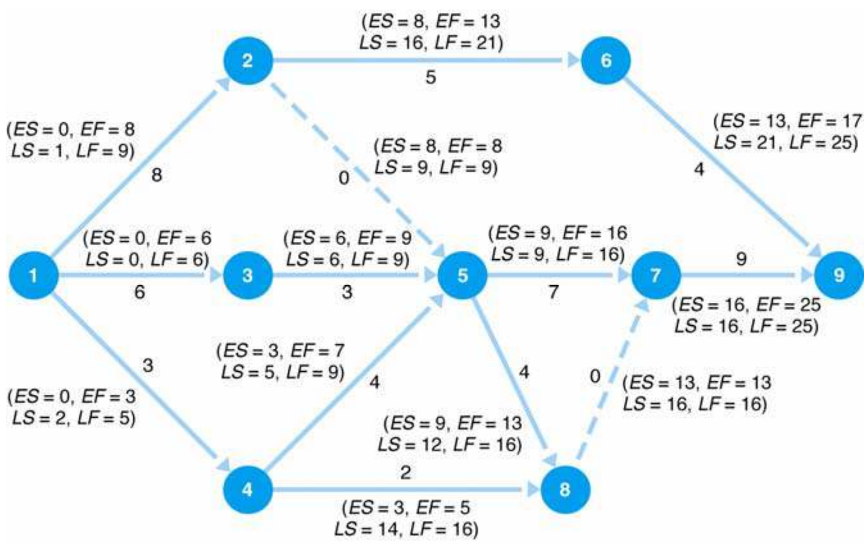
6.1.2 PERT Örneği

Aşağıdaki tabloda verilen faaliyetlerden oluşan bir proje söz konusu olsun.

Kritik yol üzerindeki faaliyetleri bulunuz. Projenin beklenen tamamlanma süresini ve varyansını hesaplayınız

Faaliyet	a	m	b	Önceki faaliyet
A	6	8	10	–
B	3	6	9	–
C	1	3	5	–
D	2	4	12	A
E	2	3	4	B
F	3	4	5	C
G	2	2	2	C
H	3	7	11	A, E, F
I	2	4	6	A, E, F
J	1	4	7	D
K	1	10	13	G, H, I





Kritik yol: 1–3–5–7–9   (B, E, H ve K faaliyetleri kritik yol üzerinde)

Projenin beklenen tamamlanma süresi: $\mathbf{t_p}$=25 hafta;

Projenin varyansı: σ_p^2=62/9   (1+1/9+16/9+4)

6.2 PROJE SÜRESİ İÇİN OLASILIK ANALİZİ

Beklenen proje süresinin *normal dağıldığı varsayılır* (merkezi limit teoremi).

Dağılımın ortalaması (μ) beklenen proje süresi, varyansı (σ^2) da projenin varyansıdır

Örnek

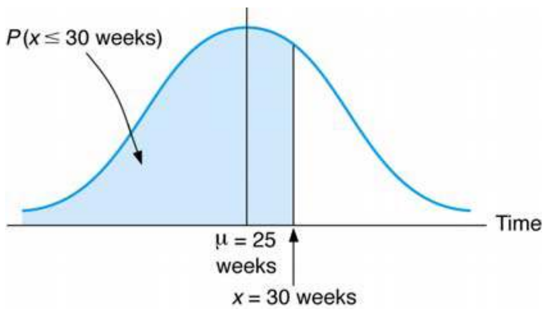
PERT örneğindeki projenin 30 haftada veya daha az sürede tamamlanma olasılığı ile 22 haftada veya daha az sürede tamamlanma olasılığı nedir?

Proje için dağılımın ortalaması (μ) 25, standart sapması (σ) 2.63'dür.

$Z = (x - \mu) / \sigma = (30–25)/2.63 = 1.90$Tabloda Z=1.9'a karşılık gelen olasılık .4713'dür.

Bu durumda projenin 30 haftada veya daha az sürede tamamlanma olasılığı

$P(x \leq 30) = .5 + .4713 = .9713$

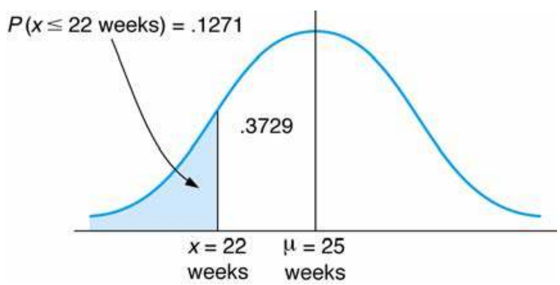


Projenin 22 haftada veya daha az sürede tamamlanma olasılığını bulmak için:

$Z = (x - \mu) / \sigma = (22–25)/2.63 = –1.14$Tabloda Z=1.14'e karşılık gelen olasılık .3729'dur.

Bu durumda projenin 22 haftada veya daha az sürede tamamlanma olasılığı

$P(x \leq 22) = .5 – .3729 = .1271$



6.3 PROJENİN DOĞRUSAL PROGRAMLAMA İLE MODELLENMESİ

Amaç projenin tamamlanabileceği süreyi enküçüklemektir.

i ve j düğümleri arasındaki bir faaliyet için

x_i : i. düğümün (olayın) en erken gerçekleşme zamanı (**ET(i)**)

x_j : j. düğümün en erken gerçekleşme zamanı

t_{ij}: faaliyetin süresi

olmak üzere n düğümlü bir proje için DP modeli

min $z = x_n – x_1$

s.t. $x_j – x_i \geq t_{ij}$ tüm faaliyetler için

$x_i \geq 0$ $i = 1,2, ..., n$

CPM Örneği için DP modeli

min $z = x_7 – x_1$

öyle ki $x_2 – x_1 \geq 12$ (A faaliyeti)

$x_3 – x_2 \geq 8$ (B faaliyeti)

$x_4 – x_2 \geq 4$ (C faaliyeti)

$x_4 – x_3 \geq 0$ (yapay faaliyet)

$x_5 – x_4 \geq 4$ (D faaliyeti)

$x_6 – x_4 \geq 12$ (E faaliyeti)

$x_6 – x_5 \geq 4$ (F faaliyeti)

$x_7 – x_6 \geq 4$ (G faaliyeti)

$x_i, x_j \geq 0$

DP modelinin çözümünün Lindo çıktısı

```
OBJECTIVE FUNCTION VALUE
        1)      36.00000
  VARIABLE        VALUE          REDUCED COST
        X7        36.000000          0.000000
        X1         0.000000          0.000000
        X2        12.000000          0.000000
        X3        20.000000          0.000000
        X4        20.000000          0.000000
        X5        28.000000          0.000000
        X6        32.000000          0.000000
       ROW   SLACK OR SURPLUS     DUAL PRICES
        A)         0.000000         -1.000000
        B)         0.000000         -1.000000
        C)         4.000000          0.000000
    YAPAY)         0.000000         -1.000000
        D)         4.000000          0.000000
        E)         0.000000         -1.000000
        F)         0.000000          0.000000
        G)         0.000000         -1.000000
```

“Gölge fiyat” = –1 ⇒ Faaliyet *kritik yol* üzerinde: A, B, Yapay, E, G (1–2–3–4–6–7)

“Amaç fonksiyonu en iyi değeri” = Projenin *tamamlanma süresi* = 36 hafta (9 ay)

6.4 PROJE SÜRESİNİN DARALTILMASI VE ZAMAN – MALİYET DEĞİŞ TOKUŞU

(Project Crashing and Time – Cost Trade-off)

Proje süresi faaliyetlere daha fazla kaynak atanarak azaltılabilir. Söz konusu atama proje maliyetini yükseltir. Bu durumda zaman ve maliyet arasında bir değiş tokuş analizi yapmak gerekir.

Bir veya daha fazla sayıda faaliyetin normal faaliyet sürelerini daha fazla kaynak kullanımı ile kısaltarak projenin tamamlanma süresinin azaltılması işleminde hangi faaliyetlerin seçileceği kararı "*proje süresinin daraltılması*" yöntemi ile verilir.

Daraltılmış faaliyet süresi ile maliyet arasında doğrusal bir ilişki olduğu varsayılır:

Örnek

Bir faaliyetin süresi 12 haftadan 7 haftaya indirilirken ilgili maliyet \$3000'den \$5000'e çıkıyorsa haftalık daraltma maliyeti (5000 – 3000) / (12 – 7) = \$400/hafta'dır.

Proje Yönetimi Örneği 1 için proje süresinin daraltılması aşağıdaki gibi olsun:

Faaliyet	Normal süre	Daraltılmış süre	Normal maliyet	Daraltılmış maliyet	Haftalık daraltma maliyeti
1 – 2	12	7	3000	5000	400
2 – 3	8	5	2000	3500	500
2 – 4	4	3	4000	7000	3000
3 – 4	0	0	0	0	0
4 – 5	4	1	500	1100	200
4 – 6	12	9	50000	71000	7000
5 – 6	4	1	500	1100	200
6 – 7	4	3	15000	22000	7000
		toplam	75000	110700	

Projeyi 30 haftada bitirmek için hangi faaliyetlerin süresi ne kadar kısaltılmalıdır? y_{ij} = i ve j düğümleri arasındaki faaliyetin süresindeki daralma (kısaltma) iken

$$\min z = 400y_{12} + 500y_{23} + 3000y_{24} + 200y_{45} + 7000y_{46} + 200y_{56} + 7000y_{67}$$

öyle ki

$y_{12} \le 5$	$y_{12} + x_2 - x_1 \ge 12$
$y_{23} \le 3$	$y_{23} + x_3 - x_2 \ge 8$
$y_{24} \le 1$	$y_{24} + x_4 - x_2 \ge 4$
$y_{34} \le 0$	$y_{34} + x_4 - x_3 \ge 0$
$y_{45} \le 3$	$y_{45} + x_5 - x_4 \ge 4$
$y_{46} \le 3$	$y_{46} + x_6 - x_4 \ge 12$
$y_{56} \le 3$	$y_{56} + x_6 - x_5 \ge 4$
$y_{67} \le 1$	$y_{67} + x_7 - x_6 \ge 4$
$x_7 - x_1 \le 30$	
$x_j, y_{ij} \ge 0$	

DP modelinin çözümü: z = 2500 (daraltma için ödenecek fazla maliyet)

$y_{12} = 5$, $y_{23} = 1$ (1–2 faaliyeti 5 hafta, 2–3 faaliyeti 1 hafta daraltılmalıdır)